IEEE Second International Conference on Data Stream Mining & Processing
August 21-25, 2018, Lviv, Ukraine

Improvement of Character Segmentation using Recurrent Neural Networks and Dynamic Programming

Valentyna Volkova
Samsung R&D Institute Ukraine (SRK)
Kyiv, Ukraine
v.volkova@samsung.com

Ivan Deriuga
Samsung R&D Institute Ukraine (SRK)
Kyiv, Ukraine
i.deriuga@samsung.com

Vadym Osadchyi
Samsung R&D Institute Ukraine (SRK)
Kyiv, Ukraine
vad.osadchiy@samsung.com

Olga Radyvonenko
Samsung R&D Institute Ukraine (SRK)
Kyiv, Ukraine
oradivonenko@gmail.com

***Abstract*—A common characteristic of all the existing online handwritten text recognition algorithms is that the character segmentation process is closely related to the recognition process. There are different approaches to segment data but all of them don't give absolutely correctly segmentation results due to specifics of handwriting data input. In this paper, we present a new approach for character segmentation improvement in online handwriting recognition which is based on using recurrent neural networks and dynamic programming. Due to online handwritten text is a sequence of points we propose to use Bidirectional Long Short-Term Memory (BLSTM) for classification of decoder outputs and dynamic programming for interpretation of classification results. Experimental evaluation shows the effectiveness of a proposed approach in increasing of segmentation quality.**

***Index Terms*—online handwriting recognition, character segmentation, recurrent neural networks, dynamic programming**

I. Introduction

Handwriting recognition applications are popular and useful in business, education and other because allow to make quickly handwritten notes and convert them into printed text. Also, handwriting input is an alternative to using keyboards for pen-based, touch-based tablets and smartphones [1] – [2].

Character segmentation is an important part of online handwriting recognition process. The goal of character segmentation is to partition of a handwritten text into segments, each containing an isolated and complete character.

Distinguish several types of segmentation for handwriting recognition: line, word and character segmentation. Each type is important and has a big influence on recognition result in general. Especially it influences on recognition such entities as formulae where the quality of relations between symbols is very important. Also, it is useful in solving of the ink beautification problem where every handwritten stroke in a text has to be modified in a better way to beautify the general text representation. Such correction depends not only on geometric characteristics of a stroke but also on a quality of character segmentation [3].

This work considers a character segmentation task as an important step in the recognition process and designs the method of segmentation improvement based on using of recurrent neural networks (RNN) and dynamic programming.

The paper has next structure: introduction, related publications with short overview represented in section 2. Section 3 describes the proposed model architecture, feature selection, and proposed dynamic programming. Experimental results are presented in section 4. The conclusion of the paper is given in Section 5.

II. Background

Character segmentation is an operation that attempts to decompose a sequence of strokes into subsequences of individual strokes. It is one of the crucial processes in a system for online handwriting recognition.

There are exist different segmentation algorithms [4] – [6]. And all of them don't give absolutely correctly segmentation results or aimed at solving a word, line segmentation problems.

For character segmentation usually are used classifiers such as SVM, BLSTM, decision trees and dynamic programming [7] – [12]. These methods are applied after removing delayed strokes from the handwritten text and potential breakpoints. Breakpoints are detected after the shape analysis of the stroke trajectory to find the best segmentation point for each character. These approaches allow finding a global optimal path of segments.

The difficulties in segmenting of handwritten text arise due to the following factors:

1) characters in cursive writing usually are connected;
2) character shaping depends on its position in the word and what characters are next;
3) neighboring characters in a word may overlap;

978-1-5386-2874-4/18/$31.00 2018 IEEE

4) delayed strokes (the dot of "i", "j" or the crossing "f", "t", "H", "F", "E");
5) the variance of writing styles.

Most of considered approached use a pre-segmentation step for decoding improvement. Specifically, character segmentation is used to speed-up the decoding by pruning word lattice on only segmentation points. The goal of this paper is to improve the character segmentation after decoding without affecting of the recognition result.

III. Proposed approach

A. Model architecture

To improve the quality of character segmentation it is proposed to use RNN which corrects the segmentation points received from the output of the decoder. Segmentation point is a transition between recognition entities (letters, digits etc.). The goal of this RNN is to classify input strokes into N classes: $N-1$ classes are potentially character segments and "0" class reserved for marking of delayed strokes. A number of classes can be different. It is better to use one class for delayed strokes and three or more for segments.

In this work we use three classes for segments and experiments have shown that such amount is sufficient to obtain a good quality of character segmentation. As result, the input sequence of points is marked as four classes. The class change in this sequence shows the beginning of a new character (Fig. 1).

Fig. 1. Example of marked by classes segments.

A general scheme of the proposed algorithm is shown in Fig. 2. Due to input handwritten data is represented as a sequence of points we propose to use BLSTM [13] for classification of decoder outputs.

According to machine learning principles, an input dataset have to be divided into train, validation and test sets and preliminary preprocessed.

To preprocess data we propose to use the following algorithms: size normalization, density normalization of points (interpolating for missing points and resampling if there is overmuch of points), Bezier smoothing, slant and skew correction [14]. Multilines have to be separated into single lines.

Preprocessed samples are fed to a handwriting recognition neural network. As already said for online text recognition we propose to use a BLSTM neural network which is well-proven in solving such type of problems.

From preprocessed samples, there are extracted features which fed to the input of recognition BLSTM.

As recognition BLSTM in this work is used decoder from the RNNLIB library [15]. RNNLIB is a recurrent neural network library for sequence learning problems which has proven particularly effective for speech and handwriting recognition.

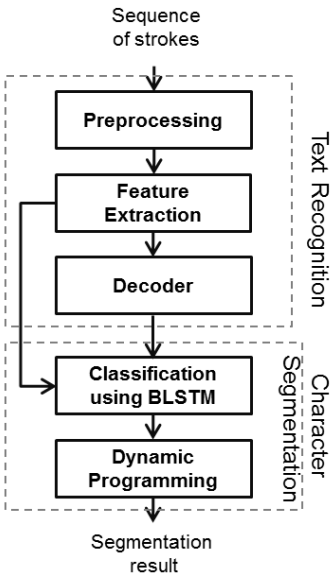

Fig. 2. General scheme of proposed approach.

After the recognition training is done we mark the obtained sequence of points according to the output of decoder (decoding results) using segmentation BLSTM (Fig. 3) as it was described earlier. Decoding results can be approved by implementation of the token passing algorithm [17]. The proposed segmentation classificator contains two BLSTM hidden layers. Every layer consists of 20 cells.

As inputs of segmentation BLSTM are four features. Three of them are extracted on the text recognition step and the last one the output of decoder (the recognition BLSTM) which represents preliminarily segmented by frames characters. More detailed description of this features is given in Section 3.2.

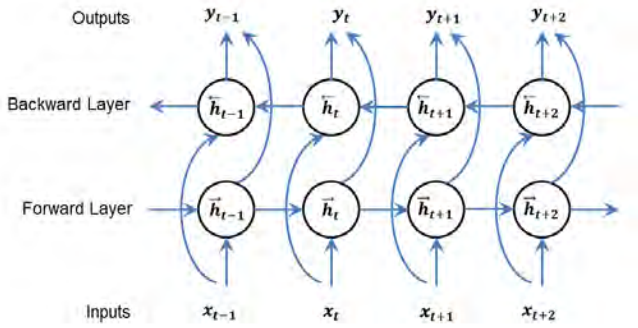

Fig. 3. Segmentation BLSTM.

The network is trained to minimize the cross-entropy error of the targets using a softmax output layer. For interpretation of obtained results, we apply the dynamic programming described in Section 3.3.

B. Feature extraction

The common set of features for character segmentation is given in the Table I.

TABLE I
FEATURES FOR CHARACTER SEGMENTATION

Feature	Description
$f_{\triangle x}$	delta x coordinates
$f_{\triangle y}$	delta y coordinates
f_{upd}	pen-up/pen-down. Means the points in the sequence when the individual strokes end
f_{output}	output of the previous stage (sequence of points marked according to output of decoder after recognition)

At the input of the segmentation neural network are fed four features: f_{output} which represents decoding results and $f_{\triangle x}$, $f_{\triangle y}$, f_{upd} obtained on the feature extraction stage for the recognition BLSTM training.

Features $f_{\triangle x}$, $f_{\triangle y}$ and f_{upd} are extracted from the input handwritten sequence of strokes on a online text recognition stage. These characteristics are classic for online handwriting recognition using RNN and are used in the recognition training.

As the fourth feature can be used a raw decoder output when each frame is marked with a letter code. Using of such feature is possible because a codebook for concrete language is limited (less than 256 characters) and thus it can be normalized. Such feature gives some segmentation improvement but is sensitive to a language.

To achieve character segmentation which doesn't depend on language codebook there can be used the binarization of decoding output. In this case segmentation points (frames) are marked as "1" and all other regular frames are marked as "0". This feature gives a smaller improvement comparing to the previous one but can be used for different language groups.

In this paper as fourth feature we propose to use the specific representation of a decoder output with its initial segmentation which has to be further improved. For this we select N segmentation classes for ordinary char segments and one special for delayed strokes ("0" class). All frames which belong to one character segment are marked with the same class. All delayed stroke frames are marked with class "0". Each character segment is marked in ordered way starting from "1" class and following class numbers, when the next non-delayed stroke segment has a next class number by module N.

Using of this feature gives better segmentation results than previous one and in this case segmentation is language independent.

C. Dynamic programming

Neural network output has to be properly interpreted. Outputs of the segmentation RNN discussed in this work can be considered as probabilities $\widetilde{P}(\omega_t)$ of a frame belonging t (point) to a segment class ω. By analogy with the output feature there was defined four segment classes: from "1" to "3" are for regular segments and "0" class is for delayed strokes. Then a character segmentation can be formulated as a classical maximization likelihood decoding problem:

$$\begin{aligned} &L(\{\omega\}_{t=\overline{1,T}};\{(x,y)\}_{t=\overline{1,T}}) = \\ &= P(\{\omega\}_{t=\overline{1,T}}|\{(x,y)\}_{t=\overline{1,T}}) = \\ &= \prod_{t=1}^{T} P(\omega_t|\{\omega\}_{t'<t};\{(x,y)\}_{t=\overline{1,T}}) = \prod_{t=1}^{T}\widetilde{P}(\omega_t), \end{aligned} \tag{1}$$

$$\{\omega_t^0\}_{t=\overline{1,T}} = \underset{\{\omega_t\}_{t=\overline{1,T}}}{\operatorname{argmax}} L(\{\omega\}_{t=\overline{1,T}};\{(x,y)\}_{t=\overline{1,T}}), \tag{2}$$

where $\{\omega_t^0\}_{t=\overline{1,T}} = \omega_1^0,\ldots,\omega_T^0$ is a sequence of segment classes with a special restriction for segment numbers K.

From one side there is known an expected segment number. It is equal to a length of a character sequence output of decoder without counting of spaces. In this case, one character corresponds to one segment. From another side we can count a character segments number K from a sequence $\{\omega\}_{t=\overline{1,T}}$ in a next way:

$$K = |S(\{\omega_t^0\}_{t=\overline{1,T}})| = |\{\widetilde{\omega}_t^0\}_{t=\overline{1,T'}}| = T', \tag{3}$$

where S is a sequence transformation which suppose removing of $\omega_{t_0}^0$ from $\{\omega_t^0\}_{t=\overline{1,T}}$ if $\omega_{t_0}^0 = 0$ or $\omega_{t_0-1}^0 = \omega_{t_0}^0$. After such transformation we obtain a reduced sequence $\{\widetilde{\omega}_t^0\}_{t=\overline{1,K}}$ with a length equal to a character segment count.

Previously many researchers noted relation between a maximum likelihood decoding and DTW algorithm [17]. Following this idea there was decided to apply dynamic programming with a logarithmic likelihood estimation function:

$$\begin{aligned} &\log L(\{\omega\}_{t=\overline{1,T}};\{(x,y)\}_{t=\overline{1,T}}) = \\ &= \sum_{t=1}^{T}\log\widetilde{P}(\omega_t) = \sum_{t=1}^{T} c_t(\omega_t), \end{aligned} \tag{4}$$

where $c_t(\omega_t) = \log\widetilde{P}(\omega_t)$.

$$\{\omega_t^0\}_{t=\overline{1,T}} = \underset{\{\omega_t\}_{t=\overline{1,T}}}{\operatorname{argmax}} \log L(\{\omega\}_{t=\overline{1,T}};\{(x,y)\}_{t=\overline{1,T}}). \tag{5}$$

In terms of dynamic programming there were defined conventions (Table II).

TABLE II
CONVENTIONS.

Designation	Description
$dp_t(k,s)$	the cost function of traversed path to segment k of time t, where $t = \overline{1,T}$
T	a general number of frames
k	a number of current segment, $k = \overline{1,K}$
K	a total number of segments
s	a binary state which represents a delayed stroke with values $s = \{0,1\}$

Relation between segment number k and segment class can be described as follows:

$$\omega = k \bmod N + 1 \text{ if } s = 1 \text{ and } \omega = 0 \text{ if } s = 0, \tag{6}$$

where N is a number of segment classes, in this paper $N = 3$. There is a special extra class for delayed strokes and thus total number of segment classes is $N + 1$.

Also, there were applied the following restrictions for dynamic programming related to segment classes possible transitions:

1) the first class can be changed only by the second or by the delayed class, or not changed;
2) the second class can be changed only by the third or the delayed classes, or not changed;
3) the third class can be changed only by the first class or the delayed classes, or not changed.
4) a delayed stroke can be only the stroke as a whole;
5) the zero class can be changed by the previous one.

Possible state transitions between classes are presented in Fig. 4.

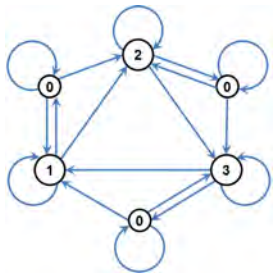

Fig. 4. Transitions between segmentation classes.

Taking into account described restrictions the traversing path for proposed dynamic programming is given in Fig. 5

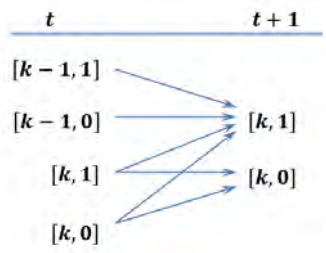

Fig. 5. Traversing path.

Recurrent equations for cost functions can be represented as follows:

Initial state:

$t = 0$

$k = 1$

$s = 1$

$dp_0(1, 1) = c_0(1)$

General step:

$$dp_{t+1}(k, 1) = \\ = \max \begin{cases} dp_t(k, 1) + c_{t+1}(k \bmod N + 1); \\ dp_t(k - 1, 1) + c_{t+1}((k - 1) \bmod N + 1); \\ dp_t(k - 1, 0) + c_{t+1}((k - 1) \bmod N + 1) \\ \text{if the stroke begins in current frame;} \\ dp_t(k, 0) + c_{t+1}(k \bmod N + 1) \\ \text{if the stroke begins in current frame,} \end{cases} \quad (7)$$

$$dp_{t+1}(k, 0) = \max \begin{cases} dp_t(k, 0) + c_{t+1}(0); \\ dp_t(k, 1) + c_{t+1}(0); \end{cases} \quad (8)$$

Next state is given by

$$(k_{t+1}, s_{t+1}) = \\ = \underset{(k,s)}{\operatorname{argmax}}(dp_{t+1}(k_t, 0), dp_{t+1}(k_t, 1), dp_{t+1}(k_{t+1}, 1)) \quad (9)$$

IV. Experiments

Experiments were conducted on IAM online database (IAMonDo) [16] and handwritten dataset (HWRD) collected for training and testing of the proposed solution.

Both datasets consist of pen trajectories collected from different writers using a whiteboard (IAMonDo) and pen on a smartphone or hand on a tablet. The HWRD contains 11297 sequences for the English language. IAMonDo contains about 3859 sequences for the English language, a number of sequences with 100% recognition accuracy is 2043. For the experiment evaluation was selected 3170 sequences with the same length and which give 95% of recognition accuracy.

Datasets were divided into train, validation and test sets and preliminary preprocessed. From preprocessed samples there were extracted features $f_{\triangle x}$, $f_{\triangle y}$ and f_{upd} (Table I). After that, they are fed to the input of recognition BLSTM which consist of two hidden layers and 100 cells in every layer. The training was accelerated using sequence bucketing and data parallelization [19]. After recognition training is done we obtain the sequence of points marked according to an output of decoder (decoding results) which represent the f_{output} feature.

As inputs of segmentation BLSTM are four features: f_{output} which represents decoding results and $f_{\triangle x}$, $f_{\triangle y}$, f_{upd} obtained on feature extraction stage for the recognition BLSTM training.

The structure of trained segmentation neural network is next: an input layer, two forward layers, one layer which combines outputs of forward layers, two backward layers, one layer which combines outputs of backward layers and an output layer. Total segmentation network consists of 7 layers.

Trained segmentation BLSTM contains 14164 weights. Training data (HWRD dataset) has 6848 sequences and 931343 frames in sequences. The average ratio of frames per sequence is 136.00. Validation data (HWRD dataset) contains 1488 sequences, 226969 frames in sequences, the average ratio of frames per sequence is 152.53.

On 1st epoch of the training classification error was equal to 20.08%, cross entropy error was equal to 76.03%. The lowest classification error was obtained on 55 epoch and is equal to 0.48%, cross entropy error is 2.22%.

Fig. 6 illustrates an example of character segmentation by RNNLIB and proposed approach.

Train and validation errors are shown in Fig. 7.

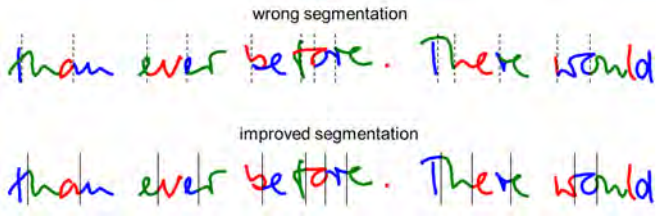

Fig. 6. Example of segmentation.

Here the first line shows segmentation results of text recognition system based on RNNLIB and the second one shows results of the proposed approach. The vertical line shows the beginning of a new segment for cases with wrong segmentation. From the figure follows that unlike the RNNLIB in proposed approach crossing characters were segmented more correctly. Such cases can be fixed using rule-based approach, but it is difficult to find all cases of wrong segmentation and to cover them by rules.

Character segmentation results are given in the Table III.

TABLE III
SEGMENTATION RESULTS

Dataset	Initial Segmentation Accuracy, %	Final Segmentation Accuracy, %	Delta, %
IAMonDO	91.35	98.75	7.4
HWRD	89.2	98.81	9.61

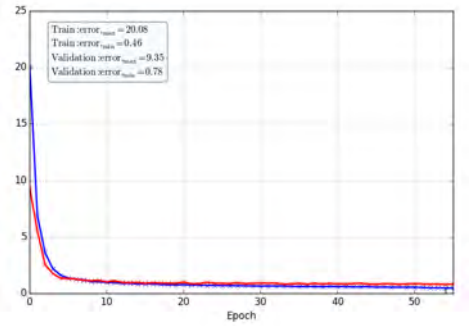

Fig. 7. Error curves.

V. CONCLUSION

In this paper, a new approach for character segmentation improvement of on-line handwritten text using recurrent neural networks and dynamic programming was proposed. Implementation of these methods to outputs of decoder gives improved segmentation results and allows to increase character segmentation accuracy after decoding without degradation of recognition results.

Experiments show that using of recurrent neural networks for classification segments gives an increase about 7% of segmentation accuracy. Implementation of proposed dynamic programming approach gave 1% of performance improvement.

The proposed approach can be applied to the ink beautification problem, formulae recognition and for editing of handwritten text or other data represented as a sequence of points where a correct segmentation is needed.

REFERENCES

[1] R. Plamondon and S. N. Srihari, "Online and off-line handwriting recognition: a comprehensive survey," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 1, pp. 63-84, 2000.

[2] C. C. Tappert, C. Y. Suen, T. Wakahara, "The state of the art in online handwriting recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 12, no. 8, pp. 787-808, 1990.

[3] P. Y. Simard, D. Steinkraus and M. Agrawala, "Ink normalization and beautification," Eighth International Conference on Document Analysis and Recognition (ICDAR'05), vol. 2, pp. 1182-1187, 2005.

[4] R.G. Casey and E. Lecolinet, "A Survey of Method and Strategies in Character Segmentation," IEEE Trans. on PAMI, vol. 18 (7), pp. 690-706, 1996.

[5] C. T. Nguyen and M. Nakagawa, "An improved segmentation of online English handwritten text using recurrent neural networks," 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), Kuala Lumpur, pp. 176-180, 2015.

[6] S. P. Naeini, M. Khademi and A. Nikookar, "A novel approach to segmentation of Persian cursive script using decision tree," International Journal of Computer Theory and Engineering, vol. 4 (3), p. 465, 2012.

[7] I. Mayire, H. Askar and T. Dilmurat, "A Dynamic Programming Method for Segmentation of Online Cursive Uyghur Handwritten Words into Basic Recognizable Units," Journal of Software, vol. 10(8), pp. 2535-2540, 2013.

[8] E. Kavallieratou, E. Stamatatos, N. Fakotakis and G. Kokkinakis, "Handwritten character segmentation using transformation-based learning," Proceedings 15th International Conference on Pattern Recognition. ICPR-2000, vol.2, pp. 634-637, 2000.

[9] R. Ghosh, "Stroke segmentation of online handwritten word using the busy zone concept," 2013 International Conference on Soft Computing and Pattern Recognition (SoCPaR), pp. 54-59, 2013.

[10] F. Naohiro, J. Tokuno and H. Ikeda, "Online character segmentation method for unconstrained handwriting strings using off-stroke features," Tenth International Workshop on Frontiers in Handwriting Recognition, IWFHR-10, pp. 361-366, 2006.

[11] N. Bhattacharya and U. Pal, "Stroke segmentation and recognition from Bangla online handwritten text," 2012 International Conference on Frontiers in Handwriting Recognition, pp. 740-745, 2012.

[12] I. Mayire, H. Askar, T. Dilmurat, "A dynamic programming method for segmentation of online cursive Uyghur handwritten words into basic recognizable units," Journal of Software, vol. 8 (10), pp. 2535-2540, 2013.

[13] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," Neural Networks, v.18 n.5-6, pp. 602-610, 2005.

[14] A. Graves, M. Liwicki, S. Fernndez, R. Bertolami, H. Bunke and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31 (5), pp. 855-868, 2009.

[15] A. Graves, "RNNLIB: A recurrent neural network library for sequence learning problems," http://sourceforge.net/projects/rnnl/, 2013.

[16] E. Indermhle, M. Liwicki and H. Bunke, "IAMonDo-database: an online handwritten document database with non-uniform contents," In Proc. Of Int. Workshop on Document Analysis Systems, pp. 97-104, 2010.

[17] S. J. Young, N. H. Russell, and J. H. S. Thornton, "Token passing: A simple conceptual model for connected speech recognition systems," Tech. Rep. CUED/F-INFENG/TR38, Cambridge University Engineering Department, 1989.

[18] F. Chunsheng, "From Dynamic Time Warping (DTW) to Hidden Markov Model (HMM) Final project report for ECE742 Stochastic Decision," 2009.

[19] V. Khomenko, O. Shyshkov, O. Radyvonenko and K. Bokhan, "Accelerating recurrent neural network training using sequence bucketing and multi-GPU data parallelization," Proceedings of the 2016 IEEE First International Conference on Data Stream Mining & Processing, pp. 100-103, 2016.