Character-Based Handwritten Text Transcription with Attention Networks

Jason Poulos[†*] and Rafael Valle[**]

[*] *Department of Political Science, University of California, Berkeley*
[**] *Department of Electrical Engineering and Computer Sciences, University of California, Berkeley*

[†] *Corresponding author*: poulos@berkeley.edu. *Funding details*: Poulos acknowledges support of the National Science Foundation Graduate Research Fellowship under Grant No. DGE 1106400 and the NVIDIA Corporation for the donation of the Titan Xp GPU used for this research.

Abstract

The paper approaches the task of handwritten text transcription with attentional encoder-decoder networks that are trained on sequences of characters. We experiment on lines of text from a popular handwriting database and compare different attention mechanisms for the decoder. The model trained with softmax attention achieves the lowest test error, outperforming several other RNN-based models. Softmax attention is able to learn a linear alignment between image pixels and target characters whereas the alignment generated by sigmoid attention is linear but much less precise. When no function is used to obtain attention weights, the model performs poorly because it lacks a precise alignment between the source and text output.

Keywords: Handwriting recognition; image-to-text; encoder-decoder networks; convolutional neural networks; attention

1 Introduction

The problem of image-to-text is to convert images to text labels. Image-to-text is an open research problem, especially for the task of transcribing lines of unconstrained (i.e., cursive or overlapping) handwritten text because it is harder to segment characters and recognize them individually (Bluche, Louradour, and Messina 2016). Moreover, transcription models must solve the problem of finding and classifying characters at each time-step without knowing the alignment between image pixels and target characters (Louradour and Kermorvant 2013).

Previous approaches to handwritten text transcription include extracting image features using a sliding window and then matching the features to character labels with a hidden Markov model (HMM) or HMM-neural network hybrid. This approach was outperformed by models that combine a single recurrent neural network (RNN) with bidirectional Long Short-Term Memory (LSTM) with a connectionist temporal classification (CTC) output layer (Graves et al. 2009; Liwicki, Graves, and Bunke 2012). HMM models were subsequently outperformed by multidimensional LSTMs (MDLSTMs), which generalize LSTMs to image inputs (Graves and Schmidhuber 2009).

Attention-based methods have been employed to assist networks in learning the correct alignment between image pixels and target characters (Cho et al. 2014). Attention improves the ability of the network to extract the most relevant information for each part of the output sequence. Moreover, attention networks are capable of modeling the language structures within the output sequence, rather than simply mapping the input to the correct output (Cho, Courville, and Bengio 2015). Attention-based MDLSTMs have recently been employed to recognize multiple lines of cursive text without explicit line segmentation and achieve results comparable to the state-of-the-art on the IAM modern handwriting database (Bluche, Louradour, and Messina 2016).

Attention-based encoder-decoder models (Cho, Courville, and Bengio 2015) have similar advantages to attention-based MDLSTMs, such as being able to handle long sequences, not having to rely on prior segmentation, and are adaptable to different domains. Encoder-decoder networks are a special variant of recurrent neural networks (RNNs) that are suitable for handling sequential

data (Cho et al. 2014). The networks encode a variable-length sequence of characters into a fixed-length vector and then decode the vector into a variable-length target label. These models are the standard for neural machine translation (Bahdanau, Cho, and Bengio 2014; Vinyals et al. 2014), and have also achieved impressive results for tasks including speech recognition (Chorowski et al. 2015) and image captioning (Xu et al. 2015).

Recently, attention-based encoder-decoder networks have been employed for recognizing text in natural images (Lee and Osindero 2016; Shi et al. 2016) and to convert images into presentational markup (Deng et al. 2016); however, the use of these models for handwritten text transcription is very recent. For instance, Bluche and Messina 2017 propose an architecture consisting of a convolutional encoder of the input image and a bidirectional Long Short-Term Memory (LSTM) decoder to predict sequences of characters. The main difference between the authors' architecture and our model is that our model uses a multilayer convolutional neural network (CNN) to extract image features and a separate BLSTM encoder to re-encode the features, which is potentially useful because the encoder can learn features such as text directionality. Another difference is that we use a unidirectional RNN decoder to predict the sequence of characters.

A unique advantage of encoder-decoder networks is that a language model can be easily be integrated on top of the decoder, while language models cannot easily be integrated into MDLSTMs (Bluche, Louradour, and Messina 2016). The language model views the input and output text lines as sequences of characters instead of words, and each character prediction is explicitly conditioned on the previous character. Developing character-aware models — i.e., using a model that views the input and output lines as a sequence of characters rather than words — for image-to-text is promising because these models are capable of making inferences about unseen source words and also generating unseen target words. In addition, character-aware models do not require large vocabularies because only characters are explicitly modeled (Ling et al. 2015).

Our primary contributions are applying character-aware attention networks to the task of recognizing lines of handwritten text and comparing different attention configurations for the decoder. The attention networks used in this paper are capable of transcribing handwritten text without the

need for producing segmentations or bounding boxes of text in images, so the model can potentially transcribe handwritten text in natural scene images.

After describing our model in the context of the image-to-text problem in Section 2, we provide details on the experimental setup and implementation in Section 3. Section 4 benchmarks our approach against competing approaches on a popular handwritten text database and also compares different attention mechanisms. Section 5 concludes and suggests directions for future research.

2 Encoder-decoder networks for image-to-text

The image-to-text problem is one of converting images to hand-transcribed sequences of discrete character labels. The source $\mathbf{x} \in \mathcal{X}$ consists of a sequence of images $x_1, x_2, \ldots, x_N$, each with height and width dimensions $H \times W$. The target $\mathbf{y} \in \mathcal{Y}$ consist of a sequence of characters, $y_1, y_2, \ldots, y_C$, where C is the length of the text and each y is within vocabulary Σ. The task is to learn the function $f(\cdot)$ that maps $\mathcal{X} \rightarrow \mathcal{Y}$ using training example pairs ($\mathbf{x}$, $\mathbf{y}$) of varying dimensions.

We employ the model proposed by Deng et al. 2016 for decompiling images into presentational markup. The model stacks a multilayer encoder and attention-based decoder on a multilayer CNN that extracts image features from the raw input $\mathbf{x}$ and arranges the features on a grid, $\mathbf{V}$.

The encoder re-encodes each row of $\mathbf{V}$ by sliding an RNN across each row of the feature grid. The row encoder RNN generates a re-encoded feature grid $\tilde{\mathbf{V}}_{h,w} = \mathrm{RNN}(\tilde{\mathbf{V}}_{h,w-1}, \mathbf{V}_{h,w})$, for rows h and columns w. Re-encoding $\mathbf{V}$ is useful for transcription tasks because the encoder can learn features such as text directionality. The encoder updates a hidden state $\mathbf{h}_t$ at each time-step t using inputs $\tilde{\mathbf{v}}_t \in \tilde{\mathbf{V}}$:

$$\mathbf{h}_t = f(\tilde{\mathbf{v}}_t, h_{t-1}), \tag{1}$$

where f is a non-linear activation function that operates on all time-steps and input lengths. The decoder is an RNN that inputs the encoder hidden state and the previous element of the output

sequence. Its hidden state is updated recursively by,

$$\mathbf{h'_t} = f\left(\mathbf{h}_{t-1}, \mathbf{y}_{t-1}\right). \tag{2}$$

The decoder hidden states $\mathbf{h'}_t$ are combined with the feature representations via linear transformation $\mathbf{W}$ to generate the distribution $\theta_t = \mathbf{h}_t \mathbf{W} \mathbf{h'}_t$, which is used to produce a context vector,

$$\mathbf{c}_t = \sum_t = a(\theta_t)\mathbf{h}_t, \tag{3}$$

where $a(\cdot)$ is the attention mechanism. Finally, vectors $\mathbf{c}_t$ and $\mathbf{h'}_t$ are concatenated together to calculate the conditional probability of the next element of the sequence,

$$P(\mathbf{y_{t+1}}|\mathbf{y_1}, \mathbf{y_t}, \tilde{\mathbf{V}}) = f\left(\mathbf{W}\mathbf{o}_t\right),$$

where $\mathbf{o}_t = f(\mathbf{W}[\mathbf{h'}_t; \mathbf{c}_t])$.

3 Experimental setup

We experiment on the widely-used IAM database (Marti and Bunke 2002), which consists of 300dpi PNG images of handwritten English text lines along with their transcriptions.[1] We binarize the images in a manner so that preserves the original grayscale information (Villegas, Romero, and Sánchez 2015), scaled to 64 pixel height, and convert to JPEG format.[2] In the IAM text line transcription task training set, there are 6,161 text lines with a maximum length of 81 characters and 79 unique characters (Table 1).

1. Implementation code is available at the repository `https://github.com/jvpoulos/Attention-OCR/`, which modifies the code of Guo and Deng 2016 to include options for different attention mechanisms, regularization, and optimization techniques.

2. We preprocess images and their transcriptions following the procedure and code of Puigcerver, Martin-Albo, and Villegas 2016. Examples of preprocessed text line images are presented in Fig. 1.

3.1 Evaluation

We measure the performance of our model by comparing the estimated transcription $\hat{\mathbf{y}}$ with the ground-truth $\mathbf{y}$. Following the standard in handwritten text transcription, we measure the Character Error Rate (CER), which is the edit distance normalized by the number of characters in the ground truth:

$$\text{CER} = \frac{\sum_t \text{Edit Distance}(y_t, \hat{y}_t)}{\sum_t |y_t|}. \tag{4}$$

In our experiments, we calculate edit distance as Levenshtein distance, or the minimum number of insertions, substitutions, and deletions required to alter the target y_t to the prediction $\hat{y}_t$ at each time-step.

3.2 Implementation

We perform a manual search with a strategy of coordinate descent (Bengio 2012) to select model hyper-parameters (e.g., target embedding size and number of hidden layers in the decoder); optimization strategies such as gradient clipping and normalization; and regularization strategies such as adding ℓ_2 regularization losses with different values of λ and adding dropout to the CNN and LSTM encoder.

We train the networks with stochastic gradient descent to learn the parameter weights and Adadelta (Zeiler 2012) to adapt the learning rate (initial rate of 1). We employ gradient norm clipping and gradient normalization at 5 in order to prevent exploding gradients. We train for 100 epochs with batch size of 4, which takes about 20 hours to run on a 16GB NVIDIA Tesla M60 GPU.[3] In order to facilitate batching, we use bucketing and padding over the image aspect ratio and text length.

Our final model, selected in terms of CER on the validation set, has the following properties:

3. The model with softmax attention took 19 hours and the models with sigmoid attention took 15 hours. We let the model with no attention train for an additional 100 epochs, taking about 30 hours.

3.2.1 CNN

The visual features extractor has seven convolutional layers, each followed by a Rectified Linear Unit (ReLU) activation function (Nair and Hinton 2010). Each convolutional layer is followed by a 2×2 max-pooling layer, except for the third, fifth, and seventh layer, which use batch normalization following the convolution. Dropout ($p = 0.5$) is applied to the inputs after the last convolutional layer.

3.2.2 Encoder-decoder

Stacked on the CNN is a single-layer bidirectional LSTM encoder with 256 hidden units and a two-layer Gated Recurrent Unit (GRU) decoder, each with 128 hidden units. Each RNN layer uses sampled softmax to handle a large target vocabulary without increasing training complexity (Jean et al. 2014). The target vocabulary Σ contains 95 characters, including case-sensitive alphanumeric characters and punctuation. We set the target embedding size to 300 because we find in the model selection phase that a larger-than-standard embedding size tends to improve validation set accuracy.

We experiment with three different attention mechanisms for the decoder: standard softmax attention, sigmoid (i.e., Bernoulli) attention, and no attention (i.e., no function is used to obtain the weights).

4 Results

Table 2 compares the performance of our model trained with three different attention configurations — softmax, sigmoid, and no attention — with competing models on the IAM handwritten text lines test set. The model with softmax attention achieves a CER of 16.9%, which outperforms the state-of-the-art models in 2012 (BLSTM + CTC) and 2013 (MDLSTM + CTC), but does not approach the current state-of-the-art model of Bluche and Messina 2017 (CNN+BLSTM+CTC). Our model trained with sigmoid attention (19.6%) and without attention (49.1%) performs considerably worse.

A direct comparison against most of the models listed in Table 2 is not possible because most of these models rely on domain-specific and word-based dictionaries and language models for decoding. Bluche (2015), for example, uses a word-based dictionary (i.e., a list of words found in the training set) and a word-based language model. The state-of-the-art model of Bluche and Messina 2017 uses a hybrid word and character-based language model.

Two papers written concurrently with our paper use attention-based encoder-decoder networks for the IAM text line transcription task and achieve better results. Sueiras et al. 2018 use an architecture very similar to ours, but train their model to transcribe words rather than sequences of characters and employ a word-based dictionary for decoding. Gui et al. 2018 train character-aware attention networks, but the architecture differs in that they use an attention-based BLSTM decoder and CTC output layer to convert predictions made by the decoder into a character sequence.

4.1 Comparing attention distributions

Figure 2 visualizes the source attention distribution for each attention mechanism. Each row traces the attention weights over the source line at each step of decoding. White values reflect intensity of attention while absence of attention is black.

The plots show that softmax attention predicts a character by focusing heavily on single characters, whereas the attention distribution for sigmoid focus on multiple characters at each time-step. Softmax attention is able to learn a linear alignment whereas the alignment generated by sigmoid attention is linear but much less precise. These results are similar to those of Kim et al. 2017, who find that softmax attention performs better than sigmoid attention on word-to-word machine translation tasks. When no function is used to obtain the attention weights, the model predicts a character by looking at the entire sequence of characters and there is no clear structure in the alignment.

Lastly, we visualize attention on the input image in order to determine how the model makes mistakes. For example, the model tends to produce errors when characters are skewed (Fig. 3 [b]), have long tails (Fig. 3 [a] and [c]), or written in uppercase cursive (Fig. 3 [d]). Fig. 4, which

provides examples of correct IAM transcriptions and visualized softmax attention, shows that the model can correctly predict illegible handwriting (Fig. 4)[b] because it leverages information from the entire input sequence.

5 Conclusion and future directions

The paper approaches the task of handwritten text transcription with attention-based encoder-decoder networks trained to handle sequences of characters rather than words. We train the model on lines of text from a popular handwriting database and experiment with different attention mechanisms. The model trained with softmax attention achieves the lowest test error (16.9%) which is twice the error attained by Gui et al. 2018, who also train character-aware attention networks, but with a CTC output layer to perform the transcription.

Our results show that softmax attention focuses heavily on individual characters when predicting characters, while sigmoid attention focuses on multiple characters at each step of the decoding. When the task is one-to-one, softmax attention is able to learn a more precise alignment at each step of the decoding whereas the alignment generated by sigmoid attention is much less precise. When the model has no attention (i.e., no function is used to obtain attention weights), the model predicts a character by looking at the entire sequence of characters and performs poorly because it lacks a precise alignment between the source and text output.

Our primary contributions are applying character-aware attention networks to the task of handwritten text line transcription and also comparing attention configurations for the decoder. Potential future work might include experimenting with an architecture based entirely on CNNs, which have recently outperformed standard RNN encoder-decoders on neural machine translation tasks (Gehring et al. 2017). CNNs have several attractive properties including parallel computation of all features for source text (Luong, Pham, and Manning 2015).

In addition, future work can apply attention networks to the problem of handwriting recognition in natural scene images (Veit et al. 2016). Previous literature has focused on recognizing

printed text in natural scene images using standard methods in computer vision for segmentation (e.g. Jaderberg et al. 2016, ,). The attention networks used in this paper are capable of transcribing handwritten text without the need for producing segmentations or bounding boxes of text in images, so the model can potentially transcribe handwritten text in natural scene images without preprocessing.

References

Bahdanau, D., K. Cho, and Y. Bengio. 2014. "Neural Machine Translation by Jointly Learning to Align and Translate." *ArXiv e-prints* (September). arXiv: 1409.0473 [cs.CL].

Bengio, Yoshua. 2012. "Practical Recommendations for Gradient-Based Training of Deep Architectures." In *Neural Networks: Tricks of the Trade,* 437–478. Springer.

Bluche, T., J. Louradour, and R. Messina. 2016. "Scan, Attend and Read: End-to-End Handwritten Paragraph Recognition with MDLSTM Attention." *ArXiv e-prints* (April). arXiv: 1604.03286 [cs.CV].

Bluche, Théodore. 2015. "Deep Neural Networks for Large Vocabulary Handwritten Text Recognition." PhD diss., Université Paris Sud-Paris XI.

———. 2016. "Joint Line Segmentation and Transcription for End-to-End Handwritten Paragraph Recognition." In *Advances in Neural Information Processing Systems,* 838–846.

Bluche, Théodore, and Ronaldo Messina. 2017. "Gated Convolutional Recurrent Neural Networks for Multilingual Handwriting Recognition." In *Proceedings of the 13th International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan,* 13–15.

Cho, K., A. Courville, and Y. Bengio. 2015. "Describing Multimedia Content using Attention-Based Encoder–Decoder Networks." *ArXiv e-prints* (July). arXiv: 1507.01053.

Cho, K., B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. 2014. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation." *ArXiv e-prints* (June). arXiv: 1406.1078 [cs.CL].

Chorowski, Jan K, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. "Attention-Based Models for Speech Recognition." In *Advances in Neural Information Processing Systems,* 577–585.

Deng, Y., A. Kanervisto, J. Ling, and A. M. Rush. 2016. "Image-to-Markup Generation with Coarse-to-Fine Attention." *ArXiv e-prints* (September). arXiv: 1609.04938 [cs.CV].

Doetsch, Patrick, Michal Kozielski, and Hermann Ney. 2014. "Fast and Robust Training of Recurrent Neural Networks for Offline Handwriting Recognition." In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on,* 279–284. IEEE.

Gehring, Jonas, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. "Convolutional Sequence to Sequence Learning." *ArXiv e-prints* (May). arXiv: 1705.03122 [cs.CL].

Graves, Alex, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. 2009. "A Novel Connectionist Dystem for Unconstrained Handwriting Recognition," 31:855–868. 5. IEEE.

Graves, Alex, and Jürgen Schmidhuber. 2009. "Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks." In *Advances in Neural Information Processing Systems,* 545–552.

Gui, Liangke, Xiaodan Liang, Xiaojun Chang, and Alexander G Hauptmann. 2018. "Adaptive Context-Aware Reinforced Agent for Handwritten Text Recognition." In *Proceedings of the British Machine Vision Conference (BMVC).*

Guo, Qi, and Yuntian Deng. 2016. *Visual Attention based OCR.* `https://github.com/da03/Attention-OCR`. GitHub repository.

Jaderberg, Max, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2016. "Reading Text in the Wild with Convolutional Neural Networks." *International Journal of Computer Vision* 116 (1): 1–20.

Jean, S., K. Cho, R. Memisevic, and Y. Bengio. 2014. "On Using Very Large Target Vocabulary for Neural Machine Translation." *ArXiv e-prints* (December). arXiv: `1412.2007 [cs.CL]`.

Kim, Y., C. Denton, L. Hoang, and A. M. Rush. 2017. "Structured Attention Networks." *ArXiv e-prints* (February). arXiv: `1702.00887 [cs.CL]`.

Kozielski, Michał, Patrick Doetsch, and Hermann Ney. 2013. "Improvements in RWTH's System for Off-line Handwriting Recognition." In *2013 12th International Conference on Document Analysis and Recognition,* 935–939. IEEE.

Kozielski, Michał, David Rybach, Stefan Hahn, Ralf Schlüter, and Hermann Ney. 2013. "Open Vocabulary Handwriting Recognition Using Combined Word-Level and Character-Level Language Models." In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on,* 8257–8261. IEEE.

Lee, Chen-Yu, and Simon Osindero. 2016. "Recursive Recurrent Nets with Attention Modeling for OCR in the Wild." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 2231–2239.

Ling, W., I. Trancoso, C. Dyer, and A. W Black. 2015. "Character-Based Neural Machine Translation." *ArXiv e-prints* (November). arXiv: `1511.04586 [cs.CL]`.

Liwicki, Marcus, Alex Graves, and Horst Bunke. 2012. "Neural Networks for Handwriting Recognition." In *Computational Intelligence Paradigms in Advanced Pattern Classification,* 5–24. Springer.

Louradour, J., and C. Kermorvant. 2013. "Curriculum Learning for Handwritten Text Line Recognition." *ArXiv e-prints* (December). arXiv: `1312.1737`.

Luong, M.-T., H. Pham, and C. D. Manning. 2015. "Effective Approaches to Attention-Based Neural Machine Translation." *ArXiv e-prints* (August). arXiv: `1508.04025 [cs.CL]`.

Marti, U-V, and Horst Bunke. 2002. "The IAM-database: an English sentence database for offline handwriting recognition." *International Journal on Document Analysis and Recognition* 5 (1): 39–46.

Nair, Vinod, and Geoffrey E Hinton. 2010. "Rectified Linear Units Improve Restricted boltzmann Machines." In *Proceedings of the 27th International Conference on Machine Learning (ICML-10),* 807–814.

Pham, Vu, Théodore Bluche, Christopher Kermorvant, and Jérôme Louradour. 2014. "Dropout Improves Recurrent Neural Networks for Handwriting Recognition." In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on,* 285–290. IEEE.

Puigcerver, Joan. 2017. "Are Multidimensional Recurrent Layers Really Necessary for Handwritten Text Recognition?" In *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on,* 1:67–72. IEEE.

Puigcerver, Joan, Daniel Martin-Albo, and Mauricio Villegas. 2016. *Laia: A Deep Learning Toolkit for HTR.* `https://github.com/jpuigcerver/Laia`. GitHub repository.

Shi, Baoguang, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. 2016. "Robust Scene Text Recognition with Automatic Rectification." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 4168–4176.

Sueiras, Jorge, Victoria Ruiz, Angel Sanchez, and Jose F Velez. 2018. "Offline Continuous Handwriting Recognition Using Sequence to Sequence Neural Networks." *Neurocomputing*.

Veit, A., T. Matera, L. Neumann, J. Matas, and S. Belongie. 2016. "COCO-Text: Dataset and Benchmark for Text Detection and Recognition in Natural Images." *ArXiv e-prints* (January). arXiv: `1601.07140 [cs.CV]`.

Villegas, Mauricio, Verónica Romero, and Joan Andreu Sánchez. 2015. "On the Modification of Binarization Algorithms to Retain Grayscale Information for Handwritten Text Recognition." In *Iberian Conference on Pattern Recognition and Image Analysis,* 208–215. Springer.

Vinyals, O., L. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. Hinton. 2014. "Grammar as a Foreign Language." *ArXiv e-prints* (December). arXiv: `1412.7449 [cs.CL]`.

Voigtlaender, Paul, Patrick Doetsch, and Hermann Ney. 2016. "Handwriting Recognition with Large Multidimensional Long Short-Term Memory Recurrent Neural Networks." In *Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on,* 228–233. IEEE.

Voigtlaender, Paul, Patrick Doetsch, Simon Wiesler, Ralf Schlüter, and Hermann Ney. 2015. "Sequence-Discriminative Training of Recurrent Neural Networks." In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on,* 2100–2104. IEEE.

Xu, Kelvin, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C Courville, Ruslan Salakhutdinov, Richard S Zemel, and Yoshua Bengio. 2015. “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention.” In *ICML,* 14:77–81.

Zeiler, M. D. 2012. “ADADELTA: An Adaptive Learning Rate Method.” *ArXiv e-prints* (December). arXiv: 1212.5701.

Figures & tables

(a) IAM original

(b) IAM preprocessed

Figure 1: IAM images are binarized in a manner so that preserves the original grayscale information. We scale the images to 64 pixel height and convert to JPEG.

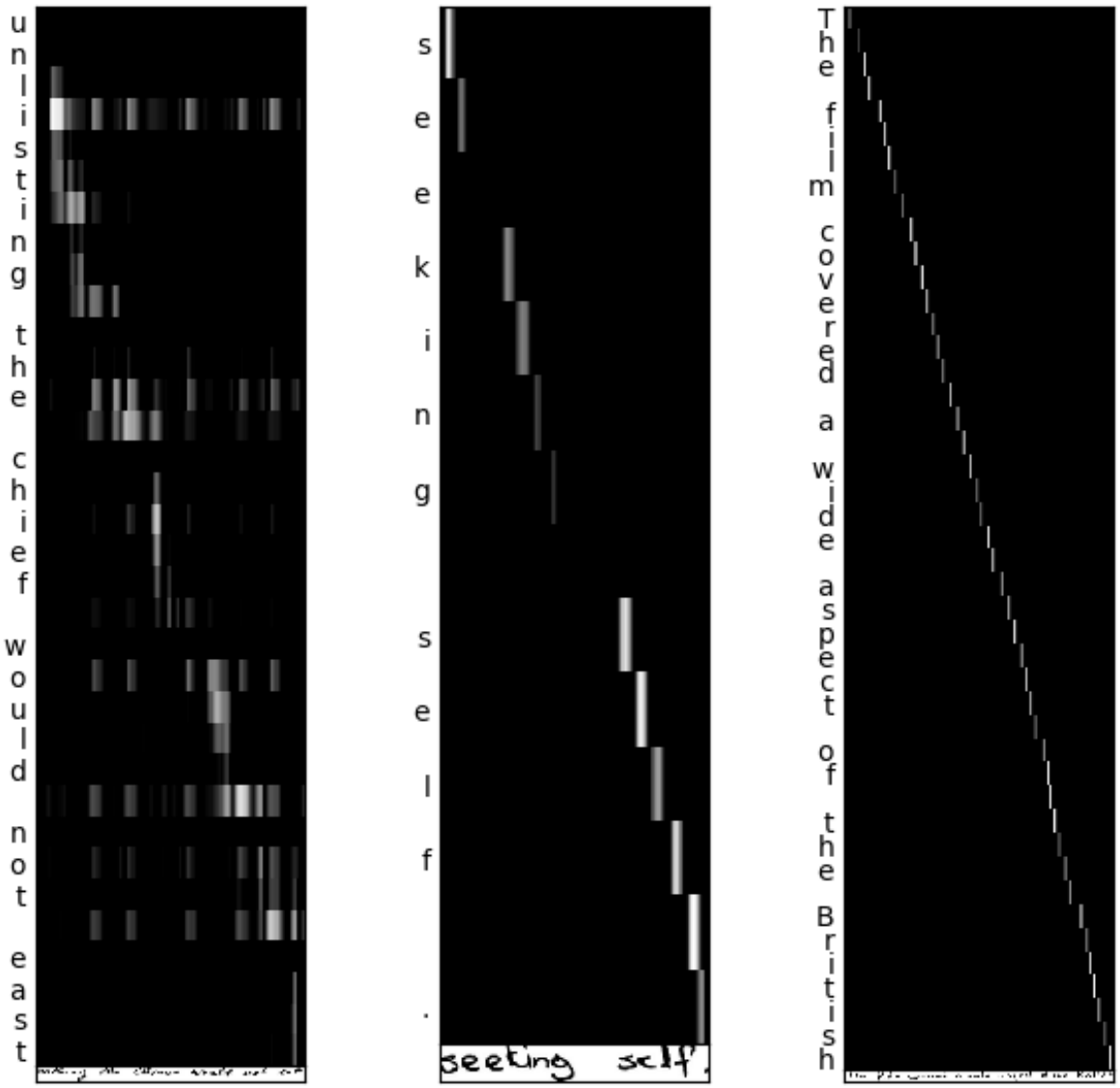
Table 1: Text line splits and language characteristics for IAM text line recognition task.

	Text lines	Unique characters	Max. length
Train	6,161	79	81
Validation	966		
Test	2,915		
Total	10,042		

Table 2: Test set CER for text line transcription task.

Model	Source	LM*	CB#	CER (%)
CNN + BLSTM + CTC	Bluche and Messina 2017	✓		3.2
MDLSTM + CTC	Voigtlaender, Doetsch, and Ney 2016	✓		3.5
MDLSTM + CTC	Bluche 2015	✓		4.4
CNN + LSTM + CTC	Puigcerver 2017	✓		4.4
LSTM + HMM	Doetsch, Kozielski, and Ney 2014	✓		4.7
LSTM + HMM	Voigtlaender et al. 2015	✓		4.8
LSTM + HMM	Kozielski, Doetsch, and Ney 2013	✓		5.1
Multi-directional LSTM + CTC	Pham et al. 2014	✓		5.1
MDLSTM + Attention	Bluche 2016	✓		5.5
CNN + LSTM + Attention$	Sueiras et al. 2018	✓		6.2
MDLSTM + CTC	Bluche, Louradour, and Messina 2016	✓		6.6
CNN + LSTM + CTC	Puigcerver, Martin-Albo, and Villegas 2016			6.7
MDLSTM + Attention	Bluche, Louradour, and Messina 2016	✓	✓	7.0
GMM/HMM	Kozielski et al. 2013			8.2
CNN + BLSTM + Attention + CTC	Gui et al. 2018	✓	✓	8.35
GMM/HMM	Kozielski et al. 2013	✓		11.1
CNN + BLSTM/GRU + Attention (softmax)	Authors	✓	✓	16.85
MDLSTM + CTC	Louradour and Kermorvant 2013	✓		17
BLSTM + CTC	Liwicki, Graves, and Bunke 2012	✓		18.2
CNN + BLSTM/GRU + Attention (sigmoid)	Authors	✓	✓	19.56
CNN + BLSTM/GRU (no attention)	Authors	✓	✓	49.16

* language model used for decoding # model is character-based $ model trained on words



(a) No attention (b) Sigmoid attention (c) Softmax attention

Figure 2: Visualization of the source attention distribution over the input image (x-axis). The y-axis is the transcription. Each row traces the attention weights over the source line at each step of decoding, in grayscale (0: black, 1: white).

(a) Actual: *say a word about it , Lester wants his*;
Predicted: *say a word about it , lester wants his*

(b) Actual: *booty , a new group of Lords might oust*;
Predicted: *booky , o new group of Lords might oust*

(c) Actual: *in the case of the single-sheet quire , an extra*;
Predicted: *in the case of the single-sheet quire , an extraa*

(d) Actual: *your substance on a complete stranger . Set*;
Predicted: *your subteance on a complete stranger , fut*

Figure 3: Incorrect IAM transcriptions and visualized softmax attention. White lines indicates the attended regions and underlines in the transcription indicate the corresponding character.

(a) Actual/predicted: *to the man she had spent so much time*

(b) Actual/predicted: *away at a rate of knots .*

(c) Actual/predicted: *texts and the Gemara explains why these ,*

(d) Actual/predicted: *he was on the verge of a new chapter in*

Figure 4: Correct IAM transcriptions and visualized softmax attention. See notes to Fig. 3.