Convolve, Attend and Spell: An Attention-based Sequence-to-Sequence Model for Handwritten Word Recognition

Lei Kang[1,2](✉), J. Ignacio Toledo[1](✉), Pau Riba[1](✉), Mauricio Villegas[2](✉), Alicia Fornés[1](✉), and Marçal Rusiñol[1](✉)

[1] Computer Vision Center, Universitat Autónoma de Barcelona, Barcelona, Spain
{lkang,jitoledo,priba,afornes,marcal}@cvc.uab.es
[2] omni:us, Berlin, Germany
{lei,mauricio}@omnius.com

Abstract. This paper proposes Convolve, Attend and Spell, an attention-based sequence-to-sequence model for handwritten word recognition. The proposed architecture has three main parts: an encoder, consisting of a CNN and a bi-directional GRU, an attention mechanism devoted to focus on the pertinent features and a decoder formed by a one-directional GRU, able to spell the corresponding word, character by character. Compared with the recent state-of-the-art, our model achieves competitive results on the IAM dataset without needing any pre-processing step, predefined lexicon nor language model. Code and additional results are available in https://github.com/omni-us/research-seq2seq-HTR.

1 Introduction

Handwriting Text Recognition (HTR) has interested the Pattern Recognition community for many years. Transforming images of handwritten text into machine readable format has an important amount of application scenarios, such as historical documents, mail-room processing, administrative documents, etc. But the inherent high variability of handwritten text, the myriad of different writing styles and the amount of different languages and scripts, make HTR an open research problem that is still challenging. With the rise of neural networks and deep learning architectures, HTR has reached, as many other applications, an important performance boost. The recognition of handwritten text was, in fact, one of the first application scenarios of convolutional neural networks, when LeCun *et al.* proposed in the late nineties such architectures [16] for recognizing handwritten digits from the MNIST dataset. In the literature, several other

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-12939-2_32) contains supplementary material, which is available to authorized users.

T. Brox et al. (Eds.): GCPR 2018, LNCS 11269, pp. 459–472, 2019.
https://doi.org/10.1007/978-3-030-12939-2_32

methods have been proposed for tackling the HTR task such as Hidden Markov Models (HMM) [4,6,11], Recurrent Neural Networks (RNN) and Connectionist Temporal Classification (CTC) [15,18,20,23,27], or nearest neighbor search methods in embedding spaces [1,14,21].

Inspired in the latest advances in machine translation [2,25], image captioning [28] or speech recognition [3,8], we believe that sequence-to-sequence models backed with attention mechanisms [5,24] have a significant potential to become the new state-of-the-art for HTR tasks. Recurrent architectures suit the temporal nature of text, written usually from left to right, and attention mechanisms have proven to be quite performant when paired with such recurrent architectures to focus on the right features at each time step. Sequence-to-sequence (seq2seq) models follow an encoder-decoder paradigm. In our case, the encoder part consists of a Convolutional Neural Network (CNN) that extracts low-level features from the written glyphs, that are then sequentially encoded by an Recurrent Neural Network (RNN). The decoder is another RNN that will decode one character at each time step, thus spelling the whole word. An attention mechanism is introduced as a bridge between the encoder and the decoder, in order to provide a high-correlated context vector that focuses on each character's features at each decoding time step.

The contributions of this work are twofold. On the one hand, we present a novel attention-based seq2seq model, whose performance is comparable to that of other state-of-the-art approaches. Our architecture does not need any pre-processing step of the handwritten text such as de-slanting, baseline normalization, etc. The proposed approach is able to recognize the handwritten texts without the need of any predefined lexicon nor a language model. On the other hand, we also provide a deep investigation for content- and location-based attention formulations, and other strategies such as attention smoothing, multinomial sampling and label smoothing. In this paper we focus on the specific task of isolated word recognition, and we present our results in the widely known offline IAM dataset, comparing our performance with a collection of different approaches from the literature.

The rest of the paper is organized as follows. Section 2 reviews the relevant works for handwritten text recognition. Afterwards, Sect. 3 introduces the proposed architecture. Section 4 presents our experimental results and performs a comparison of the proposed method against the state-of-the-art. Finally, Sect. 5 draws the conclusions and future work.

2 Related Work

Handwritten text recognition approaches can be grouped into four different categories: HMM-based approaches, RNN-based approaches, nearest neighbor-based approaches and attention-based approaches. We will discuss methods from each of these big groups below.

HMM-based approaches were the first ones to reach a reasonable performance level [9]. Bianne-Bernard *et al.* [4] built a handwriting recognizer based on HMM, decision tree and a set of expert-based questions. Bluche *et al.* [6] proposed a method of the combination of hidden Markov models (HMM) and convolutional neural networks (CNN) for handwritten word recognition. Giménez *et al.* [11] provided a method using windowed Bernoulli mixture HMMs. However, with the rise of deep learning, such HMM proposals have been outperformed.

The second group of methods corresponds to RNN-based approaches. Graves *et al.* [12] first proposed to use Long Short-Term Memory (LSTM) cells together with the Connectionist Temporal Classification (CTC) loss to train a multi-time step output recurrent neural network. Later on, in [13] he first provided the Bidirectional Long Short-Term Memory (BLSTM) and CTC model for HTR which outperformed the state-of-the-art HMM-based models. For many years, the use of LSTM with CTC was the state of the art in handwriting recognition and many different variants were proposed. Krishnan *et al.* [15] perform word spotting and recognition by employing a Spatial Transformer Network (STN), BLSTM and CTC networks. Stuner *et al.* [23] provide a BLSTM cascade model using a lexicon verification operator and a CTC loss. Wigington *et al.* [27] perform word and line-level recognition by applying their normalization and augmentation to both training and test images using a CNN-LSTM-CTC network. However, CTC implies that the output cannot have more time steps than the input, this is usually not a problem for HTR tasks, but it is a barrier to further development towards generality and robustness. In addition, CTC only allows monotonic alignments, it may be a valid assumption for word-level or line-level HTR tasks, but it lacks the possibility for further research on paragraph or even more complex article styles.

As an alternative to RNN architectures, some authors proposed to learn embeddings that will map handwritten words to an n-dimensional space in which a nearest neighbor strategy can be applied to find the most likely transcription of a word. Almazán *et al.* [1] created a fixed length and low dimensional attribute representation known as PHOC. Krishnan *et al.* [14] proposed to learn such embeddings using a deep convolutional representation. Poznanski *et al.* [21] provided a CNN-N-Gram based method as word embedding. All the above methods have proven to correctly address the problem of multiple writers, but, as far as we know, they need a predefined lexicon, so they can not recognize out of vocabulary words, which is an important drawback.

Finally, attention-based approaches have been widely used for machine translation, speech recognition and image captioning. Recently, the interest in these approaches for HTR has arisen. Bluche *et al.* [5] propose an attention-based model for end-to-end handwriting recognition, but the features from the encoding step still needed to be pre-trained using CTC loss in order to be meaningful. This work might be the first successful trial using an attention-based model. Sueiras *et al.* [24] recently proposed a seq2seq model with attention for handwritten recognition, but they impose a sliding-window approach whose window size needs to be manually tuned which limits the representative power of the CNN features by arbitrarily limiting its field of view. In addition, in the paper they introduced some changes to the widely used Bahdanau [2] content-based attention that are not properly justified. Our seq2seq model outperforms all of those previous proposals.

3 Seq2seq Model with Attention Mechanism

Our attention-based seq2seq model consists of three main parts: an encoder, an attention mechanism and a decoder. Figure 1 shows the whole architecture proposed in this work. Let us detail each of the different parts.

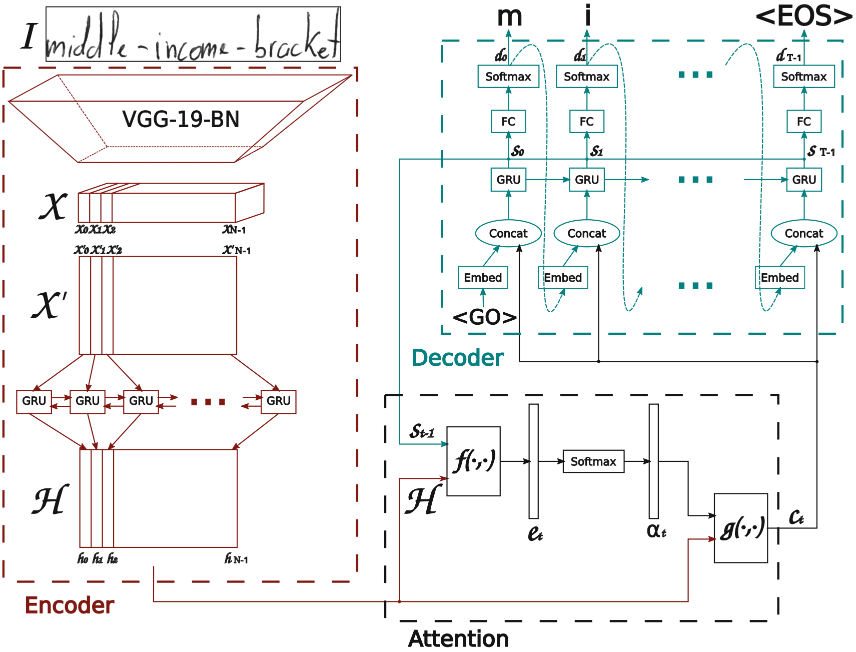


Fig. 1. Architecture of the seq2seq model with attention mechanism.

3.1 Encoder

We start with a CNN to extract visual features. Since we believe that handwritten text images are not visually as complex as real world images, we choose a reasonable CNN architecture such as the VGG-19-BN [22] and initialize it with the pre-trained weights from ImageNet. Then we introduce a multi-layered Bi-directional Gated Recurrent Unit (BGRU) which will involve mutual information and extra positional information for each column, and will encode the sequential nature of handwritten text. For VGG-19-BN network, we removed the last Max pooling layer to be able to tackle short feature sequences. So we use VGG+BGRU as an encoder to transfer the image $\mathcal{I}$ into an intermediate-level feature $\mathcal{X}$, which then is reshaped into a two-dimensional feature map $\mathcal{X}'$. The feature map $\mathcal{X}'$ can be referred as a sequence of column feature vectors $(x'_0, x'_1, \ldots, x'_{N-1})$, where N is the width of the feature map. $\mathcal{H}$ is the output of encoder which shares the same width of $\mathcal{X}'$. Each element $h_i \in \mathcal{H}$ is the output of BGRU at each time step, which will be further used to calculate attention.

3.2 Attention Mechanism

In this section we will discuss two main attention mechanisms, content-based attention and location-based attention.

Content-based Attention. The basic attention mechanism is content-based attention [2]. The intuition is to find the similarity between the current hidden state of the decoder and the word image representation feature map, thus we can find the most correlated feature vectors in the feature map of the encoder, which can be used to predict the current character at the current time step. Let us define α_t as the attention mask vector at time step t, h_i as the hidden state of the encoder at the current time step $i \in \{0, 1, \ldots, N-1\}$, s_t as the hidden state of decoder at current time step $t \in \{0, 1, \ldots, T-1\}$, where T is the maximum length of decoding characters. Then,

$$\alpha_t = \mathrm{Softmax}(e_t) \tag{1}$$

where

$$e_{t,i} = f(h_i, s_{t-1}) = w^T \tanh(Wh_i + Vs_{t-1} + b) \tag{2}$$

where w, W, V and b are trainable parameters. After obtaining the attention mask vector, the most relevant context vector can be calculated as:

$$c_t = g(\alpha_t, H) = \sum_{i=0}^{N\text{-}1} \alpha_{ti} h_i \tag{3}$$

Location-based Attention. The main disadvantage of content-based attention is that it expects positional information to be encoded in the extracted features. Hence, the encoder is forced to add this information, otherwise, content-based attention will never detect the difference between multiple feature representations of same character in different positions. To overcome it, we use an attention mechanism that takes into account the location information explicitly, *i.e.* location-based attention [8]. Thus, the content-based has been extended to be location-aware by making it take into account the alignment produced at the previous step. First we extract k vectors $l_{t,i} \in \mathbb{R}^k$ for every position i of the previous alignment α_{t-1} by convolving it with a matrix $F \in \mathbb{R}^{k \times r}$:

$$l_t = F * \alpha_{t-1} \tag{4}$$

And then, we replace Eq. 2 by:

$$e_{t,i} = f'(h_i, s_{t-1}, l_t) = w^T \tanh(Wh_i + Vs_{t-1} + Ul_{t,i} + b) \tag{5}$$

where w, W, V, U and b are trainable parameters.

Attention Smoothing. In practice, the attended area is a little narrower than the target character area of the word image. Consequently, we can infer that the model can already get the correct prediction only focusing at the narrow area. However, from the viewpoint of humans, a little wider covering area of the target character would be beneficial. For this reason, we propose to replace the Softmax Eq. 1 with the logistic sigmoid σ proposed by [8]:

$$\alpha_{t,i} = \frac{\sigma(e_{t,i})}{\sum_{i=0}^{N} \sigma(e_{t,i})} \tag{6}$$

3.3 Decoder

The decoder is a one-directional multi-layered GRUs. During each time step t, the concatenation of the embedding vector of the previous time step $\tilde{y}_{t-1}$ and the context vector c_t will be fed into the current GRU unit. The embedding vector for each character in the dataset's vocabulary comes from a look-up table matrix, which is randomly initialized and updated during the training process. The prediction of each time step t is:

$$y_t = \arg\max(\omega(s_t)) \tag{7}$$

where $\omega(\cdot)$ is a linear layer. Then we use the index to fetch the corresponding embedding vector $\tilde{y}_t$ from the look-up table matrix:

$$\tilde{y}_t = \text{Embedding}(y_t) \tag{8}$$

The decoder always starts with the start signal $\langle GO \rangle$ as first input character and ends the decoding process when the end signal $\langle EOS \rangle$ occurs or until the maximum time step T.

The previous embedding vector and current context vector are concatenated to obtain s_t, the hidden state of decoder at current time step. Thus, at each time step of the decoding, the decoder GRU can take advantage of both the information of the previous character and the potentially most relevant visual features, which will benefit the model to make correct predictions. So,

$$s_t = \text{Decoder}([\tilde{y}_{t-1}, c_t], s_{t-1}) \tag{9}$$

where $[\cdot, \cdot]$ is the concatenation of two vectors. There are two techniques that we can adopt to improve the decoding process: multi-nomial decoding and label smoothing.

Multi-nomial Decoding. Inspired by [7], during the training process, instead of choosing the character that has the highest probability from the Softmax output d_t at time step t, multiple indices can be sampled from the multi-nomial probability distribution located in the Softmax output d_t. But to keep the model simple, here we sample only one index but in a random way based on the multi-nomial probability distribution, and this index corresponds to a specific character. Although only one index has been sampled, it allows the decoder to explore other alternative decoding paths towards the final word prediction, which could make the decoder more robust and lead to better performance, although it will absolutely take longer epochs to train.

Label Smoothing. Label smoothing [26] is a regularization mechanism to prevent the model from making over-confident predictions. It encourages the model to have higher entropy at its prediction, and therefore it makes the model more adaptable and improve generalization. We regularize the groundtruth by replacing the hard 0 and 1 classification targets with targets of $\frac{\varepsilon}{k}$ and $1 - \frac{k-1}{k}\varepsilon$. In this paper, we choose the $\varepsilon = 0.4$.

4 Experiments

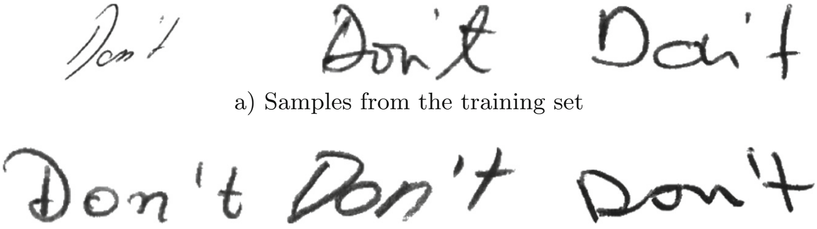
In this section, we report the experiments performed to evaluate our attention-based seq2seq model and discuss the techniques that could be potentially helpful for HTR tasks. We finally make a comparison among the state-of-the-art works.

4.1 Dataset

As the IAM Handwriting dataset [17] is the most popular one for handwritten text recognition tasks, we carried out our experiments based on it. The IAM dataset consists of 115320 isolated and labeled words written by 657 writers. For the partition, we chose the most widely used one: the RWTH Aachen partition, which consists of 55081, 8895 and 25920 words in training, validation and test sets, respectively. All of these sets are disjoint, and no writer has contributed to more than one set. We selected all the words whose segmentation are marked "OK" (even when there are some errors among the "OK" words, we still keep them), so we obtain 47981, 7554 and 20305 words in each partition. Examples of the training and test images are shown in Fig. 2.

4.2 Implementation Details

All experiments were run using the PyTorch system [19] on an NVIDIA GTX 1080 Ti. Training was done using Adam optimizer with an initial learning rate of $2 \cdot 10^{-4}$ and a batch size of 32. We set the dropout probability to be 50% for all the GRU layers except the last layer of both encoder and decoder. We have run



a) Samples from the training set

b) Samples from the testing set

Fig. 2. Samples from the IAM dataset of the word "Don't".

some experiments based on different number of layers and size of hidden state, and the final decision of these hyper-parameters will be discussed in Sect. 4.3.

All the images have been resized to a fixed height of 64 pixels while keeping the original ratio of the length/height. With the fixed height size of 64 pixels, the longest word has the length of 1011 pixels, so we padded zeros to the right of every word image so as to share the same shape of 64 × 1011.

4.3 Results

All results presented use the standard performance measures: character error rate (CER) and word error rate (WER) [10]. The CER is computed as the Levenshtein distance which is the sum of the character substitutions (S), insertions (I) and deletions (D) that are needed to transform one string into the other, divided by the total number of characters in the groundtruth word (N). Formally,

$$CER = \frac{S + I + D}{N} \tag{10}$$

Similarly, the WER is computed as the sum of the word substitutions (S_w), insertions (I_w) and deletions (D_w) that are required to transform one string into the other, divided by the total number of words in the groundtruth (N_w). Formally,

$$WER = \frac{S_w + I_w + D_w}{N_w} \tag{11}$$

Since our experiments are at word level, WER becomes the percentage of incorrectly recognized words.

At first, we need to find out relatively perfect parameters for sizes of hidden state and hidden layers of both encoder and decoder. As the hidden state of the decoder should be initialized by the encoder, we always keep the size of the hidden state and the number of hidden layers the same for both the encoder and decoder. We tried 1, 2 and 3 layers, 128, 256, 512 and 1024 sizes, being a total of 12 experiments. From the results shown in Table 1, we can observe that the relatively best parameters are 2 layers and 512 size for both the encoder and decoder.

Table 1. Validation CER comparison changing the size of the hidden state and number of layers.

Size	Number of layers		
	1	2	3
128	5.57	6.07	6.09
256	5.13	5.33	5.69
512	5.05	**5.01**	5.34
1024	5.19	5.03	5.10

Table 2. Ablation study for the proposed model tested on the IAM dataset, character error rates are computed from validation set.

Attention	AttnSmooth	Multinomial	LabelSmooth	Valid-CER	Valid-WER
Content	–	–	–	5.79	15.91
	–	–	✓	5.08	13.88
Location	–	–	–	5.49	14.74
	–	–	✓	**5.01**	**13.61**
	–	✓	–	5.53	14.53
	–	✓	✓	5.03	13.66
	✓	–	–	5.72	15.92
	✓	–	✓	5.34	14.62
	✓	✓	–	5.84	15.85
	✓	✓	✓	5.56	14.85

As detailed in Sect. 3, we explored some techniques for potential improvements. Table 2, shows that the best performance was achieved using location-based attention and label smoothing. Studying the table, we can see that the label smoothing is really helpful. The location-based attention is just slightly better than the content-based one. The reason behind this little improvement is that the use of the BGRU in the encoder can already encode some positional information to the feature map. Contrary, once we encode the positional information explicitly, the result improves. In conclusion, the location-based attention still meets our expectation.

Concerning attention smoothing and multi-nomial decoding, they seem not helping our model. On the one hand, the original Softmax attention is already good (attention visualization can be found in Figs. 3 and 4), therefore smoothing the attention may introduce noise, which could harm the model. On the other hand, multi-nomial decoding enables the proposed approach to explore new decoding paths. This exploration was expected to make our model more robust, however, it has showed that this technique is still not able to outperform our best result in the table. This probably means that the multi-nomial decoding really makes our model harder to train.

Table 3. Comparison with the state-of-the-art methods.

Idea	Method	Lexicon[a]	LM	Pre-processing	Pre-train	CER	WER
HMMs	Giménez *et al.* [11]	tr+va+te	✓	✓	–	–	25.80
	Bluche *et al.* [6]	te	✓	✓	–	–	23.70
	Bianne *et al.* [4]	tr+va+te	–	–	–	–	21.90
RNN + CTC	Mor *et al.* [18]	–	–	–	–	–	20.49
	Pham *et al.* [20]	–	–	–	–	13.92	31.48
	Krishnan *et al.* [15]	–	✓	–	Synthetic	6.34	16.19
	Wiginton *et al.* [27]	–	–	✓	–	6.07	19.07
	Stunner *et al.* [23]	2.4M	✓	–	–	**4.77**	**13.30**
Nearest Neighbor Search	Almazán *et al.* [1]	te	–	–	–	11.27	20.01
	Krishnan *et al.* [14]	te+90K	–	–	Synthetic	6.33	14.07
	Poznanski *et al.* [21]	tr+te	✓	✓	Synthetic	**3.44**	**6.45**
Attention	Bluche *et al.* [5]	–	–	–	CTC	12.60	–
	Sueiras *et al.* [24]	–	–	✓	–	8.80	23.80
	Ours	–	–	–	–	**6.88**	**17.45**

[a] Vocabulary of all words occurring in training (tr), validation (va) and test set (te). 2.4 million (2.4M) and 90 thousand (90K) words lexicon.

Table 3 shows the most popular approaches on the IAM word-level dataset, however, most of them have applied different pre-processings on the original dataset. For HMM-based approaches, Giménez *et al.* [11] corrected the slant in the image and made the gray level normalization. Bluche *et al.* [6] also corrected the slant in the image, enhanced the image contrast and added 20 white pixels on left and right to model the empty context. Bianne *et al.* [4] trained the model using all training and validation sets. These approaches have already been outperformed, since the RNN- and nearest neighbor-based approaches perform pretty well. In the case of RNN-based approaches, Mor *et al.* [18] filtered out punctuation and short words, and trained the model using training and validation sets. Krishnan *et al.* [15] has been pre-trained using synthetic data. Wiginton *et al.* [27] cleaned the punctuation and upper-cases, used the profile normalization and applied test augmentation.

Since the nearest neighbor-based approaches cannot work without lexicons, they cannot be widely used in daily or industrial use cases. In addition, Krishnan *et al.* [14] has also pre-trained using synthetic data, cleaned punctuation and upper-cases and applied test augmentation. Poznanski *et al.* [21] used a pre-trained model from synthetic data and applied test augmentation.

The bottom rows of Table 3 correspond to attention-based approaches, which are relatively new for handwriting recognition and have a significant potential for development. But Bluche *et al.* [5] has been pre-trained using CTC loss in order to get meaningful feature representation. Sueiras *et al.* [24] corrected the line skew and the slant in the images, normalized the height of the characters based on baseline and corpus line.

Among all those approaches, some of them have utilized language model (LM) explicitly. Even though no language model is used in our system, the RNN of the decoder might learn the relations between characters in the training vocabulary.

In summary, we can observe that our results are the best among the attention-based approaches and comparable to other state-of-the-art approaches especially with neither dataset pre-processing, model pre-training on synthetic dataset nor using CTC loss.

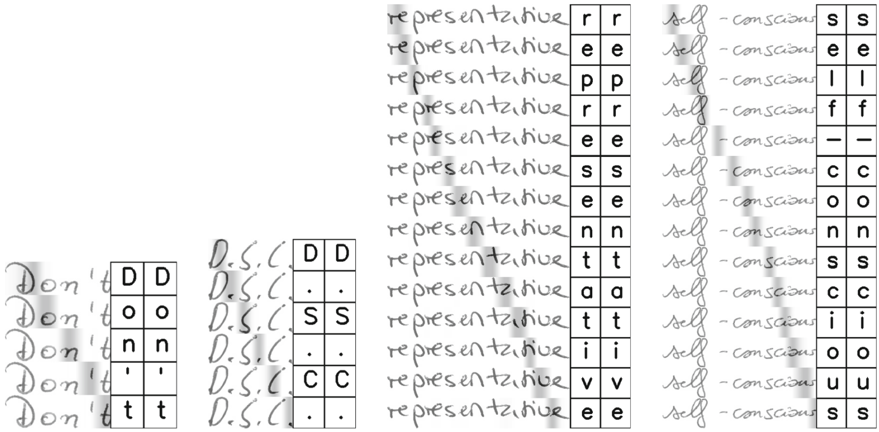


Fig. 3. Examples where the attention mechanisms correctly focuses on the right characters and we obtain a good transcription (Left: Prediction, Right: Groundtruth).

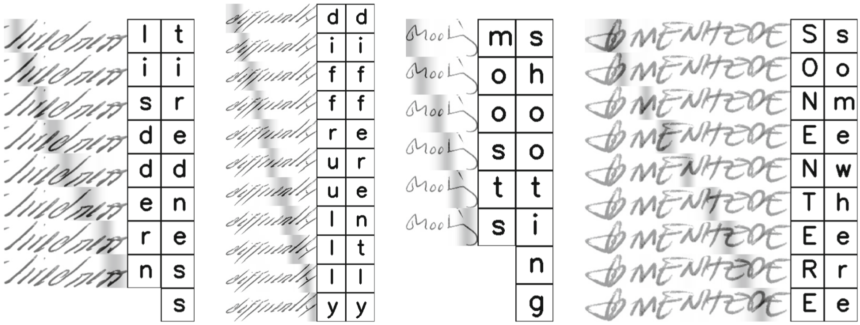


Fig. 4. Examples where we obtain an incorrect transcription (Left: Prediction, Right: Groundtruth).

4.4 Error Analysis

Some attention examples are visualized in Figs. 3 and 4. In Fig. 3, the predictions are correct and the attentions are perfectly aligned to each character as we expected. However, in Fig. 4, the errors in the first two images are due to the slant of the words and very different writing styles in comparison to the training set. Concerning the third image, if we only look at the isolated word image, actually it is hard to tell the transcription even as a human. According to this error, a language model could be used to improve the prediction. The error in the fourth image is due to the mismatch of the image and groundtruth label, while they are all upper-cases in the image but in the label all the characters are lower-cases. This last error is inevitable due to the dataset grountruth, hence, we prefer to reduce the other errors. Some approaches deployed a deslanting method or other pre-processing steps to deal with it, but there are limitations to these techniques. A video showing the evolution of such attention maps across different training epochs is provided as supplementary material.

5 Conclusion and Future Work

In this paper, we have presented Convolve, Attend and Spell, an attention-based seq2seq model for handwritten word recognition without using any of the traditional components of a HTR system, such as CTC, language model nor lexicon. It is an end-to-end system consisting of an encoder, decoder and attention mechanism. We explored various structures and strategies to improve the model, and we finally outperformed most of the state of the art methods with a 6.88% character error rate and 17.45% word error rate on IAM word-level dataset. Our future work will be focused on the application of this model to the recognition of text-lines on the IAM dataset, and to explore the incorporation of language models into the seq2seq models.

Acknowledgements. This work has been partially supported by the European Fund for Regional Development (EFRE), Pro FIT-Project "Vollautomatisierung der Wertschöpfungskette im Digitalisierungsprozess von Archivdaten" with support of IBB/EFRE in 2016/2017, the Spanish research projects TIN2014-52072-P and TIN2015-70924-C2-2-R, the grant FPU15/06264 from the Spanish Ministerio de Educación, Cultura y Deporte, the grant 2016-DI-087 from the Secretaria d'Universitats i Recerca del Departament d'Economia i Coneixement de la Generalitat de Catalunya, the Ramón y Cajal fellowship RYC-2014-16831, the AGAUR Llavor project 2016LLAV00057, the CERCA Program/Generalitat de Catalunya and RecerCaixa (XARXES, 2016ACUP-00008), a research program from Obra Social "La Caixa" with the collaboration of the ACUP. We gratefully acknowledge the support of the NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

References

1. Almazán, J., Gordo, A., Fornés, A., Valveny, E.: Word spotting and recognition with embedded attributes. IEEE Trans. Pattern Anal. Mach. Intell. **36**(12), 2552–2566 (2014)
2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
3. Bahdanau, D., Chorowski, J., Serdyuk, D., Brakel, P., Bengio, Y.: End-to-end attention-based large vocabulary speech recognition. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 4945–4949 (2016)

4. Bianne-Bernard, A.L., Menasri, F., Mohamad, R.A.H., Mokbel, C., Kermorvant, C., Likforman-Sulem, L.: Dynamic and contextual information in HMM modeling for handwritten word recognition. IEEE Trans. Pattern Anal. Mach. Intell. **33**(10), 2066–2080 (2011)
5. Bluche, T., Louradour, J., Messina, R.: Scan, attend and read: end-to-end handwritten paragraph recognition with MDLSTM attention. In: Proceedings of the IAPR International Conference on Document Analysis and Recognition, pp. 1050–1055 (2017)
6. Bluche, T., Ney, H., Kermorvant, C.: Tandem HMM with convolutional neural network for handwritten word recognition. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 2390–2394 (2013)
7. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
8. Chorowski, J.K., Bahdanau, D., Serdyuk, D., Cho, K., Bengio, Y.: Attention-based models for speech recognition. In: Proceedings of the International Conference on Neural Information Processing Systems, pp. 577–585 (2015)
9. España-Boquera, S., Castro-Bleda, M.J., Gorbe-Moya, J., Zamora-Martinez, F.: Improving offline handwritten text recognition with hybrid HMM/ANN models. IEEE Trans. Pattern Anal. Mach. Intell. **33**(4), 767–779 (2011)
10. Frinken, V., Bunke, H.: Continuous handwritten script recognition. In: Doermann, D., Tombre, K. (eds.) Handbook of Document Image Processing and Recognition, pp. 391–425. Springer, London (2014). https://doi.org/10.1007/978-0-85729-859-1_12
11. Giménez, A., Khoury, I., Andrés-Ferrer, J., Juan, A.: Handwriting word recognition using windowed Bernoulli HMMs. Pattern Recogn. Lett. **35**, 149–156 (2014)
12. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the International Conference on Machine Learning, pp. 369–376 (2006)
13. Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., Schmidhuber, J.: A novel connectionist system for unconstrained handwriting recognition. IEEE Trans. Pattern Anal. Mach. Intell. **31**(5), 855–868 (2009)
14. Krishnan, P., Dutta, K., Jawahar, C.: Deep feature embedding for accurate recognition and retrieval of handwritten text. In: Proceedings of the International Conference on Frontiers in Handwriting Recognition, pp. 289–294 (2016)
15. Krishnan, P., Dutta, K., Jawahar, C.: Word spotting and recognition using deep embedding. In: Proceedings of the IAPR International Workshop on Document Analysis (2018)
16. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
17. Marti, U.V., Bunke, H.: The IAM-database: an English sentence database for offline handwriting recognition. Int. J. Doc. Anal. Recogn. **5**(1), 39–46 (2002)
18. Mor, N., Wolf, L.: Confidence prediction for lexicon-free OCR. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision, pp. 218–225 (2018)
19. Paszke, A., et al.: Automatic differentiation in PyTorch (2017)
20. Pham, V., Bluche, T., Kermorvant, C., Louradour, J.: Dropout improves recurrent neural networks for handwriting recognition. In: Proceedings of the International Conference on Frontiers in Handwriting Recognition, pp. 285–290 (2014)

21. Poznanski, A., Wolf, L.: CNN-N-gram for handwriting word recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2305–2314 (2016)
22. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
23. Stuner, B., Chatelain, C., Paquet, T.: Handwriting recognition using cohort of LSTM and lexicon verification with extremely large lexicon. CoRR, vol. abs/1612.07528 (2016)
24. Sueiras, J., Ruiz, V., Sanchez, A., Velez, J.F.: Offline continuous handwriting recognition using sequence to sequence neural networks. Neurocomputing **289**, 119–128 (2018)
25. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Proceedings of the International Conference on Neural Information Processing Systems, pp. 3104–3112 (2014)
26. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)
27. Wigington, C., Stewart, S., Davis, B., Barrett, B., Price, B., Cohen, S.: Data augmentation for recognition of handwritten words and lines using a CNN-LSTM network. In: Proceedings of the IAPR International Conference on Document Analysis and Recognition, pp. 639–645 (2017)
28. Xu, K., et al.: Show, attend and tell: Neural image caption generation with visual attention. In: Proceedings of the International Conference on Machine Learning, pp. 2048–2057 (2015)