Extended Overview of DocILE 2023: Document Information Localization and Extraction

Štěpán Šimsa[1,*,†], Michal Uřičář[1,*,†], Milan Šulc[2], Yash Patel[3], Ahmed Hamdi[4], Matěj Kocián[1], Matyáš Skalický[1], Jiří Matas[3], Antoine Doucet[4], Mickaël Coustaty[4] and Dimosthenis Karatzas[5]

[1]*Rossum, Křižíkova 148/34, 186 00 Prague, Czech Republic*
[2]*Second Foundation, Na Florenci 15, 110 00 Prague, Czech Republic*
[3]*Visual Recognition Group, CTU in Prague, Karlovo náměstí 13, 121 35 Prague, Czech Republic*
[4]*University of La Rochelle, 23 Avenue Albert Einstein, 17031 La Rochelle, France*
[5]*Computer Vision Center, Universitat Autónoma de Barcelona, 08193 Cerdanyola del Vallès, Barcelona, Spain*

Abstract

This paper provides an overview of the DocILE 2023 Competition, its tasks, participant submissions, the competition results and possible future research directions. This first edition of the competition focused on two Information Extraction tasks, Key Information Localization and Extraction (KILE) and Line Item Recognition (LIR). Both of these tasks require detection of pre-defined categories of information in business documents. The second task additionally requires correctly grouping the information into tuples, capturing the structure laid out in the document. The competition used the recently published DocILE dataset and benchmark that stays open to new submissions. The diversity of the participant solutions indicates the potential of the dataset as the submissions included pure Computer Vision, pure Natural Language Processing, as well as multi-modal solutions and utilized all of the parts of the dataset, including the annotated, synthetic and unlabeled subsets. This is an extended version of the condensed overview paper [1].

Keywords

Information Extraction, Computer Vision, Natural Language Processing, Optical Character Recognition, Document Understanding

1. Introduction

Documents, such as invoices, purchase orders, contracts, and financial statements, are a major form of communication between businesses. Extraction of the key information from such

CLEF 2023: Conference and Labs of the Evaluation Forum, September 18–21, 2023, Thessaloniki, Greece
*Corresponding author.
†These authors contributed equally.
stepan.simsa@rossum.ai (Š. Šimsa); michal.uricar@rossum.ai (M. Uřičář)
https://github.com/rossumai/docile (Š. Šimsa)
0000-0001-6687-1210 (Š. Šimsa); 0000-0002-2606-4470 (M. Uřičář); 0000-0002-6321-0131 (M. Šulc); 0000-0001-9373-529X (Y. Patel); 0000-0002-8964-2135 (A. Hamdi); 0000-0002-0124-9348 (M. Kocián); 0000-0002-0197-7134 (M. Skalický); 0000-0003-0863-4844 (J. Matas); 0000-0001-6160-3356 (A. Doucet); 0000-0002-0123-439X (M. Coustaty); 0000-0001-8762-4454 (D. Karatzas)

documents is an essential task, as they contain a wealth of valuable information critical for day-to-day decision-making, compliance, and operational efficiency.

Machine learning techniques, particularly those based on deep learning, natural language processing, and computer vision, have shown great promise in a number of document understanding tasks [2, 3, 4, 5, 6, 7, 8, 9, 10, 11], such as understanding of forms [12, 13, 14], receipts [7, 15], tables [16, 17, 18], or invoices [19, 20, 21]. Another approach to document understanding is question answering [22, 23].

The DocILE competition and lab at CLEF 2023 called for contributions to the DocILE benchmark [24], which focuses on the practically oriented tasks of Key Information Localization and Extraction (KILE) and Line Item Recognition (LIR), as defined in [25].

This paper provides an overview of the first run of the DocILE competition, summarizing the participants solutions and their final results, as well as a breakdown of the results with respect to certain information, e.g., with respect to zero-shot/few-shot/many-shot layouts in the training or with respect to text extractions, which are not otherwise checked in the main evaluation metric. This is an extended version of the condensed overview paper [1].

The paper is structured as follows: Section 2 describes the DocILE dataset, its acquisition and distribution to individual subsets; Section 3 summarizes the DocILE competition tasks and their respective evaluation process; all competing methods submitted to the competition are briefly described in Section 4; results from the competition, their breakdown and discussion are provided in Section 5; finally, Section 6 concludes the paper.

2. Data

The competition was based on the DocILE [24] dataset of business documents, which consists of three distinct subsets: *annotated*, *unlabeled*, and *synthetic*. The *annotated* set comprises $6,680$ real business documents sourced from publicly available platforms, which have been carefully annotated. The *unlabeled* set consists of a massive collection of $932,467$ real business documents also obtained from publicly available sources, intended for unsupervised pre-training purposes. The dataset draws its documents from two public data sources: UCSF Industry Documents Library [26] and Public Inspection Files (PIF) [27]. UCSF Industry Documents Library is a digitalized archive of documents created by industries that impact public health, while PIF consists of public files of American broadcast stations, specifically focusing on political campaign ads. The documents were retrieved in a PDF format, and various selection criteria were applied to ensure the quality and relevance of the dataset. The *synthetic* set comprises $100,000$ documents generated using a proprietary document generator. These synthetic documents are designed to mimic the layout and structure of 100 fully annotated real business documents from the annotated set.

Participants were allowed to use the $5,180$ training samples, 500 validation samples and the full synthetic and unlabeled dataset. The remaining $1,000$ documents form the test set. Usage of external document datasets or models pre-trained on such datasets was forbidden in the competition, while datasets and pre-trained models from other domains — such as images from ImageNet [28] or texts from BooksCorpus [29] — were allowed.

For each document, the dataset contains the original PDF file and OCR pre-computed using

the DocTR [30] library achieving excellent recognition scores in [31]. Annotations are provided for documents in the annotated and synthetic sets and include field annotations for the two competition tasks, KILE and LIR, as well as additional metadata: original source of the document, layout cluster ID[1], table grid annotation, document type, currency, page count and page image sizes. Annotations for the test set are not publicly available.

3. Tasks and Evaluation

The competition had two tracks, one for each of the two tasks, KILE and LIR, respectively. The goal of both of these tasks is to detect semantic fields in the document, i.e., for each category (field type) localize all the text boxes that have this semantic meaning and extract the corresponding text. For LIR, fields have to be additionally grouped into Line Items, i.e., tuples representing a single item. For a more formal definition, refer to [25], where the tasks were first defined. An example document with annotations for KILE and LIR is illustrated in Figure 1.

The DocILE benchmark is hosted on the Robust Reading Challenge portal[2]. As the test set annotations remain private, the only way to compare the solutions on the test set is to make a submission to the benchmark. During the competition, participants did not see the results or even their own score, so they had to select the final solution without gathering any info about the test set.

To focus the competition on the most important part of the two tasks, which is the semantic understanding of the values in the documents, only the localization part was evaluated. This means the tasks can be framed as object detection tasks, with LIR additionally requiring the grouping of the detected objects into Line Items. Therefore, standard object detection metrics are employed, with Average Precision (AP) as the main metric for KILE and F1 as the main metric for LIR. A predicted and a ground truth field are matching if they have the same field type and if they cover the same text in the document, as explained in detail in Figure 2. For LIR the fields also need to belong to corresponding Line Items, where this correspondence is found with a matching that maximizes the total number of matched fields, as shown in Figure 3.

Extracting the text of the localized fields is an obvious extension of the two tasks whose precision is also important. Therefore, both tracks in the benchmark have a separate leaderboard, where the extracted text is compared with the annotated text for each matched field pair and an exact match is required to count the pair as a true positive pair.

The benchmark also contains additional leaderboards for zero-shot, few-shot and many-shot evaluation. This is the same evaluation as in the main leaderboard but evaluated only on a subset of the test documents. Specifically, it is evaluated on documents from layout clusters that have zero (zero-shot), one to three (few-shot) or four and more (many-shot) samples available for training (i.e., in the training or validation set). These test subsets contain roughly 250, 250 and 500 documents, respectively. This enables a more detailed analysis of the methods and helps to understand which methods generalize better to new document layouts and which can better overfit to clusters with many examples available for training.

[1]Clusters are formed by documents that have similar visual layout and placement of semantic information in this layout.

[2]https://rrc.cvc.uab.es/?ch=26

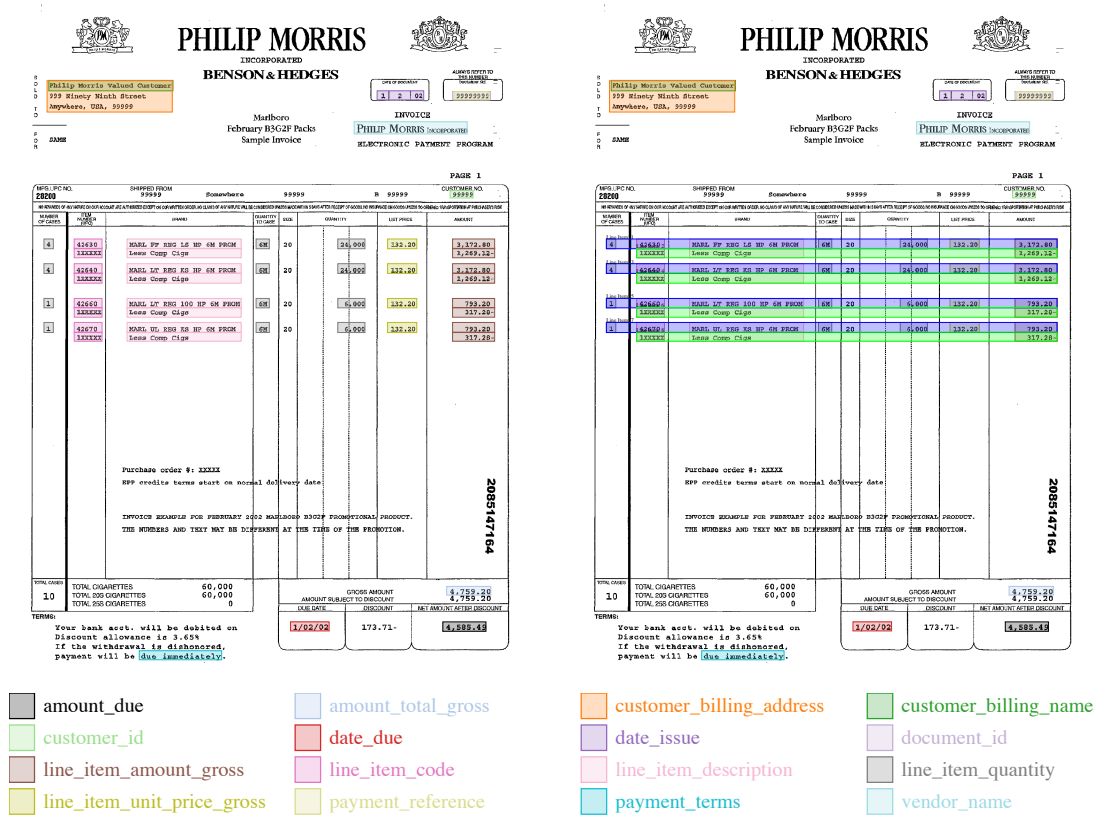


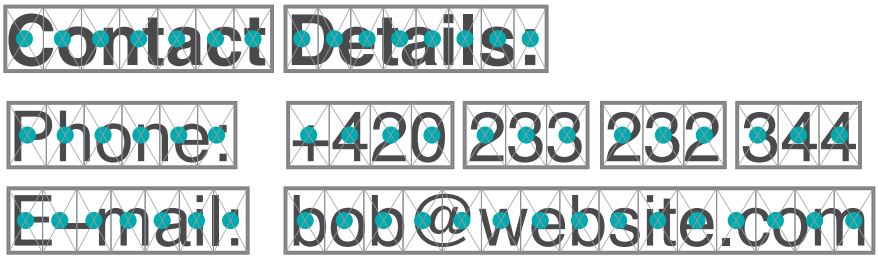
Figure 1: An example DocILE document with KILE and LIR annotations emphasized. Left: field annotations. Right: Line Item areas displayed by alternating blue and green. Bottom: color legend for KILE and LIR field types. The image is taken from [24].

4. Submissions

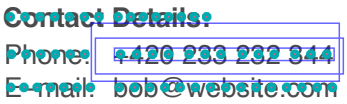
The competitions received contributions from 5 teams for the KILE task and 4 teams in the LIR task. See Figure 4 to compare this with the number of dataset downloads and competition registrations. We briefly present all the submitted methods in an alphabetical order.

4.1. GraphDoc — USTC-iFLYTEK, China

The team from the University of Science and Technology of China and iFLYTEK AI Research, China submitted a method [33], which jointly solves both KILE and LIR tasks. Their approach is based on an ensemble of a modified GraphDoc [11] tailored for the purpose of the DocILE competition, pre-trained on the DocILE unlabeled set and consequently fine-tuned on the training set. Both competition tasks are handled like Named Entity Recognition (NER), followed by a special *Merger* module, which operates on the attention layer from the GraphDoc model and the merging strategy is therefore learned, unlike in the baseline method. The authors noticed the inherent nature of the KILE and LIR task and exploited it naturally — the word



(a) Each pre-computed OCR word is split uniformly into pseudo-character boxes based on the number of characters. Pseudo-Character Centers are the centers of these boxes.



(b) Correct extraction examples.

(c) Incorrect extraction examples.

Figure 2: Correct and incorrect bounding box predictions of the phone number are shown in 2b and in 2c, respectively. A predicted field matches the location of a ground truth field if their bounding boxes cover the same text. More precisely, the fields must contain exactly the same Pseudo-Character Centers defined in 2a. Note: in 2b, only one of the predictions would be considered correct if all three boxes were predicted. Images are taken from [32].

tokens are merged to instances by the first level Merger module and then the second Merger module operates on these instances for the line item classes and merges them into final line items. The proposed method still uses some level of a rule-based post-processing, which is based on the observation of data: 1) some field annotations contain only part of the detected text boxes from DocTR and need to be manually split (such as `currency_code_amount_due` fields that usually contains only the symbol '$'); 2) some symbols are frequently detected as part of the OCR word box, but excluded from the annotations (such as the symbol '#'); 3) Text boxes that are far apart rarely belong to the same instance, or to the same line item.

Besides the contribution on the model side, the authors also devoted some effort to improve the OCR detections provided, by removing the detections with low confidence and by running DocTR [30] on scaled-up images ($1.25\times$, $1.5\times$, and $1.75\times$) and aggregating the found text boxes to improve the recall of the OCR detections. The OCR detections are also re-ordered, similarly as in the baseline methods, in the top-down left-right reading order.

Since the proposed method uses multi-modal input (text, layout, vision), we can put it into a category of combination of NLP and CV.

4.2. LiLT — University of Information Technology, Vietnam

The team from University of Information Technology, Vietnam submitted a method based on the baselines with a layout-aware backbone LiLT [34]. The authors decided to re-split the provided dataset to 80% for training and 20% for validation (original ratio was 90% and 10%,

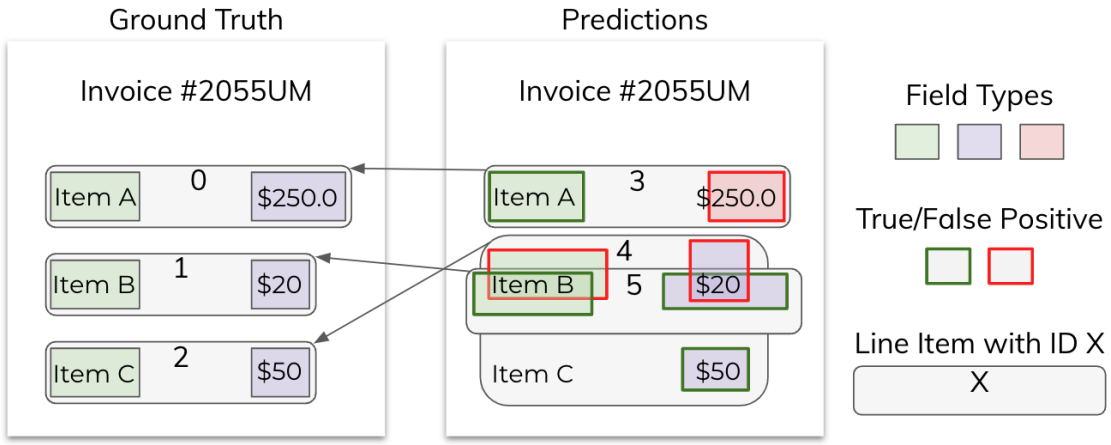


Figure 3: Visualization of Line Item (LI) matching. Both the annotations and predictions consist of three line items where LI 3 and LI 0 are clearly matched together. The two fields of "Item B" are detected both as part of LI 4 and LI 5, so greedy assignment might assign LI 4 to LI 1, leading to only three matched fields in total. Instead, maximum matching assigns LI 4 to LI 2 and LI 5 to LI 1, leading to four matched fields overall.

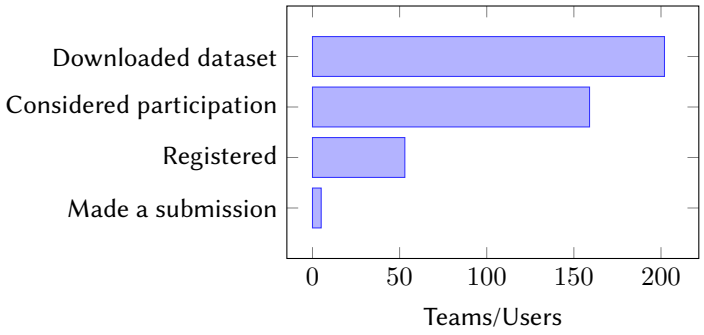


Figure 4: In the four and half months since its release on February 1, the DocILE dataset was downloaded by over 200 unique users. While 159 of them indicated they are considering participating in the competition, only 53 teams actually registered and 5 of them made a submission to the benchmark. Based on the feedback from a few of these teams, we attribute this to the tight schedule and to the competitiveness of the baselines, as they were not so easy to beat.

respectively), arguing that the original split was leading to a poor generalization. Another contribution was filtering out low-confident OCR detections. There is no mention of the usage of either the synthetic or the unlabeled sets of the DocILE dataset in the manuscript.

Unfortunately, despite competing in both KILE and LIR tasks, the authors submitted a manuscript describing only the solution for the LIR task. Since the backbone LiLT uses a combination of text and layout input, we categorize it as a pure NLP solution.

The review process of the authors' manuscript discovered a violation of the benchmark rules due to the usage of the prohibited pre-trained checkpoint for the LiLT backbone. The authors

used the checkpoint from training on the IIT-CDIP [35] dataset, which is a document dataset. Therefore we had to remove this method from the official leaderboard of the competition and the benchmark.

4.3. Union-RoBERTa — University of Information Technology, Vietnam

The team from University of Information Technology, Vietnam submitted a method [36] which is heavily based on the provided baselines. Their method, coined as Union-RoBERTa, is an ensemble of two provided baselines [24] with a plain RoBERTa trained from scratch on the synthetic and training data using Fast Gradient Method. They use the affirmative strategy for the ensemble (hence the Union in the name) and follow it by an additional merging of fields based on distance with a threshold tuned on the validation set. This ensemble is then used to generate pseudo-labels for $10,000$ samples from the unlabeled set which are then used for additional pre-training of the three models followed by an additional training on the training set. Although there is not much novelty in the proposed method, it is a nice example how well-established practices can yield significant improvements.

The proposed method participated in the KILE task only. Since the method is based on RoBERTa models, we put it into a pure NLP category.

4.4. ViBERTGrid — Ricoh Software Research Center, China

The team from Ricoh Software Research Center, China submitted a method based on token classification with ViBERTGrid [37], followed by a distance-based merging procedure. The team participated in both KILE and LIR tasks. However, the results were below baselines for both tasks and the authors decided not to submit a manuscript with further details. We can only guess, based on the provided description with ViBERTGrid, that the method was a combination of NLP and CV.

We noticed that the method probably suffers from not using the adequate score (all detections were using the same score 1.0) which could explain why AP is significantly lower compared to the other methods, while F1 measure on the KILE task is in the middle of the ranking, as seen in Figure 5a and discussed more in Section 5.5.

4.5. YOLOv8 — University of West Bohemia, Czech Republic

The team from University of West Bohemia, Czech Republic submitted a method [38] based on the combination of YOLOv8 [39] and CharGrid [40] with modifications, such as splitting the word boxes to pseudo-characters, not using the one number encoding of a character directly but a three numbers encoding instead, and concatenating the image with the CharGrid representation. The authors did not leverage synthetic nor unlabeled parts of the dataset, but they used augmentations during training. Due to the faster training procedure, they decided to use just random translation for augmentation, even though the best results in ablation study were observed when mosaicking was applied. The method works quite well on the KILE task (where it even achieves the highest F1) but falls behind on the LIR task. The latter is attributed to the increased number of false positive detections.

This contribution is purely based on computer vision.

5. Results and Discussion

The results for the KILE and LIR tasks, including the baselines from the DocILE dataset paper [24], are displayed in Figures 5a and 5b, respectively. We can see that while on the KILE task participants approaches clearly outperform the provided baseline by a large margin on the main evaluation metric (AP), on the LIR task, there is not such a big improvement, except for the GraphDoc based approach. The baseline methods are marked with ⊟ symbol.

Interestingly, for the KILE task, the secondary metric (F1) does not seem to be correlated with the primary metric (AP) and several of the methods, including the baselines, are comparatively much better on F1 than on AP. In fact, the YOLOv8 based approach outperforms the otherwise winning GraphDoc in F1 metric. This might be related to the fact that AP takes into account the score assigned to individual predictions, while F1 does not, and that some teams focused on assigning good scores to predictions more than others, as discussed in Section 5.5.

In the LIR task, there is some correlation between the primary metric, which in this task is F1, and the secondary metric (AP), with a slight violation for the GraphDoc based method.

Considering the achieved metric values, we can say that the DocILE benchmark poses very challenging tasks, because the best results on both KILE and LIR tasks are below 80% of the respective quality metric.

5.1. Text Extraction Evaluation

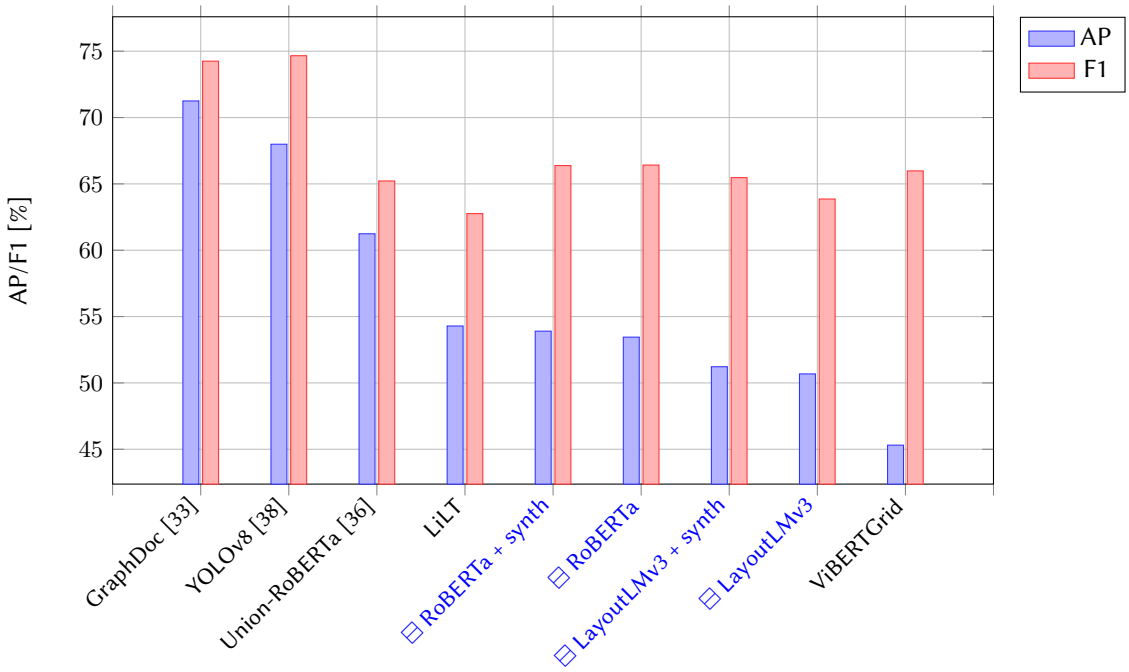
Figure 6 summarizes the results when text extractions are checked in the evaluation. Note, that this was intentionally not done in the main evaluation, which focuses more on the localization part, so that participants do not have to focus on optimizing the OCR solution for text read out. However, in a real-world system, this would likely be the main metric for evaluation and therefore we present results of all of the competing methods when this strict text comparison is employed. By definition, all methods are performing worse on both KILE and LIR task, compared to the main localization-only evaluation. Also both AP and F1 metrics show less variance for all competing methods. Unfortunately, the YOLOv8 based method did not provide the text outputs (which was not required for the competition), so we cannot evaluate this method properly.

The KILE task, summarized in Figure 6a, shows that the GraphDoc still outperforms all the other competitors. However, the margin is not as big as in the final evaluation.

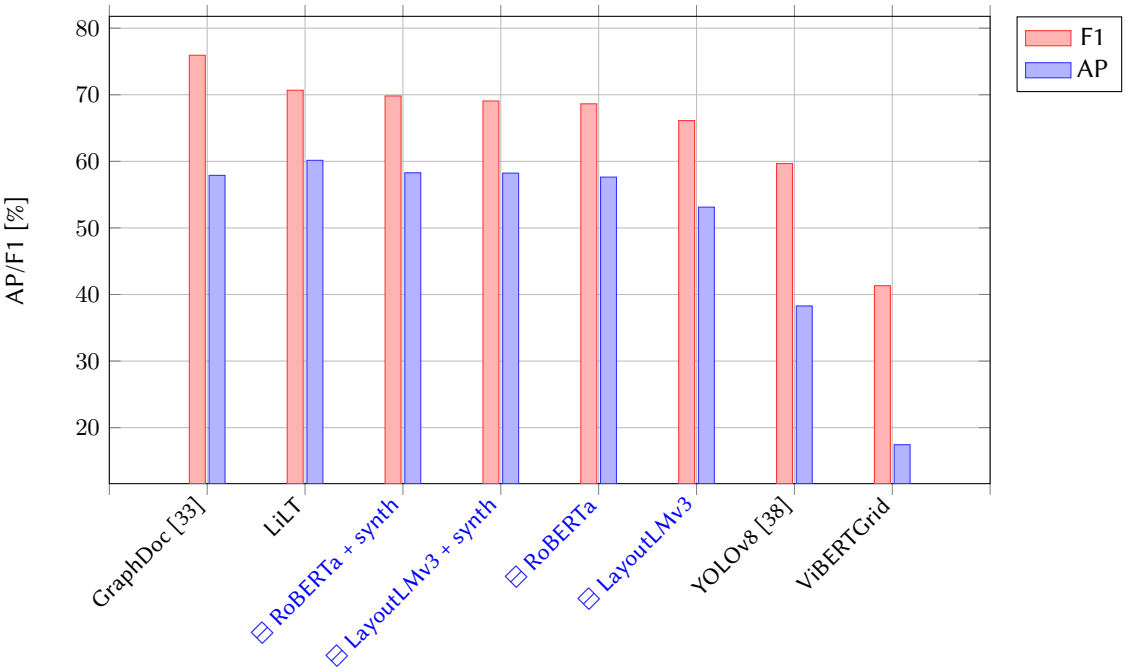
The LIR task is summarized in Figure 6b. Surprisingly, the GraphDoc based method, which was winning in the main evaluation, and which kept its position for the KILE task, is now lagging behind quite significantly. We believe this might be attributed to the lack of effort invested to the text read-out after merging.

5.2. Evaluation on zero/few/many-shot layouts

In this section, we present a break-down of the evaluation with respect to the document layouts seen/unseen during training, hence providing hints about how the particular method generalizes. We have three distinct categories for this evaluation: 1) zero-shot, formed by document layouts that were not in the training nor validation sets; 2) few-shot, which is formed by document layouts that have 1–3 samples in the training and validation subset of the DocILE dataset; 3) many-shot, with 4 or more samples in the training and validation subset.

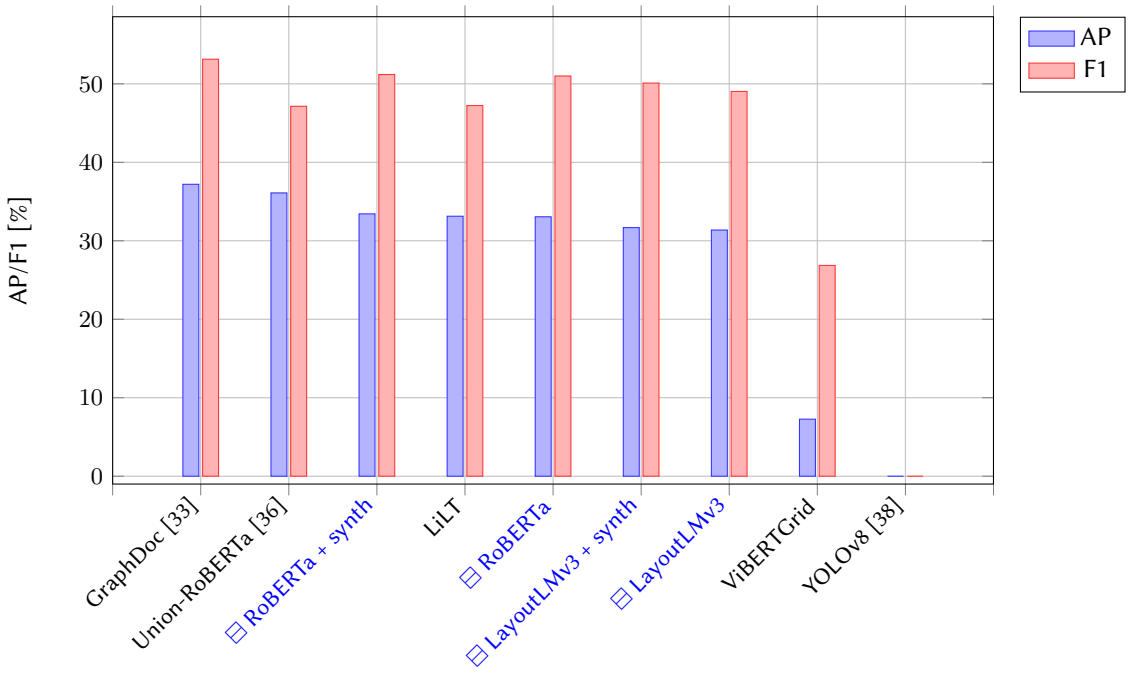


(a) KILE overall results

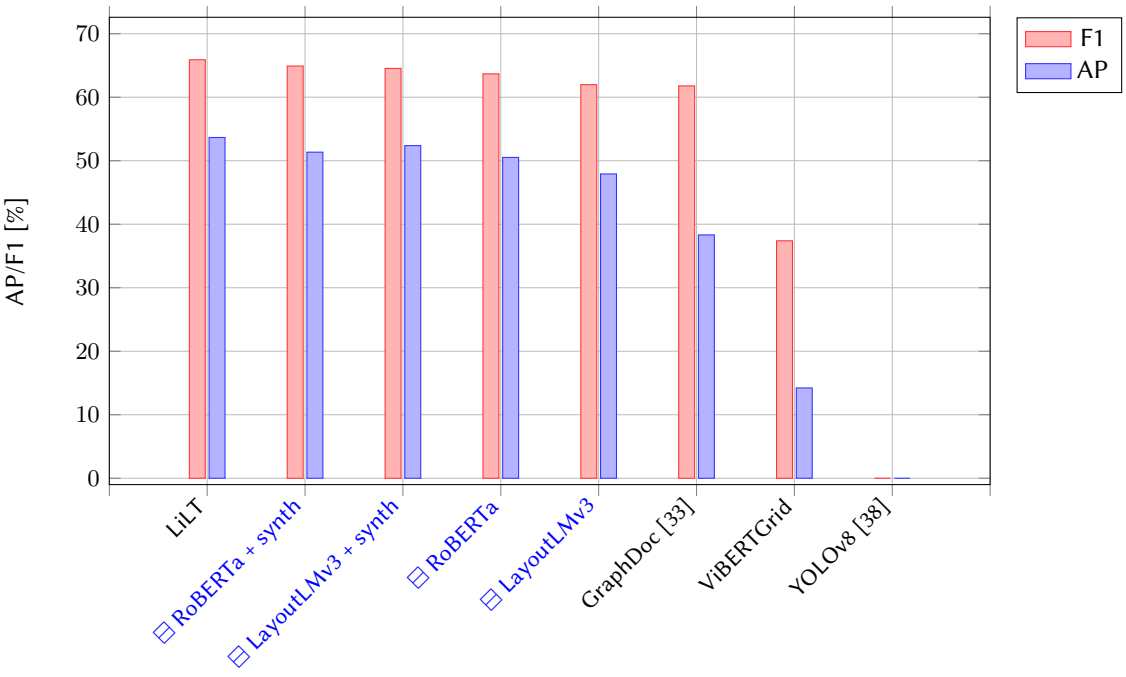


(b) LIR overall results

Figure 5: Final results of the DocILE'23 competition for Task 1: KILE (5a) and Task 2: LIR (5b).



(a) KILE text extraction results



(b) LIR text extraction results

Figure 6: Text extraction results for Task 1: KILE (6a) and Task 2: LIR (6b).

In Figure 7, we show the results of the first category — zero-shot. For the KILE task (Figure 7a), we can see that GraphDoc is still a clear winner with a relatively high margin. However, interestingly, YOLOv8 performs much worse, compared to the overall results. This might be attributed to the fact that this method did not leverage the unlabeled part of the DocILE dataset and therefore is more prone to overfitting. The RoBERTa baseline performs better than RoBERTa with supervised pre-training on synthetic data, which might be caused by the fact that synthetic documents are based on selected layouts from the training set and these layouts are not present in the zero-shot test subset, although we do not see the same effect in the case of LayoutLMv3 or the LIR task. Union-RoBERTa gets to the second place; considering it is basically an ensemble of the baselines, this might be an indicator that ensembling can also improve generalization properties. It is also worth mentioning that ViBERTGrid is very good in generalization when the F1 measure is concerned.

The LIR task (Figure 7b) shows similar results — GraphDoc remains on the first place, LiLT lost its second position to RoBERTa with supervised pre-training on synthetic data and LayoutLMv3 baseline pre-trained on synthetic data swapped its position with RoBERTa baseline. Note, that for both tasks, the results are significantly worse for the zero-shot setup compared to the overall results, showing a room for improvement with respect to generalization of all competing methods.

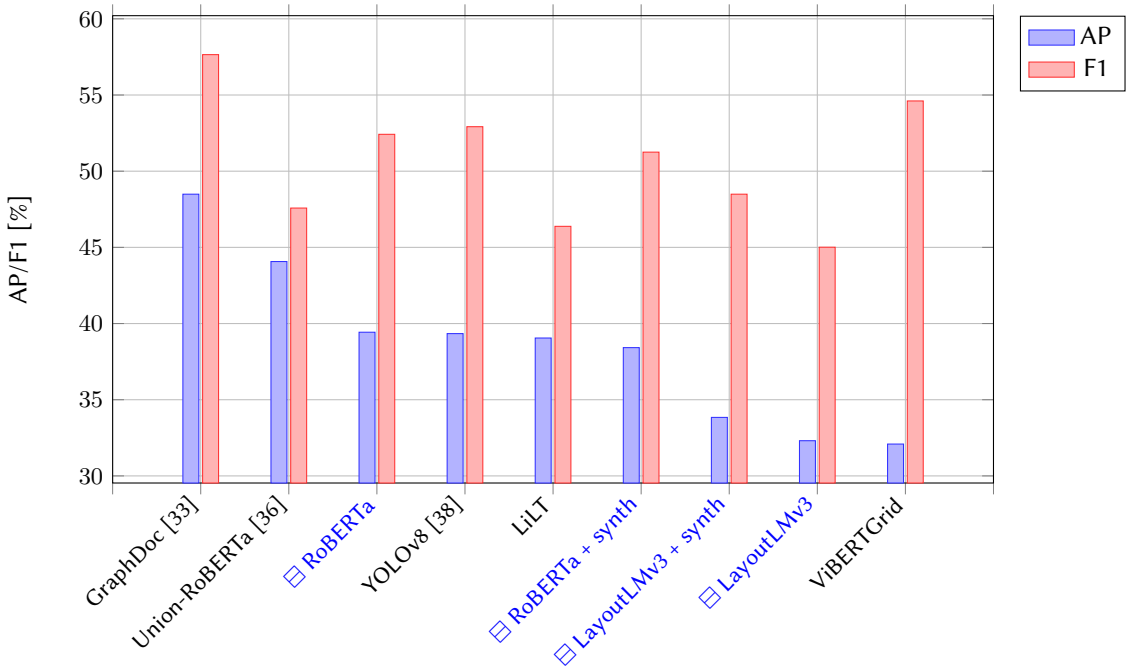
The results of the few-shot evaluation are in Figure 8. The KILE task (Figure 8a) shows that only a few similar layouts during training can help significantly. We see, that YOLOv8 gets back to the second place, RoBERTa+synth baseline improves significantly. It is also worth mentioning that all methods improve both the AP and F1 metrics by roughly 10%, compared to the zero-shot setup, with some exceptions with even a better improvement, and ViBERTGrid, which has a lower improvement.

In the LIR task (Figure 8b), we can see that all methods get closer to each other, similarly as it was in the overall evaluation. However, what is really surprising is that the results for zero-shot variant were actually slightly better than the results for few-shot. Also, the LiLT benefits from seeing at least a few similar layouts during training much more than GraphDoc and overtakes its first position. Also RoBERTa baseline is slightly better than RoBERTa+synth.

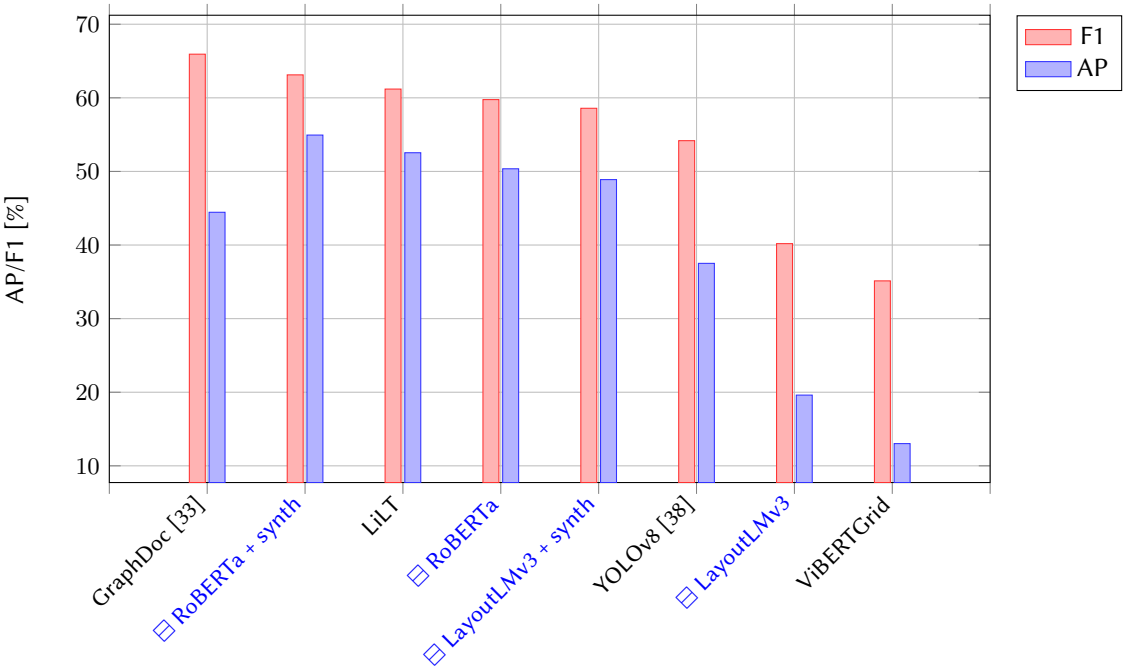
In Figure 9, we show the results for the many-shot scenario. For the KILE task (Figure 9a), it can be seen that the order of competing methods converges to the same one as for the overall results, with the only exception of LayoutLMv3 and LayoutLMv3+synth baselines, which are swapped. We can also see, that the results are roughly 10% better than for the overall case, which is not surprising, since the overall case contains also unseen layout examples. For the LIR task (Figure 9b), we see a similar trend, but the improvement is not that significant. Interestingly, the LayoutLMv3+synth baseline gets to the second place outperforming both LiLT and RoBERTa+synth baselines. However, we should point out that the results of these methods are very close.

5.3. Breakdown based on document source

The number of documents of each layout cluster in the training, validation and unlabeled subsets are depicted in Figure 13 and Figure 14 for the UCSF and PIF document source type subsets, respectively.

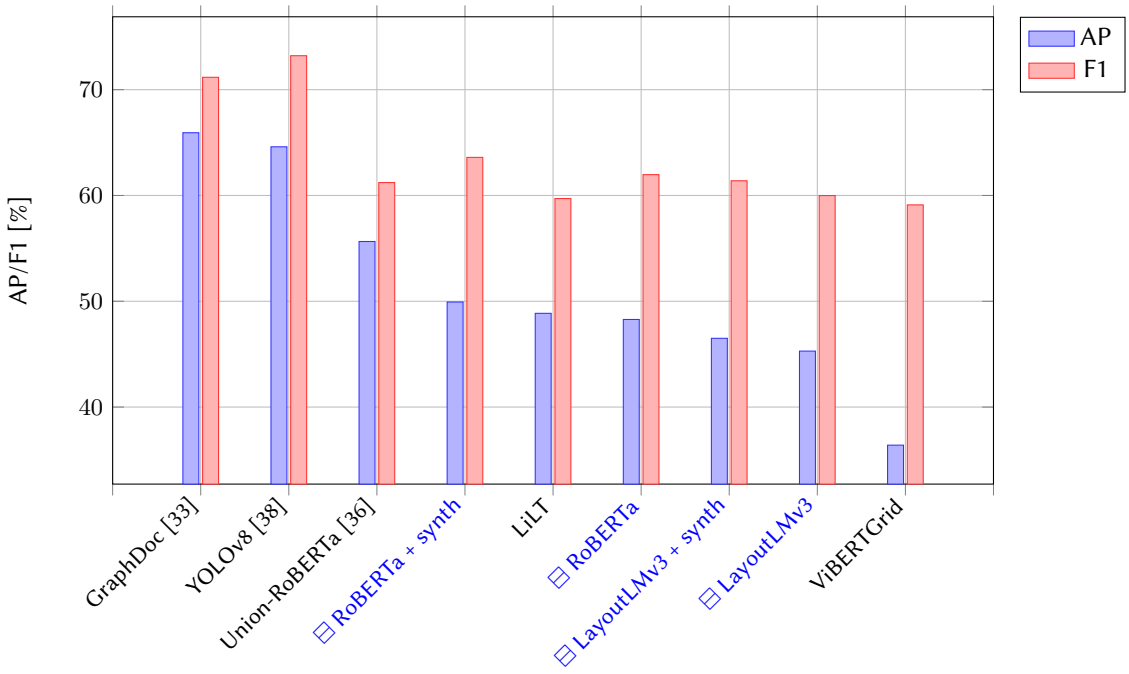


(a) KILE zero-shot results

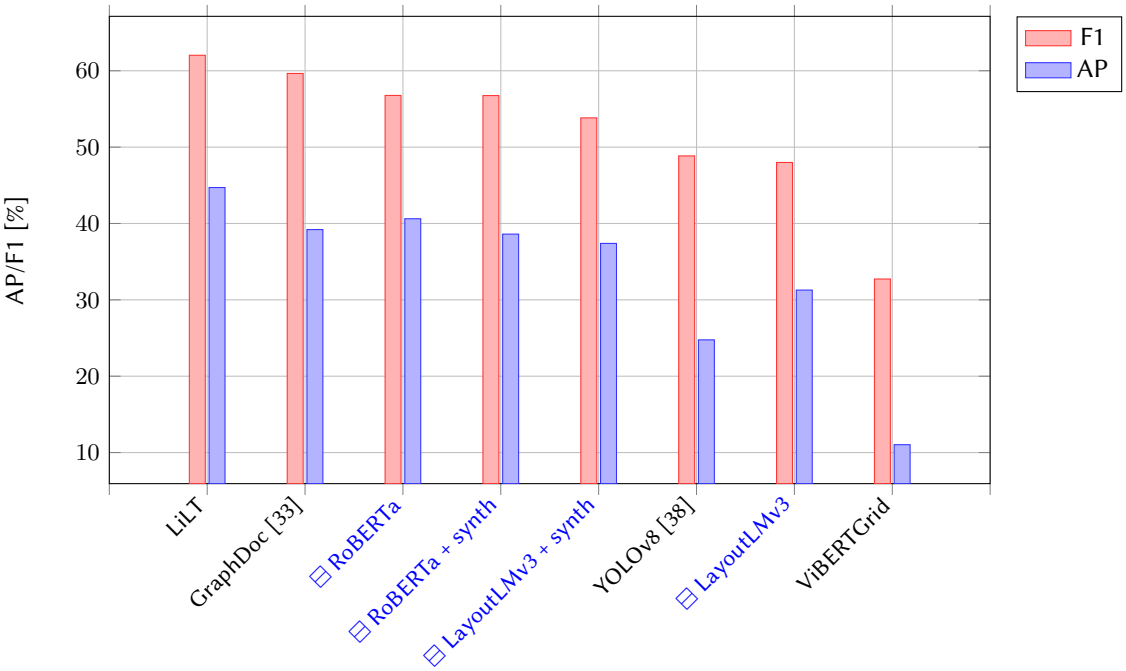


(b) LIR zero-shot results

Figure 7: Results on the zero-shot subset for Task 1: KILE (7a) and Task 2: LIR (7b).

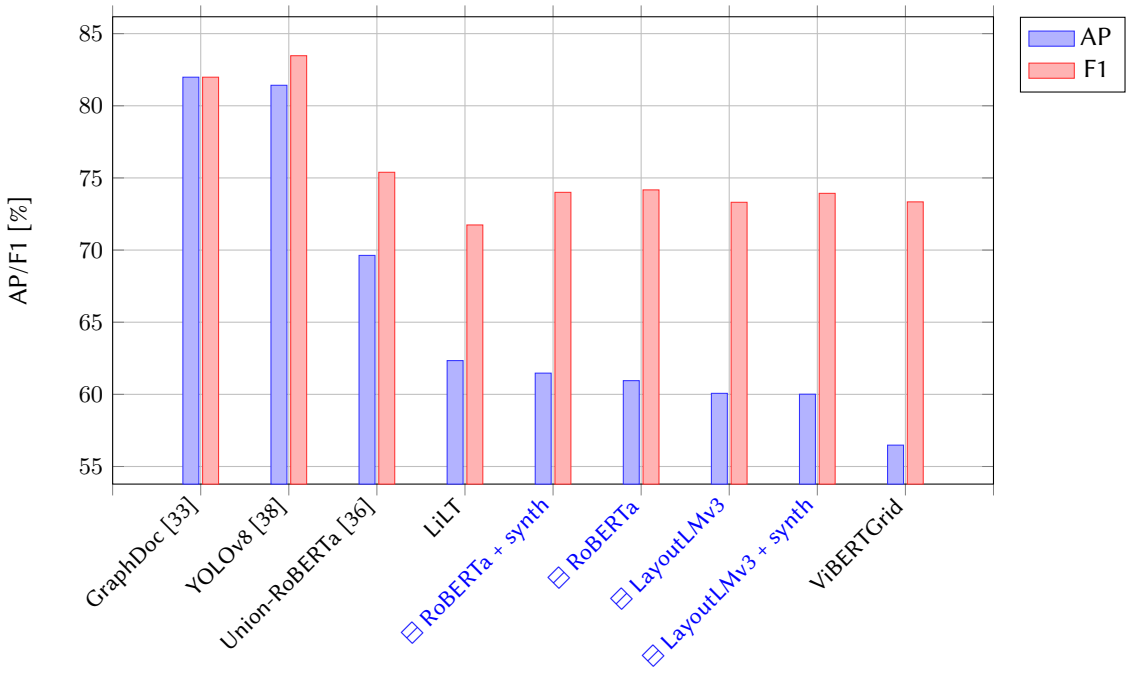


(a) KILE few-shot results

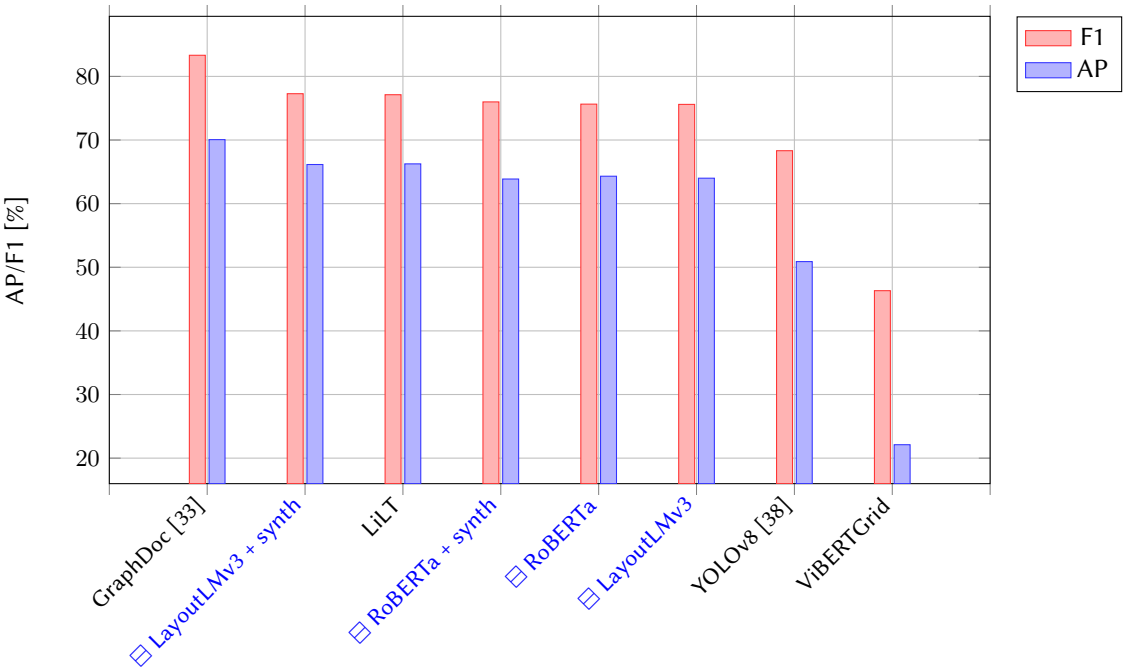


(b) LIR few-shot results

Figure 8: Results on the few-shot subset for Task 1: KILE (8a) and Task 2: LIR (8b).



(a) KILE many-shot results



(b) LIR many-shot results

Figure 9: Results on the many-shot subset for Task 1: KILE (9a) and Task 2: LIR (9b).

The distribution of the number of document pages in the training, validation and unlabeled sets are depicted in Figure 11 and Figure 12 for the UCSF and PIF document sources subsets, respectively.

Figure 10 depicts the breakdown of results based on the document source type and also in combination with zero/few/many-shot layout analysis for both KILE and LIR tasks of the winning solution [33]. From the graphs, we can see that documents from the PIF source are posing a bigger problem to the method which is interesting since UCSF has bigger variance in the number of different layouts. This noticable difference might be attributed to the fact that PIF documents are more frequently multi-paged. There might be some non-trivial changes in document layout thanks to the transition from one page to another, especially when tables are concerned.

5.4. Using synthetic and unlabeled data

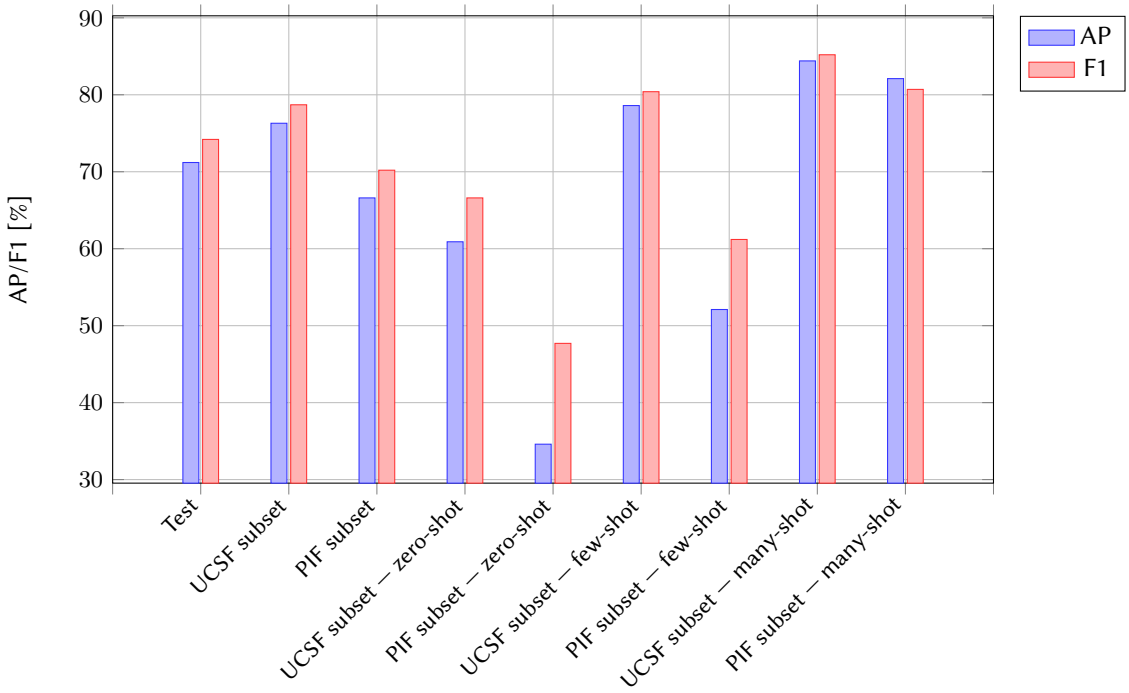
According to the submitted participant papers, only GraphDoc and partly also Union-RoBERTa (they used only $10,000$ samples) leveraged the unlabeled part of the DocILE dataset. We believe that the reason for not using the unlabeled data was mainly relatively tight time constraints. It is visible that GraphDoc-based method wins in almost all comparisons with the exception of the few-shot (Figure 8b) and text extraction (Figure 6b) LIR tasks. However, it is hard to judge if this could be attributed to the usage of the unlabeled data.

Only the authors of Union-RoBERTa report the usage of the synthetic part of the DocILE dataset. Again, the reason for not using the provided synthetic data might be time constraints. From the baselines point of view, we see that using the synthetic data helps in most situations, with a few exceptions like the zero-shot KILE task (Figure 7a) and the few-shot LIR task (Figure 8b), where RoBERTa performs better than RoBERTa+synth. However, simultaneously, the LayoutLMv3+synth outperforms LayoutLMv3. But we should point out that in these cases the differences are not very big.

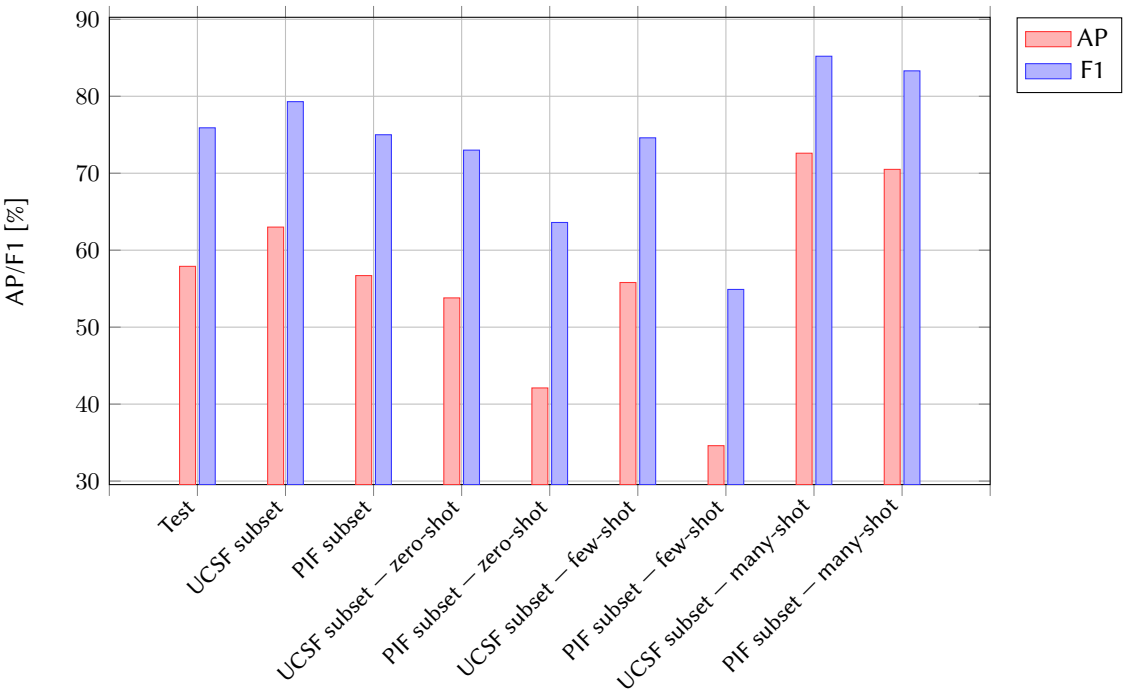
5.5. Importance of score for Average Precision

While for the F1 metric the score assigned to individual predictions is ignored, it plays an important role for the AP metric. In AP, predicted fields are first sorted by the score, then the precision-recall pairs are computed iteratively and finally the metric itself is the average precision achieved for different recall thresholds. Therefore, if we can ensure that there are more true positives among the predictions with higher score than among the predictions with lower score, the precision will increase for lower recall thresholds and remain similar for higher recall thresholds, when compared to the case when scores are random.

To prove this point, we can look at two examples. The ViBERTGrid method used the same score for all predictions and it achieves very poor results on AP compared to its results on F1, as can be seen in Figure 5. On the other hand, in the participant paper of the GraphDoc method, they argue that the prediction score is important for the AP metric and they show that by using a carefully selected score they achieve a 13.6% higher result on AP on the validation set compared to using the same score for all predictions. We can see in Figure 5 that for the KILE task GraphDoc has the smallest difference between the AP and F1 metrics of all the methods.



(a) GraphDoc [33] breakdown of results based on the document source and zero/few/many-shot layout for the KILE task.



(b) GraphDoc [33] breakdown of results based on the document source and zero/few/many-shot layout for the LIR task.

Figure 10: Breakdown of results based on the document source type and in combination with zero/few/many-shot layouts of the winning solution GraphDoc [33] for Task 1: KILE (10a) and Task 2: LIR (10b).

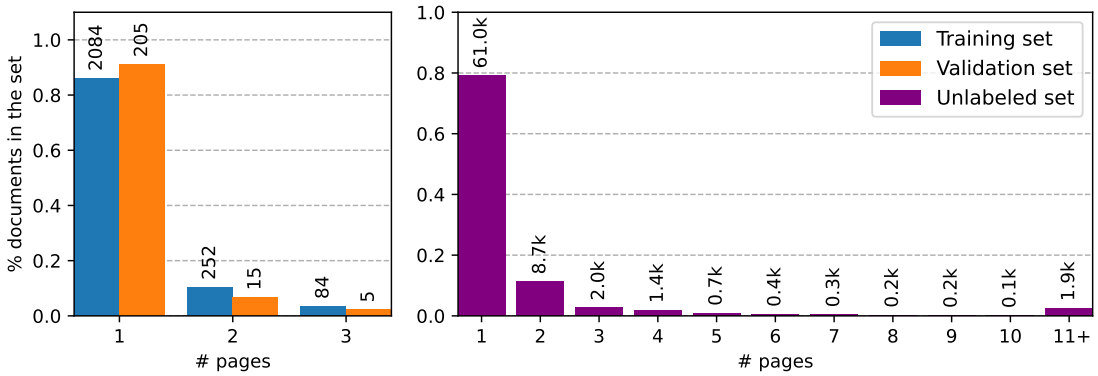


Figure 11: Distribution of the number of document pages in the training, validation and unlabeled sets for the *UCSF* source type subset. The numbers of documents are displayed above the bars.

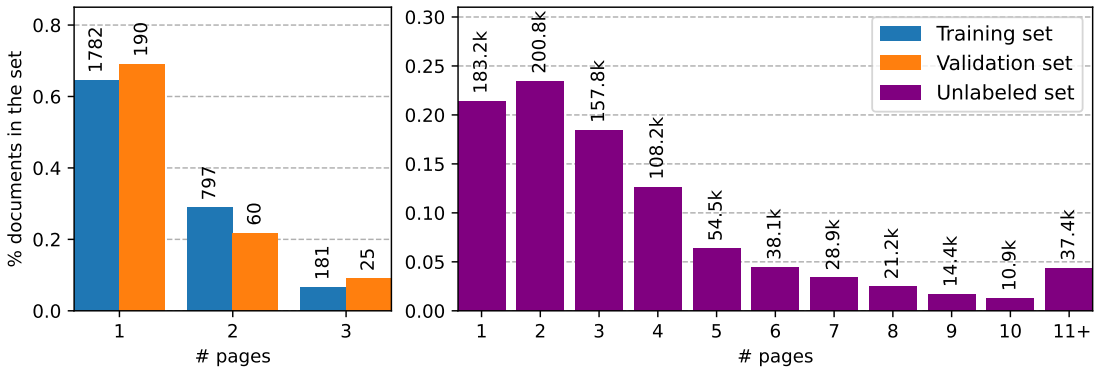


Figure 12: Distribution of the number of document pages in the training, validation and unlabeled sets for the *PIF* source type subset. The numbers of documents are displayed above the bars.

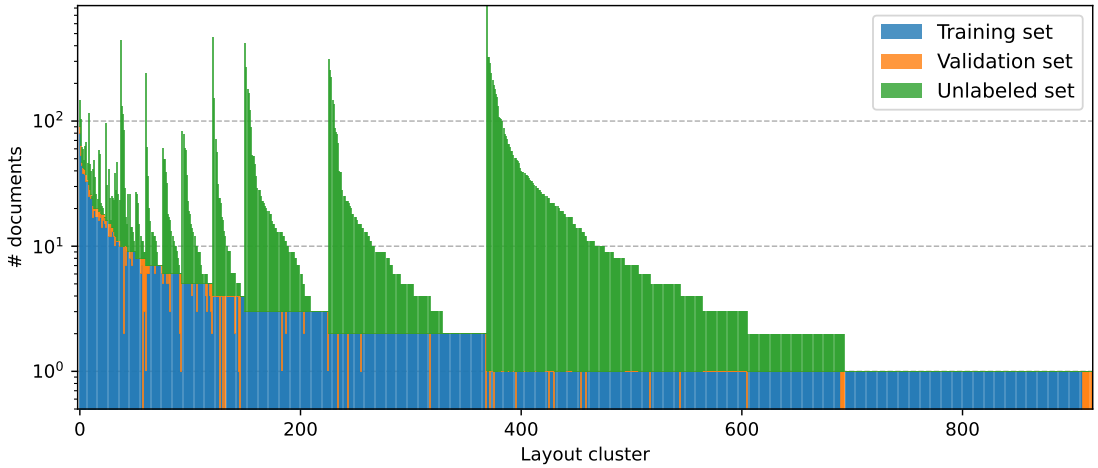


Figure 13: The number of documents of each layout cluster in the training, validation and unlabeled subsets with *UCSF* document source type, on a logarithmic scale. Most of the clusters have less than 10 documents.

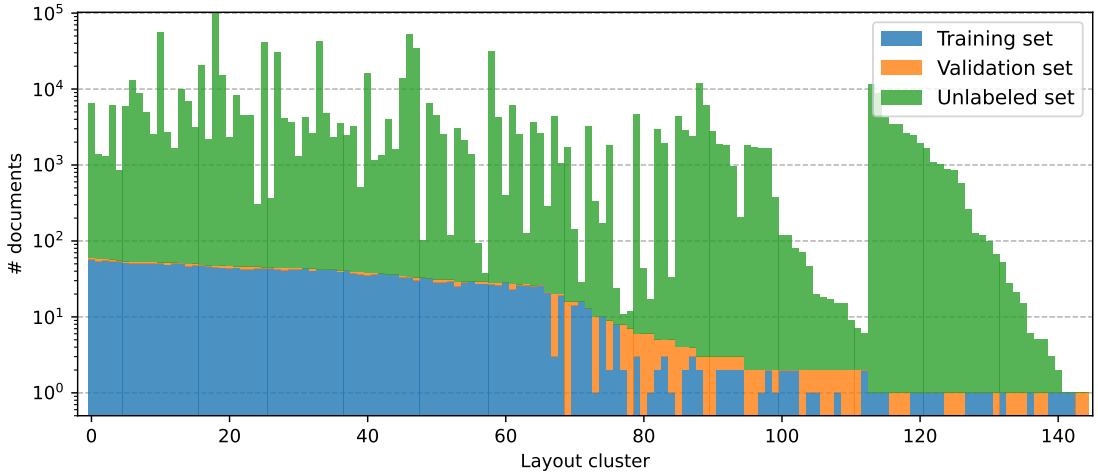


Figure 14: The number of documents of each layout cluster in the training, validation and unlabeled subsets with *PIF* document source type, on a logarithmic scale. Some clusters have up to 100k documents.

This is not the case for LIR, maybe because here AP was not the main evaluation metric and so less focus might have been given to assigning a correct score to the predictions in this case.

Since there is a noticeable difference between the behaviour of the AP and F1 metrics, benchmark submissions are allowed to mark some fields with the flag `use_only_for_ap` to include it only for the AP computation, while excluding it from the F1 computation. Unfortunately, no submission have utilized this feature so we cannot evaluate its effect.

5.6. Potential gain from hand-crafted post-processing rules

In the GraphDoc [33] submission the authors use multiple post-processing rules to mitigate some common errors. Two of these rules deal with the problem that granularity of the input OCR words is not always good enough for some specific field types:

($) When a word box has a predicted field type `currency_code_amount_due` and contains the symbol `'$'`, return just the value `'$'` and split the bounding box.
(#) For field types with the suffix `_id`, when the predicted text starts with the symbol `'#'`, remove this symbol and split the bounding box.

In this section, we explore what is the potential impact of such heuristic rules and verify whether we see a gap between GraphDoc [33] and the other methods on the impacted field types. In the following analysis, we will consider a method that uses the OCR provided with the dataset and that creates final fields by taking a union of bounding boxes of several of the input OCR word (their snapped version). Although this is not true for YOLOv8 [38] that predicts bounding boxes directly, the other methods more or less follow these criteria.

First, let us generalize the two rules above. We say a field is a *split field* if there is no word token that matches the field location, i.e., that covers the same set of Pseudo-Character-Centers (PCCs) as defined in Figure 2, and if there exists a word token that covers a superset of the PCCs covered by the field. The number of split fields for each field type in the training and validation sets are listed in Tables 1 and 2. For KILE, a total of 7.5 % of fields are split fields, while for LIR it is 3.3 % of all fields. Since handling these cases usually affects both precision and recall, it has the potential to improve the final metrics by several percentage points.

Now let us focus specifically on the rules ($) and (#) listed above. We say a split field follows the rule ($) if its text is equal to just the symbol `'$'` and it is covered by a word that contains this symbol in its text. In the training and validation set, the affected field types are only `currency_code_amount_due` and `line_item_currency` as shown in Table 3. This represents 4.9 % of all KILE fields and less than 0.1 % of all LIR fields.

We say a split field follows the rule (#) if there is a word covering this field that has exactly the same text with an additional symbol `'#'` prepended at the beginning. The number of split fields satisfying the rule (#) for each affected field type is listed in Table 4. In total, this represents 0.5 % of all KILE fields and less than 0.1 % of all LIR fields and as noticed by GraphDoc [33], most of these fields have a type with the suffix `_id`.

Let us now verify whether we see the impact of the GraphDoc [33] post-processing rules on the test set predictions. In Figure 15 we compare all of the methods on the whole test subset, on the `currency_code_amount_due` field type (only for KILE) and on field types with the

Table 1
List of all split fields in the training and validation sets for the KILE task.

field type	split fields
currency_code_amount_due	89.8 % (3593/4000)
vendor_tax_id	26.6 % (202/760)
bank_num	21.0 % (22/105)
tax_detail_rate	20.3 % (15/74)
vendor_registration_id	13.5 % (7/52)
bic	12.9 % (4/31)
account_num	8.9 % (12/135)
customer_tax_id	7.5 % (3/40)
customer_id	7.4 % (155/2108)
customer_order_id	7.3 % (46/626)
order_id	7.2 % (244/3374)
document_id	6.5 % (401/6141)
vendor_order_id	4.3 % (10/233)
date_issue	4.0 % (251/6214)
date_due	3.3 % (29/884)
amount_paid	3.2 % (14/432)
payment_reference	3.2 % (6/187)
vendor_email	2.8 % (18/648)
tax_detail_gross	1.8 % (10/542)
amount_due	1.8 % (113/6125)
tax_detail_net	1.7 % (9/519)
amount_total_gross	1.7 % (102/5966)
amount_total_tax	1.7 % (11/654)
tax_detail_tax	1.7 % (10/601)
amount_total_net	1.4 % (12/838)
payment_terms	0.7 % (15/2295)
vendor_name	0.4 % (26/7354)
customer_billing_name	0.1 % (9/6142)
customer_other_name	0.1 % (1/1463)
vendor_address	0.0 % (2/6634)
micro accuracy	7.5 % (5352/71513)

suffix _id. As expected, we see that rule ($) gives GraphDoc a big edge over most of the other methods. Exceptions are YOLOv8 [38], which does not have the same limitations connected to the OCR input, and ViBERTGrid, which also demonstrates decent performance on this field type, but the reasons behind its success are unknown to us, as we have not received a paper describing this method in more detail. For (#) we do not see GraphDoc outperforming the other methods (when compared to the results on all field types) on either of the two tasks, which matches the observations from the analysis on the training and validation set above.

From these results it is apparent that the small trick of extracting just the symbol '$' out of the word boxes predicted to have the class currency_code_amount_due, pushed the GraphDoc [33] results on KILE several percentage points up compared to most of the other methods.

Table 2
List of all split fields in the training and validation sets for the LIR task.

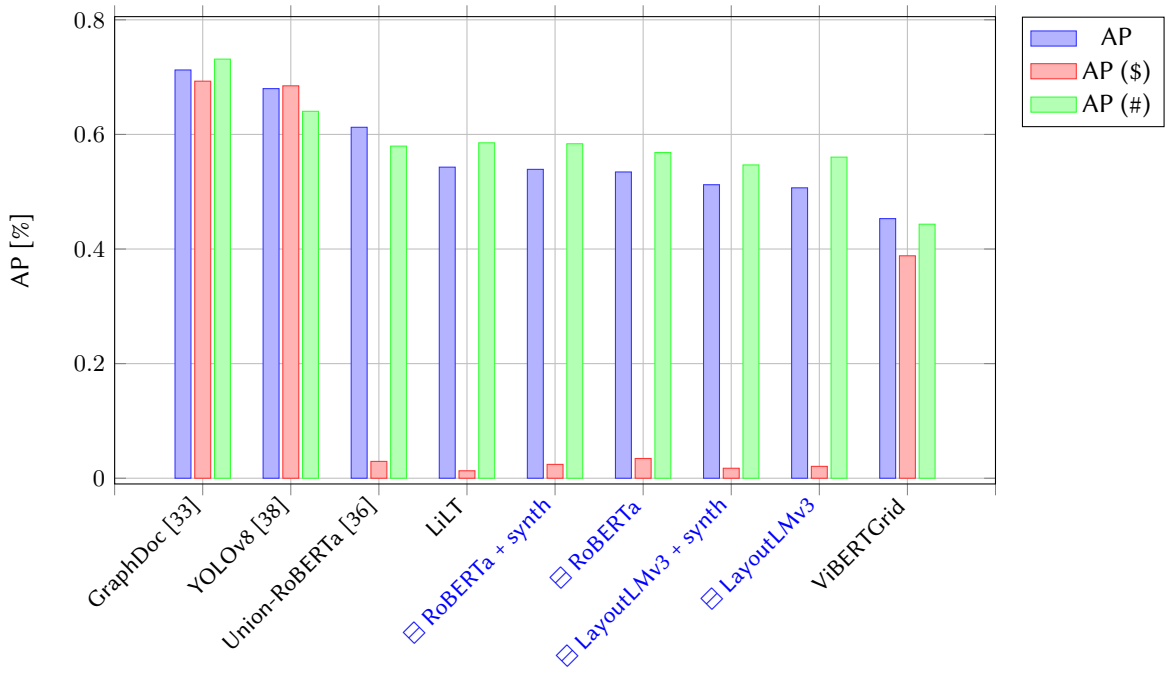
field type	split fields
line_item_order_id	24.4 % (69/283)
line_item_units_of_measure	20.4 % (312/1533)
line_item_position	8.9 % (905/10189)
line_item_date	5.7 % (1842/32405)
line_item_currency	5.2 % (62/1196)
line_item_weight	4.4 % (4/91)
line_item_discount_amount	3.5 % (2/57)
line_item_unit_price_gross	3.3 % (636/19324)
line_item_code	2.3 % (204/8904)
line_item_unit_price_net	2.2 % (67/3009)
line_item_amount_net	2.1 % (90/4235)
line_item_quantity	1.6 % (359/22993)
line_item_amount_gross	1.5 % (345/23734)
line_item_description	1.0 % (273/28617)
line_item_person_name	0.2 % (1/496)
micro accuracy	3.3 % (5171/157469)

Table 3
List of all split fields in the training and validation sets satisfying rule ($).

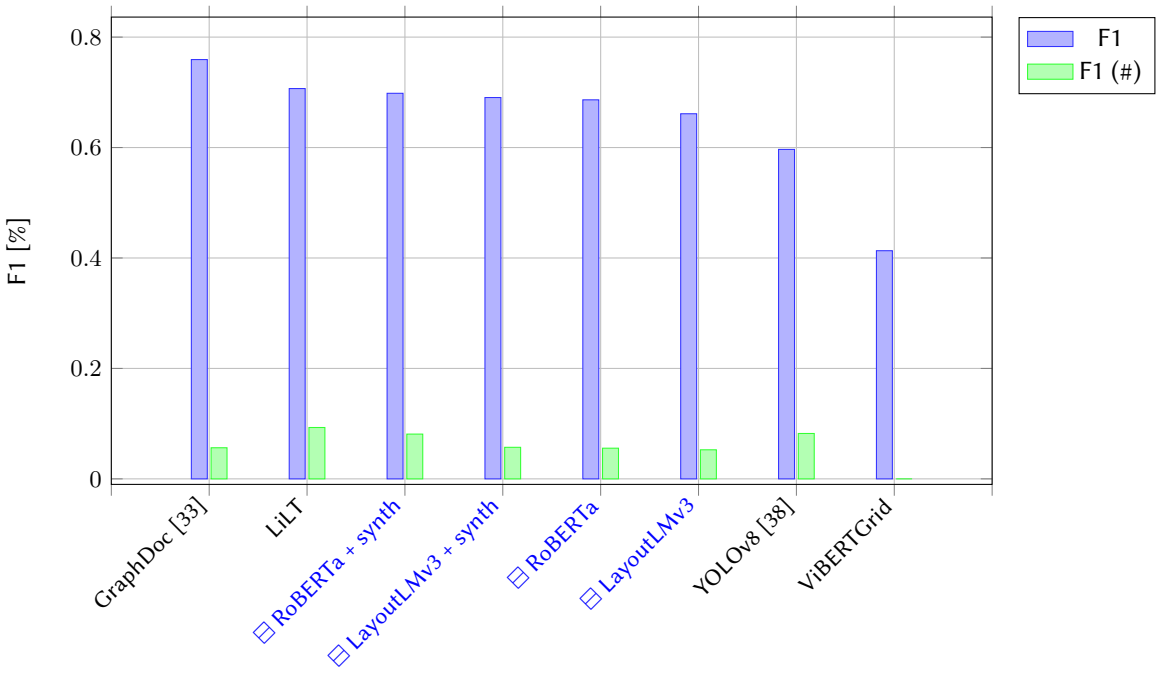
task	field type	split fields
KILE	currency_code_amount_due	86.8 % (3470/4000)
LIR	line_item_currency	5.1 % (61/1196)
KILE	micro accuracy	4.9 % (3470/71513)
LIR	micro accuracy	0.0 % (61/157469)

Table 4
List of all split fields in the training and validation sets satisfying rule (#).

task	field type	split fields (#)
KILE	vendor_tax_id	10.9 % (83/760)
KILE	vendor_registration_id	5.8 % (3/52)
KILE	bank_num	5.7 % (6/105)
KILE	customer_tax_id	5.0 % (2/40)
LIR	line_item_order_id	3.2 % (9/283)
KILE	document_id	2.6 % (159/6141)
KILE	account_num	1.5 % (2/135)
KILE	customer_id	1.4 % (29/2108)
KILE	order_id	1.4 % (46/3374)
KILE	customer_order_id	1.3 % (8/626)
KILE	payment_reference	1.1 % (2/187)
LIR	line_item_position	0.0 % (2/10189)
KILE	micro accuracy	0.5 % (340/71513)
LIR	micro accuracy	0.0 % (11/157469)



(a) KILE evaluated with AP on different subsets of field types. AP: all field types, AP ($): field type `currency_code_amount_due`, AP (#): all KILE field types with suffix `_id`.



(b) LIR evaluated with F1 on different subsets of field types. F1: all field types, F1 (#): field type `line_item_order_id`.

Figure 15: Evaluation of the methods on field types affected by the GraphDoc splitting heuristics for Task 1: KILE (15a) and Task 2: LIR (15b).

6. Conclusion

We presented the first edition of the DocILE 2023 competition, which consisted of two tracks: KILE and LIR. Both tasks consist of detection of pre-defined categories of information in business documents. The latter task additionally requires groupping the information into tuples. In the end, we obtained 5 submissions for KILE and 4 submissions for LIR. The diversity of the chosen approaches shows the potential of the DocILE dataset and benchmark, which spans the domains of computer vision, layout analysis, and natural language processing. Unsurprisingly, some of the submissions used a multi-modal approach. The values of the respective error metrics indicate that the benchmark is non-trivial and the problems are far from being solved.

The benchmark remains open to new submissions, leaving it as a springboard for future research and for the document understanding community. To point out just a few possible research questions for this benchmark: 1) How to best use the unlabeled and synthetic datasets (as most of the solutions did not focus on these parts of the dataset)? 2) Is it possible to better utilize the fact that many documents share the same layout and push the performance on the few-shot subset closer to the performance on the many-shot subset? 3) Which parts of the tasks are better solved by pure NLP solutions (such as the baselines), which are better solved by pure CV solutions (such as YOLOv8) and do the multi-modal solutions (such as GraphDoc) already utilize both of the modalities to their full potential or is one of the modalities still under-utilized?

References

[1] Š. Šimsa, M. Uřičář, 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] W. Lin, Q. Gao, L. Sun, Z. Zhong, K. Hu, Q. Ren, Q. Huo, Vibertgrid: a jointly trained multi-modal 2d document representation for key information extraction from documents, in: ICDAR, 2021.

[3] A. R. Katti, C. Reisswig, C. Guder, S. Brarda, S. Bickel, J. Höhne, J. B. Faddoul, Chargrid: Towards understanding 2d documents, in: EMNLP, 2018.

[4] Y. Xu, M. Li, L. Cui, S. Huang, F. Wei, M. Zhou, Layoutlm: Pre-training of text and layout for document image understanding, in: KDD, 2020.

[5] Y. Xu, Y. Xu, T. Lv, L. Cui, F. Wei, G. Wang, Y. Lu, D. Florencio, C. Zhang, W. Che, et al., Layoutlmv2: Multi-modal pre-training for visually-rich document understanding, ACL (2021).

[6] Y. Huang, T. Lv, L. Cui, Y. Lu, F. Wei, Layoutlmv3: Pre-training for document ai with unified text and image masking, in: ACM-MM, 2022.

[7] T. Hong, D. Kim, M. Ji, W. Hwang, D. Nam, S. Park, Bros: A pre-trained language model

focusing on text and layout for better key information extraction from documents, in: AAAI, 2022.
[8] R. Tanaka, K. Nishida, S. Yoshida, Visualmrc: Machine reading comprehension on document images, in: AAAI, 2021.
[9] R. Powalski, Ł. Borchmann, D. Jurkiewicz, T. Dwojak, M. Pietruszka, G. Pałka, Going full-tilt boogie on document understanding with text-image-layout transformer, in: ICDAR, 2021.
[10] Z. Tang, Z. Yang, G. Wang, Y. Fang, Y. Liu, C. Zhu, M. Zeng, C. Zhang, M. Bansal, Unifying Vision, Text, and Layout for Universal Document Processing, arXiv (2022).
[11] Z. Zhang, J. Ma, J. Du, L. Wang, J. Zhang, Multimodal pre-training based on graph attention network for document understanding, IEEE Transactions on Multimedia (2022).
[12] B. Davis, B. Morse, S. Cohen, B. Price, C. Tensmeyer, Deep visual template-free form parsing, in: ICDAR, 2019.
[13] M. Hammami, P. Héroux, S. Adam, V. P. d'Andecy, One-shot field spotting on colored forms using subgraph isomorphism, in: ICDAR, 2015.
[14] J. Zhou, H. Yu, C. Xie, H. Cai, L. Jiang, irmp: From printed forms to relational data model, in: HPCC, 2016.
[15] Z. Huang, K. Chen, J. He, X. Bai, D. Karatzas, S. Lu, C. V. Jawahar, ICDAR2019 competition on scanned receipt OCR and information extraction, in: ICDAR, 2019.
[16] J. Herzig, P. K. Nowak, T. Müller, F. Piccinno, J. M. Eisenschlos, Tapas: Weakly supervised table parsing via pre-training, arXiv (2020).
[17] S. Schreiber, S. Agne, I. Wolf, A. Dengel, S. Ahmed, Deepdesrt: Deep learning for detection and structure recognition of tables in document images, in: ICDAR, 2017.
[18] X. Zhong, J. Tang, A. Jimeno-Yepes, Publaynet: Largest dataset ever for document layout analysis, in: ICDAR, 2019.
[19] D. Lohani, A. Belaïd, Y. Belaïd, An invoice reading system using a graph convolutional network, in: ACCV workshops, 2018.
[20] B. P. Majumder, N. Potti, S. Tata, J. B. Wendt, Q. Zhao, M. Najork, Representation learning for information extraction from form-like documents, in: ACL, 2020.
[21] P. Riba, A. Dutta, L. Goldmann, A. Fornés, O. Ramos, J. Lladós, Table detection in invoice documents by graph neural networks, in: ICDAR, 2019.
[22] M. Mathew, D. Karatzas, C. Jawahar, DocVQA: A dataset for vqa on document images, in: WACV, 2021.
[23] M. Mathew, V. Bagal, R. Tito, D. Karatzas, E. Valveny, C. Jawahar, InfographicVQA, in: WACV, 2022.
[24] Š. 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, 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.
[25] M. Skalický, Š. Šimsa, M. Uřičář, M. Šulc, Business document information extraction: Towards practical benchmarks, in: CLEF, 2022.
[26] Web, Industry Documents Library, https://www.industrydocuments.ucsf.edu/, ????. Accessed: 2022-10-20.

[27] Web, Public Inspection Files, https://publicfiles.fcc.gov/, ????. Accessed: 2022-10-20.
[28] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, IJCV (2015).
[29] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, S. Fidler, Aligning books and movies: Towards story-like visual explanations by watching movies and reading books, in: ICCV, 2015.
[30] Mindee, docTR: Document Text Recognition, https://github.com/mindee/doctr, 2021.
[31] K. Olejniczak, M. Šulc, Text Detection Forgot About Document OCR, in: CVWW, 2023.
[32] Š. Šimsa, M. Šulc, M. Skalický, Y. Patel, A. Hamdi, DocILE 2023 Teaser: Document Information Localization and Extraction, in: J. Kamps, L. Goeuriot, F. Crestani, M. Maistro, H. Joho, B. Davis, C. Gurrin, U. Kruschwitz, A. Caputo (Eds.), Advances in Information Retrieval - 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2-6, 2023, Proceedings, Part III, volume 13982 of *Lecture Notes in Computer Science*, Springer, 2023, pp. 600–608. doi:10.1007/978-3-031-28241-6_69.
[33] Y. Wang, J. Du, J. Ma, P. Hu, Z. Zhang, J. Zhang, USTC-iFLYTEK at DocILE: a Multi-modal approach using Domain-specific GraphDoc, in: M. Aliannejadi, G. Faggioli, N. Ferro, M. Vlachos (Eds.), Working Notes of CLEF 2023 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 18th - to - 21st, CEUR Workshop Proceedings, CEUR-WS.org, 2023.
[34] J. Wang, L. Jin, K. Ding, Lilt: A simple yet effective language-independent layout transformer for structured document understanding, ACL (2022).
[35] D. Lewis, G. Agam, S. Argamon, O. Frieder, D. Grossman, J. Heard, Building a test collection for complex document information processing, in: SIGIR, 2006.
[36] B. G. Tran, D.-N. M. Bao, K. G. Bui, H. V. Duong, D. H. Nguyen, H. M. Nguyen, Union-RoBERTa: RoBERTas Ensemble Technique for Competition on Document Information Localization and Extraction, in: M. Aliannejadi, G. Faggioli, N. Ferro, M. Vlachos (Eds.), Working Notes of CLEF 2023 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 18th - to - 21st, CEUR Workshop Proceedings, CEUR-WS.org, 2023.
[37] W. Lin, Q. Gao, L. Sun, Z. Zhong, K. Hu, Q. Ren, Q. Huo, ViBERTgrid: A Jointly Trained Multi-modal 2D Document Representation for Key Information Extraction from Documents, in: J. Lladós, D. Lopresti, S. Uchida (Eds.), 16th International Conference on Document Analysis and Recognition, ICDAR 2021, Lausanne, Switzerland, September 5-10, 2021, Proceedings, Part I, volume 12821 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 548–563. URL: https://doi.org/10.1007/978-3-030-86549-8_35. doi:10.1007/978-3-030-86549-8_35.
[38] J. Straka, I. Gruber, Object Detection Pipeline Using YOLOv8 for Document Information Extraction, in: M. Aliannejadi, G. Faggioli, N. Ferro, M. Vlachos (Eds.), Working Notes of CLEF 2023 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 18th - to - 21st, CEUR Workshop Proceedings, CEUR-WS.org, 2023.
[39] G. Jocher, A. Chaurasia, J. Qiu, YOLO by Ultralytics, 2023. URL: https://github.com/ultralytics/ultralytics.
[40] A. R. Katti, C. Reisswig, C. Guder, S. Brarda, S. Bickel, J. Höhne, J. B. Faddoul, CharGrid: Towards Understanding 2D Documents, in: E. Riloff, D. Chiang, J. Hockenmaier, J. Tsujii

(Eds.), Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018, Association for Computational Linguistics, 2018, pp. 4459–4469. URL: https://aclanthology.org/D18-1476/.