CONTEXT-DEPENDENT BLSTM MODELS. APPLICATION TO OFFLINE HANDWRITING RECOGNITION

Youssouf Chherawala, Partha Pratim Roy, Mohamed Cheriet

Synchromedia Laboratory, École de Technologie Supérieure, Montreal (QC), Canada

ABSTRACT

The BLSTM model has been recently introduced for sequence labeling tasks and provides state-of-the-art performance for handwriting recognition. Its recurrent connections integrate the context at the feature level over a long range. Nevertheless, this context is not explicitly modeled at the label level. Explicit context-modeling strategies have been applied to HMMs with improvement of the recognition rate. In this paper, we study the effect of context modeling on the performance of the BLSTM model. The baseline BLSTM, with context-independent character label, is compared with two context-dependent BLSTM, one modeling the left context and the other the right context. The results show that context-dependent models provide an improvement of the recognition rate, and demonstrate the ability of the BLSTM model to deal with a large number of models, without clustering. We tested our models on the RIMES database of Latin script documents.

Index Terms— Handwriting recognition, BLSTM, context-dependent model, RIMES database

1. INTRODUCTION

The bidirectional long short-term memory (BLSTM) [1] neural network has been recently introduced for sequence labeling tasks and provides state-of-the-art performance for handwriting recognition. It shows better results than the well-known HMM model in recent studies, thanks to its discriminative training and recurrent connections that integrate the context at the feature level over a long range. Nevertheless, this context is not explicitly modeled at the label level. Explicit context modeling strategies have been applied to HMMs with improvement of the recognition rate. Indeed, in cursive scripts, the shape of a character is dependent of its context, usually described by the characters preceding and following it. This is illustrated in Figure 1. Schüßler [2] used a hierarchy of context, from monographs, bigraphs, trigraphs and word models during the HMM recognition. Fink et al. [3] showed that the improvement using context-specific models for handwriting recognition is marginal compared to its application to speech recognition. El-Hajj [4] applied context-dependent models to Arabic handwriting recognition, where the context is defined by the presence of an overlapping character in the sliding window. Natarajan [5] applied left and right context in their multi-lingual offline handwriting recognition system. One issue of context-dependent models is the lack of data to accurately learn each model. Bianne-Bernard et al. [6] proposed a knowledge driven strategy for state clustering to efficiently tackle this limitation.

In this paper, we study the effect of context modeling on the performance of the BLSTM model. It is the main novelty of the paper and we are the first to investigate on it. The baseline BLSTM, with context-independent character label is compared with two context-dependent BLSTM, one modeling the left context and the other the right context. The results show that context-dependent models provide an improvement of the recognition rate, and demonstrate the ability of the BLSTM model to deal with a large number of models, without model clustering. In this paper, we are specifically interested in Latin-script documents.

The paper is organized as follows. The background on the BLSTM model and the type of context models investigated are presented in Section 2. The preprocessing steps are detailed in Section 3. The experiments are detailed and discussed in Section 4. Finally, the conclusions are drawn in Section 5.
2. BLSTM

The BLSTM is a recurrent neural network, that is, connections between nodes form a directed cycle. It provides a ‘memory’ of the previous network internal state. In this section, we first describe the LSTM hidden layer and the CTC output layer, both forming the BLSTM network, followed by our context modeling strategies.

2.1. Long Short-Term memory (LSTM) layer

The LSTM network nodes have a specific architecture, referred as memory block. Each memory block contains a memory cell, and its interaction with the rest of the network is controlled by three gates, namely: an input gate, an output gate and a forget gate. This allows the memory cell to preserve its state over a long range of time and to model the context at the feature level. The 1D sequence recognition is improved by processing the input signal in both directions, i.e., one layer processes the signal in forward direction while another layer processes it in backward direction. The output of both layers is combined at the next layer as a feature map. Similarly to convolutional neural network architecture, it is possible to have multiple forwards and backward layers in each LSTM layer as well as multiple feature maps at the output layer, and to stack multiple LSTM layers using max-pooling subsampling.

2.2. Connectionist Temporal Classification (CTC) layer

Usually, most of the RNNs require pre-segmented training data or postprocessing to transform its outputs into transcriptions. To avoid such process, the CTC output layer has been designed for sequence labeling. This layer is trained to predict the probability of an output label sequence given an input sequence. The output activation function provides the probability to observe each character for each sequence time. The objective function of CTC is defined as the negative log probability to observe each character for each sequence. The output activation function provides the probability of an output label sequence given an input sequence. The objective function of CTC is defined as the negative log probability of the network correctly labeling the entire training set.

Once the network is trained, the labeling of an unknown sequence of observation $O$ is performed by choosing the word $\hat{w}$ from a given lexicon having the highest conditional probability:

$$\hat{w} = \arg \max_w P(w|O)$$

(1)

2.3. Context modeling strategies

In this paper we compare the performance of the BLSTM model with three different context-modeling strategies. The first is just the context-independent (CI) character model, and it will serve as our baseline BLSTM. The second model considers the left context (LC) of each character, i.e., the character preceding the current one. The third model considers the right context (RC), i.e., the character following the current one. The French word ‘transmis’ is modeled by the following label sequence for each strategy:

- **Context independent**: $w = \text{t’ t’ a’ n’ s’ m’ i’ s’}$.
- **Left context**: $w = \text{#t’ t’ ra’ an’ ns’ sm’ mi’ is’}$.
- **Right context**: $w = \text{tr’ ra’ an’ ns’ sm’ mi’ is’ s#}$.

where # represents the word boundary.

3. PREPROCESSING AND FEATURE EXTRACTION

Grey-level images are binarized using the Otsu algorithm [7]:

$$T = \arg \max_T \left( \frac{\sigma_{\text{bet}}^2}{\sigma_{\text{tot}}^2} \right)$$

(2)

where $T$ is the threshold, and $\sigma_{\text{bet}}^2, \sigma_{\text{tot}}^2$ are the between-class and total variances respectively. Binarization allows the extraction of the word shape. Then, the image is transformed into a sequence of observation $O = \{o_1, o_2, \ldots, o_N\}$, by scanning the image from left to right and extracting the feature proposed by Marti and Bunke [8] from each column. More precisely, each observation $o_i$ is made of a 9D vector of distribution features from the following categories:

- **Global features**: the number of black pixels, their center of gravity and second order moment.
- **Profile features**: the upper and lower profiles, the number of black to white pixel transition, and the distance between the upper and lower profiles.
- **Dynamic features**: the gradient of the upper and lower profiles with respect to the previous column.

This feature has been chosen for its state-of-the-art performance for cursive handwriting recognition [9], this choice is further motivated in the experimental section. The main parts of the preprocessing are illustrated in Figure 2. No other prepossessing steps have been applied to the image, such as deslanting or skew correction, although they could have improved the recognition result.

4. EXPERIMENTS

4.1. Database and protocol

For our experiments, we used the RIMES database [10]. It is composed of French words split into training, validation and test sets, containing 59,203, 7,542, and 7,464 images respectively. We used the 2009 version of this database; the lexicon is composed solely of words from the test set and contains 1,612 words.

We used the following protocol for the training of the BLSTM models used in our experiments. Five instances of
each model are trained because of the random initialization of the neural network parameters. The BLSTM learning rate has been set to $10^{-4}$ with a momentum of 0.9. The training stops if there is no improvement of the character-level error rate on the validation set for 20 epochs. We used the RNNLIB implementation of the BLSTM [11].

### 4.2. Results and discussions

In a first experiment, we justify the choice of the Marti and Bunke feature. For this, we compare three context-independent BLSTM models trained with different features. The first feature is the Marti and Bunke one, described in Section 3. The second is the MDLSTM neural network [12], which is a multidimensional variant of the BLSTM whose hidden layers act as trainable 2D image filters. The last one is the local gradient histogram (LGH) [13] which has been inspired by the computer-vision literature and is similar to the SIFT feature. The recognition results are shown in Table 1. The Marti and Bunke feature obtains a recognition rate of more than 1.3% on average above the other features. Therefore, it is a good candidate for our context-dependent models and we use it for the next experiment.

The CI BLSTM model is compared with the LC and RC BLSTMs. The results are shown in Table 2. The LC model provides a recognition rate improvement of 0.9% over the baseline CI model (difference considered statistically significant under the $t$ test) and of 0.4% over the RC model. Both context-dependent models perform better than the context-independent model. Also, LC provides a better modeling than RC; this was expected because of the left to right writing direction.

If we have a closer look at the statistics of the models (Table 3), we notice that the BLSTM is able to cope with a large number of context-dependent models, which improve the recognition rate without increasing the processing time. This is remarkable because all the context-dependent models have been considered without any clustering.

The improvement provided by context-dependent models for the HMM and BLSTM models are compared in Table 4. The relative word error rate ($\Delta \text{WER} = (\text{CD} - \text{CI}) / (100 - \text{CI}) \times 100$) improvement range between 6.3% to 23.4% for HMM, and it is of 9.1% for the LC BLSTM. The effect of context-dependent models on the BLSTM and HMM is in the same range. It would be possible to further improve the $\Delta \text{WER}$ of the BLSTM by using trigraph modeling (left and right context simultaneously) combined with an advanced model clustering technique [6].

We show in Table 5 qualitative recognition results for dif-

### Table 1. Comparison of different features for the BLSTM model. (Best values highlighted.)

<table>
<thead>
<tr>
<th>Repetition</th>
<th>Marti-Bunke</th>
<th>MDLSTM</th>
<th>LGH</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>90.0</td>
<td>87.9</td>
<td>88.1</td>
</tr>
<tr>
<td>2</td>
<td>89.9</td>
<td>87.6</td>
<td><strong>88.7</strong></td>
</tr>
<tr>
<td>3</td>
<td>89.9</td>
<td>89.4</td>
<td>88.7</td>
</tr>
<tr>
<td>4</td>
<td><strong>90.7</strong></td>
<td><strong>89.7</strong></td>
<td>88.6</td>
</tr>
<tr>
<td>5</td>
<td>90.1</td>
<td>89.6</td>
<td>88.5</td>
</tr>
<tr>
<td>Average</td>
<td>90.1 ± 0.3</td>
<td>88.8 ± 1.0</td>
<td>88.5 ± 0.3</td>
</tr>
</tbody>
</table>

### Table 2. Recognition accuracy (%) with different contexts. (Best values highlighted.)

<table>
<thead>
<tr>
<th>Repetition</th>
<th>CI</th>
<th>LC</th>
<th>RC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>90.0</td>
<td>90.2</td>
<td>90.3</td>
</tr>
<tr>
<td>2</td>
<td>89.9</td>
<td>90.9</td>
<td><strong>91.3</strong></td>
</tr>
<tr>
<td>3</td>
<td>89.9</td>
<td>90.8</td>
<td>91.1</td>
</tr>
<tr>
<td>4</td>
<td><strong>90.7</strong></td>
<td>91.4</td>
<td>90.8</td>
</tr>
<tr>
<td>5</td>
<td>90.1</td>
<td><strong>91.6</strong></td>
<td>89.4</td>
</tr>
<tr>
<td>Average</td>
<td>90.1 ± 0.3</td>
<td>91.0 ± 0.5</td>
<td>90.6 ± 0.8</td>
</tr>
</tbody>
</table>

### Table 3. Models statistics

<table>
<thead>
<tr>
<th>Context type</th>
<th>Number of models</th>
<th>Avg. proc. time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI</td>
<td>80</td>
<td>145</td>
</tr>
<tr>
<td>LC</td>
<td>1378</td>
<td>134</td>
</tr>
<tr>
<td>RC</td>
<td>1375</td>
<td>137</td>
</tr>
</tbody>
</table>
Table 4. Comparison of the recognition rate (%) improvement with context-dependent models (CD) with HMM.

<table>
<thead>
<tr>
<th>Model</th>
<th>CI</th>
<th>CD</th>
<th>∆ WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM-Natarajan [5]</td>
<td>47.2</td>
<td>50.7</td>
<td>6.6</td>
</tr>
<tr>
<td>HMM-Fink [3]</td>
<td>76.0</td>
<td>77.5</td>
<td>6.3</td>
</tr>
<tr>
<td>HMM-Bianne-Bernard [6]</td>
<td>73.0</td>
<td>79.3</td>
<td>23.4</td>
</tr>
<tr>
<td>Proposed BLSTM</td>
<td>90.1</td>
<td>91.0</td>
<td>9.1</td>
</tr>
</tbody>
</table>

Different context modeling. The LC and RC models provide better recognition than the CI model for slanted handwriting and for images with ambiguous character segmentation. Therefore, context modeling has the potential to accurately model slanted handwriting. However, the context-dependent models show poor performance for printed handwriting style. Context modeling has no influence in such cases because the characters are independent to each other’s. Using a proper style identification technique, it would be possible to direct each image to the most appropriate model (CI, LC, or RC) for a better recognition rate. Finally, images with bad word segmentation, disproportionate characters and ambiguous character shapes are not recognized by any model.

5. CONCLUSIONS

In this paper, the effect of context-dependent models on the BLSTM neural network has been investigated for the first time. We explored the left-context and right-context models. The results show a relative error rate improvement of 9.1% with the left-context model compared to the context-independent one. This result has been obtained without model clustering and with no impact on the processing time. For future work, it will be interesting to see if the BLSTM would benefit from trigraph context. It leads to a total of 5472 context-dependent models on the RIMES database. One possible challenge will be to accurately learn such a high number of character models.

6. ACKNOWLEDGMENTS

The authors would like to thank NSERC of Canada for their financial support.

7. REFERENCES


