Scan, Attend and Read: End-to-End Handwritten Paragraph Recognition with MDLSTM Attention

Théodore Bluche Jérôme Louradour Ronaldo Messina
A2iA SAS
39 rue de la Bienfaisance
75008 Paris
{tb,rm}@a2ia.com

***Abstract*—We present an attention-based model for end-to-end handwriting recognition. Our system does not require any segmentation of the input paragraph. The model is inspired by the differentiable attention models presented recently for speech recognition, image captioning or translation. The main difference is the implementation of covert and overt attention with a multi-dimensional LSTM network. Our principal contribution towards handwriting recognition lies in the automatic transcription without a prior segmentation into lines, which was critical in previous approaches. Moreover, the system is able to learn the reading order, enabling it to handle bidirectional scripts such as Arabic. We carried out experiments on the well-known IAM Database and report encouraging results which bring hope to perform full paragraph transcription in the near future.**

I. Introduction

In offline handwriting recognition, the input is a variable-sized two dimensional image, and the output is a sequence of characters. The cursive nature of handwriting makes it hard to first segment characters to recognize them individually. Methods based on isolated characters were widely used in the nineties [1], [2], and progressively replaced by the sliding window approach, in which features are extracted from vertical frames of the line image [3]. This method transforms the problem into a sequence to sequence transduction one, while potentially encoding the two-dimensional nature of the image by using convolutional neural networks [4] or by defining relevant features [5].

The recent advances in deep learning and the new architectures allowed to build systems that can handle both the 2D aspect of the input and the sequential aspect of the prediction. In particular, Multi-Dimensional Long Short-Term Memory Recurrent Neural Networks (MDLSTM-RNNs [6]), associated with the Connectionist Temporal Classification (CTC [7]) objective function, yield low error rates and became the state-of-the-art model for handwriting recognition, winning most of the international evaluations in the field [8], [9], [10].

Up to now, current systems require segmented text lines, which are rarely readily available in real-world applications. A complete processing pipeline must therefore rely on automatic line segmentation algorithms in order to transcribe a document. We propose a model for multi-line recognition, built upon the recent "attention-based" methods, which have proven successful for machine translation [11], image caption generation [12], [13], or speech recognition [14], [15]. This proposal follows the longstanding and successful trend of making less and less segmentation hypotheses for handwriting recognition. Text recognition state-of-the-art moved from isolated character to isolated word recognition, then from isolated words to isolated lines recognition, and we now suggest to go further and recognize full pages without explicit segmentation.

Our domain of application bears similarities with the image captioning and speech recognition tasks. We aim at selecting the relevant parts of an input signal to sequentially generate text. Like in image captioning, the inputs are images. Similarly to the speech recognition task, we want to predict a monotonic and potentially long sequence of characters. In fact, we face here the challenges of both tasks. We need an attention mechanism that should look for content at specific location and in a specific order. Moreover, in multi-line recognition, the reading order is encapsulated. For example, in Latin scripts, we have a primary order from left to right, and a secondary order from top to bottom. We deal here with a complex problem involving long two-dimensional sequences.

Previous models make sequential predictions over the width of the image, with an horizontal step size fixed by the model. They have to resort to tricks to transform the 2D input image into a character sequence, such as sliding window and Hidden Markov Models, or collapsing representations and CTC, making it impossible to handle multiple lines of text. Those approaches need the text to be already segmented into lines to work properly. Moreover, the length of the predicted sequence, the reading order and the positions of predictions are directly embeded into the architecture. Here, the sequence generation and extraction of information from the multi-dimensional input are decoupled. The system may adjust the number of predictions and arbitrarily and iteratively select any part of the input. Furthermore, since the model makes no assumption about the reading order, it could be applied without any change to languages with different reading order, such as Arabic (right-to-left, or even bidirectional when mixed with Latin script) or some Asian languages (top-to-bottom).

The remaining of this paper is organized as follows. In Section II, we present the baseline MDLSTM-RNN model for state-of-the-art handwriting recognition. The proposed model is described in Section III, and is compared to existing attention-basel models in Section IV. A series of experiments on the IAM database is reported in Section V. We provide a short discussion of the model in Section VI, before concluding the paper in Section VII.

II. Handwriting Recognition with MDLSTM and CTC

Multi-Dimensional Long Short-Term Memory recurrent neural networks (MDLSTM-RNNs) were introduced in [6] for unconstrained handwriting recognition. They generalize the LSTM architecture to multi-dimensional inputs. An overview of the architecture is shown in Fig. 1. The image is presented to four MDLSTM layers, one layer for each scanning direction. The LSTM cell inner state and output are computed from the states and outputs of previous positions in the horizontal and vertical directions.

Each LSTM layer is followed by a convolutional layer, with a step size greater than one, subsampling the feature maps. As in usual convolutional architectures, the number of features computed by these layers increases as the size of the feature maps decreases. At the top of this network, there is one feature map for each label. A collapsing layer sums the features over the vertical axis, yielding a sequence of prediction vectors, effectively delaying the 2D to 1D transformation just before the character predictions, normalized with a softmax activation.

In order to transform the sequence of T predictions into a sequence of $N \leq T$ labels, an additionnal *non-character* – or blank – label is introduced, and a simple mapping is defined in order to obtain the final transcription. The connectionist temporal classification objective (CTC [7]), which considers all possible labellings of the sequence, is applied to train the network to recognize a line of text.

The paradigm collapse/CTC already encodes the monotonicity of the prediction sequence, and allows to recognize characters from 2D images. In this paper, we propose to go beyond single line recognition, and to directly predict character sequences, potentially spanning several lines in the input image. To do this, we replace the collapse and CTC framework with an attention-based decoder.

III. An Attention-Based Model for End-to-End Handwriting Recognition

The proposed model comprises an encoder of the 2D image of text, producing feature maps, and a sequential decoder that predicts characters from these maps. The decoder proceeds by combining the feature vectors of the encoded maps into a single vector, used to update an intermediate state and to predict the next character in the sequence. The weights of the linear combination of the feature vectors at every timestep are predicted by an attention network. In this work the attention is implemented with a MDLSTM network.

The whole architecture, depicted in Fig. 2, computes a fully differentiable function, which parameters can be trained with backpropagation. The optimized cost is the negative log-likelihood of the correct transcription:

$$\mathcal{L}(\mathcal{I}, \mathbf{y}) = -\sum_t \log p(y_t|\mathcal{I}) \tag{1}$$

where $\mathcal{I}$ is the image, $\mathbf{y} = y_1, \cdots, y_T$ is the target character sequence and $p(\cdot|\mathcal{I})$ are the outputs of the network.

In the previous architecture (Fig. 1), we can see the MDLSTM network as a **feature extraction module**, and the last collapsing and softmax layers as a way to predict sequences. Taking inspiration from [12], [15], [13], we keep the MDLSTM network as an encoder of the image $\mathcal{I}$ into high-level features:

$$e_{i,j} = Encoder(\mathcal{I}) \tag{2}$$

where (i, j) are coordinates in the feature maps, and we apply an attention mechanism to read character from them.

The **attention mechanism** provides a summary of the encoded image at each timestep in the form of a weighted sum of feature vectors. The attention network computes a score for the feature vectors at every position:

$$z_{(i,j),t} = Attention(\mathbf{e}, \alpha_{t-1}, s_{t-1}) \tag{3}$$

We refer to $\alpha_t = \{\alpha_{(i,j),t}\}_{(1\leq i\leq W,\ 1\leq j\leq H)}$ as the attention map at time t, which computation depends not only on the encoded image, but also on the previous attention map, and on a state vector s_{t-1}. The attention map is obtained by a softmax normalization:

$$\alpha_{(i,j),t} = \frac{e^{z_{(i,j),t}}}{\sum_{i',j'} e^{z_{(i',j'),t}}} \tag{4}$$

In the literature of attention-based models, we find two main kinds of mechanisms. The first one is referred to as "location-based" attention. The attention network in this case only predicts the position to attend from the previous attended position and the current state (*e.g.* in [16], [17]):

$$\alpha_{(i,j),t} = Attention(\alpha_{t-1}, s_{t-1}) \tag{5}$$

The second kind of attention is "content-based". The attention weights are predicted from the current state, and the encoded features, *i.e.* the network looks for relevant content (*e.g.* in [11], [12]):

$$\alpha_{(i,j),t} = Attention(\mathbf{e}, s_{t-1}) \tag{6}$$

We combine these two complementary approaches to obtain the attention weights from both the content and the position, similarly to Chorowski et al. [15], who compute convolutional features on the previous attention weights in addition to the content-based features.

In this paper, we combine the previous attention map with the encoded features through an MDLSTM layer, which can keep track of position and content (Eqn. 3). With this architecture, the attention potentially depends on the context of the whole image. Moreover, the LSTM gating system allows the network to use the content at one location to predict the attention weight for another location. In that sense, we can see this network as implementing a form of both overt and covert attention.

The state vector s_t allows the model to keep track of what it has seen and done. It is an ensemble of LSTM cells, whose inner states and outputs are updated at each timestep:

$$s_t = LSTM(s_{t-1}, g_t) \tag{7}$$

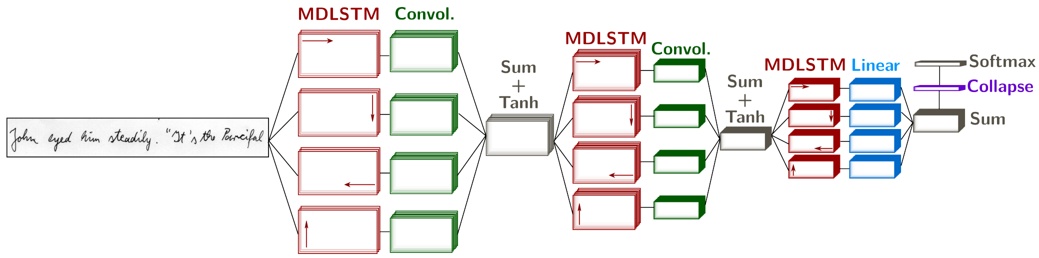

Fig. 1: MDLSTM-RNN for handwriting recognition, alternating LSTM layers in four directions and subsampling convolutions. After the last linear layer, the feature maps are collapsed in the vertical dimension, and character predictions are obtained after a softmax normalization.

where g_t represents the summary of the image at time t, resulting from the attention given to the encoder features:

$$g_t = \sum_{i,j} \alpha_{(i,j),t} e_{i,j} \tag{8}$$

and is used both to update the state vector and to predict the next character.

The final component of this architecture is a **decoder**, which predicts the next character given the current image summary and state vector:

$$y_t = Decoder(s_t, g_t) \tag{9}$$

The end of sequence is predicted with a special token EOS. In this paper, the decoder is a simple multi-layer perceptron with one hidden layer ($\tanh$ activation) and a softmax output layer.

IV. Related Work

Our system is based on the idea of [11] to learn to align and transcribe for machine translation. It is achieved by coupling an encoder of the input signal and a decoder predicting language tokens with an attention mechanism, which selects from the encoded signal the relevant parts for the next prediction.

It bears many similarity with the attention-based models for speech recognition [14], [15]. Indeed, we want to predict text from a sensed version of natural language (audio in speech recognition, image of handwritten text here). As for speech recognition, we need to deal with long sequences. Our network also has LSTM recurrences, but we use MDLSTM units to handle images, instead of bi-directional LSTMs. This is a different way of handling images, compared with the attention-based systems for image captioning for example [12], [13].

Contrary to some attention models like DRAW [18] or spatial transformer networks [19], our model does not select and transform a part of the input by interpolation, but only weights the feature vectors and combines them with a sum. We do not explicitely predict the coordinates of the attention, as done in [20].

In similar models of attention, the weights are either computed from the content at each position individually (e.g. in [14], [13]), from the location of the previous attention (e.g. in [16], [17]) or from a combination of both (e.g. in [15], [17]). In our model, the content of the whole image is explicitely taken into account to predict the weight at every position, and the location is implicitely considered through the MDLSTM recurrences.

Attention models have been applied to the recognition of sequences of symbols (e.g. in [20], [21] for MNIST or SVHN digits, and [22], [23] for scene text OCR on cropped words), and recently to handwriting recognition. Most notably, Doetsch et al. [24] apply an attention neural network to the output of a BLSTM network operating on frames extracted from a text line image with a sliding window. In the model we propose, we apply the network directly to paragraph images, and we do not predict the number of characters in advance.

Finally, our system is quite similar to the one proposed by Bluche [25] in its architecture. The main difference is that we apply the attention at the character level, allowing the model to learn any reading order. This should also allow its application to full document images.

V. Experiments

A. Experimental Setup

We carried out the experiments on the popular IAM database, described in details in [26], consisting of images of handwritten English text documents. They correspond to English texts exctracted from the LOB corpus. 657 writers produced between 1 and 59 handwritten documents. The training set comprises 747 documents (6,482 lines, 55,081 words), the validation set 116 documents (976 lines, 8,895 words) and the test set 336 documents (2,915 lines, 25,920 words). The texts in this database typically contain 450 characters in about nine lines. In 150 dpi images, the average character has a width of 20px.

The baseline corresponds to the architecture presented in Fig. 1, with 4, 20 and 100 units in MDLSTM layers, 12 and 32 units in convolutional layers, and dropout after every MDLSTM as presented in [27]. The last linear layer has 80 outputs, and is followed by a collapse layer and a softmax normalization. In the attention-based model, the encoder has the same architecture as the baseline model, without the collapse and softmax. The attention network has 16 or 32 hidden LSTM units in each direction followed by a linear layer with one output. The state LSTM layer has 128 or 256 units, and the decoder is an MLP with 128 or 256 tanh neurons. The networks are trained with RMSProp [28] with a base learning rate of 0.001 and mini-batches of 8 examples. We measure the Character Error Rate (CER%), i.e. the edit distance normalized by the number of characters in the ground-truth.

B. The Usual Word and Line Recognition Tasks

We first trained the model to recognize words and lines. The inputs are images of several consecutive words from the IAM database. The encoder network has the standard architecture presented in Section II, with dropout after each LSTM layer [27] and was pre-trained on IAM database with CTC. The results are presented in Table I. We see that the models tend to be better on longer inputs, and the results for complete lines are not far from the baseline performance.

TABLE I: Multi-word recognition results (CER%).

Model	Inputs	CER (%)
MDLSTM + CTC	Full Lines	6.6
Attention-based	1 word	12.6
	2 words	9.4
	3 words	8.2
	4 words	7.8
	Full Lines	7.0

In Fig. 3, we display the attention map and character predictions as recognition proceeds. We see that attention effectively shifts from one character to the next, in the proper reading order.

C. Learning Line Breaks

Next, we evaluate the ability of this model to read multiple lines, *i.e.* to read all characters of one line before finding the next line. This is challenging because it has to consider two levels of reading orders, which is crucial to achieve whole paragraph recognition without prior line segmentation.

We started with a synthetic database derived from IAM, where the images of words or sequences of words are stacked

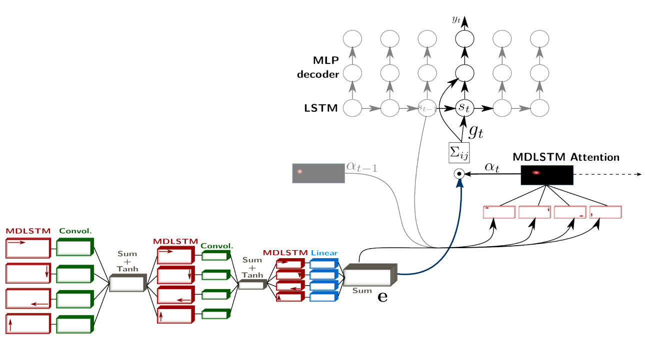

Fig. 2: Proposed architecture. The encoder network has the same architecture as the standard network of Fig. 1. At each timestep, the feature maps, along with the previous attention map and state features are fed to the attention network which outputs new weights at each position. The weighted sum of the encoder features is given to the state LSTM, and to the decoder.

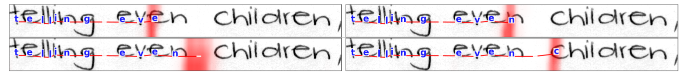

Fig. 3: Visualization of the attention weights at each timestep for multiple words. The attention map is interpolated to the size of the input image. The outputs of the network at each timestep are displayed in blue.

to represent two short lines. The results (character error rate – CER) are presented in Table II. Again, the system is better with longer inputs. The baseline from the previous section does not apply here anymore, and the error rate with two lines is worse than with a single line, but still in a reasonable range.

TABLE II: Multi-line recognition results (CER%).

Two lines of...	CER (%)
1 words	11.8
2 words	11.1
3 words	10.9
Full Lines	9.4

We show in Fig. 4 the outputs of the decoder and of the attention network on an example of two lines of one word. We observe that the system learnt to look for the second line when the first line is read, with an attention split between the end of the first line and the beginning of the second line.

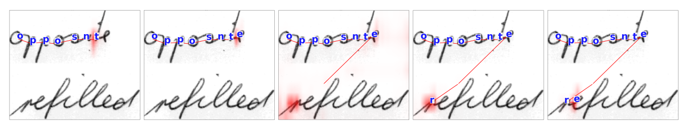

Fig. 4: Visualization of the attention weights at each timestep for multiple lines. The attention map is interpolated to the size of the input image.

D. Towards Paragraph Recognition

Training this system on paragraphs raises several challenges. The model still has to learn to both align and recognize, but the alignment problem is much more complex. A typical paragraph from IAM contains 450 characters on 9 text lines. Moreover, the full backpropagation though time must cover those 450 timesteps, on images that are significantly bigger than the line images, which is prohibitive in terms of memory usage.

To tackle these challenges, we modified the training procedure in several ways. First, we truncated the backpropagation through time of the decoder to 30 timesteps in order to adress the memory issue. Note that although 30 timesteps was chosen so that intermediate activations fit in memory even for full paragraphs, it roughly corresponds to half a line, or 4-5 words, and we suppose that it is sufficient to learn the relevant dependencies. Then, instead of using only full paragraphs (there are only 747 in the training set), we added the single lines and all concatenations of successive lines. To some extent, this may be seen as data augmentation by considering different crops of paragraphs.

Finally, we applied several levels of curriculum learning [29]. One of these is the strategy proposed by [30], which samples training examples according to their target length. It prefers short sequences at the beginning of training (e.g. single lines) and progressively adds longer sequences (paragraphs). The second curriculum is similar to that of [20]: we train only to recognize the first few characters at the beginning. The targets are the first $N \times epoch$ characters, with $N = 50$, i.e. first 50 during the first epoch, then first 100, and so on. Note that 50 characters roughly correspond to the length of one line. This strategy amounts to train to recognize the first line during the first epoch, then the first two lines, and so on.

The baseline here is the MDLSTM network trained with CTC for single lines, applied to the result of automatic line segmentation. We present in Table III the character error rates obtained with different input resolutions and segmentation algorithms. Note that the line segmentation on IAM is quite easy as the lines tend to be clearly separated.

TABLE III: Character Error Rates (%) of CTC-trained RNNs on ground-truth (GT) lines and automatic segmentation of paragraphs with different resolutions.

DPI	Line segmentation				Attention-based
	GT	Projection	Shredding	Energy	*(this work)*
90	18.8	24.7	19.8	20.8	-
150	10.3	17.2	11.1	11.8	16.2
300	6.6	13.8	7.5	7.9	-

We trained the attention-based model on 150 dpi images. In Fig. 5, we show some examples of paragraphs being transcribed by the network. We report the character error rates on inputs corresponding to all possible sub-paragraphs of one to twelve lines from the development set in Fig. 6. The *Paragraphs* column corresponds to the set of actual complete paragraphs, individually depicted as blue dots in the other columns. Note that for a few samples, the attention jumped back to a previous line at some point, causing the system to transcribe again a whole part of the image. In those cases, the insertion rate was very high and the final CER sometimes above 100%.

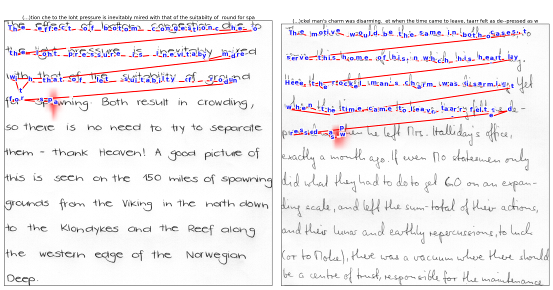

Fig. 5: Transcribing full paragraphs of text. Character predictions are located at the center of mass of the attention weights.

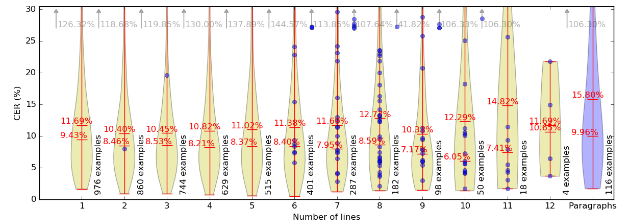

Fig. 6: Character Error Rates (%) of the proposed model trained on multiple lines (sub-paragraphs from IAM database), evaluated with inputs containing different number of lines. The medians and means accross all examples are displayed in red. The blue dots are actual complete paragraphs.

VI. Discussion

The results we present in this paper are promising and show that recognizing full paragraphs of text without an explicit segmentation into lines is feasible. Not only can we hope to perform full paragraph recognition in the near future, but we may also envision the recognition of complex documents. The attention mechanism would then be a way of performing document layout analysis and text recognition within a single end-to-end system.

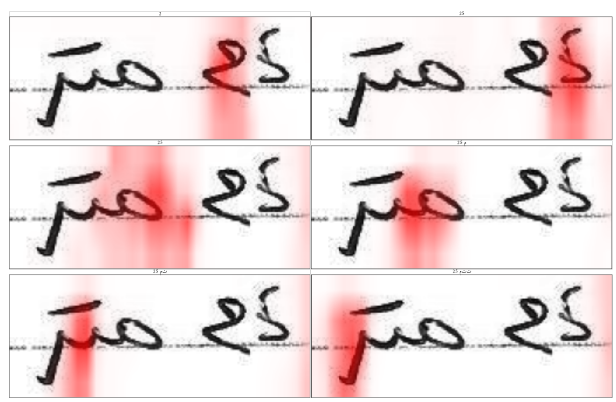

Fig. 7: Attention steps for a bidirectional Arabic input (left to right, top to bottom).

We also carried out preliminary experiments on Arabic text lines The model effectively reads from right to left, and manages to handle bidirectional reading order in mixed Arabic/Latin inputs in several images. In Fig. 7, we see that the attention model correctly reads *"25"* from left to right, and then starts reading the Arabic word from right to left.

We managed to handle long and complex sequences without resorting to tricks like in [15]. Maybe the use of LSTMs, particularly through their gating mechanism helped. Yet future plans include to try those tricks in order to see if the results improve. However, it should be noted that careful training was required for the models to converge. The curriculum seemed to make a difference, and training first on couple of lines before switching to full paragraphs had a big impact. Training with RMSProp was also quite important. That said, the IAM database represents a relatively small amount of data, and it would be interesting to see whether end-to-end paragraph training from scratch is helped by more data.

In this version of the model, the prediction is not explicitely conditioned on the previous character, as for example in [15], and the integration of a language model is more complicated than with classical models trained with CTC. This should be addressed in future work. Finally, the presented system is very slow due to the computation of attention for each character in turn. The time and memory consumption is prohibitive for most industrial applications, but learning how to read whole paragraphs might open new directions of research in the field.

VII. Conclusion

In this paper, we have presented a method to transcribe complete paragraphs of text without an explicit line segmentation. The system is based on MDLSTM-RNNs, widely applied to transcribe isolated text lines, and is inspired from the recent attention-based models. The proposed model is able to recognize multiple lines of text, and to learn encapsulated reading orders. It is not limited to handwritten Latin scripts, and could be applied without change to other languages (such as Chinese or Arabic), write type (e.g. printed text), or more generally image-to-sequence problems.

Unlike similar models, the decoder output is not conditioned on the previous token. Future work will include this architectural modification, which would enable a richer decoding with a beam search. On the other hand, we proposed an MDLSTM attention network, which computes attention weights taking into account the context of the whole image, and merging location and content information. The results are encouraging, and prove that explicit line segmentation is not necessary, which we believe is an important contribution towards end-to-end handwriting recognition.

References

[1] Y. Bengio, Y. LeCun, C. Nohl, and C. Burges, "Lerec: A NN/HMM hybrid for on-line handwriting recognition," *Neural Computation*, vol. 7, no. 6, pp. 1289–1303, 1995.

[2] S. Knerr, E. Augustin, O. Baret, and D. Price, "Hidden Markov model based word recognition and its application to legal amount reading on French checks," *Computer Vision and Image Understanding*, vol. 70, no. 3, pp. 404–419, 1998.

[3] A. Kaltenmeier, T. Caesar, J. M. Gloger, and E. Mandler, "Sophisticated topology of hidden Markov models for cursive script recognition," in *Document Analysis and Recognition, 1993., Proceedings of the Second International Conference on*. IEEE, 1993, pp. 139–142.

[4] T. Bluche, H. Ney, and C. Kermorvant, "Feature Extraction with Convolutional Neural Networks for Handwritten Word Recognition," in *12th International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2013, pp. 285–289.

[5] A.-L. Bianne, F. Menasri, R. Al-Hajj, C. Mokbel, C. Kermorvant, and L. Likforman-Sulem, "Dynamic and Contextual Information in HMM modeling for Handwriting Recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 10, pp. 2066 – 2080, 2011.

[6] A. Graves and J. Schmidhuber, "Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks," in *Advances in Neural Information Processing Systems*, 2008, pp. 545–552.

[7] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *International Conference on Machine learning*, 2006, pp. 369–376.

[8] S. Brunessaux, P. Giroux, B. Grilhères, M. Manta, M. Bodin, K. Choukri, O. Galibert, and J. Kahn, "The Maurdor Project: Improving Automatic Processing of Digital Documents," in *Document Analysis Systems (DAS), 2014 11th IAPR International Workshop on*. IEEE, 2014, pp. 349–354.

[9] J. A. Sánchez, V. Romero, A. Toselli, and E. Vidal, "ICFHR 2014 HTRtS: Handwritten Text Recognition on tranScriptorium Datasets," in *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2014.

[10] A. Tong, M. Przybocki, V. Maergner, and H. El Abed, "NIST 2013 Open Handwriting Recognition and Translation (OpenHaRT13) Evaluation," in *11th IAPR Workshop on Document Analysis Systems (DAS2014)*, 2014.

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

[12] K. Cho, A. Courville, and Y. Bengio, "Describing multimedia content using attention-based encoder-decoder networks," *Multimedia, IEEE Transactions on*, vol. 17, no. 11, pp. 1875–1886, 2015.

[13] K. Xu, J. Ba, R. Kiros, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," *arXiv preprint arXiv:1502.03044*, 2015.

[14] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," *arXiv preprint arXiv:1508.01211*, 2015.

[15] 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*, 2015, pp. 577–585.

[16] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[17] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," *arXiv preprint arXiv:1410.5401*, 2014.

[18] K. Gregor, I. Danihelka, A. Graves, and D. Wierstra, "DRAW: A recurrent neural network for image generation," *arXiv preprint arXiv:1502.04623*, 2015.

[19] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 2008–2016.

[20] J. Ba, V. Mnih, and K. Kavukcuoglu, "Multiple object recognition with visual attention," *arXiv preprint arXiv:1412.7755*, 2014.

[21] S. K. Sønderby, C. K. Sønderby, L. Maaløe, and O. Winther, "Recurrent spatial transformer networks," *arXiv preprint arXiv:1509.05329*, 2015.

[22] C.-Y. Lee and S. Osindero, "Recursive recurrent nets with attention modeling for ocr in the wild," *arXiv preprint arXiv:1603.03101*, 2016.

[23] B. Shi, X. Wang, P. Lv, C. Yao, and X. Bai, "Robust scene text recognition with automatic rectification," *arXiv preprint arXiv:1603.03915*, 2016.

[24] P. Doetsch, A. Zeyer, and H. Ney, "Bidirectional decoder networks for attention-based end-to-end offline handwriting recognition," in *Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on*. IEEE, 2016, pp. 361–366.

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

[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*, vol. 5, no. 1, pp. 39–46, 2002.

[27] V. Pham, T. Bluche, C. Kermorvant, and J. Louradour, "Dropout improves recurrent neural networks for handwriting recognition," in *14th International Conference on Frontiers in Handwriting Recognition (ICFHR2014)*, 2014, pp. 285–290.

[28] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural Networks for Machine Learning*, vol. 4, 2012.

[29] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *ICML*, 2009, p. 6.

[30] J. Louradour and C. Kermorvant, "Curriculum learning for handwritten text line recognition," in *International Workshop on Document Analysis Systems (DAS)*, 2014.