Attention networks for image-to-text

Jason Poulos[†] Rafael Valle[‡]

Abstract

The paper approaches the problem of image-to-text with attention-based encoder-decoder networks that are trained to handle sequences of characters rather than words. We experiment on lines of text from a popular handwriting database with different attention mechanisms for the decoder. The model trained with softmax attention achieves the lowest test error, outperforming several other RNN-based models. Our results show that softmax attention is able to learn a linear alignment whereas the alignment generated by sigmoid attention is linear but much less precise.

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 [5]. Image-to-text models must solve the problem of finding and classifying characters at each timestep without knowing the alignment between image pixels and target characters [24].

Previous approaches to the problem 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 edged out 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 [14, 23], and subsequently, multidimensional LSTMs (MDLSTMs), which generalize LSTMs to image inputs [15].

Attention-based methods have been employed to assist networks in learning the correct alignment between image pixels and target characters [7]. 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 [6]. 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 [5].

[†]Graduate Student, Department of Political Science, University of California, Berkeley. Email: poulos@berkeley.edu. Poulos acknowledges support of the National Science Foundation Graduate Research Fellowship (Grant No. DGE 1106400).

[‡]Graduate Student, Center for New Music and Audio Technologies, University of California, Berkeley, CA. Email: rafaelvalle@berkeley.edu.

Attention-based encoder-decoder models [6] 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 [7]. 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 [1, 31], and have also achieved impressive results for tasks including speech recognition [8] and image captioning [32].

Encoder-decoder networks have not yet to our knowledge been used for handwriting recognition. 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 [5]. 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 [22].

Our primary contributions to the problem of image-to-text are applying character-aware attention networks to the task of recognizing lines of handwritten text and comparing different attention configurations for the decoder.

In Section 2, we describe the model in the context of the image-to-text problem. Section 3 describes the experimental setup; Section 4 provides results; Section 5 concludes and contemplates future work.

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.

2.1. Model

We use the encoder-decoder model proposed by [10], which stacks a multilayer encoder and attention-based decoder on a multilayer convolutional neural network (CNN). The CNN extracts image features from the raw input $\mathbf{x}$ and arranges the features on a grid, $\mathbf{V}$. The encoder takes the form of a bidirectional Long Short-Term Memory (LSTM) network, and re-encodes each row of $\mathbf{V}$, generating a re-encoded feature grid $\tilde{\mathbf{V}}_{h,w} = \text{LSTM}(\tilde{\mathbf{V}}_{h,w-1}, \mathbf{V}_{h,w})$, for rows h and columns w. Re-encoding $\mathbf{V}$ is important for handwriting recognition because the LSTM encoder can learn features such as text directionality.

The encoder updates a hidden state $\mathbf{h}_t$ at each timestep 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 timesteps 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,

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

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,

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

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 IAM database [1] [26], which consists of 300dpi PNG images of handwritten English text lines along with their transcriptions. We binarize the images in a manner so that preserves the original grayscale information [30], scaled to 64 pixel height, and convert to JPEG format.[2] Examples of preprocessed text line images are presented in Fig. 1.

Evaluation is made by comparing the estimated transcription $\hat{\mathbf{y}}$ with the ground-truth $\mathbf{y}$. Following the standard in handwritten text line recognition, we measure the Character Error Rate (CER), or the edit distance normalized by the number of characters in the ground truth:

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

(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.

In practice, edit distance is the 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 timestep.

3.1. Implementation

We perform a manual search with a strategy of coordinate descent [2] to select model hyperparameters (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. Our final model, selected in terms of validation set CER, has the following properties:[3]

CNN A seven-layer CNN alternating with max-pooling layers. Each convolutional layer uses batch normalization and a ReLU activation function [27] with weights initialized using Xavier initialization [13]. Dropout ($p = 0.5$) is applied to the inputs after the last convolutional layer.

[1] http://www.fki.inf.unibe.ch/databases/iam-handwriting-database

[2] We preprocess images and their transcriptions following the procedure and code of [29].

[3] Implementation code is available at the repository https://github.com/jvpoulos/Attention-OCR/, which modifies the code of [16].

Encoder-Decoder Stacked on the CNN is a single-layer bidirectional LSTM encoder (256 hidden units) and a two-layer Gated Recurrent Unit (GRU) [9] decoder (128 hidden units). Each RNN layer uses sampled softmax to handle a large target vocabulary without increasing training complexity [18]. The target vocabulary Σ contains 95 characters, including case-sensitive alphanumeric characters and punctuation, and the target embedding size is 300.

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

Networks are trained with stochastic gradient descent to learn the parameter weights and Adadelta [33] 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, using bucketing and padding over the image aspect ratio and text length in order to facilitate batching.[4]

4. Results

Table 1 compares the performance of our model trained with three different attention configurations — softmax, sigmoid, and no attention — to previous models on the IAM 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. The model trained with sigmoid attention (19.6%) and without attention (49.1%) do considerably worse.

A direct comparison against previous models is not possible because most of these models rely on domain-specific and word-based dictionaries and language models for decoding. The state-of-the-art model, 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 closest model to ours is a MDLSTM encoder and a softmax attention-enhanced bidirectional LSTM decoder (MDLSTM + Attention) [5]. While MDLSTM + Attention inputs and outputs at the character-level, the decoder output is not conditioned on the previous character. The primary difference is that the model does not require a CNN to extract visual features because the encoder is capable of inputting images. MDLSTM + Attention is also trained with curriculum learning and with a slightly larger training set.[5]

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 timestep. 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 [19], 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

[4] The training process takes about 20 hours on a 16GB NVIDIA Tesla M60 GPU. 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.

[5] All of Bluche's results are trained on a set of 6,482 lines, validated on 976 lines, and tested on 2,915 lines.

Model	Source	LM	CB	CER (%)
MDLSTM + CTC	[3]	✓		4.4
LSTM + HMM	[11]	✓		4.7
HMM/LSTM	[20]	✓		5.1
Multi-directional LSTM + CTC	[28]	✓		5.1
MDLSTM + Attention	[4]	✓		5.5
MDLSTM + CTC	[5]	✓		6.6
CNN + LSTM + CTC	[29]			6.7
MDLSTM + Attention	[5]	✓	✓	7.0
GMM/HMM	[21]			8.2
GMM/HMM	[21]	✓		11.1
CNN + LSTM/GRU + Attention (softmax)		✓	✓	16.85
MDLSTM + CTC	[24]	✓		17
BLSTM + CTC	[23]	✓		18.2
CNN + LSTM/GRU + Attention (sigmoid)		✓	✓	19.56
CNN + LSTM/GRU (No Attention)		✓	✓	49.16

Table 1: Test set CER for text line recognition task. LM indicates whether a statistical language model is used for decoding. CB indicates whether the model is purely character-based (i.e., views inputs and outputs as characters instead of words). Highlighted cells indicate our results.

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. 4 [b]), have long tails (Appendix Fig. 4 [a] and [c]), or written in uppercase cursive (Appendix Fig. 4 [d]).

5. Conclusion

The paper approaches the problem of image-to-text with attention-based encoder-decoder networks that are 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 more than twice the error of the most comparable model (MDLSTM + Attention). The primary difference is that MDLSTM + Attention may be more computationally efficient because it does not require a CNN to extract visual features.

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

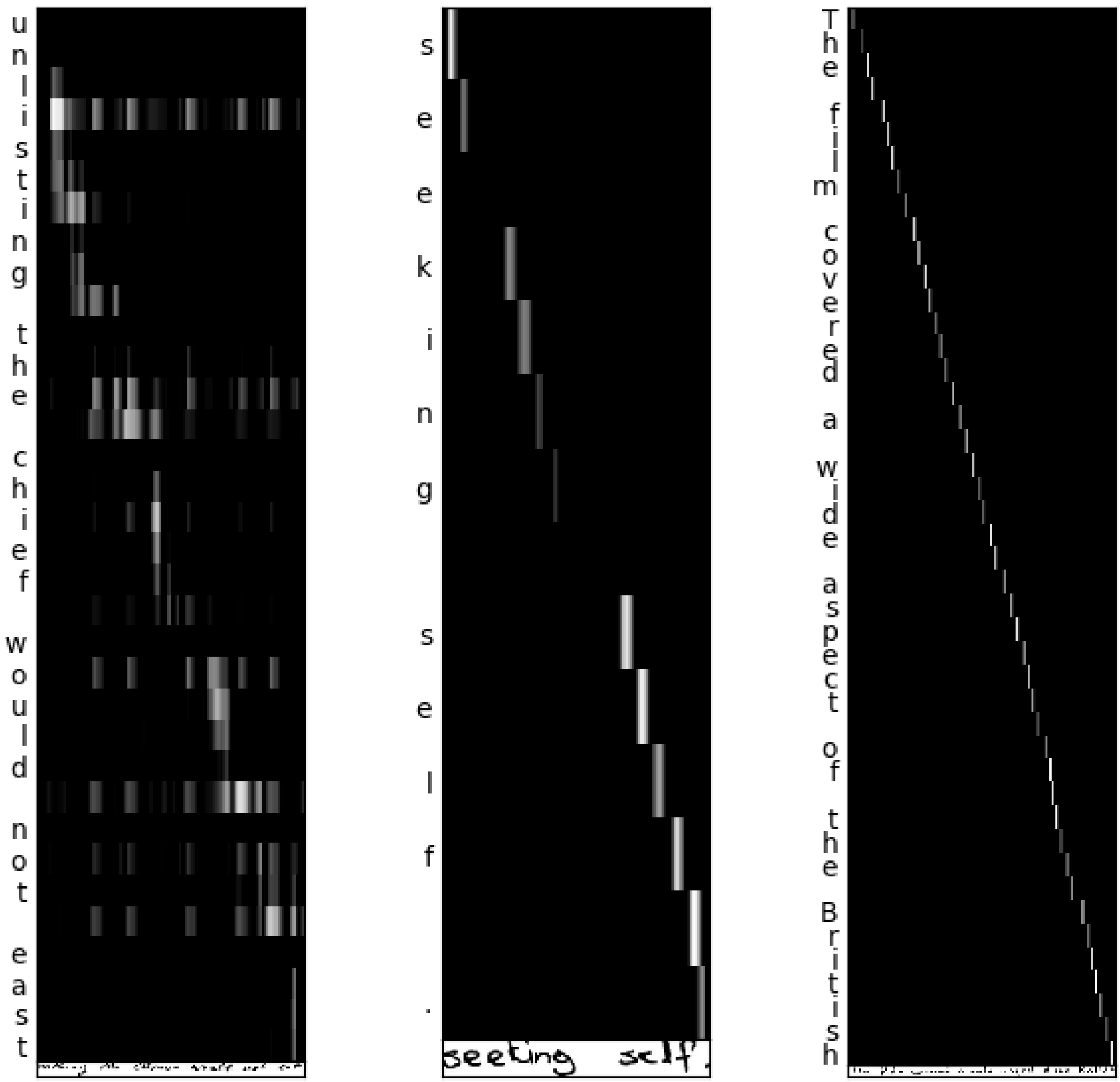

Figure 2: Visualization of the source attention distribution for the no attention (left), sigmoid (center) and softmax (right) mechanisms over the input image (x-axis). The y-axis is the transcription. Each row traces the attention weights over the source line at each step of decoding, in grayscale (0: black, 1: white). The correct transcription for the no attention mechanism is: *nothing the Chinese would not eat .*

problem of image-to-text and also comparing attention configurations for the decoder. Potential future work includes experimenting with an architecture based entirely on CNNs, which have recently outperformed standard RNN encoder-decoders on neural machine translation tasks [12]. CNNs have several attractive properties including parallel computation of all features for source text [25].

Lastly, future work can apply attention networks to the problem of handwriting recognition in the "wild." Previous literature has focused on recognizing printed text (e.g., store signs) in natural scene images using standard methods in computer vision [17]. The attention networks used in this paper are capable of recognizing handwritten text without the need for producing segmentations or bounding boxes of text in images, so that the model can potentially transcribe handwritten text in natural scene images.

References

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

[2] Y. Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural Networks: Tricks of the Trade*, pages 437–478. Springer, 2012.

[3] T. Bluche. *Deep Neural Networks for Large Vocabulary Handwritten Text Recognition*. PhD thesis, Université Paris Sud-Paris XI, 2015.

[4] T. Bluche. Joint line segmentation and transcription for end-to-end handwritten paragraph recognition. In *Advances in Neural Information Processing Systems*, pages 838–846, 2016.

[5] 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.

[6] K. Cho, A. C. Courville, and Y. Bengio. Describing multimedia content using attention-based encoder-decoder networks. *CoRR*, abs/1507.01053, 2015.

[7] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[8] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio. Attention-based models for speech recognition. In *Advances in Neural Information Processing Systems*, pages 577–585, 2015.

[9] 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.

[10] Y. Deng, A. Kanervisto, and A. M. Rush. What you get is what you see: A visual markup decompiler. *arXiv preprint arXiv:1609.04938*, 2016.

[11] P. Doetsch, M. Kozielski, and H. Ney. Fast and robust training of recurrent neural networks for offline handwriting recognition. In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, pages 279–284. IEEE, 2014.

[12] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin. Convolutional Sequence to Sequence Learning. *ArXiv e-prints*, May 2017.

[13] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Aistats*, volume 9, pages 249–256, 2010.

[14] 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*, 31(5):855–868, 2009.

[15] A. Graves and J. Schmidhuber. Offline handwriting recognition with multidimensional recurrent neural networks. In *Advances in neural information processing systems*, pages 545–552, 2009.

[16] Q. Guo and Y. Deng. Visual attention based OCR. https://github.com/da03/Attention-OCR, 2016.

[17] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision*, 116(1):1–20, 2016.

[18] S. Jean, K. Cho, R. Memisevic, and Y. Bengio. On Using Very Large Target Vocabulary for Neural Machine Translation. *ArXiv e-prints*, Dec. 2014.

[19] Y. Kim, C. Denton, L. Hoang, and A. M. Rush. Structured attention networks. *CoRR*, abs/1702.00887, 2017.

[20] M. Kozielski, P. Doetsch, and H. Ney. Improvements in rwth's system for off-line handwriting recognition. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 935–939. IEEE, 2013.

[21] M. Kozielski, D. Rybach, S. Hahn, R. Schlüter, and H. Ney. 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*, pages 8257–8261. IEEE, 2013.

[22] W. Ling, I. Trancoso, C. Dyer, and A. W. Black. Character-based neural machine translation. *CoRR*, abs/1511.04586, 2015.

[23] M. Liwicki, A. Graves, and H. Bunke. Neural networks for handwriting recognition. In *Computational intelligence paradigms in advanced pattern classification*, pages 5–24. Springer, 2012.

[24] J. Louradour and C. Kermorvant. Curriculum learning for handwritten text line recognition. *CoRR*, abs/1312.1737, 2013.

[25] M. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. *CoRR*, abs/1508.04025, 2015.

[26] U.-V. Marti and H. Bunke. The iam-database: an english sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 5(1):39–46, 2002.

[27] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[28] V. Pham, T. Bluche, C. Kermorvant, and J. Louradour. Dropout improves recurrent neural networks for handwriting recognition. In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, pages 285–290. IEEE, 2014.

[29] J. Puigcerver, D. Martin-Albo, and M. Villegas. Laia: A deep learning toolkit for HTR. https://github.com/jpuigcerver/Laia, 2016. GitHub repository.

[30] M. Villegas, V. Romero, and J. A. Sánchez. On the modification of binarization algorithms to retain grayscale information for handwritten text recognition. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 208–215. Springer, 2015.

[31] O. Vinyals, L. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. Hinton. Grammar as a foreign language. *arXiv preprint arXiv:1412.7449*, 2014.

[32] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, volume 14, pages 77–81, 2015.

[33] M. D. Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

Appendices

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

Table 2: Text line splits and language characteristics for IAM text line recognition task [29]. The text lines of all data sets are mutually exclusive (i.e., each writer contributes to only one set).

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

(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 4: Incorrect IAM transcriptions and visualized softmax attention. See notes to Appendix Fig. 4.