Challenges in end-to-end neural scientific table recognition

Yuntian Deng*
Harvard University
dengyuntian@seas.harvard.edu

David Rosenberg
Bloomberg
drosenberg44@bloomberg.net

Gideon Mann
Bloomberg
gmann16@bloomberg.net

***Abstract*—In recent years, end-to-end trained neural models have been applied successfully to various optical character recognition (OCR) tasks. However, the same level of success has not yet been achieved in end-to-end neural scientific table recognition, which involves multi-row/multi-column structures and math formulas in cells. In this paper, we take a step forward to full end-to-end scientific table recognition by constructing a large dataset consisting of 450K table images paired with corresponding LaTeX sources. We apply a popular attentional encoder-decoder model to this dataset and show the potential of end-to-end trained neural systems, as well as associated challenges.**

***Keywords*-digital images; optical character recognition; text extraction**

I. Introduction

While neural models, especially attentional encoder-decoders [1], [2], have been widely used in image captioning and optical character recognition (OCR) problems [3], [4], [5], there's still not much work on the recognition of scientific documents, which contain tables, charts and text. Previous work either uses end-to-end approaches for small scale problems, such as formula recognition [6], [7], [8], graph recognition [9], word recognition [10] and text paragraph recognition [3], or breaks the problem into separate detection and recognition phases [11], as done by the commercial system INFTY [12]. While carefully designed pipelines can achieve impressive results, they typically rely on manually specified rules and components [13]. End-to-end approaches, on the other hand, can learn to perform the task from supervised examples, but have their own challenges, including: 1) the scarcity of datasets large enough to support end-to-end learning, and 2) the memory constraint of GPU's, which limit the image sizes and label sequence lengths.

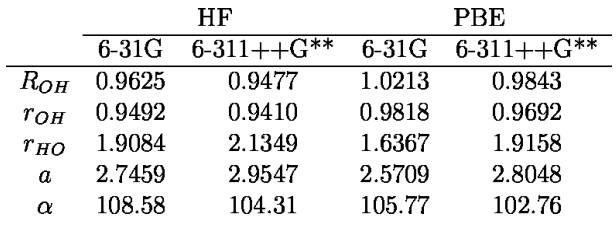
In this paper, we aim to take a small step towards fully end-to-end neural scientific document recognition. Due to the memory challenges involved with applying modern neural models to an image of a full page of a document, we compromise and consider the task of table image recognition, on which training can fit modern GPU memory, yet is much larger and more structurally sophisticated than paragraph/-graph/formula recognition problems that have come before. To make end-to-end learning possible, we construct a large table recognition dataset TABLE2LATEX-450K consisting of 450k examples. An example from our dataset is shown in Fig. 1. Due to the scale and complexity of this table recognition task (some of the tables are almost as large as a full page), we believe solving this task is a challenging and worthwhile step towards full scientific document OCR. Our goal is to evaluate a current state-of-the-art end-to-end neural approach on this particular dataset, to see how far these models are from being practical, and to see what challenges still need to be addressed. Our constructed dataset is available at https://github.com/bloomberg/TABLE2LATEX.

This paper is organized as follows: in Section II we discuss related work. In Section III we describe the details of our dataset TABLE2LATEX-450K. In Section IV we specify the end-to-end neural baseline we evaluate. Section V presents our results and Section VI concludes the paper.

*Work done while interning in the Office of the CTO at Bloomberg.

II. Related Work

The ICDAR 2013 table competition task considers the same problem of table recognition [14], [15]. However, that task only includes 150 tables, which is too small for neural end-to-end training. Furthermore, in that task the table cell contents are available by parsing the provided PDF files, while our dataset only provides the rendered images, which makes it a more challenging task. There are also two similarly constructed datasets: the IM2LATEX-100K dataset [6] and the PDFME-10K dataset [13]. IM2LATEX-100K dataset is also constructed from arXiv articles and provides 83.9k labeled digital formulas in its training set. PDFME-10K is collected from ACM papers and designed for formula recognition in PDFs [13]. However, these two datasets only focus on math formula recognition, which is a sub-problem of table recognition. The images for all datasets mentioned so far, including ours, are digitally rendered. There is also a handwritten math formula recognition dataset provided by the CROHME shared tasks [16]. The full training set consists of 8.8k labeled handwritten formulas, which is usually insufficient for training end-to-end systems, and researchers usually resort to external data [16], [6]. Other, less closely related datasets that deal with document OCR are the IAM databse [10], [17], the RIMES dataset [18], CEDAR [19], IRONOFF [20], IFN/ENIT [21], and AHDB [22].



	HF		PBE	
	6-31G	6-311++G**	6-31G	6-311++G**
R_{OH}	0.9625	0.9477	1.0213	0.9843
r_{OH}	0.9492	0.9410	0.9818	0.9692
r_{HO}	1.9084	2.1349	1.6367	1.9158
a	2.7459	2.9547	2.5709	2.8048
α	108.58	104.31	105.77	102.76

(a)

```
\begin{tabular}{c c c c c}
& \multicolumn{2}{c}{HF} & \multicolumn{2}{c}{PBE}\\
\cline{2-3}\cline{4-5}
& 6-31G & 6-311++G** & 6-31G & 6-311++G** \\
\hline
$R_{OH}$ & 0.9625 & 0.9477 & 1.0213 & 0.9843 \\
$r_{OH}$ & 0.9492 & 0.9410 & 0.9818 & 0.9692 \\
$r_{HO}$ & 1.9084 & 2.1349 & 1.6367 & 1.9158 \\
$a$ & 2.7459 & 2.9547 & 2.5709 & 2.8048 \\
$\alpha$ & 108.58 & 104.31 & 105.77 & 102.76 \\
\end{tabular}
```

(b)

Figure 1: An example from TABLE2LATEX-450K. (a): a table image. (b): the corresponding LaTeX .

One limitation of our work is that TABLE2LATEX-450K is digitally rendered from LaTeX, which makes the font appear mostly uniform and limits the applicability of a trained model to different domains, such as different font sizes and varying baselines. However, prior work [6] shows that it is possible to finetune a trained recognition neural model on digital images to adapt to handwritten ones.

Hidden Markov Models were originally used to solve these OCR problems [23], [24], [25], [26]. Parsing based approaches were also very popular in formula recognition [27], [28]. Later work integrated neural networks into such systems [29], [30], [31], [32], [33]. Connectionist Temporal Classification (CTC) is widely used in OCR because of its ability to label unsegmented sequence data with recurrent neural networks (RNN's) [34], which fits well with OCR problems due to the lack of character-level alignments [35]. However, CTC relies on the assumption that the characters are positioned monotonically [36], thus it is not suitable for cases such as formula recognition [6] and the table recognition task introduced in this paper. The attentional encoder-decoder approaches [2] do not make any such alignment assumptions and can perform language modeling at the same time (which CTC cannot). This model architecture achieves state-of-the-art performances in image captioning tasks [4], and recently researchers have been using this type of approach for OCR as well [6], [7], [8], [10]. Due to its flexibility, we apply this type of approach to solving the table recognition task proposed in this paper.

III. TABLE2LATEX-450K

Our dataset, TABLE2LATEX-450K, collects a large corpus of real-world tables. This dataset consists of about 450k items, each comprising LaTeX source code for a table and the corresponding rendered image. In generating the dataset, we closely follow the practices in IM2LATEX-100K [6]. An overview of our dataset statistics is in Table I. Our dataset and all preprocessing scripts are publicly available at https://github.com/bloomberg/TABLE2LATEX.

We downloaded the LaTeX source for all the arXiv articles from Jul 1991 to Nov 2016, and we extracted source code for tables by matching the regular expression `\begin{tabular}(.*?)\end{tabular}`. We extracted 940,279 tables from 298,168 articles and rendered them with *pdflatex*. Sources that fail to compile are excluded (mostly due to customized commands), resulting in 465,957 tables. Then the rendered PDF files are converted to PNG format[1]. We then crop the table area, pad with 8 pixels on all sides, and finally we downsample by a factor of 2 (downsampling by 3 will cause some subscripts to be indistinguishable). In order to facilitate batching images with same sizes during training, we pad images of similar sizes to the same size[2]. The image size distribution is shown in Fig. 2. For standardized experiment setup, the dataset is separated into a training set (447,321 tables), validation set (9,322 tables) and test set (9,314 tables). Unlike in [6], here we split training/validation/test sets at the article level, since tables in the same article are usually similar, and we'd like to assess generalization to previously unseen articles. The table sources range from 3 to 63,635 characters, with mean 646 and median 917 (excluding `\begin{tabular}` and `\end{tabular}`).

Table I: TABLE2LATEX-450K basic statistics.

#samples	#train	447,321
	#validation	9,322
	#test	9,314
	#total	465,957
table image	median width	512
	median height	192
table LaTeX	median #characters	917
	median #tokens	364

Along with the dataset, we also provide a tokenized and normalized version of the LaTeX source code. Normalization reduces the ambiguities inherent in LaTeX and may make it easier for neural networks to learn a consistent mapping from images to labels. Tokenization helps reduce the length of the label sequence compared to predicting one character at a time. We also provide a detokenization script, which

[1]Following [6], we used the *ImageMagick* `convert` tool with parameters `-density 200 -quality 100`

[2]The groups we used are (128, 64), (128, 128), (256, 64), (256, 128), (256, 256), (384, 128), (384, 192), (384, 384), (512, 128), (512, 256), (512, 512), (640, 192), (640, 320), (640, 640). Those with larger sizes are kept as-is.

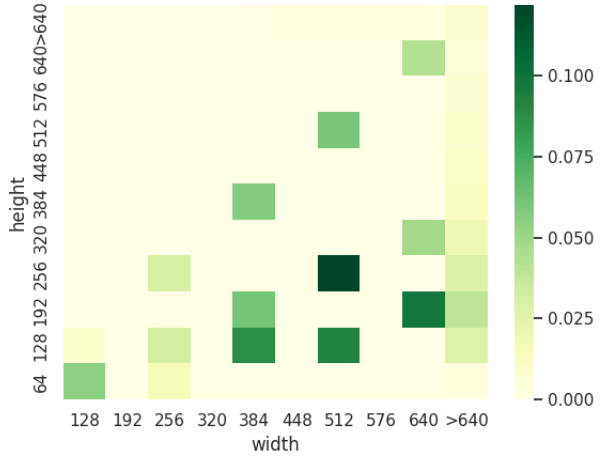


Figure 2: The heatmap of image sizes in TABLE2LATEX-450K (after cropping and downsampling). Note that a US-letter paper (8.5 by 11 inches) would correspond to width 854 and height 1105). The clusters are due to our padding scheme which facilitates batching on GPU's.

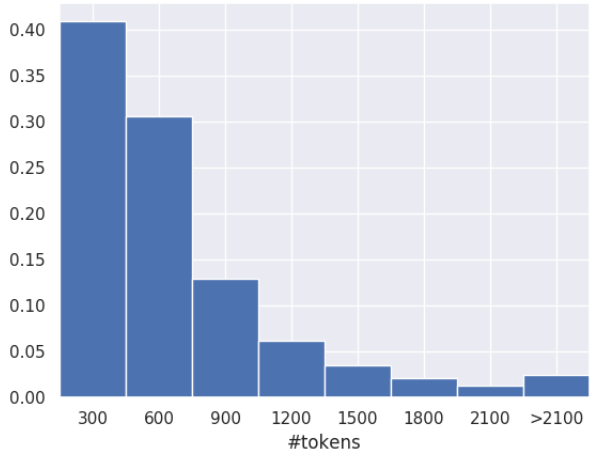


Figure 3: Histogram of the number of tokens in TABLE2LATEX-450K table LaTeX sources.

maps a token sequence back to a character sequence that can be compiled. Tokenization and normalization are based on plasTeX[3], with slight modifications to make the compiled images of normalized table LaTeX sources the same as the original tables (98.96% match rate on the validation set using our provided detokenization scripts). The distribution of the label sequence length (in tokens) across the entire dataset is shown in Fig. 3.

IV. Baseline Model

The model we propose as the baseline for comparison on this dataset is the IM2TEX model [6], which was originally used for formula recognition. The model architecture is shown in Figure 4, and we describe some details below. At a high level, the model consists of an encoder portion, which represents the input image as a set of position-specific features, and a decoder portion, which produces the label sequence (i.e. the LaTeX code) conditioned on the features from the encoder. The model is trained end-to-end to maximize the likelihood of the data.

Encoder: In the first stage of the encoder, a convolutional neural network (CNN) encodes the image into a feature map. The feature map has spatial dimensions proportional to the original image size, but scaled down by a factor of 8 due to the max pooling layers. Each position in the feature map has length 512 (i.e. it's a 512-channel feature map). This feature map is denoted by V in Figure 4. In the older model of [4], this feature map was the end of the encoder module. However, the IM2TEX model adds an extra stage at this point. Each row of the feature map is re-encoded using a bidirectional LSTM to model context [37], [38]. The state at each position of the LSTM becomes the re-encoding for the corresponding position in the CNN feature map. The LSTM state dimension size is determined on the validation set, and the LSTMs share the same parameters for all rows, except for the initial state, which is learned separately for each row. The re-encoding of the feature map is referred to as $\tilde{V}$ in Figure 4. It was found in [6] that this additional LSTM stage in the encoder provides a significant performance boost for equation recognition.

Decoder: IM2TEX uses a standard LSTM decoder with attention, as presented in [2]. To generate each token, a different distribution of attention on the re-encoded feature map is used. Technically, the decoder network produces a distribution on token sequences. We use beam search to find an approximation to the most-probable token sequence.

V. Experiments

A. Implementation Details

We base our implementation on OpenNMT [39], which is a framework for sequence-to-sequence modeling implemented in Torch [40]. The experiments are run on a single Tesla P100 GPU with 16GB of memory. We use the same CNN encoder as in the original IM2TEX work [6], but we tune over the number of decoder hidden units from $\{128, 256, 512\}$, initial learning rate from $\{0.01, 0.05, 0.1\}$, and batch size from $\{2, 4\}$ (we cannot use larger batch sizes for training due to memory constraints). Images larger than 640×640, tables with more than 700 tokens, 250 layout tokens or 40 cell tokens are excluded from the training and validation sets, although they are kept in the test set. We use half-precision floats whenever possible to reduce the memory footprint.

[3]http://tiarno.github.io/plastex/

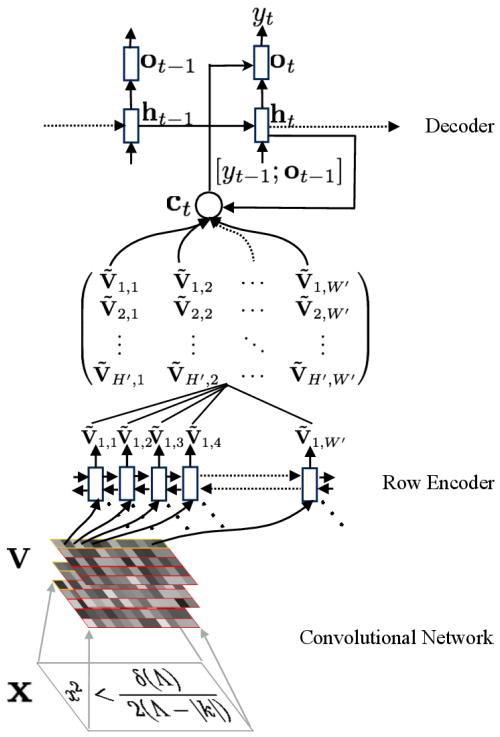


Figure 4: Network structure of IM2TEX. The encoder portion comprises a convolutional network that produces a feature map V, followed by a bidirectional LSTM that re-encodes the feature map into $\tilde{V}$. A standard LSTM decoder with attention is used to produce the final token sequence. *Image courtesy of IM2LATEX-100K [6]*

We train the model with SGD for three epochs. We evaluate the model every 2,000 mini-batches and select the best one using validation perplexity. Full training takes from 10 to 30 days, depending on the number of hidden units in the model.

At test time, we decode with beam size 5 and maximum search length 1500. Images of size larger than $(640, 640)$ are ignored, which makes exact match accuracies upper-bounded by 80%[4].

We evaluate the performance of table recognition using BLEU score [41], a text-based metric, and exact match accuracy, an image-based metric. BLEU score is a metric originally designed for machine translation, but in recent years researchers have been using it to evaluate text recognition as well. Our image-based metric, exact match accuracy, is calculated by rendering and comparing whether the predicted images are *exactly* the same as the original images[5].

[4]Our initial experiments suggest that even including those large images doesn't improve exact match accuracy.

[5]Finer grained image-based metrics may be desirable but are hard to define. Graph edit distance might be a promising future direction.

Table II: Model Performance on TABLE2LATEX-450K.

Model	BLEU	Exact Match Accuracy (%)
IM2TEX	40.33	32.40

Table III: Model Performance breakdown by table structure.

Structure	Frequency (%)	Exact Match (%)
Multi-row, multi-col	1.39	8.53
Multi-row, no multi-col	1.08	13.86
Multi-col, no multi-row	20.65	9.88
No multi-row, no multi-col	76.88	39.14

B. Results

Table II shows the performance of the neural image-to-text model IM2TEX. IM2TEX reaches 32.40% exact match accuracy on this task, which shows the potentials of end-to-end neural approaches on this table recognition task, considering the difficulty of recognizing the entire table correctly. To get a better understanding of the type of errors made by the model, we break down the different kinds of errors: 18.24% are due to ignored images (those larger than 640×640), 22.61% are incorrect in either the number of rows or the number of columns, 11.95% are incorrect in the number of columns, and 15.44% are incorrect in the number of rows. Conditioned on predicting correctly the number of rows and columns, 54.78% would be completely correct predictions. Another natural question to ask is if the model is able to recognize tables with multi-row/multi-column structures. To answer this question, we divide the tables in the test dataset into different categories based on if multi-row or multi-column is present, and we break down the exact match accuracy accordingly. The results are shown in Table III, which confirms that tables with multi-row or multi-column structures are generally harder to recognize correctly, but they are not always mis-classified.

Fig. 5 shows some typical errors of the IM2TEX model. While there is some success in interpreting the behavior of neural networks [42], [43], it is generally hard to analyze why errors occur in a rigorous way. Nevertheless, we present some hypotheses for what might lead to the errors we observe. First, we observed that headers are prone to alignment errors (Fig. 5, the second and the third pairs). We hypothesize that this is because predictions are generated auto-regressively, while in order to align headers with columns, we need to look into future predictions. While the RNN decoder could in theory maintain different alignment possibilities and penalize the wrong ones after finishing decoding the actual columns, it is hard for RNNs (including LSTMs and GRUs [44]) to effectively capture dependencies in long distances [45]. To remedy this issue, a self-attention mechanism might help [46], or we can apply a task-specific constraint during beam search which prunes

subspace	NKA	NKA$_{-1}$	Broyden	JFNK η				
				0.1	0.01	0.001	EW1	EW2
30	31	52	39	53	61	61	53	51
20	31	55	53	53	61	61	53	51
10	32	–	127	53	76	76	70	66
5	34	–	73	88	89	90	116	72

Diagram		Tubular type	Period	Multiplicity
$\widetilde{\mathbf{A}}_{p,q}$		(p,q)	$\mathbf{m.c.m.}(p,q)$	$\frac{pq-(p+q)}{\mathbf{m.c.d.}(p,q)}$
$\widetilde{\mathbf{D}}_n$	n odd	$(n-2,2,2)$	$2(n-2)$	$2(n-3)$
$\widetilde{\mathbf{D}}_n$	n even	$(n-2,2,2)$	$n-2$	$n-3$
$\widetilde{\mathbf{E}}_6$		$(3,3,2)$	6	5
$\widetilde{\mathbf{E}}_7$		$(4,3,2)$	12	11
$\widetilde{\mathbf{E}}_8$		$(5,3,2)$	30	29

E_{lev} [keV]	J^π	I_{EC} [%] TAS	HR	$I_{EC+\beta+}$ [%] TAS	HR	B(GT) [$g_A^2/4\pi$] TAS
151.31 [1]	$(1)^-$	~ 0	< 17	~ 0	< 21	~ 0
158.15 [1]	$(0,1)^-$		2.1(12)		2.6(15)	
210.55	$(1)^-$	~ 0	< 17	~ 0	< 21	~ 0
274.17	$(1,2)^-$	~ 0	* [2]	~ 0	* [2]	~ 0
372.75[1]	$(0^-, 1^-)$		1.3(5)		1.6(6)	
376.26[1]	$(1)^-$	~ 0	4.6(16)	~ 0	5.4(19)	~ 0
416.68	-	0.6(3)	< 19	0.7(4)	< 22	0.0008(4)
495.07[1]	(1^-)		< 19		< 22	
539.81[1]	(0,1)	1.1(5)	1.7(4)	1.3(5)	2.0(4)	0.002(1)
598.33	(1^-)	~ 0	< 9.7	~ 0	< 11	~ 0
738.99	$(0^-, 1)$	~ 0	< 9.9	~ 0	< 11	~ 0
890.72[1]	$(0^-, 1)$		3.8(6)		4.1(6)	
942.21[1]	1^+	43.9(10)	38(5)	48.5(11)	41(5)	0.081(4)
1235.5	1	10.6(3)	4.3(6)	11.3(4)	4.5(6)	0.024(2)
1854.5	1	5.9(9)	0.7(2)	6.0(12)	0.7(2)	0.022(2)
520-3700	-	92(1)	-	98(1)	-	0.32(4)
1880-3700	-	18(1)	-	18(1)	-	0.14(3)
TOTAL		94(2)	99.6(4)	100(2)	100.0(4)	0.32(4)

subspace	~~NKA~~	~~NKA$_{-1}$~~	~~Broyden~~	JFNK η				
				0.1	0.01	0.001	EW1	EW2
30	31	52	39	53	61	61	53	51
20	31	55	53	53	61	61	53	51
10	32	–	127	53	76	76	70	66
5	34	–	73	88	89	90	116	72

Diagram	Tubular type	Period	Multiplicity	
$\overline{\Lambda}_{p,q}$		(p,q)	$\mathbf{m.c.m.}(p,q)$	$\frac{pJ-(p+q)n.c.d.(p,q)}{}$
$\overline{\mathbf{D}}_n$	n	odd	$(n-2,2,2)$	$2(n-2)$
$2(n-3)$				
$\overline{\mathbf{D}}_n$	n	even	$(n-2,2,2)$	$n-2$
$n-3$				
$\overline{\mathbf{E}_6}$		$(3,3,2)$	6	5
$\overline{\mathbf{E}_7}$		$(4,3,2)$	12	11
$\overline{\mathbf{E}_8}$		$(5,3,2)$	30	29

E_{lev} [keV]	J^π	I_{EC} [%] TAS	$I_{EC+\beta+}$ [%] HR	B(GT) [$g_A^2/4\pi$] TAS	HR	TAS
151.31 [1]	$(1)^-$	~ 0	< 17	~ 0	< 21	~ 0
158.15 [1]	$(0,1)^-$		2.1(12)		2.6(15)	
210.55	$(1)^-$	~ 0	< 17	~ 0	< 21	~ 0
274.17	$(1,2)^-$	~ 0	* [2]	~ 0	* [2]	~ 0
372.75[1]	$(0^-, 1^-)$		1.3(5)		1.6(6)	
376.26[1]	$(1)^-$	~ 0	4.6(16)	~ 0	5.4(19)	~ 0
416.68	-	0.6(3)	< 19	0.7(4)	< 22	0.0008(4)
495.07[1]	(1^-)		< 19		< 22	
539.81[1]	(0,1)	1.1(5)	1.7(4)	1.3(5)	2.0(4)	0.002(1)
598.33	(1^-)	~ 0	< 9.7	~ 0	< 11	~ 0
738.99	$(0^-, 1)$	~ 0	< 9.9	~ 0	< 11	~ 0
890.72[1]	$(0^-, 1)$		3.8(6)		4.1(6)	
942.21[1]	1^+	43.9(10)	38(5)	48.5(11)	41(5)	0.081(4)
1235.5	1	10.6(3)	4.3(6)	11.3(4)	4.5(6)	0.024(2)
1854.5	1	5.9(9)	0.7(2)	6.0(12)	0.7(2)	0.022(2)
520-3700	-	92(1)	-	98(1)	-	0.32(4)
1880-3700	-	18(1)	-	18(1)	-	0.14(3)
TOTAL		94(2)	99.6(4)	100(2)	100.0(4)	0.32(4)

Figure 5: Typical errors of the IM2TEX model. Left: ground truth. Right: rendered predictions. In the first pair, the range of $\backslash cline$ is mispredicted; in addition, the model used $\backslash multirow$ while the ground truth used $\backslash raisebox$ to generate the multirow effect. In the second pair, the model makes various mistakes, e.g., it didn't realize that there is an additional column between *Diagram* and *Tabular type*, ˜ was recognized as ¯, and it treated the cell *n odd* as two separate cells leading to an extra cell in that row. In the last pair, the table headers are not aligned correctly with the columns.

inconsistent number of alignment tabs per row with the predicted number of columns. Secondly, we found many $\backslash cline$ range mistakes. This is not surprising, because $\backslash cline$ needs to be predicted in the beginning of a table row, while determining the range requires knowing the positions of each column, which only seems plausible after finishing the decoding of that row. In addition to using self-attention mechanism to alleviate this issue, we hypothesize that iterative refinement might help [47], [48], [49]. Finally, we noticed that IM2TEX's recognition length is only 74.45% of the length of the ground truth (excluding the blank predictions due to large image sizes), which is a well-known problem of sequence-to-sequence learning, and researchers usually use a length penalty to encourage longer predictions [50].

C. Challenges in document level end-to-end OCR

Although table images are of a scale close to that of a full page document (see Fig. 2), documents are still about 2.3 times larger than our largest table used in training (assuming US letter paper). In order to train an end-to-end neural system for document recognition, we expect to face three categories of challenges, based on our table recognition results.

Resource Challenge: In our table recognition task, with 16G memory and half precision floating point, training an IM2TEX model with decoder size 256 (9.65 million parameters, our best model found on validation set) can use at most a batch size of 2. If we want to train on document images, memory would be a big issue. This issue has been raised in the literature, and some workarounds include lower memory models [51], model parallelism [52], gradient checkpointing [53] and inplace operations [54]. Furthermore, training an

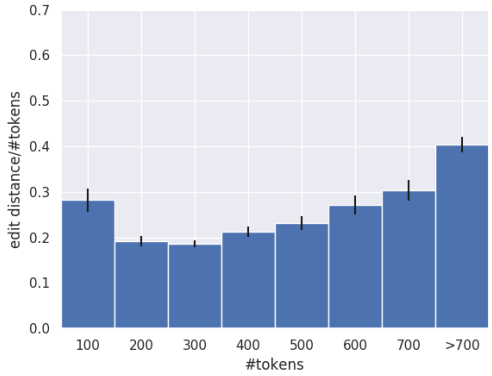


Figure 6: Edit distance error rate versus number of tokens. The error bar shows 95% confidence interval.

IM2TEX model takes more than 10 days (our best validation loss is obtained on the 22nd day), and training on document recognition is likely to take even longer. Curriculum learning may help accelerate training [55].

Modeling Challenge: Documents have a compositional structure, so ideally we should take advantage of that in the model. In table LaTeX, a natural decomposition of generation is decoding the layout tokens followed by individual cell contents. However, our initial experiment using a tree-structured decoder [56] was not very successful, with 5.65% exact-match accuracy, so we might need to come up with new architectures or training procedures to better utilize the structure information.

Performance Challenge: IM2TEX got an impressive 32.40% accuracy on the table recognition task, but we expect the performance to be worse if we work with full page documents. In Fig. 6 we can see that the average error rate grows with the length of the ground-truth LaTeX code, which suggests we may see higher error rates for full pages. Besides, the errors might accumulate over time steps, because during training at each decoding step the model is trained on the ground truth prefix rather than its own predictions (commonly referred to as teacher forcing [57]), leading to train/test inconsistencies. We might need to use techniques like scheduled sampling [58] or iterative error feedback to get further performance boosts [48].

VI. Conclusions and Future Work

In this paper, we take a step towards solving scientific document recognition problems with neural networks. To make training feasible on current hardware, we resort to the slightly more tractable problem of table recognition. Due to the lack of existing large datasets supporting end-to-end learning, we constructed a dataset TABLE2LATEX-450K, consisting of about 450K pairs of table LaTeX sources from arXiv and the corresponding compiled images.

We applied an existing neural image recognition approach IM2TEX [6] to this task. IM2TEX reaches 32.40% exact-match accuracy, showing a promising potential to be applied to full-page document recognition if we had more powerful hardware.

While this neural approach reaches impressive performance on the table recognition task, it is still far from ready for full-page document recognition. The challenges we found in our experiments point us to several possible future directions: reducing memory footprint to meet resources constraints, designing models that better capture the compositional label structure in tables/documents and employing refinement methods to boost performance. We hope future research from the community will pursue these directions and others, and that the TABLE2LATEX-450K task will be a useful benchmark task in that process.

References

[1] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[2] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[3] T. Bluche, J. Louradour, and R. Messina, "Scan, attend and read: End-to-end handwritten paragraph recognition with mdlstm attention," *arXiv preprint arXiv:1604.03286*, 2016.

[4] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International Conference on Machine Learning*, 2015, pp. 2048–2057.

[5] B. Shi, X. Bai, and S. Belongie, "Detecting oriented text in natural images by linking segments," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2550–2558.

[6] Y. Deng, A. Kanervisto, J. Ling, and A. M. Rush, "Image-to-markup generation with coarse-to-fine attention," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 980–989.

[7] J. Zhang, J. Du, S. Zhang, D. Liu, Y. Hu, J. Hu, S. Wei, and L. Dai, "Watch, attend and parse: An end-to-end neural network based approach to handwritten mathematical expression recognition," *Pattern Recognition*, vol. 71, pp. 196–206, 2017.

[8] A. D. Le and M. Nakagawa, "Training an end-to-end system for handwritten mathematical expression recognition by generated patterns," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1. IEEE, 2017, pp. 1056–1061.

[9] M. Cliche, D. Rosenberg, D. Madeka, and C. Yee, "Scatteract: Automated extraction of data from scatter plots," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2017, pp. 135–150.

[10] J. Sueiras, V. Ruiz, A. Sanchez, and J. F. Velez, "Offline continuous handwriting recognition using sequence to sequence neural networks," *Neurocomputing*, vol. 289, pp. 119–128, 2018.

[11] X. Chen and A. L. Yuille, "Detecting and reading text in natural scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2004, p. 366.

[12] M. Suzuki, F. Tamari, R. Fukuda, S. Uchida, and T. Kanahori, "Infty: an integrated ocr system for mathematical documents," in *Proceedings of the 2003 ACM symposium on Document engineering*. ACM, 2003, pp. 95–104.

[13] X. Zhang, L. Gao, K. Yuan, R. Liu, Z. Jiang, and Z. Tang, "A symbol dominance based formulae recognition approach for pdf documents," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1. IEEE, 2017, pp. 1144–1149.

[14] H. Mouchere, C. Viard-Gaudin, R. Zanibbi, U. Garain, D. H. Kim, and J. H. Kim, "Icdar 2013 crohme: Third international competition on recognition of online handwritten mathematical expressions," in *2013 12th International Conference on Document Analysis and Recognition*. IEEE, 2013, pp. 1428–1432.

[15] M. Göbel, T. Hassan, E. Oro, and G. Orsi, "A methodology for evaluating algorithms for table understanding in pdf documents," in *Proceedings of the 2012 ACM symposium on Document engineering*. ACM, 2012, pp. 45–48.

[16] H. Mouchere, C. Viard-Gaudin, R. Zanibbi, and U. Garain, "Icfhr 2014 competition on recognition of on-line handwritten mathematical expressions (crohme 2014)," in *2014 14th International Conference on Frontiers in Handwriting Recognition*. IEEE, 2014, pp. 791–796.

[17] U.-V. Marti and H. Bunke, "The iam-database: an english sentence database for offline handwriting recognition," *International Journal on Document Analysis and Recognition*, vol. 5, no. 1, pp. 39–46, 2002.

[18] E. Augustin, M. Carré, E. Grosicki, J.-M. Brodin, E. Geoffrois, and F. Prêteux, "Rimes evaluation campaign for handwritten mail processing," in *International Workshop on Frontiers in Handwriting Recognition (IWFHR'06),*, 2006, pp. 231–235.

[19] S. N. Srihari, S.-H. Cha, H. Arora, and S. Lee, "Individuality of handwriting: a validation study," in *Proceedings of Sixth International Conference on Document Analysis and Recognition*. IEEE, 2001, pp. 106–109.

[20] C. Viard-Gaudin, P. M. Lallican, S. Knerr, and P. Binter, "The ireste on/off (ironoff) dual handwriting database," in *Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR'99 (Cat. No. PR00318)*. IEEE, 1999, pp. 455–458.

[21] H. El Abed and V. Margner, "The ifn/enit-database-a tool to develop arabic handwriting recognition systems," in *2007 9th International Symposium on Signal Processing and Its Applications*. IEEE, 2007, pp. 1–4.

[22] S. Al-Ma'adeed, D. Elliman, and C. A. Higgins, "A data base for arabic handwritten text recognition research," in *Proceedings eighth international workshop on frontiers in handwriting recognition*. IEEE, 2002, pp. 485–489.

[23] H. Bunke, S. Bengio, and A. Vinciarelli, "Offline recognition of unconstrained handwritten texts using hmms and statistical language models," *IEEE transactions on Pattern analysis and Machine intelligence*, vol. 26, no. 6, pp. 709–720, 2004.

[24] A. Schlapbach and H. Bunke, "Off-line handwriting identification using hmm based recognizers," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 2. IEEE, 2004, pp. 654–658.

[25] A.-L. Bianne-Bernard, F. Menasri, R. A.-H. Mohamad, C. Mokbel, C. Kermorvant, and L. Likforman-Sulem, "Dynamic and contextual information in hmm modeling for handwritten word recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 10, pp. 2066–2080, 2011.

[26] T. Berg-Kirkpatrick and D. Klein, "Improved typesetting models for historical ocr," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, vol. 2, 2014, pp. 118–123.

[27] D. Pru_a and V. Hlavác, "Mathematical formulae recognition using 2d grammars," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, vol. 2. IEEE, 2007, pp. 849–853.

[28] F. Alvaro, J.-M. Benedi *et al.*, "Recognition of printed mathematical expressions using two-dimensional stochastic context-free grammars," in *2011 International Conference on Document Analysis and Recognition*. IEEE, 2011, pp. 1225–1229.

[29] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 5, pp. 855–868, 2009.

[30] S. Espana-Boquera, M. J. Castro-Bleda, J. Gorbe-Moya, and F. Zamora-Martinez, "Improving offline handwritten text recognition with hybrid hmm/ann models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 4, pp. 767–779, 2011.

[31] V. Frinken, A. Fischer, R. Manmatha, and H. Bunke, "A novel word spotting method based on recurrent neural networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 2, pp. 211–224, 2012.

[32] V. Pham, T. Bluche, C. Kermorvant, and J. Louradour, "Dropout improves recurrent neural networks for handwriting recognition," in *2014 14th international conference on frontiers in handwriting recognition*. IEEE, 2014, pp. 285–290.

[33] S. F. Rashid, A. Akmal, M. Adnan, A. A. Aslam, and A. Dengel, "Table recognition in heterogeneous documents using machine learning," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1. IEEE, 2017, pp. 777–782.

[34] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.

[35] R. Messina and J. Louradour, "Segmentation-free handwritten chinese text recognition with lstm-rnn," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2015, pp. 171–175.

[36] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai, "Robust scene text recognition with automatic rectification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4168–4176.

[37] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[38] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[39] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush, "Opennmt: Open-source toolkit for neural machine translation," *arXiv preprint arXiv:1701.02810*, 2017.

[40] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A matlab-like environment for machine learning," in *BigLearn, NIPS Workshop*, 2011.

[41] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 311–318.

[42] Y. Belinkov, N. Durrani, F. Dalvi, H. Sajjad, and J. Glass, "What do neural machine translation models learn about morphology?" *arXiv preprint arXiv:1704.03471*, 2017.

[43] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1885–1894.

[44] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[45] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[46] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[47] L. Neumann and J. Matas, "Efficient scene text localization and recognition with local character refinement," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2015, pp. 746–750.

[48] R. Novak, M. Auli, and D. Grangier, "Iterative refinement for machine translation," *arXiv preprint arXiv:1610.06602*, 2016.

[49] J. Lee, E. Mansimov, and K. Cho, "Deterministic non-autoregressive neural sequence modeling by iterative refinement," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 1173–1182.

[50] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[51] S. B. Blumberg, R. Tanno, I. Kokkinos, and D. C. Alexander, "Deeper image quality transfer: Training low-memory neural networks for 3d images," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2018, pp. 118–125.

[52] Z. Jia, M. Zaharia, and A. Aiken, "Beyond data and model parallelism for deep neural networks," *arXiv preprint arXiv:1807.05358*, 2018.

[53] T. Chen, B. Xu, C. Zhang, and C. Guestrin, "Training deep nets with sublinear memory cost," *arXiv preprint arXiv:1604.06174*, 2016.

[54] S. Rota Bulò, L. Porzi, and P. Kontschieder, "In-place activated batchnorm for memory-optimized training of dnns," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5639–5647.

[55] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 41–48.

[56] L. Dong and M. Lapata, "Language to logical form with neural attention," *arXiv preprint arXiv:1601.01280*, 2016.

[57] A. M. Lamb, A. G. A. P. Goyal, Y. Zhang, S. Zhang, A. C. Courville, and Y. Bengio, "Professor forcing: A new algorithm for training recurrent networks," in *Advances In Neural Information Processing Systems*, 2016, pp. 4601–4609.

[58] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 1171–1179.