Union-RoBERTa: RoBERTas Ensemble Technique for ICDAR 2023 Competition on Document Information Localization and Extraction

Notebook for the <UIT@AICLUB_TAB> Lab at CLEF 2023

Bao Gia Tran[1], Duy-Ngo Minh Bao[1], Khanh Gia Bui[1], Huy Viet Duong[1], Dang Hai Nguyen[1] and Hieu Minh Nguyen[1]

[1]*University of Information Technology - VNUHCM, Ho Chi Minh City, Vietnam*

Abstract

ICDAR organized a competition called DocILE for 2023 with the topic of information extraction from documents - a problem that 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 these 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 different versions of $RoBERTa$ along with a post-processing step, which helped us achieve a Top 3 position in the competition ranking board. We published our work at https://github.com/xbaotg/DocILE/

Keywords

ICDAR, DocILE, RoBERTa, Ensemble, Post-Processing, 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.

The International Conference on Document Analysis and Recognition (ICDAR) organized a competition on extracting information from business documents for 2023 called DocILE 2023 [1][2]. The participating teams will be provided with a dataset of invoice-like documents such

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)

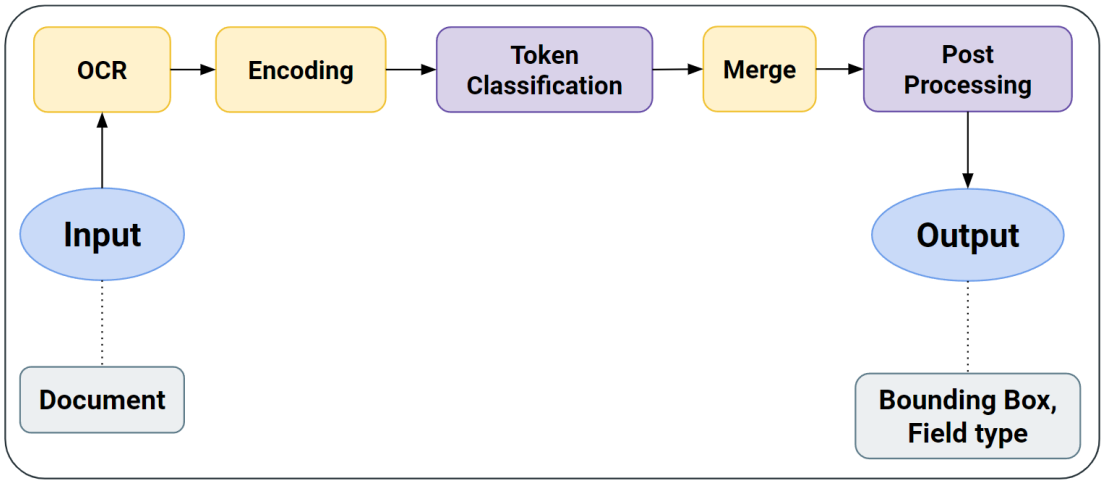

Figure 1: Pipeline of our working process, what we worked on are described as purple

as tax invoice, order, purchase order, receipt, sales order, proforma invoice, credit note, utility bill and debit note [3]. In Track 1, also known as KILE, participants are required to develop algorithms that can accurately segment the content of these documents and classify them into different categories.

Our efforts to combine different versions of $RoBERTa$ - a large language model that used to achieves state-of-the-art results on GLUE, RACE, and SQuAD in Natural Language Processing (NLP) [4] - together with a post processing step showed a relatively good result, achieving an AP of 0.6124 on the competition dataset [5]. Our primary focus was mainly on optimizing some aspects of the task, including the dataset, model, and model combinations. At first, the input data is a set of PDF pages containing invoices processed using DocTR (OCR model provided by the organizer), from which the bounding boxes and content of the information are obtained. Afterward, we classify those obtained information using Token Classification models. Finally, we merge those information based on their field type. We will discuss each component in detail in the following sections. The flowchart of our working process is shown in Figure 1.

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.

Based on the idea of set union mentioned in [7], we propose a new way to ensemble in order

to address the recall issue. Specifically, if any of the models predict that certain information belongs to a certain data field type, we will assume that information actually belongs to that data field type.

2.2. Pseudo-labeling

As described in [3], there is an abnormally huge amount of unlabeled data compared to the small amount of labeled data. We believe that utilizing this unlabeled data will significantly improve the performance of the models. This leads us to the idea of using semi-supervised learning methods.

[8] is a more effective method compared to other methods such as [9], [10], and [11]. We propose a different way to implement pseudo labeling technique for multiple models that are used for ensemble:

- Models will be trained on the labeled dataset.
- We use the ensemble technique to predict labels for the unlabeled dataset, which is then called as pseudo-labeled dataset.
- Pre-train the model on the pseudo-labeled dataset.
- Fine-tune the model on the training dataset.

Here, we pre-train the model on the pseudo-labeled dataset instead of mixing it with the labeled dataset as the pseudo-labeled dataset is a dataset that we have little control over, and it may contain cases that are completely different from the training dataset. Pre-training the model on the pseudo-labeled dataset 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:

- $box_1\ (x_{\text{1left}}, y_{\text{1top}}, x_{\text{1right}}, y_{\text{1bottom}})$
- $box_2\ (x_{\text{2left}}, y_{\text{2top}}, x_{\text{2right}}, y_{\text{2bottom}})$
- $x_{\text{1center}} = \frac{x_{\text{1left}} + x_{\text{1right}}}{2};\ y_{\text{1center}} = \frac{y_{\text{1top}} + y_{\text{1bottom}}}{2}$

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	1000	-
Synthetic	100,000	-
unlabeled	932,000	-

- $x_{\text{2center}} = \frac{x_{\text{2left}} + x_{\text{2right}}}{2}$; $y_{\text{2center}} = \frac{y_{\text{2top}} + y_{\text{2bottom}}}{2}$
- $distance = \sqrt{(x_{\text{1center}} - x_{\text{2center}})^2 + (y_{\text{1center}} - y_{\text{2center}})^2}$

If their distance is below or equal to the threshold, we will merge those two bounding boxes into a new one:

$$box_{final}\ (\min(x_{\text{1left}}, x_{\text{2left}}), \min(y_{\text{1top}}, y_{\text{2top}}), \max(x_{\text{1right}}, x_{\text{2right}}), \max(y_{\text{1bottom}}, y_{\text{2bottom}}))$$

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:

- 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 based on the results, comparing the performance between different models.

Table 2
Result obtained from training $We\text{-}RoBERTa$ with FGM

Model	AP
$We\text{-}RoBERTa$	0.562
$Ours\text{-}RoBERTa$	0.557
$Base\text{-}RoBERTa$	0.566

Table 3
Result obtained from training 3 $RoBERTa$ with Lion Optimizer

Model	AP
$We\text{-}RoBERTa$	0.562
$Ours\text{-}RoBERTa$	0.566
$Base\text{-}RoBERTa$	0.565

3.2. Model

We use 3 $RoBERTa$ models, 2 models trained on the Synthetic + Train dataset are provided by organizer [12], and the remaining model is taken directly from [13]. We will refer to these models with different names for easy distinction as follows:

- $Base\text{-}RoBERTa$: the baseline model $roberta\_base\_with\_synthetic\_pretraining$
- $Ours\text{-}RoBERTa$: the baseline model $roberta\_ours\_with\_synthetic\_pretraining$
- $We\text{-}RoBERTa$: the model that has not been trained on any dataset

3.3. Fast Gradient Method

For $We\text{-}RoBERTa$, we retrain it from scratch using the FGM [14] 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 as described in [15], this technique helps the model to be more generalized, so we still keep and apply it with other methods.

3.4. Lion Optimizer

For all 3 models, we replaced the default optimizer, from AdamW to Lion Optimizer [16], and trained for an additional 300 epochs. However, only $Ours\text{-}RoBERTa$ showed significant improvement, while the other models are mostly unchanged. Nevertheless, we will still use Lion Optimizer for the methods below because it converges much faster than AdamW [16]. Table 3 shows results obtained.

3.5. Ensemble

We perform Ensemble using different methods:

- Average output of classifier layers [6]

Table 4
Result obtained from Ensembling 3 $RoBERTa$

Method	AP
Average	0.580
Max-Voting	0.576
Union	0.607

Table 5
Result obtained from changing post processing threshold

Threshold (%)	$We\text{-}RoBERTa$ (AP)	$Ours\text{-}RoBERTa$ (AP)	$Base\text{-}RoBERTa$ (AP)
12	0.606	0.603	0.606
13	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

- Max-Voting [6]
- Union, proposed above

For each method, we ensemble the following models:

- $We\text{-}RoBERTa$ trained with FGM technique
- $Ours\text{-}RoBERTa$ trained with Lion Optimizer
- $Base\text{-}RoBERTa$

As shown in Table 4, our proposed Ensemble by using the Union method yields significantly better results than commonly used methods such as Max-Voting, and Average.

3.6. Post Processing

We performed the Post Processing method proposed above on each of the 3 models with different percentage thresholds of the document width.

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 the 14.5% as the default threshold. When this Post Processing method is combined with the methods we mentioned above, it significantly improves the results.

3.7. Pseudo Label

Due to the large number of documents, we will divide the unlabeled dataset into chunks using the following formula:

$$chunk_x\text{: document has index in } [10000x; 10000(x+1))$$

Table 6
Result obtained from training models with Pseudo-label

Model	AP on Train Data	AP on Train Data + chunk_0
$We\text{-}RoBERTa$	0.562	0.568
$Ours\text{-}RoBERTa$	0.566	0.57
$Base\text{-}RoBERTa$	0.566	0.576

Starting with the chunk_0 dataset, we preprocess it as follows:

- Remove documents belonging to clusters = -1 (documents that do not appear in the Train dataset).
- Remove documents with a size larger than 3000 pixels in any dimension.
- Remove rotated documents.

We then ensemble the 3 models trained on the Train dataset, combining with the Post Processing. Afterward, we predict on the unlabeled chunk_0 dataset to generate pseudo annotations for that dataset, which we call pseudo_0 before using that dataset to pretrain the 3 models with the following hyperparameters:

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

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

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

The addition of Pseudo-label 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.

4. Result

Overall, our results increased significantly compared to the baseline (+14%) as shown in Table 7 and Table 8. However, but we believe there are still many things we can do to further improve the results:

- Use a more powerful text detection and recognition model.
- Use models that incorporate layout features such as LayoutLMv3, LiLT, etc.

Table 7
Performance of different models on the DocILE competition task

Model	Baseline	+FGM	+Lion Optimizer	+chunk0
$We\text{-}RoBERTa$	-	0.562	0.562	0.568
$Ours\text{-}RoBERTa$	0.557	-	0.566	0.57
$Base\text{-}RoBERTa$	0.566	-	0.565	0.576

Table 8
Performance of different models on the DocILE competition task

Model	+Ensemble	+Post Processing
$We\text{-}RoBERTa + Ours\text{-}RoBERTa + Base\text{-}RoBERTa$	0.608	0.644

5. Conclusion

We present a solution to tasks required in Track 1 KILE of DocILE 2023 organized by ICDAR, which involved the development of an ensemble of dissimilar versions of $RoBERTa$ along with a post processing step. Our method focused on optimizing various aspects of the task, including dataset, model, and model ensemble. This combination shows its effectiveness when it outperformed the original $RoBERTa$ on information extraction from business documents and a relatively good result on the competition dataset with a AP of 0.6124 which secure our position as Top 3 in the ranking board.

References

[1] Šimsa, M. Šulc, M. Uřičář, Y. Patel, A. Hamdi, M. Kocián, M. Skalický, J. Matas, A. Doucet, M. Coustaty, D. Karatzas, Docile: Document information localization and extraction (rossum.ai), https://docile.rossum.ai/, 2023.

[2] I. A. for Pattern Recognition, The 17th international conference on document analysis and recognition, https://icdar2023.org/, 2023.

[3] Šimsa, M. Šulc, M. Uřičář, Y. Patel, A. Hamdi, M. Kocián, M. Skalický, J. Matas, A. Doucet, M. Coustaty, D. Karatzas, Docile benchmark for document information localization and extraction, arXiv preprint arXiv:2302.05658 (2023).

[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] Robust Reading Competition, Results - document information localization and extraction, https://rrc.cvc.uab.es/?ch=26&com=evaluation&task=1, 2023.

[6] P. Sanagapati, Ensemble learning techniques tutorial, https://www.kaggle.com/code/pavansanagapati/ensemble-learning-techniques-tutorial, 2021.

[7] F. Hausdorff, Set theory, volume 119, American Mathematical Soc., 2021.

[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.

[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.

[12] R. AI, Docile baselines, https://github.com/rossumai/docile/tree/main/baselines, 2021.

[13] 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.

[14] D. N. Huu, Generate adversarial examples using fgm, https://www.kaggle.com/code/ducnh279/generate-adversarial-examples-using-fgm, 2023.

[15] T. Miyato, A. M. Dai, I. Goodfellow, Adversarial training methods for semi-supervised text classification, arXiv preprint arXiv:1605.07725 (2016).

[16] 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).