Using BLSTM for Interpretation of 2D Languages – Case of Handwritten Mathematical Expressions

Ting ZHANG* — **Harold MOUCHERE*** — **Christian VIARD-GAUDIN***

* *LUNAM/IRCCyN/IVC - UMR CNRS 6597*
Université de Nantes
Nantes, France

Cet article fait également partie des Rencontres Jeunes Chercheurs (RJC).

RÉSUMÉ. *Nous proposons une extension à l'utilisation classique des réseaux de type BLSTM pour leur permettre de traiter des données provenant de langages graphiques bidimensionnels tels que les formules mathématiques manuscrites. La solution proposée repose sur un parcours respectant l'ordre temporel des traits. Il en résulte une séquence alternant les étiquettes de symboles et les étiquettes des relations spatiales. Dans le cas des expressions purement linéaires (1-D), nous utilisons l'étiquette « Right » pour permettre la segmentation entre les symboles. Pour une extension au cas des expressions véritablement bidimensionnels (2-D), nous utilisons autant de nouvelles étiquettes qu'il y a de relations spatiales différentes entre les sous-expressions. Il en résulte que les réseaux BLSTM permettent de résoudre à la fois la tâche de reconnaissance de symboles et celle de segmentation. Une telle approche est nouvelle dans le domaine de la reconnaissance des expressions mathématiques.*

ABSTRACT. *In this work, we study how to extend the capability of BLSTM networks to process data which are not only text strings but graphical two-dimensional languages such as handwritten mathematical expressions. The proposed solution aims at transforming the mathematical expression description into a sequence including at the same time symbol labels and relationship labels, so that classical supervised sequence labeling with recurrent neural networks can be applied. For simple one-dimensional (1-D) expression, we use the $Right$ label to segment one symbol from the next one, as with the standard blank label for regular text. For genuine two-dimensional (2-D) expressions, we introduce additional specific labels assigned to each of the different possible spatial relationships that exist between sub-expressions. As a result, BLSTM network is able to perform at the same time the symbol recognition task and the segmentation task, which is a new perspective for the mathematical expression domain.*

MOTS-CLÉS : *Reconnaissance d'expressions mathématiques, écriture manuscrite, réseau récurrent, BLSTM.*

KEYWORDS: *Handwritten mathematical expression, online handwriting, recurrent neural network, BLSTM.*

1. Introduction

Online Handwritten Mathematical Expressions (MEs) recognition has been an active research field for years, especially after the rapid development of touch screen devices. There are two main sources of handwritten ME : offline (images of manuscript documents or video of blackboards) and online. We focus on online documents which come from sensitive screens (interactive whiteboards, smaller touch screens or tablets, etc.). An online document consists of one or more strokes which are sequences of points sampled from the trajectory of the writing tool between a pen-down and a pen-up. On-line handwritten ME recognition involves the automatic interpretation of these temporal sequence of 2-D points into structured sets of symbols. To realize these, both a large set of symbol classes and symbol arrangements in a two-dimensional layout need to be recognized. ME recognition still exhibits a big challenge to researchers.

Generally, ME recognition involves three interdependent tasks (Zanibbi et Blostein, 2012) : (1) Symbol segmentation, which consists in grouping strokes that belong to the same symbol ; (2) symbol recognition, the task of labeling the symbol to assign each of them a symbol class ; (3) structural analysis, its goal is to identify spatial relations between symbols and with the help of a grammar to produce a mathematical interpretation. These three problems can be solved sequentially or jointly. In the first case, the tasks are processed sequentially which means the error from the previous stage will be propagated to the next stage. The alternative solutions run these three tasks concurrently and aim at making a final decision of the formula using global information (Awal *et al.*, 2014 ; Álvaro *et al.*, 2014b ; Álvaro *et al.*, 2016). These approaches seem reasonable and exhibit out performances as these three problems are highly interdependent by nature. Human beings recognize symbols with the help of structure, and vice versa.

Current solutions process each task separately but need to consider context. Context is included to classical classifier (MLP, SVM, etc.) with specific features or language model (grammar). Bidirectional Long Short-term Memory (BLSTM) network naturally takes this context into account because it can access the contextual information from both the future and the past in an unlimited range. The advanced recurrent neural network BLSTM has been proved to outperform other classifiers in tasks like handwritten text recognition (Graves *et al.*, 2009), handwritten math symbol recognition (Álvaro *et al.*, 2013 ; Álvaro *et al.*, 2014a), printed text recognition (Ray *et al.*, 2015) and keyword spotting (Frinken *et al.*, 2014). However, the move from text recognition to mathematical expression recognition is far from being straightforward since ME has the structure of 2 dimensions. In this study, we explored ME recognition using BLSTM model, not simply as a symbol classifier exploiting a pre-segmented fragment as proposed in (Álvaro *et al.*, 2014a), but as a system able to label a global sequence, solving at the same time the segmentation and the recognition tasks.

This research domain has been boosted by the Competition on Recognition of Handwritten Mathematical Expressions (CROHME), which began as part of the International Conference on Document Analysis and Recognition (ICDAR) in 2011.

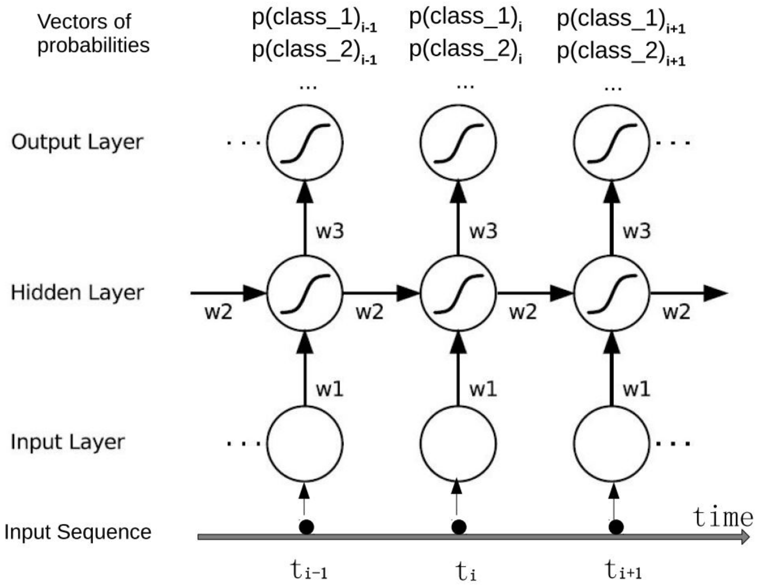


Figure 1. *An unfolded single-directional recurrent network.*

It provides a platform for researchers to test their methods and compare them, and then facilitate the progress in this field. It attracts increasing participation of research groups from all over the world. In this work, the provided data and tools will be used and results will be compared to participants.

The remainder of the paper is organized as follows. Section II introduces BLSTM architecture briefly. Section III describes the representation of ME. Section IV presents recognition strategies at stroke level. Features for on-line ME are showed in Section V. In Section VI, we report the experiments and results. Finally, conclusions and future works are presented.

2. BLSTM

Recurrent neural networks (RNNs) can use contextual information and therefore are suitable for sequence labeling (Graves *et al.*, 2012). In order to visualise the RNN and understand how it operates, we unfolded it along the input sequence and illustrated a part of the unfolded one in Figure 1. Here, each node represents a layer of network units at a single time-step. The input at a single time-step is a feature vector ; for the whole process, the input is a time sequence of feature vectors. The output at a single time-step is a probability vector of which each element is the probability of belonging to some class ; the overall output is a sequence of probability vectors. As can be seen, the output of each step depends on both the current input and the information from the

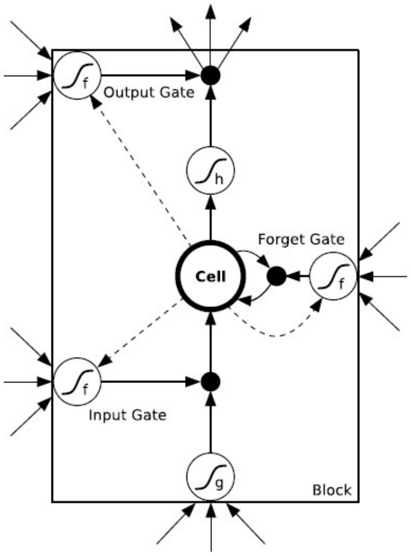


Figure 2. *LSTM memory block with one cell, figure extracted from (Graves* et al.*, 2012).*

previous time-step. The same weights (w1, w2, w3) are reused at every time-step. The training process of RNN requires the ground-truth for each time-step because it needs to compute the error to do back propagation. With respect to recognition process, since the network only outputs local classifications, some form of post-processing need to be done.

An important benefit of RNNs is their ability to use contextual information. Unfortunately, for standard RNN architectures, the range of context that can be accessed is quite limited. Long short-term memory (LSTM) (Hochreiter et Schmidhuber, 1997) was proposed to address this weakness. An LSTM network is the same as a standard RNN, except that the summation units in the hidden layer are replaced by memory blocks, as shown in Figure 2. Each block contains one or more self-connected memory cells and three multiplicative units (the input, output and forget gates). The three gates collect activation from inside and outside the block and control the activation of the cell via multiplications. The input and output gates multiply the input and output of the cell while the forget gate multiplies the cell's previous state. The only output from the block to the rest of the network emanates from the output gate multiplication.

Standard RNNs process sequences in temporal order, as presented in Figure 1, which means they can only access the past context, ignoring the information from the future. However, future context is vital for recognition task likewise. An elegant solution to this problem is bidirectional recurrent neural networks (BRNNs) (Schuster et

Paliwal, 1997), which presents every training sequence forward and backward to two separate recurrent hidden layers. Replacing the summation units in the hidden layer of BRNN generates BLSTM (Graves et Schmidhuber, 2005). Combining the advantages of BRNN and LSTM, BLSTM offers access to long range context in two directions. In the nature of things, this advanced architecture outperformed other models in several tasks (Graves *et al.*, 2009 ; Álvaro *et al.*, 2013 ; Álvaro *et al.*, 2014a).

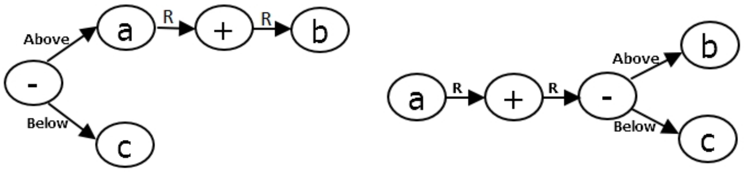
3. The Representation of ME

The recurrent neural network BLSTM processes each training sequence in temporal order (forward and backward). That means if we want to use the BLSTM model directly into ME recognition, each expression should be stored as a 1-D sequence. The proposed solution is to extract a reference path in a graph describing the ME. As a graph, we present in the next section how to model a ME with a primitive (stroke) Label Graph (LG).

3.1. *Primitive Label Graph*

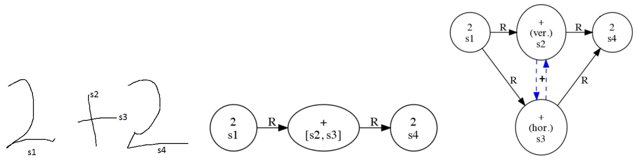
Structures can be depicted at three different levels : symbolic, object and primitive (Zanibbi *et al.*, 2013). In the case of handwritten ME, the corresponding levels are expression, symbol and stroke. It is possible to describe a ME at the symbol level using a Symbol Relation Tree (SRT), while if needed to go down at the stroke level, a primitive Label Graph can be derived from the SRT. In SRT, nodes represent symbols and labels on the edges indicate the relationships between symbols. Examples can be found in Figure 3. LG contains the same information as SRT but at the stroke level. In LG, nodes represent strokes, while labels on the edges encode either segmentation information or layout information. As relationships are defined at the level of symbols, all strokes in a symbol have the same input and output edges. Consider the simple expression '2 + 2' written using four strokes (two strokes for '+') in Figure 4a. The corresponding SRT and LG are shown in Figure 4b and Figure 4c respectively. As Figure 4c illustrates, nodes are labeled with the class of the symbol. A dashed edge corresponds to segmentation information ; it indicates that a pair of strokes belongs to the same symbol. In this case, the edge label is the same as the common symbol label. On the other hand, the non-dashed edges define spatial relationships between nodes and are labeled with one of the different possible relationships between symbols.

The spatial relationships as defined in the CROHME competition are : $Right$, $Above$, $Below$, $Inside$ (for square root), $Superscript$, $Subscript$. For the case of nth-Roots, like $\sqrt[3]{x}$, we define that the symbol 3 is $Above$ the square root and x is $Inside$ the square root. The limits of an integral and summation are designated as $Above$ or $Superscript$ and $Below$ or $Subscript$ depending on the actual position of the bounds. In addition, the label '_' is used to denote that there is no relation between two symbols.



(a) (b)

Figure 3. *(a) the symbol relation tree of* $\frac{a+b}{c}$ *; (b) the symbol relation tree of* $a + \frac{b}{c}$*. 'R' is for left-right relationship.*



(a) (b) (c)

Figure 4. *(a) '2 + 2' written with four strokes ; (b) the symbol relation tree of '2 + 2' ; (c) the label graph of '2 + 2'. The four strokes are indicated as s1, s2, s3, s4 in writing order. (ver.) and (hor.) are added to differentiate the vertical and the horizontal strokes for '+'. 'R' is for left-right relationship.*

3.2. *Storing expression as 1-D sequence*

As mentioned, each expression should be stored into a 1-D sequence in order to be suitable for BLSTM. There are several solutions to generate a 1-D sequence from a tree but none of them allows to rebuild systematically the tree structure except Euler string which requires additional brackets. We propose to generate a 1-D sequence following the time sequence of strokes in LG to represent an expression. This time sequence is not always the best solution, but the simplest and the most intuitive by nature. From Figure 4c, the corresponding 1-D sequence is $\{s1, edge_{s1\_s2}, s2, edge_{s2\_s3}, s3, edge_{s3\_s4}, s4\}$ labeled as $\{2, R, +, +, +, R, 2\}$, where $edge_{si\_sj}$ denotes the edge from si to sj. This sequence alternates the node labels $\{2, +, 2\}$ and the edge labels $\{R, +, R\}$. This proposal works well on linear expressions. For example, given $\{2, R, +, +, +, R, 2\}$, we can regenerate the correct LG with adding the edges from $s1$ to $s3$ and from $s2$ to $s4$ (following the rule that all strokes in a symbol have the same input and output edges), also the edge from $s3$ to $s2$. It can also deal with a part of 2-D expressions. With P^{eo} shown in Figure 5a, the sequence is $\{P, P, P, Superscript, e, R, o\}$. All the spatial relationships are covered in it and naturally a correct LG can be regenerated. However, this kind of sequence

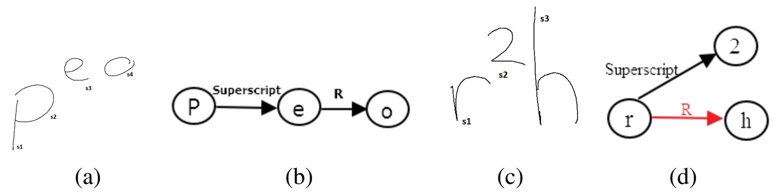


Figure 5. *(a) P^{eo} written with four strokes ; (b) the SRT of P^{eo} ; (c) r^2h written with three strokes ; (d) the SRT of r^2h, the red edge can not be generated by the time sequence of strokes.*

fails on a number of 2-D expressions. Figure 5c presents a failed case. According to time order, 2 and h are neighbors but there is no edge between them as can be seen on Figure 5d. In this case, we write the sequence as $\{r, Superscript, 2, \_, h\}$. The $Right$ relationship existing between r and h drawn with red color in Figure 5d is missing in the previous sequence. For the training process, it means missing a training sample for relationship. For the recognition part, it means this expression can never be recognized fully. Being aware of this limitation, the 1D time sequence of strokes is used to train the BLSTM and the outputted sequence of labels during recognition will be completed to generate a LG graph as much as possible.

Expressions can be regarded as the arrangements of math symbols in two-dimensional layout according to some grammars. To feed the inputs of the BLSTM, it is important to scan as well the points belonging to the strokes themselves (on-paper points) and also the points separating one stroke from the next one (in-air points). We expect that the on-paper points will be labeled with corresponding symbol labels and that the in-air points will be assigned with one of the possible edge labels. Thus, besides re-sampling points from strokes, we also re-sample points from the straight line which links two strokes. In the rest of this paper, $strokeD$ and $strokeU$ are used to indicate a re-sampled pen-down stroke and a re-sampled pen-up stroke for convenience.

Given each expression, we first re-sampled points both from strokes and between two successive strokes in the writing order. 1-D unlabeled sequence can be described as $\{strokeD_1, strokeU_2, strokeD_3, strokeU_4, ..., strokeD_K\}$ with K is the number of re-sampled strokes. Note that if s is the number of visible strokes, $K = 2 * s - 1$. Ground-truth of each point is required for BLSTM training process. The labels of the points from $strokeD_i$ should be assigned with the label of the corresponding node in LG ; The label of the points from $strokeU_i$ should be assigned with the label of the corresponding edge in LG. If no corresponding edge exists, the label is '_'.

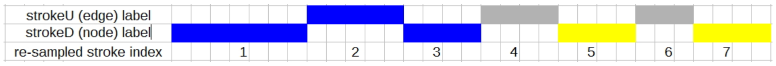


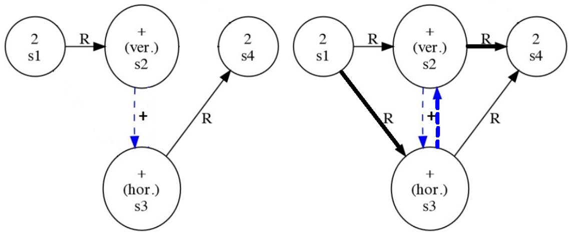
Figure 6. *The illustration for the decision of the label of $strokeU$. Gray color represents one of 6 relationships or '_' while other colors indicate different symbols. In this example, 3 symbols are recognized : strokes grouped as* $\{1, 3\}, \{5\}, \{7\}$.

4. Recognition Strategies

As Figure 1 shows, since the RNN outputs local classifications for each point, some form of post-processing need to be done. The connectionist temporal classification (CTC) (Graves *et al.*, 2006) technology is a good choice for sequence transcription tasks. It outputs the probabilities of the complete sequences directly but do not provide the alignment between the inputs and the target labels. In our case, we need the labels of strokes to obtain a LG. Thus, we propose to make decisions on stroke ($strokeD$ or $strokeU$) level instead of sequence level (as CTC) or point level. As BLSTM can access the contextual information from two directions in an unlimited range, the output probability of each point is not a local decision. With the same method taken by Alex Graves for isolated handwritten digits recognition using a multidimensional RNN with LSTM hidden layers in (Graves *et al.*, 2012), we choose for each stroke the label which has the highest cumulative probability over the entire stroke. Suppose that p_{ij} is the probability of outputting the i_{th} label at the j_{th}time step. Then, for each stroke k, the probability of outputting the i_{th} label can be computed as $ps_{ik} = \sum_{j=1}^{|s_k|} p_{ij}$, where $|s_k|$ is the number of points of stroke k. Finally, the label with the highest ps_{ik} is selected.

Also, we add two constraints when making decisions on stroke level : (1) the label of $strokeD$ should be one of the symbol labels, excluding the relationship labels, (2) the label of $strokeU_i$ is divided into 2 cases, if the labels of $strokeD_{i-1}$ and $strokeD_{i+1}$ are different, it should be one of the six relationships or '_' ($strokeU_4$ in Figure 6) ; otherwise, it should be relationships, '_' or symbol. A symbol label assigned to an edge means that the corresponding pair of nodes (strokes) belongs to the same symbol ($strokeU_2$ in Figure 6) while '_' or relationship refers to 2 strokes belonging to 2 symbols ($strokeU_6$ in Figure 6).

After recognition, post-processing should be done in order to rebuild the LG, i.e. adding edges. Supposing that the sample shown in Figure 4 is correctly recognized, the output sequence should be $\{2, R, +, +, +, R, 2\}$ as illustrated in Figure 7a. According to the rule that all strokes in a symbol have the same input and output edges, we add the edges from $s1$ to $s3$ and from $s2$ to $s4$. We also add the edge from $s3$ to $s2$, where the double-direction edge represents the segmentation. The rebuilded LG is available in Figure 7b.



(a) (b)

Figure 7. *(a) The correct recognized sequence of '2 + 2' written with four strokes ; (b) the rebuilded LG of '2 + 2', added edges are depicted as bold.*

5. Features

A stroke is a sequence of points sampled from the trajectory of a writing tool between a pen-down and a pen-up at a fixed interval of time. Then an additional re-sampling is performed with a fixed spatial step to get rid of the writing speed. The number of re-sampling points depends on the size of expression. For expression which has 3 or more strokes, we re-sample with $10 \times (length/avrheight)$ points ; otherwise, we re-sample with $10 \times (length/avrdiagonal)$. Here, $length$ refers to the length of all the strokes (including distance between successive strokes) and $avrheight$ $(avrdiagonal)$ refers to the average height (diagonal) of the bounding boxes of all the strokes in an expression. Then, the whole expression is re-scaled, preserving the aspect ratio, into the normalized rectangle : $[-w/2, w/2] \times [-h/2, h/2]$ to be robust with regard to size variation, where $w = width/avrwidth$ and $h = height/avrheight$, while $width$ $(height)$ refers to the width (height) of the bounding box of the entire expression.

Subsequently, we compute five features per point, which are quite close to the state of art (Álvaro *et al.*, 2013 ; Awal *et al.*, 2014). For every point $p(x, y)$ we obtained 5 features in the following format :

$$[\sin\theta, \cos\theta, \sin\phi, \cos\phi, \text{PenUD}]$$

with :

- $sin\theta$, $cos\theta$ are the sine and cosine directors of the tangent of the stroke at point p(x, y) ;
- $\phi = \Delta\theta$, defines the change of direction at point p(x, y) ;
- PenUD refers to the state of pen-down or pen-up.

6. Experiments

We use the RNNLIB library (Graves, 2013) for training a BLSTM model. For each training process, the network having the best classification error on validation data set is saved. Then, we test this network on the test data set. The Label Graph Evaluation library (LgEval) (Mouchère *et al.*, 2014) is adopted to evaluate the recognition output.

Three experiments are performed in this paper. Experiment 1 only focus on expressions which do not include 2-D spatial relations. In the second one, we begin to test our method on 2-D expressions the depth of which is limited to 1. It imposes that two sub-expressions having a spatial relationship ($Above, Below, Inside, Superscript, Subscript$) should be left-right expressions. Finally, all kinds of expressions available are included in experiment 3.

Network architecture and configuration are as following :

- The input layer size : 5
- The output layer size : the number of class
- The hidden layers : 2 layers, the forward and backward, each contains 100 single-cell LSTM memory blocks
- The weights : initialized uniformly in [-0.1, 0.1]
- The momentum : 0.9

This configuration has obtained good results in both handwritten text recognition (Graves *et al.*, 2009) and handwritten math symbol classification (Álvaro *et al.*, 2013 ; Álvaro *et al.*, 2014a).

6.1. *Experiment 1*

We select the expressions with only left-right relations from CROHME 2014 training and test data. 2609 expressions are available for training, about one third of the full set ; 265 expressions for testing. In this case, there are 91 classes of symbols. Next, we split the training set into the new training set and validation set, 90% for training and 10% for validation. The output layer size in this experiment is 93 (91 symbol classes + $Right$ + $NoRelation$). In left-right expressions, $NoRelation$ will be used for indicating delayed strokes.

6.2. *Experiment 2*

In this part, according to the rule described before, 5820 expressions are selected for training from CROHME 2014 train set ; 674 expressions for test from CROHME

2014 test set. Also, we divide 5820 expressions into the new training set and validation set, 90% for training and 10% for validation. The output layer size in this experiment is 100 (93 symbol classes + 6 relationships + $NoRelation$).

6.3. *Experiment 3*

We use the complete data set from CROHME 2014, 8834 expressions for training and 982 expressions for test. Also, we divide 8834 expressions for training (90%) and validation (10%). The output layer size in this experiment is 108 (101 symbol classes + 6 relationships + $NoRelation$).

6.4. *Discussion*

The evaluation results on symbol level for the 3 experiments are provided in Table 1 including recall ('Rec.') and precision ('Prec.') rates for symbol segmentation ('Segments'), symbol segmentation and recognition ('Seg+Class'), spatial relationship classification ('Tree Rels.'). A correct spatial relationship between two symbols requires that both symbols are correctly segmented and with the right relationship label.

As can be seen, the results in 'Segments' and 'Seg+Class' of experiment 1 are a bit lower compared to experiment 2 and 3. It is a reasonable phenomenon given that the training data set in experiment 1 is not large enough. The results in 'Segments' and 'Seg+Class' do not present a big difference in experiment 2 and 3. The results for 'Tree Rels.' are declining in these 3 experiments. It is understandable as the number of missed relationships grows with the complexity of expressions with the limitation of our method. The results of experiment 3 are comparable to the results of CROHME 2014 because the same training and testing data sets are used. The second part of Table 1 gives the symbol level evaluation results of the top 4 participants in CROHME 2014 sorting by the recall rate for correct symbol segmentation. The best 'Rec.' of 'Segments' and 'Seg+Class' reported by CROHME 2014 are 98.42% and 93.91% respectively. Ours are 92.14% and 82.82%, both ranked 3 out of 8 systems (7 participants in CROHME 2014). Our solution presents competitive results on symbol recognition task and segmentation task even though the symbols with delayed strokes were missed. However, our proposal, at that stage, shows poor performances compared to the 7 participants on relationship detection and recognition task which is not surprising because the existing of limitation that some relationships were missed.

Table 2 shows the recognition rates at the global expression level with no error, and with at most one to three errors in the labels of LG. This metric is very strict. For example one label error can happen only on one stroke symbol or in the relationship between two one-stroke symbols ; a labeling error on a 2-strokes symbol leads to 4 errors (2 nodes labels and 2 edges labels). The recognition rate with no error on CROHME 2014 test set is 12.22%. The best one and worst one reported by CROHME

Table 1. *The symbol level evaluation results on CROHME 2014 test set, including the experiment results in this work and CROHME 2014 participant results (Top 4 by recall of Segments).*

exp.	Segments (%)		Seg + Class (%)		Tree Rels. (%)	
	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.
1	90.44	80.81	78.69	70.31	80.58	72.26
2	92.71	85.86	82.47	76.38	64	70.51
3	**92.14**	**84.68**	**82.82**	**76.12**	**59.52**	**67.53**
system	CROHME 2014 participant results (Top 4)					
III	98.42	98.13	93.91	93.63	94.26	94.01
I	93.31	90.72	86.59	84.18	84.23	81.96
VII	89.43	86.13	76.53	73.71	71.77	71.65
V	88.23	84.20	78.45	74.87	61.38	72.70

2014 are 62.68% and 15.01%. With regard to the recognition rate with ≤ 3 errors, 4 participants are between 27% and 37% and our result is 28.51%.

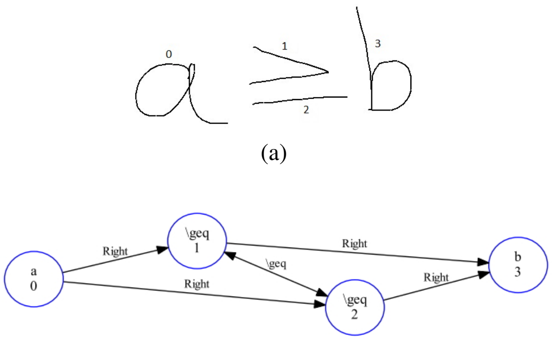
Table 2. *The expression level evaluation results on CROHME 2014 test set, including the experiment results in this work and CROHME 2014 participant results (Top 2)*

exp.	correct (%)	<= 1 error	<= 2 errors	<= 3 errors
1	30.57	47.92	52.45	53.21
2	16.32	25.22	32.64	37.39
3	12.22	19.56	24.95	28.51
system	CROHME 2014 participant results (Top 2)			
III	62.68	72.31	75.15	76.88
I	37.22	44.22	47.26	50.20

We present a correctly recognized sample and an incorrectly recognized sample in Figure 8 and Figure 9 respectively.

7. Conclusion

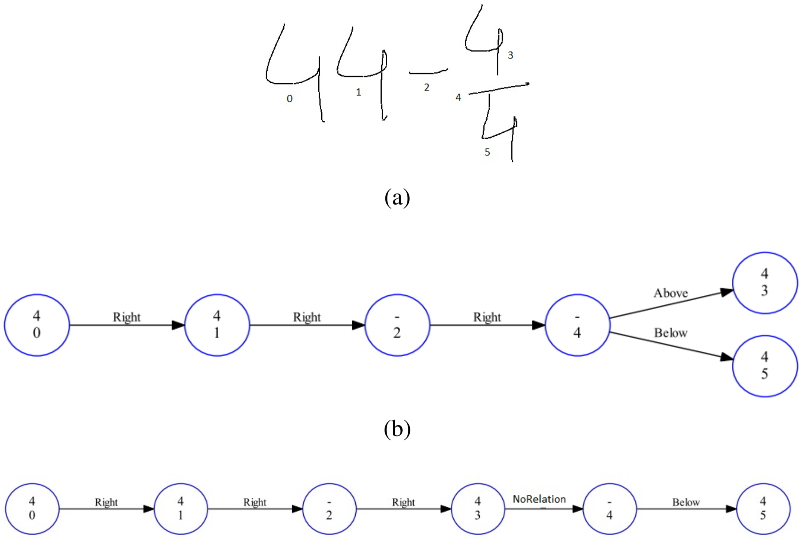
The capability of BLSTM networks to process graphical two-dimensional languages such as handwritten mathematical expressions is explored in this study. We represent the mathematical expression with a 1-D sequence derived from a path following the stroke writing order and using the labels extracted from the LG, so that classical supervised sequence labeling with recurrent neural networks can be applied. Our solution presents competitive results with CROHME 2014 on symbol recognition task and segmentation task. Proposing a global solution to perform segmentation, recognition and interpretation, with no dedicated stages, is a major advantage of the proposed solution. To some extent, at the present time, it fails on relationship recog-



(a)

(b)

Figure 8. *(a) $a \geq b$ written with four strokes ; (b) the rebuilded LG of $a \geq b$ according to the recognition result, all labels read correct.*



(a)

(b)

(c)

Figure 9. *(a) $44 - \frac{4}{4}$ written with six strokes ; (b) the ground-truth LG ; (c) the rebuilt LG according to the recognition result. Three edge errors occurred : the $Right$ relation between strokes 2 and 4 was missed and the relation between strokes 3 and 4 was wrongly recognized, also, the edge between strokes 2 and 3 should be labeled as $NoRelation$.*

nition task. This is primarily due to the proposed expression description. Currently, a single path following the time sequence of strokes in LG is used to represent the expression. Actually, some important relationships are discarded. In future, a better description solution will be proposed such that less relationships will be missed. Several paths can be merged to build a more robust decision which takes into account most of possible stroke configurations in time and space.

Remerciements

We would like to thank the China Scholarship Council for the financial support for Ting ZHANG PhD studentship at Nantes university.

8. Bibliographie

Álvaro F., Sánchez J.-A., Benedí J.-M., « Classification of on-line mathematical symbols with hybrid features and recurrent neural networks », *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, IEEE, p. 1012-1016, 2013.

Álvaro F., Sánchez J.-A., Benedí J.-M., « Offline Features for Classifying Handwritten Math Symbols with Recurrent Neural Networks », *Pattern Recognition (ICPR), 2014 22nd International Conference on*, IEEE, p. 2944-2949, 2014a.

Álvaro F., Sánchez J.-A., Benedí J.-M., « Recognition of on-line handwritten mathematical expressions using 2d stochastic context-free grammars and hidden markov models », *Pattern Recognition Letters*, vol. 35, p. 58-67, 2014b.

Álvaro F., Sánchez J.-A., Benedí J.-M., « An integrated grammar-based approach for mathematical expression recognition », *Pattern Recognition*, vol. 51, p. 135-147, 2016.

Awal A.-M., Mouchère H., Viard-Gaudin C., « A global learning approach for an online handwritten mathematical expression recognition system », *Pattern Recognition Letters*, vol. 35, p. 68-77, 2014.

Frinken V., Fischer A., Baumgartner M., Bunke H., « Keyword spotting for self-training of BLSTM NN based handwriting recognition systems », *Pattern Recognition*, vol. 47, n^{o} 3, p. 1073-1082, 2014.

Graves A., « Rnnlib : A recurrent neural network library for sequence learning problems », 2013.

Graves A. *et al.*, *Supervised sequence labelling with recurrent neural networks*, vol. 385, Springer, 2012.

Graves A., Fernández S., Gomez F., Schmidhuber J., « Connectionist temporal classification : labelling unsegmented sequence data with recurrent neural networks », *Proceedings of the 23rd international conference on Machine learning*, ACM, p. 369-376, 2006.

Graves A., Liwicki M., Fernández S., Bertolami R., Bunke H., Schmidhuber J., « A novel connectionist system for unconstrained handwriting recognition », *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, n^{o} 5, p. 855-868, 2009.

Graves A., Schmidhuber J., « Framewise phoneme classification with bidirectional lstm networks », *Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on*, vol. 4, IEEE, p. 2047-2052, 2005.

Hochreiter S., Schmidhuber J., « Long short-term memory », *Neural computation*, vol. 9, n^{o} 8, p. 1735-1780, 1997.

Mouchère H., Viard-Gaudin C., Zanibbi R., Garain U., « Icfhr 2014 competition on recognition of on-line handwritten mathematical expressions (crohme 2014) », *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, IEEE, p. 791-796, 2014.

Ray A., Rajeswar S., Chaudhury S., « Text recognition using deep BLSTM networks », *Advances in Pattern Recognition (ICAPR), 2015 Eighth International Conference on*, IEEE, p. 1-6, 2015.

Schuster M., Paliwal K. K., « Bidirectional recurrent neural networks », *Signal Processing, IEEE Transactions on*, vol. 45, n^{o} 11, p. 2673-2681, 1997.

Zanibbi R., Blostein D., « Recognition and retrieval of mathematical expressions », *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 15, n^{o} 4, p. 331-357, 2012.

Zanibbi R., Mouchère H., Viard-Gaudin C., « Evaluating structural pattern recognition for handwritten math via primitive label graphs », *IS&T/SPIE Electronic Imaging*, International Society for Optics and Photonics, p. 865817-865817, 2013.