Data-Efficient Information Extraction from Documents with Pre-Trained Language Models

No author

May 31, 2021

Abstract

Like for many text understanding and generation tasks, pre-trained languages models have emerged as a powerful approach for extracting information from business documents. However, their performance has not been properly studied in data-constrained settings which are often encountered in industrial applications. In this paper, we show that LayoutLM, a pre-trained model recently proposed for encoding 2D documents, reveals a high sample-efficiency when fine-tuned on public and real-world Information Extraction (IE) datasets. Indeed, LayoutLM reaches more than 80% of its full performance with as few as 32 documents for fine-tuning. When compared with a strong baseline learning IE from scratch, the pre-trained model needs between 4 to 30 times fewer annotated documents in the toughest data conditions. Finally, LayoutLM performs better on the real-world dataset when having been beforehand fine-tuned on the full public dataset, thus indicating valuable knowledge transfer abilities. We therefore advocate the use of pre-trained language models for tackling practical extraction problems.

Keywords: Pre-training, Language models, Business documents, Information extraction, Document Understanding, Document Intelligence, Few-shot learning, Intermediate learning

1 Introduction

Business documents are files that describe all the internal and external transactions occurring in a company. Such documents cover a wide variety of types, including invoices, purchase orders, receipts, vendor contracts, financial reports and employment agreements. To cope with the increasing volume of business documents to process, academic and industrial practitioners have leveraged AI techniques to automatically read, understand and interpret them [24]. This research topic, recently referred to as Document Intelligence (DI), comprises multiple disciplines ranging from Natural Language Processing, Computer Vision over Information Retrieval to Knowledge Representation and Reasoning among others.

Nowadays, business documents are still often distributed in non-machine-readable formats such as images of scanned documents or PDFs filled with unstructured data [6]. One crucial task in Document Intelligence is thus to parse the text of these documents to retrieve valuable semantic information. It may be extracting the value of fields that repeatedly appear in the documents, e.g. the total amount in restaurant receipts [16] or analyzing the structure of forms by identifying all their key-value pairs [17]. To tackle the diversity and complexity of document structure and content, current Information Extraction (IE) approaches employ deep neural networks that learns from annotated documents. Yet, as for many tasks in DI, labeling documents is a challenge in IE since it involves significant human expertise in the targeted application domain [24]. Besides, the extraction objectives are highly specific to the type of documents to process, hindering the reusability of a trained IE model. In [26, 32], the authors obtain high-quality annotations from the end users of commercialized document automation software but those users expect to rapidly leverage the benefits of automated IE. Therefore, DI practitioners usually seek to minimize the amount of supervision required to design performing automation tools, especially knowing the wide spectrum of document types that a company may receive or emit.

Following the current trend in the NLP field, a number of works [35, 28, 36, 14] have proposed language models that are pre-trained on large collections of documents and then fine-tuned and evaluated on several document analysis tasks such as information extraction but also document-level classification and visual question answering. Their pre-trained models have considerably outperformed the previous state-of-the art models that were trained from scratch, whether they are evaluated on benchmarks with large-scale [13] or relatively restrained [17, 16, 27] annotated sets for training. However, this comparison has not been conducted in even more data-constrained settings that are encountered in practical applications of IE models. In this paper, we aim to quantify to what extent the pre-trained models are sample-efficient for IE tasks by comparing LayoutLM [35] — a pre-trained language model recently proposed for encoding 2D documents — with two models without pre-training. We present three main findings that we experimentally validated using the public SROIE benchmark [16] as well as a private real-world dataset:

- The pre-trained LayoutLM exhibits remarkable few-shot learning capabilities for IE, reaching more than 80% of its full performance with as few as 32 documents for fine-tuning.
- This model is significantly more data-efficient than a strong non-pretrained baseline in the lowest data regimes, hitting the same levels of extraction performance with around 30 times fewer samples for the real-world dataset.
- Finally, the pre-trained model displays helpful knowledge transfer between IE tasks since learning beforehand to extract information on the full SROIE dataset improves the performance of up to 10 % when fine-tuning the model on the private dataset.

Corroborating the data efficiency of such models already observed in other NLP tasks [15, 4, 2], our results show that using pre-trained models dramatically reduces the amount of annotations required for achieving satisfying performance which is appreciable for industrial IE systems.

2 Related works on Information Extraction (IE)

2.0.1 Fully supervised models

Historically tackled by rule-based approaches [3, 21], the IE task has lately been dominated by machine learning based solutions [5]. Most ML approaches first employ an encoder, usually a few neural network layers, to obtain contextualized high-level representations of all the tokens of the document. Then, a decoder module composed of a couple of dense layers is immediately applied to these representations to classify each token according to the type of information that it carries. Most works adopting this sequence labeling approach for extracting information have focused on constituting more powerful representations of the document tokens. The first encoders to appear were recurrent neural networks [26, 33] that operate on an uni-dimensional arrangement of tokens. Later, encoders that explicitly consider the two dimensional structure of business documents have been proposed, thus leveraging physical layout information. These methods either represent a document as a graph of tokens [22, 29, 37, 10] or a regularly shaped grid on which the tokens are embedded [18, 8, 39, 7]. Some convolutional layers are then applied to these models of document to obtain the token representations. In addition to better understanding the document layout, some authors [18, 25] also include the pixel values of the document images in the input for capturing clues not conveyed by the text modality such as table ruling lines, logos and stamps.

In all these extraction models, the whole set of their parameters, except perhaps the token embeddings [8], are learned in a fully supervised task-specific way. Specifically, they are attributed random values at the beginning of the model training. The parameters values are then updated by directly minimizing the cross-entropy loss on the target IE dataset. While being successful for most IE tasks, this results in a costly process since a massive amount of weights need to be learned from scratch.

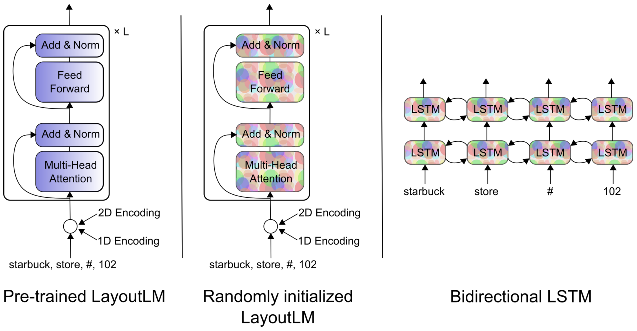


Figure 1: The different architectures used in our experiments for encoding documents. From left to right: Transformer-based LayoutLM [35] with pre-trained weights, LayoutLM with random initialization and a 2-layer bidirectional LSTM also randomly initializated.

2.0.2 Pre-trained models

Since the recent development of language modeling techniques [9, 2], NLP models for understanding and generating text are not learned from scratch anymore [30]. Rather, the mainstream approach to reach state-of-the-art performance on many downstream tasks is to adapt the parameters of models that have already learned powerful representations of the language. Such pre-training is performed in a self-supervised way on a large quantity of text data. Starting from LayoutLM [35], pre-trained models that were originally operating on serialized text have been extended to process the spatially distributed text contained in business documents, e.g. text blocks and tables. To that end, positional embedding vectors relative to their absolute 2D coordinates are included into the token representations that are given to the Transformer encoder. Before fine-tuning the model on the downstream tasks like the fully supervised models, LayoutLM is first pre-trained on millions of document pages [20] using a self-supervised Masked Visual-Language Modeling (MVLM) task that naturally expands the main pre-training objective of BERT [9].

This work further inspires other language models dedicated to two dimensional documents. While the image modality was introduced only at the fine-tuning stage in LayoutLM, later models [28, 14, 36] include visual descriptors from convolutional layers directly into the token representations used for pre-training. These recent works mainly focus on adding new pre-training objectives complementing MVLM to more effectively mix the text, layout and image modalities when learning the document representations, for example the topic-modeling and document shuffling tasks of [28], the Sequence Positional Relationship Classification (SPRC) objective [34], the text-image alignment and matching tasks leveraged in [36] and the 2D area-masking strategy from [14]. Moreover, [36, 14] both modify the computation of the self-attention scores to better encompass the relative positional relationships among the tokens of the document. Finally, [28] has resorted to page index embeddings and the Longformer's [1] self-attention that scales linearly with the sequence length in order to process multi-page and longer documents.

All these pre-trained models largely surpass fully supervised models and have established state-of-the-art performance on multiple document understanding benchmarks, including common information extraction datasets [17, 16, 27]. Yet, all the experiments have been performed with the full training set of the downstream tasks for fine-tuning, thus not studying the potential of pre-trained models to learn IE with few annotated data compared to models without such pre-training. Our contribution consists here in showing how pre-trained models can lead to a performance gain on low-resource downstream IE tasks.

3 Models

In our experiments, we follow the sequence labeling approach for performing IE. The evaluated models are composed of an encoder delivering contextualized representations of the tokens and a linear classifier that decodes this sequence of representations to extract information. All models only differ by their encoder.

3.1 Encoder

As shown in Figure 1, we use three different networks for encoding the business documents. We compare a pre-trained encoder with two fully supervised encoders.

3.1.1 Pre-trained model

As pre-trained model, we use LayoutLM from [35] since this is the only IE work that publicly releases their pre-trained model parameters. We use its base-uncased version[1] which consists of a 12-layer Transformer with a hidden size of 768 and 12 attention heads per layer, resulting in 113 millions weights. It is built upon the BERT base-uncased model with 4 additional embedding vectors to represent the position of each token in the document page. This 2D positional encoding, coupled with a pre-training task that strongly binds the token's semantic representation with their surrounding, allows LayoutLM to take advantage of the structure of the documents. Although proposed in their paper for the fine-tuning stage, we do not leverage the image modality since it brings marginal improvements for IE. We thus solely rely on the text and its layout for constructing token embeddings. We refer the reader to their paper for more details about its architecture and pre-training stage.

3.1.2 Fully supervised models

For fully supervised models, we use 2 encoders that are trained from scratch on the IE tasks. First, we reuse the LayoutLM model but we discard pre-training and randomly initialize all its parameters. However, as

[1] `https://github.com/microsoft/unilm/tree/master/layoutlm`

confirmed by our early experiments, this encoder version performs poorly in low-resource settings due to its massive amount of parameters to learn from scratch. Secondly, we propose a smaller fully supervised baseline that has shown success in past IE works [26, 33]. This is a 2-layer bidirectional LSTM network (BLSTM) with a 128 hidden size. We reuse the same sub-word tokenizer as LayoutLM and employ only textual embeddings for tokens. The resulting model contains 8.5 millions parameters.

Following standard practises, Transformer and embedding layers are respectively initialized with a truncated normal and Gaussian distributions. BLSTM layers resort to Glorot initialization [12].

3.2 Decoder

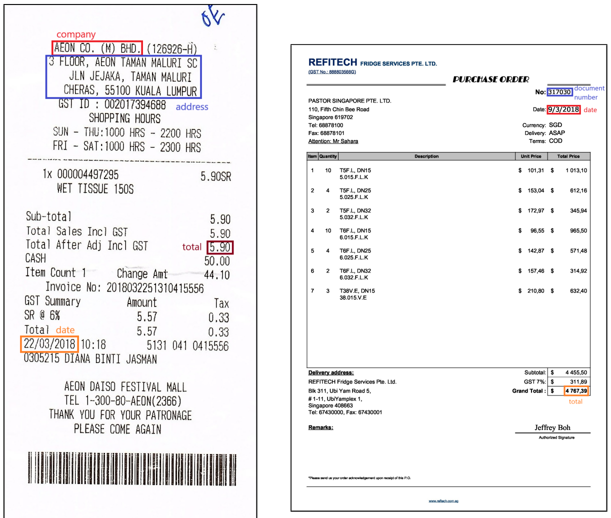
On top of each of these 3 encoders, we add a dense softmax layer to predict the information type carried by each document token. Since the fields to extract can be spread over multiple tokens, the `BIESO` labeling scheme [31] is utilized to denote the beginning (`B`), continuation (`I`) and end (`E`) of a field value while `S` classes stand for single token values. This results in 4 output classes per field, with the additional class `O` for tokens not conveying any relevant information. At inference time, we determine the class of a token by getting its highest probability and reduce the resulting list of `BIESO` classes to obtain the field level predictions. If a document has more than 512 tokens, its text is split in multiple sequences that are independently processed by the extraction model.

4 Datasets

As illustrated in Figure 2, we consider two IE datasets that cover different document types and extraction objectives.

4.1 Scanned Receipts OCR and Information Extraction (SROIE)

We train and evaluate the models on the public SROIE dataset [16] containing restaurant receipts. We only consider its information extraction task that aims to retrieve the name and address of the company issuing the receipt, the total amount and date. The dataset gathers 626 receipts for training and 347 receipts for test. We further randomly split the training set to constitute a validation set of 26 receipts. While not stated in [16], the document issuers are shared between the training and test sets.



AEON CO. (M) BHD. (126926-H)
3 FLOOR, AEON TAMAN MALURI SC
JLN JEJAKA, TAMAN MALURI
CHERAS, 55100 KUALA LUMPUR
GST ID : 002017394688
SHOPPING HOURS
SUN - THU:1000 HRS - 2200 HRS
FRI - SAT:1000 HRS - 2300 HRS

1x 000004497295 5.90SR
WET TISSUE 150S

Sub-total 5.90
Total Sales Incl GST 5.90
Total After Adj Incl GST 5.90
CASH 50.00
Item Count 1 Change Amt 44.10
Invoice No: 2018032251310415556
GST Summary Amount Tax
SR @ 6% 5.57 0.33
Total 5.57 0.33
22/03/2018 10:18 5131 041 0415556
0305215 DIANA BINTI JASMAN

AEON DAISO FESTIVAL MALL
TEL 1-300-80-AEON(2366)
THANK YOU FOR YOUR PATRONAGE
PLEASE COME AGAIN

(a) receipt from SROIE

REFITECH FRIDGE SERVICES PTE. LTD.
(GST No.: 8888035666G)

PURCHASE ORDER

No: 317030
Date: 9/3/2018

PASTOR SINGAPORE PTE. LTD.
110, Fifth Chin Bee Road
Singapore 619702
Tel: 68878100
Fax: 68878101
Attention: Mr Sahara

Currency: SGD
Delivery: ASAP
Terms: COD

Item	Quantity	Description	Unit Price	Total Price
1	10	T5F.L, DN15 5.015.F.L.K	$ 101,31	$ 1 013,10
2	4	T5F.L, DN25 5.025.F.L.K	$ 153,04	$ 612,16
3	2	T5F.L, DN32 5.032.F.L.K	$ 172,97	$ 345,94
4	10	T6F.L, DN15 6.015.F.L.K	$ 96,55	$ 965,50
5	4	T6F.L, DN25 6.025.F.L.K	$ 142,87	$ 571,48
6	2	T6F.L, DN32 6.032.F.L.K	$ 157,46	$ 314,92
7	3	T38V.E, DN15 38.015.V.E	$ 210,80	$ 632,40

Delivery address:
REFITECH Fridge Services Pte. Ltd.
Blk 311, Ubi Yam Road 5,
1-11, UbiYamplex 1,
Singapore 408663
Tel: 67430000, Fax: 67430001

Subtotal: $ 4 455,50
GST 7%: $ 311,89
Grand Total : $ 4 767,39

Remarks:

Jeffrey Boh
Authorized Signature

*Please send us your order acknowledgement upon receipt of this P.O.

www.refitech.com.sg

(b) purchase order from PO-51k

Figure 2: A document sample for each dataset alongside their expected field values to extract. For PO-51k, we show a fictive purchase order due to privacy reasons.

Each receipt is given the ground-truth value for the four targeted fields. The comparison with the model predictions is made in terms of exact matching of strings, leading to precision, recall and F1 score metrics[2]. For the sake of readability, we only report the F1 scores averaged over all the targeted fields. To establish the `BIESO` labels, we look for the receipt words matching the ground-truth field values. For the total amount, a value may match different sets of words, e.g. the amounts without taxes or after rounding. If so, we select the bottom most occurrence having the keyword `total` in its line.

We use the provided OCR results containing a list of text segments and their bounding boxes. As noticed by many submissions in the leaderboard including LayoutLM's authors, they contain a number of brittle text recognition errors, e.g. a comma interpreted as a dot. This highly impacts the evaluation results based on exact matching. Therefore, following previous works, we manually fix them in the test set while we perform fuzzy matching for deriving the token labels in the training set. The order of text segments being sometimes faulty, we also re-arrange them from top-to-bottom.

[2] The metric values are obtained at: `https://rrc.cvc.uab.es/?ch=13&com=evaluation&task=3`

4.2 Real-world purchase orders (PO-51k)

To prove the efficiency of the IE models, we also conduct experiments on a private dataset composed of $51,000$ English purchase orders that were processed on a commercial document automation solution. We split the dataset in 40k, 1k and 10k documents for training, validation and test sets. Unlike SROIE, these three subsets contain different document issuers, respectively 6200, 870 and 1700 issuers. This induces that for a large portion of the test set, the layout and content organization of documents have not been seen at training time.

We aim to extract 3 different fields among these purchase orders: the document number, the date and the total amount. The ground truth for these fields is directly provided by the end-users of the automation software, ensuring high-quality annotations. We employ the same methodology as in SROIE for evaluating the models. Text of documents is retrieved thanks to a commercial OCR engine.

Since LayoutLM is not designed for handling multipage documents, we only consider the first page of documents. Because of this limitation, there may be no value to predict for a target field. In practice, roughly 25% of the documents miss a total amount on the first page while only 10% of the documents are affected for the two other fields.

5 Experiments

5.1 Experiment settings

We use the following settings in all our experiments. To evaluate data efficiency, we restrict the training set to 8, 16, 32, 64, 128, 256 and 600 randomly selected documents for both datasets. For PO-51k, we additionally study the extraction performance when training with 2k, 8k and 40k samples. We repeat each experiment 5 times, each time with different random seeds and thus different selected training documents. We plot the average μ of the 5 F1 scores as well as the shaded region $[\mu - \sigma, \mu + \sigma]$ for representing the standard deviation σ. We use a log scale over the number of training documents to better visualize the lowest-resource regimes.

As in [35], we use the Adam optimizer with an initial learning rate of 5e-5, linearly decreasing it to 0 as we reach the maximum number of training steps. For the BLSTM model, we employ a higher initial learning rate of 5e-3 since the former value was not giving a good convergence. For each run, we set the maximum number of training steps to 1k for the pre-trained LayoutLM and 2k for models without pre-training. We proceed to early stopping on the validation set to choose the model checkpoint to evaluate or use for a further training run. We employ a batch size of 8 for all runs in SROIE. For PO-51k, we set the batch size to 16 for all runs, except for 8 and 40k training docs where we fix it to respectively 8 and 32 in order to see at least once each training document. Following the results of language models fine-tuning in low-resource settings [15], we update the entire model in all runs.

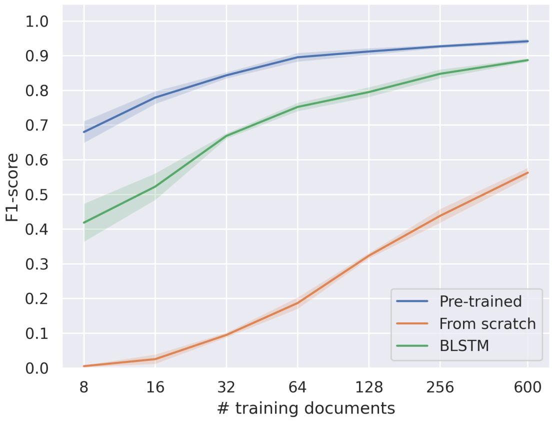


Figure 3: Few-shot extraction performance on the SROIE [16] test set for the pre-trained LayoutLM [35] against its randomly initialized version and a BLSTM network.

All training runs are performed on a single 12 Go TITAN XP GPU. We have released the code for reproducing the experiments on the SROIE dataset[3].

5.2 Few-shot learning

For both datasets, we first study the performance when the models independently learn the IE task from a few annotated samples. After initializing them from scratch or from pre-trained weights, we fine-tune the models for variable numbers of training documents. We report below their results on the whole test set.

5.2.1 SROIE

We show F1 scores for the SROIE dataset in the Figure 3. We first notice that we get to an average F1 score of 0.9417 when the pre-trained LayoutLM is fine-tuned on 600 receipts. This is in accordance with the

[3] `https://github.com/clemsage/unilm`

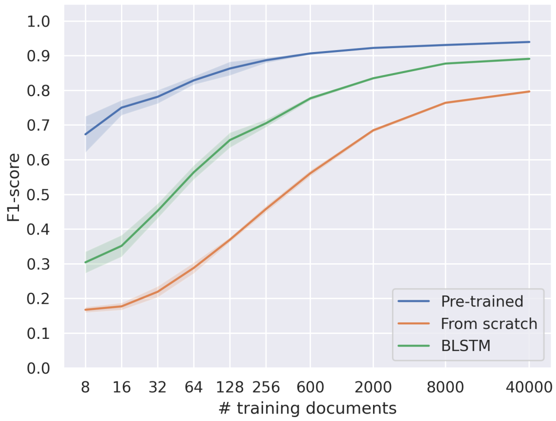


Figure 4: Few-shot extraction performance on the PO-51K test set for the pre-trained LayoutLM [35] against its randomly initialized version and a BLSTM network.

0.9438 F1 score reported in its paper [35] when considering the 626 documents of the original training set. The model convergence is really fast, hitting 90% of its full performance with only 32 documents, i.e. a 18 times smaller training set.

Unsurprisingly, we observe that the pre-trained LayoutLM achieves significantly better performance than fully supervised models whatever the number of training documents. Yet, the fewer training documents we make use of, the larger is the difference of F1 score between these two classes of models. For instance, even if the BLSTM network reaches a near similar level of performance with 600 documents (0.8874 against 0.9417), it performs significantly worse than LayoutLM in more data-constrained regimes: the gap of F1 score attains 0.2612 for 8 training receipts. This is even more noticeable for the randomly initialized LayoutLM which completely fails to extract the fields when trained with 8 documents. When offered the full training set, the model does not even outperform its pre-trained counterpart that makes use of only 8 documents.

As expected [38], the performance variance is greater in the lowest data regimes. Yet, the pre-training effectively reduces the variance, making pre-trained models less dependent on the choice of fine-tuning

5.2.2 PO-51k

We show F1 scores for the PO-51k dataset in the Figure 4. We observe similar learning curves for all models, including the pre-trained model that hits 92% of its maximal performance with only 128 samples, i.e. 312 times fewer training documents. In the lowest data regimes, the gap between LayoutLM and the fully supervised baselines is even wider than for SROIE. Indeed, the difference with the BLSTM model is on average of 0.37 F1 score until 32 documents while it was on average of 0.23 points for SROIE. The BLSTM trained with 600 documents performs on par with LayoutLM fine-tuned on only 32 documents, i.e. a order of magnitude less annotations. We also note that this real-world dataset is notoriously more complex than SROIE since a few hundreds documents are not enough to achieve full convergence of the F1 scores. We finally underline the sample inefficiency of LayoutLM trained from scratch with a F1 score at 40k training documents that still lags behind both its pre-trained counterpart and the BLSTM.

On both datasets, we have confirmed that the pre-training stage extensively reduces the amount of annotations needed to reach specific performance for downstream IE tasks.

5.3 Intermediate learning

In these experiments, we analyze to what extent learning to extract information from given documents decreases the annotation efforts for later performing IE on another document distribution. Specifically, we first fine-tune the pre-trained LayoutLM on the SROIE task using its full training set and then transfer the resulting model on the PO-51k dataset and study its few-shot performance. This simulates an actual use case where a practitioner leverages publicly available data to later tackle IE in more challenging industrial environments.

Since the fields to extract are not identical between the SROIE and PO-51k tasks, we remove the final classifier layer on top of LayoutLM after the fine-tuning on SROIE. We replace it with a randomly initialized layer that matches the number of fields in PO-51k. Even if this imposes to learn the decoder parameters from scratch between the two IE tasks, there are only a few thousands compared to the million weights of the encoder. We therefore hope that LayoutLM can still transfer some knowledge from SROIE to PO-51k tasks.

5.3.1 SROIE to PO-51k

We compare the few-shot performance on PO-51k when having firstly fine-tuned on SROIE with the results obtained when directly employing the pre-trained LayoutLM weights. We show results of these intermediate learning experiences in Figure 5.

We note that the fine-tuning on SROIE considerably improves the extraction for few PO-51k examples with

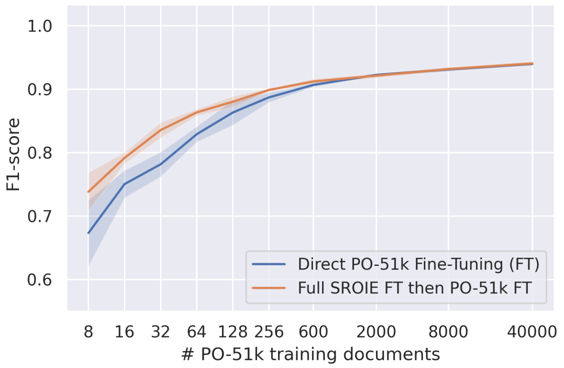


Figure 5: Test F1 scores of pre-trained LayoutLM when transferring extraction knowledge from SROIE to PO-51k tasks. The IE performance is always improved by resorting to SROIE as an intermediate task, the boost being significant with few available PO-51k documents for fine-tuning.

a boost of 0.065 (+10%) F1 score for 8 documents. For 600 examples or more, the effect of intermediate learning disappears with a performance indistinguishable from directly fine-tuning on PO-51k. Fine-tuning beforehand on the SROIE dataset also helps to reduce the variance when it is significant: between 8 to 32 PO-51k documents, the mean standard deviation decreases from 0.031 to 0.017 (-45%) when resorting to intermediate learning.

Therefore, if the amount of annotated documents at their disposal is limited, we encourage IE practitioners not to directly fine-tune the pre-trained models on their task but first use publicly available IE datasets to enhance performance.

6 Conclusion

In this paper, we showed that pre-trained language models are highly beneficial for extracting information from few annotated documents. On a public dataset as well as on a more demanding industrial application, such a pre-trained approach consistently outperformed two fully supervised models that learn from scratch the IE task. We finally demonstrated that pre-training brings additional improvements when transferring knowledge from an IE task to another.

In the future, we will further investigate the potential of pre-trained models for intermediate learning. Under the current sequence labeling paradigm, the decoder still needs to be learned from scratch for each IE task, presumably hindering the transferability of extraction knowledge between downstream tasks. We hypothesize that resorting to decoders with reusable weights may help to better leverage the knowledge learned from the intermediate IE task. We have particularly in mind the question answering format [11] which has already shown success for zero-shot relation extraction [19]. We also plan to confirm that the sample efficiency of pre-trained models is observed for other document analysis tasks such as document level classification [13] or visual question answering [23].

References

[1] I. Beltagy, M. E. Peters, and A. Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[2] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[3] F. Cesarini, M. Gori, S. Marinai, and G. Soda. INFORMys: A flexible invoice-like form-reader system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(7):730–745, 1998.

[4] Z. Chen, H. Eavani, W. Chen, Y. Liu, and W. Y. Wang. Few-shot NLG with pre-trained language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 183–190, Online, July 2020. Association for Computational Linguistics.

[5] L. Chiticariu, Y. Li, and F. Reiss. Rule-based information extraction is dead! long live rule-based information extraction systems! In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 827–832, 2013.

[6] B. Cohen and M. York. Ardent partners' accounts payable metrics that matter in 2020. Technical report, Ardent Partners, 2020. `http://ardentpartners.com/2020/ArdentPartners-AP-MTM2020-FINAL.pdf`.

[7] T. A. N. Dang and D. N. Thanh. End-to-end information extraction by character-level embedding and multi-stage attentional u-net. In *30th British Machine Vision Conference 2019, BMVC*

2019, Cardiff, UK, September 9-12, 2019, page 96. BMVA Press, 2019.

[8] T. I. Denk and C. Reisswig. Bertgrid: Contextualized embedding for 2d document representation and understanding. In *Workshop on Document Intelligence at NeurIPS 2019*, 2019.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[10] R. Gal, S. Ardazi, and R. Shilkrot. Cardinal graph convolution framework for document information extraction. In *Proceedings of the ACM Symposium on Document Engineering 2020*, pages 1–11, 2020.

[11] M. Gardner, J. Berant, H. Hajishirzi, A. Talmor, and S. Min. Question answering is a format; when is it useful? *arXiv preprint arXiv:1909.11291*, 2019.

[12] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[13] A. W. Harley, A. Ufkes, and K. G. Derpanis. Evaluation of deep convolutional nets for document image classification and retrieval. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 991–995. IEEE, 2015.

[14] T. Hong, D. Kim, M. Ji, W. Hwang, D. Nam, and S. Park. BROS: A pre-trained language model for understanding texts in document, 2021.

[15] J. Howard and S. Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[16] Z. Huang, K. Chen, J. He, X. Bai, D. Karatzas, S. Lu, and C. Jawahar. Icdar2019 competition on scanned receipt ocr and information extraction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1516–1520. IEEE, 2019.

[17] G. Jaume, H. K. Ekenel, and J. Thiran. FUNSD: A dataset for form understanding in noisy scanned documents. In *2nd International Workshop on Open Services and Tools for Document Analysis, OST@ICDAR 2019, Sydney, Australia, September 22-25, 2019*, pages 1–6. IEEE, 2019.

[18] A. R. Katti, C. Reisswig, C. Guder, S. Brarda, S. Bickel, J. Höhne, and J. B. Faddoul. Chargrid: Towards understanding 2d documents. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4459–4469, 2018.

[19] O. Levy, M. Seo, E. Choi, and L. Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, 2017.

[20] D. Lewis, G. Agam, S. Argamon, O. Frieder, D. Grossman, and J. Heard. Building a test collection for complex document information processing. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 665–666, 2006.

[21] Y. Li, R. Krishnamurthy, S. Raghavan, S. Vaithyanathan, and H. V. Jagadish. Regular expression learning for information extraction. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 21–30, Honolulu, Hawaii, Oct. 2008. Association for Computational Linguistics.

[22] X. Liu, F. Gao, Q. Zhang, and H. Zhao. Graph convolution for multimodal information extraction from visually rich documents. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, pages 32–39, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[23] M. Mathew, D. Karatzas, and C. Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF Winter*

Conference on Applications of Computer Vision (WACV), pages 2200–2209, January 2021.

[24] H. Motahari, N. Duffy, P. Bennett, and T. Bedrax-Weiss. A report on the first workshop on document intelligence (di) at neurips 2019. *ACM SIGKDD Explorations Newsletter*, 22(2):8–11, 2021.

[25] R. B. Palm, F. Laws, and O. Winther. Attend, copy, parse end-to-end information extraction from documents. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 329–336. IEEE, 2019.

[26] R. B. Palm, O. Winther, and F. Laws. Cloudscan - a configuration-free invoice analysis system using recurrent neural networks. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, pages 406–413. IEEE, 2017.

[27] S. Park, S. Shin, B. Lee, J. Lee, J. Surh, M. Seo, and H. Lee. CORD: A consolidated receipt dataset for post-OCR parsing. In *Workshop on Document Intelligence at NeurIPS 2019*, 2019.

[28] S. Pramanik, S. Mujumdar, and H. Patel. Towards a multi-modal, multi-task learning based pre-training framework for document representation learning. *arXiv preprint arXiv:2009.14457*, 2020.

[29] Y. Qian, E. Santus, Z. Jin, J. Guo, and R. Barzilay. GraphIE: A graph-based framework for information extraction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 751–761, 2019.

[30] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, pages 1–26, 2020.

[31] L. A. Ramshaw and M. P. Marcus. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer, 1999.

[32] C. Sage, A. Aussem, V. Eglin, H. Elghazel, and J. Espinas. End-to-end extraction of structured information from business documents with pointer-generator networks. In *Proceedings of the Fourth Workshop on Structured Prediction for NLP*, pages 43–52, Online, Nov. 2020. Association for Computational Linguistics.

[33] C. Sage, A. Aussem, H. Elghazel, V. Eglin, and J. Espinas. Recurrent neural network approach for table field extraction in business documents. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1308–1313. IEEE, 2019.

[34] M. Wei, Y. He, and Q. Zhang. Robust layout-aware IE for visually rich documents with pre-trained language models. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 2367–2376. ACM, 2020.

[35] Y. Xu, M. Li, L. Cui, S. Huang, F. Wei, and M. Zhou. LayoutLM: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1192–1200, 2020.

[36] 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. *arXiv preprint arXiv:2012.14740*, 2020.

[37] W. Yu, N. Lu, X. Qi, P. Gong, and R. Xiao. PICK: Processing key information extraction from documents using improved graph learning-convolutional networks. *arXiv preprint arXiv:2004.07464*, 2020.

[38] T. Zhang, F. Wu, A. Katiyar, K. Q. Weinberger, and Y. Artzi. Revisiting few-sample BERT fine-tuning. In *International Conference on Learning Representations*, 2021.

[39] X. Zhao, Z. Wu, and X. Wang. CUTIE: Learning to understand documents with convolutional universal text information extractor. *arXiv preprint arXiv:1903.12363*, 2019.