MRD: A Memory Relation Decoder for Online Handwritten Mathematical Expression Recognition

Jiaming Wang[1], Qing Wang[1], Jun Du[1,2](✉), Jianshu Zhang[1], Bin Wang[3], and Bo Ren[3]

[1] University of Science and Technology of China, Hefei, Anhui, China
{jmwang66,xysszjs}@mail.ustc.edu.cn, {qingwang2,jundu}@ustc.edu.cn
[2] Guangdong Artificial Intelligence and Digital Economy Laboratory (Pazhou Lab), Guangzhou, China
[3] Youtu Lab, Tencent, Hefei, Anhui, China
{bingolwang,timren}@tencent.com

Abstract. Recently, attention based encoder-decoder methods have been widely used in online handwritten mathematical expression recognition, which achieve significant improvements compared to traditional methods. The encoder-decoder methods usually employ string decoders to generate the recognition result, which are not well matched for tree-structured languages like math expression. A novel sequential relation decoder (SRD) was introduced to recognize the online mathematical expression as a math tree, which can be decomposed into a subtree sequence and each subtree consists of a relation node and two symbol nodes (related symbol node and primary symbol node). However, the alignments between these two symbol nodes were implemented by spatial attention probabilities, leading to incorrect recognition if spatial attention is not accurate. In this paper, we propose a memory relation decoder (MRD), equipped with a memory based attention model to determine the correspondence between two symbol nodes. Specifically, at each decoding step, this memory based attention finds the corresponding primary symbol node in the memory and treats it as the related symbol node, which actually achieves the alignments between two symbol nodes in an explicit manner. Besides, we propose to introduce global visual information while calculating attention probabilities to help alleviate the ambiguous problems in online handwritten mathematical expression recognition. Evaluated on a benchmark published by CROHME competition, the proposed approach can substantially outperform previous encoder-decoder methods.

Keywords: Handwritten mathematical expression recognition · Encoder-decoder · Tree structure · Memory based attention

J. Lladós et al. (Eds.): ICDAR 2021, LNCS 12823, pp. 39–54, 2021.
https://doi.org/10.1007/978-3-030-86334-0_3

1 Introduction

Handwritten mathematical expression recognition (HMER) plays an essential role in electronic technology documents, machine scoring and many other applications. As mathematical expression is a complicated two-dimensional structure (inherent tree structure) [3,6,15], HMER usually meets more challenges than Chinese text recognition or other sequence recognition problems, which are usually written in one direction and the alignments between input and output are monotonic, i.e., the correspondence between the input and output shares the same order.

The main problems of HMER can be roughly divided into two branches [7], namely symbol recognition and structural analysis. Symbol recognition denotes grouping strokes which belong to the same symbol and then determines the class of each symbol. Structural analysis denotes generating the most likely math tree based on the symbol recognition. Traditional methods usually solve these two problems separately or jointly, namely sequential or global methods. While contextual information is not fully utilized and symbol recognition errors will be inherited afterwards to structural analysis in sequential methods [1,22], global methods [2,4] seem to be more suitable as they optimize symbol recognition and structural analysis concurrently, but previous sequential methods usually outperform traditional global methods.

Recently, several researches [20,23,26] proposed a global way to recognize a mathematical expression as a LaTeX string instead of a math tree since the LaTeX string and the math tree are actually one to one correspondence and can be converted into each other equivalently. As deep learning came into prominence, attention based encoder-decoder methods were widely used in sequence to sequence learning, such as machine translation [11,13,18], speech recognition [5,8] and so on. Online handwritten mathematical expression recognition can also be treated as a sequence to sequence problem and attention based encoder-decoder methods [19,23] can be employed to generate a LaTeX string as the recognition result. These encoder-decoder methods can usually achieve better performance than traditional methods due to their powerful modeling capabilities and free of pre-defined grammar or symbol segmentation.

However, using LaTeX strings as the recognition results of handwritten mathematical expressions will meet several problems [24,25]. Therefore, [24] proposed a sequential relation decoder (SRD), which obtained recognition results in math tree formats using encoder-decoder methods and can be trained in an end-to-end manner. Specifically, SRD decomposed the complete math tree into a subtree sequence. At each step, SRD can generate a subtree, containing a relation node and two symbol nodes (first generated a primary symbol node and then a related symbol node based on the obtained primary symbol node). Besides, SRD employed spatial attention probabilities to acquire alignments between primary symbol nodes and related symbol nodes. [25] proposed a tree decoder, which employed a memory based attention model to achieve the alignments between primary symbol nodes and related symbol nodes instead.

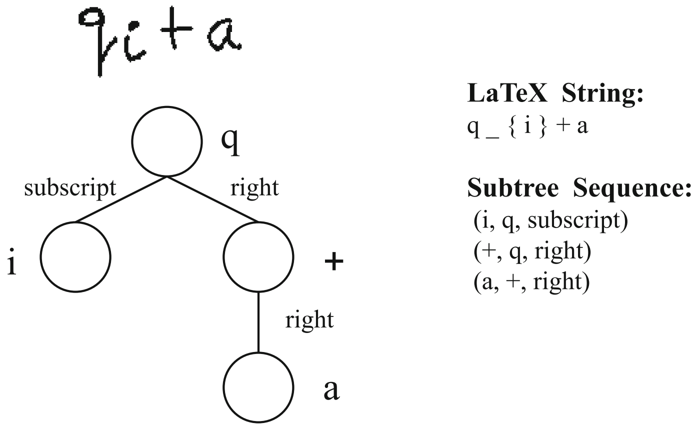

Fig. 1. An example of handwritten mathematical expression, which can be represented as a LaTeX string or a subtree sequence.

In this work, we propose a memory relation decoder (MRD) for online handwritten mathematical expression recognition, which also recognizes the mathematical expression as a tree structure. As shown in Fig. 1, the complete math tree can be decomposed into a subtree sequence and MRD can generate a subtree at each decoding step and finally all the subtrees can be utilized to compose the complete tree. More specifically, MRD first generates a related symbol node using a predicted related GRU, a related GRU and a related attention model. Then a primary symbol node, following depth-first order, is generated by a predicted primary GRU, a primary GRU and a primary attention model. Moreover, we propose to insert a global visual feature into original features to help alleviate ambiguous problems in online handwritten mathematical expression recognition. Based on two obtained symbol nodes, a relation node can be predicted, indicating the attribute between related and primary symbol nodes. Unlike SRD, we employ an improved version of memory based attention model [25] to achieve alignments between related symbol nodes and primary symbol nodes, which additionally exploits related and primary context vectors. This memory based attention model actually determines which primary symbol node that the related symbol node should be corresponded to at each decoding step in an explicit manner. Furthermore, two attention guiders, namely related attention guider and primary attention guider are employed to help guide the learning of related attention and primary attention.

The main contributions of this paper can be summarized as:

– A memory relation decoder (MRD) is proposed for online handwritten mathematical expression recognition, which significantly outperforms both previous string decoders and tree-structured decoders.
– We introduce global visual information inserted in attention models to help alleviate the ambiguous problems.

– We demonstrate the effectiveness of memory based attention and global visual information through complete experimental analysis.

2 The Proposed Approach

In this section, we elaborate the overall system for online handwritten mathematical expression recognition, which consists of an encoder and a memory relation decoder (MRD). The encoder employs a stacked RNNs to extract high-level features from handwriting traces. Then, the memory relation decoder is introduced to generate a subtree at each decoding step t, including a related symbol node, a primary symbol node and a relation node. The two symbol nodes are determined by related decoder and primary decoder, respectively. Then the relation node can be determined by both related symbol node and primary symbol node. We employ a memory based attention model to implement the alignments between related symbol nodes and primary symbol nodes in an explicit manner, which is necessary to reconstruct the complete tree by the generated subtree sequence. Besides, we introduce a global visual feature to alleviate ambiguous problems and two attention guiders to help guide the learning of related attention and primary attention.

2.1 Encoder

The raw data of online handwritten mathematical expression recognition is handwriting traces collected during the writing procedure. Following [20], we first normalize the traces and then obtain an 8-dimensional feature vector for each point i as follows:

$$\mathbf{x}_i = [x_i, y_i, \Delta x_i, \Delta y_i, \Delta' x_i, \Delta' y_i, \text{strokeFlag1}, \text{strokeFlag2}] \tag{1}$$

where $\Delta x_i = x_{i+1} - x_i$, $\Delta y_i = y_{i+1} - y_i$, $\Delta' x_i = x_{i+2} - x_i$, $\Delta' y_i = y_{i+2} - y_i$. The last two terms indicate the pen status, which record whether the point is the last one of a stroke, i.e., $[1, 0]$ means pen-down while $[0, 1]$ means pen-up. Then, this sequence of 8-dimensional feature vectors is considered as the input of the encoder.

As shown in Fig. 2, to capture contextual information from input, we employ a stack of recurrent neural networks (RNN) with gated recurrent units (GRU). Besides, we actually adopt bidirectional GRU instead of unidirectional GRU as both past and future contextual information are useful for recognition. The bidirectional GRU will scan the input forwards and backwards with two separate GRU layers and concatenate their hidden states. The final output of stacked GRUs is an annotation sequence of variable length, which is referred as $\mathbf{A} = \{\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_L\}$ and $\mathbf{a}_i \in \mathbb{R}^{D_1}$.

Besides, we believe that the attention models can benefit from global visual information as it can help alleviate ambiguous problems in online handwritten mathematical expression recognition. Therefore, we first convert handwriting

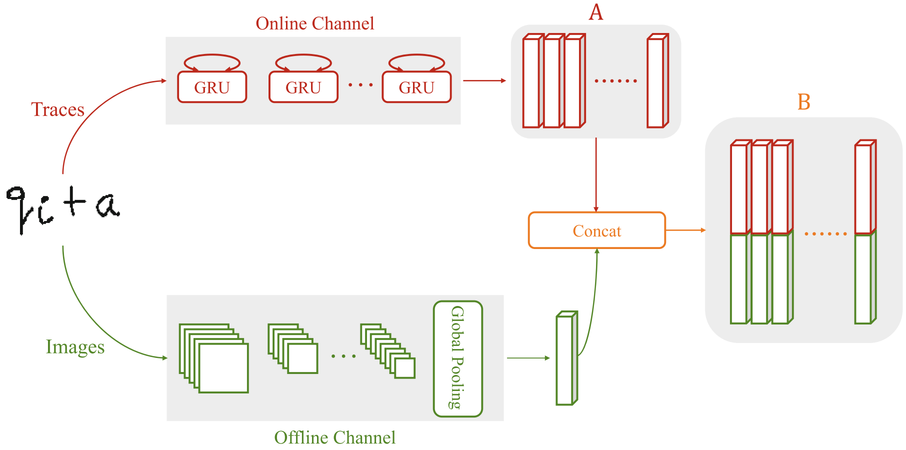

Fig. 2. The architecture of the encoder.

traces into a static image [20] and then employ an additional convolution neural networks (CNN) with dense blocks [12] to extract visual features, which is a tensor with size $H \times W \times D_2$. A global average pooling layer is built on top of CNN to obtain the global visual feature, $\mathbf{fea} \in \mathbb{R}^{D_2}$. We combine this global visual feature with each hidden state as follows:

$$\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_1, \cdots, \mathbf{b}_L\} \qquad \mathbf{b}_i = \mathrm{Concat}\left(\mathbf{a}_i, \mathbf{fea}\right) \tag{2}$$

where $\mathbf{b}_i \in \mathbb{R}^D$ and $D = D_1 + D_2$. Overall, the encoder can extract an annotation sequence A used to compute context vectors and an annotation sequence B used in attention models. The implementation details of the encoder can be seen in Sect. 3.1.

2.2 Memory Relation Decoder

As shown in Fig. 1, the target of memory relation decoder (MRD) is to generate a complete math tree for recognition, which can be decomposed into a variable length subtree sequence:

$$\mathbf{Y} = \left\{\left(\mathbf{y}_1^{\mathrm{r}}, \mathbf{y}_1^{\mathrm{p}}, \mathbf{y}_1^{\mathrm{re}}\right), \left(\mathbf{y}_2^{\mathrm{r}}, \mathbf{y}_2^{\mathrm{p}}, \mathbf{y}_2^{\mathrm{re}}\right), \cdots, \left(\mathbf{y}_T^{\mathrm{r}}, \mathbf{y}_T^{\mathrm{p}}, \mathbf{y}_T^{\mathrm{re}}\right)\right\} \tag{3}$$

where T denotes the total number of subtrees and each subtree t contains a related symbol node $\mathbf{y}_t^{\mathrm{r}}$, a primary symbol node $\mathbf{y}_t^{\mathrm{p}}$ and a relation node $\mathbf{y}_t^{\mathrm{re}}$. These subtrees can be generated by several unidirectional GRUs step by step, using two annotation sequences $\mathbf{A}$ and $\mathbf{B}$ extracted from the encoder.

Note that there are three rules to confirm that the predicted subtree sequence can reconstruct the complete tree: (i) the subtree sequence is serialized by traversing the complete tree following a depth-first order. (ii) every primary symbol node must have a corresponding related symbol node and only occur once. (iii) each related symbol node must be selected from existing primary symbol nodes.

Related Decoder. As shown in Fig. 3, to decode the related symbol node, we employ two GRUs, namely predicted related GRU and related GRU with a related attention model, which can be represented as follows:

$$\hat{\mathbf{s}}_t^{\mathrm{r}} = \mathrm{PRGRU}\left(\mathbf{y}_{t-1}^{\mathrm{p}}, \mathbf{s}_{t-1}^{\mathrm{p}}\right) \tag{4}$$

$$\mathbf{c}_t^{\mathrm{r}} = f_{\mathrm{ratt}}\left(\hat{\mathbf{s}}_t^{\mathrm{r}}, \mathbf{A}, \mathbf{B}\right) \tag{5}$$

$$\mathbf{s}_t^{\mathrm{r}} = \mathrm{RGRU}\left(\mathbf{c}_t^{\mathrm{r}}, \hat{\mathbf{s}}_t^{\mathrm{r}}\right) \tag{6}$$

where PRGRU and RGRU denote predicted related GRU and related GRU, respectively. f_{ratt} denotes related attention model, considering handwriting information and global visual information at the same time, which is designed as:

$$\mathbf{F}^{\mathrm{r}} = \mathbf{Q}^{\mathrm{r}} * \sum\nolimits_{\tau=1}^{t-1} \boldsymbol{\alpha}_{\tau}^{\mathrm{r}} \tag{7}$$

$$e_{tj}^{\mathrm{r}} = \boldsymbol{\nu}_{\mathrm{r}}^{\mathrm{T}} \tanh\left(\mathbf{W}_{\mathrm{att}}^{\mathrm{r}} \hat{\mathbf{s}}_t^{\mathrm{r}} + \mathbf{U}_{\mathrm{att}}^{\mathrm{r}} \mathbf{b}_j + \mathbf{U}_{\mathrm{F}}^{\mathrm{r}} \mathbf{f}_j^{\mathrm{r}}\right) \tag{8}$$

$$\alpha_{tj}^{\mathrm{r}} = \frac{\exp\left(e_{tj}^{\mathrm{r}}\right)}{\sum_k \exp\left(e_{tk}^{\mathrm{r}}\right)} \tag{9}$$

$$\mathbf{c}_t^{\mathrm{r}} = \sum\nolimits_{j=1}^{L} \alpha_{tj}^{\mathrm{r}} \mathbf{a}_j \tag{10}$$

where $*$ denotes a convolution layer and $\mathbf{f}_j^{\mathrm{r}}$ denotes the j-th element of F, which is utilized as a coverage vector to help alleviate the lack of coverage in the standard attention model. α_{tj}^{r} denotes the related attention probability of j-th element at decoding step t while $\mathbf{c}_t^{\mathrm{r}}$ denotes related context vector at decoding step t. $\mathbf{a}_j$ and $\mathbf{b}_j$ are j-th elements of $\mathbf{A}$ and $\mathbf{B}$, respectively. Note that we employ $\mathbf{B}$ to compute attention probabilities as global visual information can help solve ambiguous problems in online handwritten mathematical expression recognition. Instead, we only adopt original features $\mathbf{A}$ to compute related context vector because these features are enough when accurate trace points are selected by attention model. Another reason is that global visual information is unsuitable to be considered at each decoding step t as only local visual information corresponded to current predicted symbol is needed.

Primary Decoder. After obtaining related symbol node $\mathbf{y}_t^{\mathrm{r}}$ and related decoder hidden state $\mathbf{s}_t^{\mathrm{r}}$, we can generate primary context vector $\mathbf{c}_t^{\mathrm{p}}$ and primary decoder hidden state $\mathbf{s}_t^{\mathrm{p}}$. As shown in Fig. 3, the architecture of primary decoder is similar with related decoder, which consists of predicted primary GRU and primary GRU:

$$\hat{\mathbf{s}}_t^{\mathrm{p}} = \mathrm{PPGRU}\left(\mathbf{y}_t^{\mathrm{r}}, \mathbf{s}_t^{\mathrm{r}}\right) \tag{11}$$

$$\mathbf{c}_t^{\mathrm{p}} = f_{\mathrm{patt}}\left(\hat{\mathbf{s}}_t^{\mathrm{p}}, \mathbf{A}, \mathbf{B}\right) \tag{12}$$

$$\mathbf{s}_t^{\mathrm{p}} = \mathrm{PGRU}\left(\mathbf{c}_t^{\mathrm{p}}, \hat{\mathbf{s}}_t^{\mathrm{p}}\right) \tag{13}$$

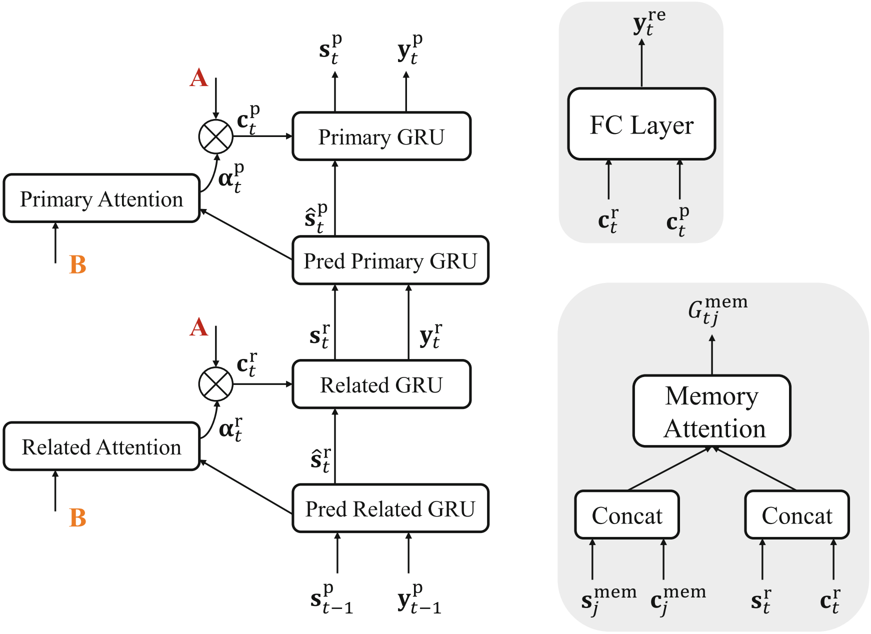
$\mathbf{s}_t^{\mathrm{p}}$
$\mathbf{y}_t^{\mathrm{p}}$
A
$\mathbf{c}_t^{\mathrm{p}}$
Primary GRU
$\boldsymbol{\alpha}_t^{\mathrm{p}}$
Primary Attention
$\hat{\mathbf{s}}_t^{\mathrm{p}}$
B
Pred Primary GRU
A
$\mathbf{s}_t^{\mathrm{r}}$
$\mathbf{y}_t^{\mathrm{r}}$
$\mathbf{c}_t^{\mathrm{r}}$
Related GRU
$\boldsymbol{\alpha}_t^{\mathrm{r}}$
Related Attention
$\hat{\mathbf{s}}_t^{\mathrm{r}}$
Pred Related GRU
B
$\mathbf{s}_{t-1}^{\mathrm{p}}$
$\mathbf{y}_{t-1}^{\mathrm{p}}$
$\mathbf{y}_t^{\mathrm{re}}$
FC Layer
$\mathbf{c}_t^{\mathrm{r}}$
$\mathbf{c}_t^{\mathrm{p}}$
G_{tj}^{mem}
Memory Attention
Concat
Concat
$\mathbf{s}_j^{\mathrm{mem}}$
$\mathbf{c}_j^{\mathrm{mem}}$
$\mathbf{s}_t^{\mathrm{r}}$
$\mathbf{c}_t^{\mathrm{r}}$

Fig. 3. The architecture of the decoder, which consists of a related decoder and a primary decoder. The right part illustrates the prediction of relation node and the memory based attention model.

where PPGRU and PGRU denote predicted primary GRU and primary GRU, respectively. f_{patt} has the same structure with f_{ratt} but the parameters are not shared. Besides, different from related decoder, we will additionally compute the probability of each primary symbol node $\mathbf{y}_t^{\mathrm{p}}$ by feeding the concatenation of related symbol node $\mathbf{y}_t^{\mathrm{r}}$, primary decoder hidden state $\mathbf{s}_t^{\mathrm{p}}$ and primary context vector $\mathbf{c}_t^{\mathrm{p}}$ into a fully connected layer with a softmax activation function:

$$p\left(\mathbf{y}_t^{\mathrm{p}}\right) = \operatorname{softmax}\left(\mathbf{W}_{\mathrm{out}}^{\mathrm{p}}\left(\mathbf{y}_t^{\mathrm{r}}, \mathbf{s}_t^{\mathrm{p}}, \mathbf{c}_t^{\mathrm{p}}\right)\right) \tag{14}$$

Then the classification loss of primary symbol node, namely the training loss of primary decoder part is defined as:

$$\mathcal{L}_{\mathrm{p}} = -\sum_t \log p\left(w_t^{\mathrm{p}}\right) \tag{15}$$

where w_t^{p} denotes the ground-truth primary symbol node at decoding step t.

In addition, to generate a subtree at each decoding step t, we still need to compute the relation node, which describes the attribute between related symbol node and primary symbol node, such as right, above, superscript and so on. This relation node is computed by feeding the concatenation of related context vector $\mathbf{c}_t^{\mathrm{r}}$ and primary context vector $\mathbf{c}_t^{\mathrm{p}}$ into a fully connected layer with a softmax activation function:

$$p^{\mathrm{re}}\left(\mathbf{y}_t^{\mathrm{re}}\right) = \operatorname{softmax}\left(\mathbf{W}_{\mathrm{out}}^{\mathrm{re}}\left(\mathbf{c}_t^{\mathrm{r}}, \mathbf{c}_t^{\mathrm{p}}\right)\right) \tag{16}$$

Then we define the training loss of the relation part as:

$$\mathcal{L}_{\mathrm{re}} = -\sum\nolimits_t \log p\left(v_t\right) \tag{17}$$

where v_t denotes the ground-truth relation node at decoding step t.

Memory Based Attention. In the above sections, we have introduced how to compute the probability of primary symbol node $\mathbf{y}_t^{\mathrm{p}}$ at each decoding step t, which can be used to determine $\mathbf{y}_t^{\mathrm{p}}$ during the testing stage. Nevertheless, we do not determine related symbol nodes in this way as there is no explicit order for related symbol nodes while primary symbol nodes always follow the depth-first order. In contrast, we adopt a memory based attention model to help determine related symbol node $\mathbf{y}_t^{\mathrm{r}}$ at each decoding step t.

Specifically, we can get the related decoder hidden state $\mathbf{s}_t^{\mathrm{r}}$, related context vector $\mathbf{c}_t^{\mathrm{r}}$, primary decoder hidden state $\mathbf{s}_t^{\mathrm{p}}$, primary context vector $\mathbf{c}_t^{\mathrm{p}}$ and primary symbol node $\mathbf{y}_t^{\mathrm{p}}$ at each decoding step t. During decoding, we append the concatenation of primary decoder state $\mathbf{s}_t^{\mathrm{p}}$ and primary context vector $\mathbf{c}_t^{\mathrm{p}}$ into the key memory and append the primary symbol node $\mathbf{y}_t^{\mathrm{p}}$ into the value memory.

To determine the related symbol node at decoding step t, as shown in Fig. 3, we employ a memory based attention using the concatenation of related decoder hidden state $\mathbf{s}_t^{\mathrm{r}}$ and related context vector $\mathbf{c}_t^{\mathrm{r}}$ as query and the concatenation of primary decoder hidden state $\mathbf{s}_t^{\mathrm{p}}$ and primary context vector $\mathbf{c}_t^{\mathrm{p}}$ as key. Then, the attention probabilities can be computed as:

$$\mathbf{G}_{tj}^{\mathrm{mem}} = \sigma\left(\nu_{\mathrm{mem}}^T\left(\tanh\left(\mathbf{W}_{\mathrm{mem}}\mathbf{z}_t^{\mathrm{r}} + \mathbf{U}_{\mathrm{mem}}\mathbf{z}_j^{\mathrm{mem}}\right)\right)\right) \tag{18}$$

where $\mathbf{z}_t^{\mathrm{r}}$ denotes the concatenation of the related decoder state $\mathbf{s}_t^{\mathrm{r}}$ and related context vector $\mathbf{c}_t^{\mathrm{r}}$, $\mathbf{z}_j^{\mathrm{mem}}$ denotes the j-th element in the key memory.

In the training stage, we define the training loss of related decoder part as a binary classification loss:

$$\mathcal{L}_{\mathrm{r}} = -\sum\nolimits_t\sum\nolimits_j\left[\bar{\mathbf{G}}_{tj}^{\mathrm{mem}}\log\left(\mathbf{G}_{tj}^{\mathrm{mem}}\right) + \left(1-\bar{\mathbf{G}}_{tj}^{\mathrm{mem}}\right)\log\left(1-\mathbf{G}_{tj}^{\mathrm{mem}}\right)\right] \tag{19}$$

where $\bar{\mathbf{G}}_{tj}^{\mathrm{mem}}$ denotes the ground-truth of the alignment between related symbol node $\mathbf{y}_t^{\mathrm{r}}$ and primary symbol node $\mathbf{y}_j^{\mathrm{p}}$. In other words, $\bar{\mathbf{G}}_{tj}^{\mathrm{mem}}$ is 1 when t-th related symbol node is aligned to the j-th element of the memory, otherwise 0.

In the testing stage, we choose $\mathbf{y}_{\hat{j}}^{\mathrm{p}}$, $\hat{j} = \operatorname{argmax}\left(\mathbf{G}_{tj}^{\mathrm{mem}}\right)$ in the value memory as the related symbol node at decoding step t.

Symbol Node Attention Guider. The alignment accuracy between input and output provided by attention models are important for recognition. However, how to train attention properly remains challenging. Therefore, we employ two symbol node attention guiders, namely related attention guider and primary attention guider, to help guide the learning of related attention and primary attention models. These two guiders can be implemented as the oracle alignment information can be acquired in the training stage in online handwritten mathematical expression recognition, namely which annotation features should be aligned when decoding each related symbol node $\mathbf{y}_t^{\mathrm{r}}$ and each primary symbol node $\mathbf{y}_t^{\mathrm{p}}$. We design these two symbol node attention guiders as:

$$\mathcal{L}_{\mathrm{rali}} = -\sum_t \sum_j \left[\bar{\alpha}_{tj}^{\mathrm{r}} \log\left(\alpha_{tj}^{\mathrm{r}}\right) + \left(1-\bar{\alpha}_{tj}^{\mathrm{r}}\right) \log\left(1-\alpha_{tj}^{\mathrm{r}}\right)\right] \tag{20}$$

$$\mathcal{L}_{\mathrm{pali}} = -\sum_t \sum_j \left[\bar{\alpha}_{tj}^{\mathrm{p}} \log\left(\alpha_{tj}^{\mathrm{p}}\right) + \left(1-\bar{\alpha}_{tj}^{\mathrm{p}}\right) \log\left(1-\alpha_{tj}^{\mathrm{p}}\right)\right] \tag{21}$$

where $\bar{\alpha}_{tj}^{\mathrm{r}}$ and $\bar{\alpha}_{tj}^{\mathrm{p}}$ denote the ground-truth alignments of related attention and primary attention models. These two guiders will be regarded as additional losses of the total training loss, namely related alignment loss and primary alignment loss, respectively.

3 Experiments

In this section, we design a set of experiments to evaluate the effectiveness of the proposed method on CROHME benchmark [14,17], which is currently the most widely used dataset for online handwritten mathematical expression recognition. We use CROHME 2014 training set as our training set, which consists of 8836 handwritten mathematical expressions and CROHME 2014 testing set as our testing set, which has 986 handwritten mathematical expressions. There are totally 101 math symbol classes and 6 math relations (above, below, right, inside, superscript (sup), subscript (sub)). To prove the generalization and robustness, we also evaluate our proposed method on CROHME 2016 and CROHME 2019 testing sets, which consist of 1147 expressions and 1199 expressions, respectively.

3.1 Training and Testing Details

Training. The overall model can be trained in an end-to-end manner and the training target is to minimize the weighted summation of the related decoder loss, primary decoder loss, relation loss, related alignment loss and primary alignment loss, which can be represented as follows:

$$O = \lambda_1 \mathcal{L}_{\mathrm{r}} + \lambda_2 \mathcal{L}_{\mathrm{p}} + \lambda_3 \mathcal{L}_{\mathrm{re}} + \lambda_4 \mathcal{L}_{\mathrm{rali}} + \lambda_5 \mathcal{L}_{\mathrm{pali}} \tag{22}$$

We set $\lambda_1 = \lambda_2 = \lambda_3 = 1$ as we believe that the prediction of the related symbol node, primary symbol node and relation node are equally important. Besides, we set $\lambda_4 = \lambda_5 = 0.1$ for alignment losses, which can be regarded as the regularization losses. For encoder, we employ 4 stacked GRU layers to extract high-level features. Each GRU layer is bidirectional and has 256 forward and 256 backward GRU units. There are two pooling layers of factor 2 on the top 2 GRU layers. Besides, the CNN is the same as DenseNet-99 in [19], which consists of three dense blocks and each block has 16 3×3 and 16 1×1 convolution layers. As for decoder, PRGRU, RGRU, PPGRU, PGRU are all unidirectional GRU layers and each layer has 256 GRU units. The attention dimension of related attention and primary attention models are both 512. The kernel size of coverage model is

7×1 and the number of output channel is 256. The embedding dimension is set to 256. We train our model by AdaDelta algorithm and the corresponding hyperparameters are set as $\rho = 0.95$, $\epsilon = 10^{-8}$. All the experiments are implemented with Pytorch and on a single NVIDIA Tesla 1080Ti 11G GPU.

Testing. In the testing stage, we expect to obtain the most likely subtree sequence. As we do not have the ground-truth related symbol nodes and primary symbol nodes during testing, we employ a hierarchical version of beam search algorithm [9] of beam size 3. Specifically, at each decoding step, 3 most likely previous primary symbol nodes (i.e., 3 hypotheses) are maintained to compute the current related symbol nodes. Then each hypothesis is expanded with 3 most likely current related symbol nodes and these current related symbol nodes are utilized to compute the current primary symbol nodes. In total, we have $3 \times 3 = 9$ hypotheses kept and then choose 3 beams according to the combined likelihood of related symbol node and primary symbol node, which are used for the next step decoding.

3.2 Recognition Performance

Table 1. Performance Comparison on CROHME 2014 testing set (in %). ExpRate denotes the percentage of predicted mathematical expressions matching the ground truth. ≤1 s. error and ≤2 s. error denote the expression recognition accuracies with at most one and two errors. StruRate only focuses on whether the structure is correctly recognized and ignores symbol recognition errors.

System	ExpRate	≤1 s. error	≤2 s. error	StruRate
I	37.2	44.2	47.3	–
II	25.7	33.2	35.9	–
III	26.1	33.9	38.5	–
WYGIWYS [10]	35.9	–	–	–
PAL [21]	39.7	–	–	–
WAP [26]	48.4	66.1	70.2	70.1
TAP [23]	48.5	63.3	67.3	67.2
TAP + WAP + LM [23]	61.2	75.5	77.7	–
SRD [24]	50.6	57.9	62.1	–
TD [25]	49.1	64.2	67.8	68.6
MRD1	**55.2**	**70.0**	**73.4**	**73.3**
MRD2	**55.8**	**72.0**	**75.3**	**75.3**

In this section, we first compare the proposed MRD based encoder-decoder system with other state-of-the-arts, including traditional methods, string decoder based encoder-decoder methods and tree-structured decoder based encoder-decoder methods on CROHME 2014 testing set. As shown in Table 1, we list the best 3 systems in CROHME 2014 competition [16], which only used official datasets. We also list the recognition performance of 5 string decoder based encoder-decoder systems and the details can be seen in [10,21,23,26]. SRD and TD are tree-structured decoder based encoder-decoder systems [24,25]. MRD1 denotes the MRD based encoder-decoder system without global visual feature while MRD2 uses global visual feature. Note that although TAP + WAP + LM can achieve a high result, it actually ensembled three TAP, three WAP and three GRU-based language models, which is not fairly comparable. Apart from expression recognition rate (ExpRate), we also adopt those with at most one, two object-level errors (≤1 s. error, ≤2 s. error) and structural recognition rate (StruRate) as additional metrics to further conduct the effectiveness of the proposed methods.

It is obvious that tree-structured decoder based encoder-decoder systems can outperform string decoder based encoder-decoder systems. Furthermore, MRD1 can achieve a significant improvement compared with SRD/TD and the ExpRate improvement is more than 5%, which demonstrates that our memory based attention model can implement more accurate alignments between related symbol nodes and primary symbol nodes and accordingly improve performance. Besides, MRD2 can still improve the performance compared with MRD1, proving the necessary of the global visual information. The improvements for ≤1 s. error, ≤2 s. error and StruRate are more significant and further conduct the effectiveness of the proposed MRD.

Table 2. Performance comparison on CROHME 2016 and CROHME 2019 testing sets (in %).

Dataset	System	ExpRate	≤1 s. error	≤2 s. error	StruRate
CROHME16	Tokyo	43.9	50.9	53.7	61.6
	São Paulo	33.4	43.5	49.2	57.0
	Nantes	13.3	21.0	28.3	21.5
	WAP	46.8	64.6	65.5	66.2
	TAP	44.8	59.7	62.8	63.1
	TAP + WAP + LM	57.0	72.3	75.6	–
	SRD	46.6	–	–	–
	TD	48.5	62.3	65.3	65.9
	MRD1	**51.3**	**65.9**	**68.9**	**69.2**
	MRD2	**52.5**	**68.4**	**71.5**	**71.7**
CROHME19	WAP	48.1	63.5	67.2	68.0
	TAP	44.2	58.8	62.7	63.6
	SRD	45.9	–	–	–
	TD	51.4	66.1	69.1	69.8
	MRD1	**52.3**	**67.3**	**70.2**	**70.8**
	MRD2	**53.6**	**68.9**	**72.1**	**72.3**

To confirm the generalization of the proposed MRD, we also compare MRD with competition systems and other state-of-the-art systems on both CROHME 2016 [17] and CROHME 2019 [14] testing sets in Table 2. The systems Tokyo, São Paulo and Nantes denote the best 3 systems of all submitted systems in CROHME 2016 competition using only official dataset and we do not list the results of submitted systems in CROHME 2019 competition as they all use additional training sets or other strategies such as ensemble. It is obvious that MRD1 can still achieve better performance compared to both string decoder based encoder-decoder systems and other tree-structured decoder based encoder-decoder systems. Similarly, MRD2 can further outperform MRD1 and the improvement is larger on these two testing sets with more ambiguous problems.

3.3 Visualization Analysis

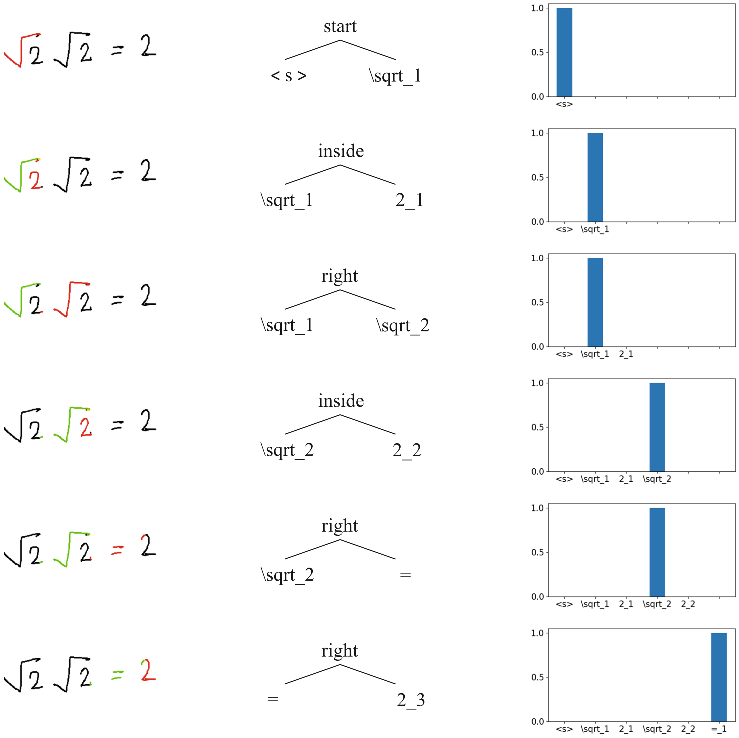

Fig. 4. An example of how MRD generates a complete tree step by step. For each step, from left to right, we show the attention visualization of related attention and primary attention, the predicted subtree and the attention visualization of memory based attention. Memory based attention actually achieves the alignments between related symbol nodes and primary symbol nodes in an explicit manner, illustrated in the right part of the figure (Color figure online).

In Sect. 3.2, we have demonstrated that the proposed MRD can outperform both previous string decoder and tree-structured decoder. In this section, we further show how MRD achieves to generate a complete tree as the recognition result for online handwritten mathematical recognition. As shown in Fig. 4, we show the attention visualization results of related attention, primary attention and memory based attention at each decoding step.

Specifically, each line in Fig. 4 denotes a decoding step, which has three parts. The left part shows the related attention result in green color and the primary attention result in red color. The middle part shows the corresponding subtree, including the related symbol node, primary symbol node and relation node. For example, the related symbol node, primary symbol node and relation node of the subtree in the second line are "\sqrt", "2" and "inside", respectively. Note that we use "\sqrt_1" instead of "\sqrt", "2_1" instead of "2" to distinguish other same symbols in this expression. The right part shows the result of memory based attention. As described in Sect. 2.2, at each decoding step, memory based attention is designed to determine which primary symbol node that the related symbol node should be corresponded to. We take the third line as an example. At this decoding step, there are already three primary symbol nodes in the memory, which are appended in the previous steps. Then, the memory based attention will compute a probability over these symbols, which is represented as the vertical coordinate. Obviously, the probability of primary symbol node, "\sqrt_1" is the largest. Therefore, we select "\sqrt_1" as the related symbol node at this step, indicating both the symbol class and the alignment between the related symbol node and primary symbol node. This memory based attention is very accurate and the probability distribution is very close to the ground-truth probability distribution (the value is nearly 1 or 0).

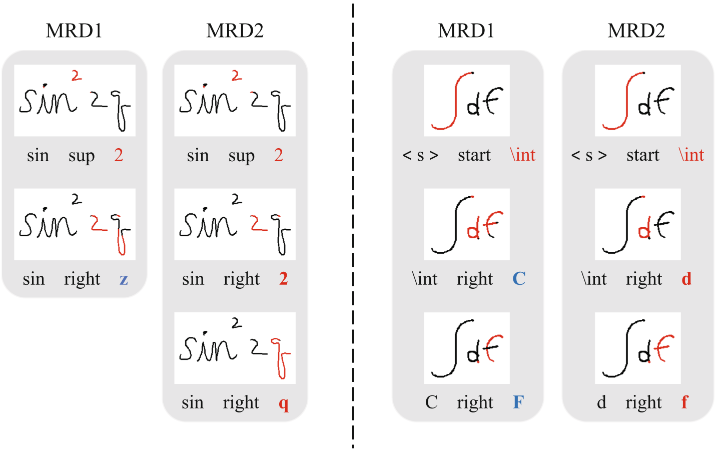

Fig. 5. Two examples to show the effectiveness of global visual information, which helps generate more accurate attention results. The incorrect recognition results are shown in blue color (Color figure online).

Furthermore, compared with MRD1, MRD2 equipped with global visual information can acquire more accurate attention results. As shown in Fig. 5, we show two examples that MRD2 can correctly recognize while MRD1 not. Note that we only show the primary attention results in red color as related symbol nodes are actually selected from primary symbol nodes. In the left example, MRD1 incorrectly recognizes "2 q" as "z" as MRD1 attends both "2" and "q" simultaneously. Therefore, the redundant parts make MRD1 misidentify "2" as "z" and the one step attention omission makes MRD1 miss "q". However, MRD2 can acquire more accurate attention with global visual information and recognize correctly. The similar observation can be seen in the right example.

4 Conclusion

In this study, we propose a memory relation decoder (MRD) for online handwritten mathematical expression recognition. To alleviate the ambiguous problems, we further introduce global visual information, which can help generate more accurate attention results. The proposed MRD can achieve significant improvements compared to string decoder based encoder-decoder methods and other tree-structured decoder base encoder-decoder methods on a benchmark published by CROHME competition, including CROHME 2014, 2016 and 2019 testing sets. Through attention visualization, we show how the proposed MRD implements the alignments between related symbol nodes and primary symbol nodes in an explicit manner and how the global visual information helps achieve better spatial attention results, which can both improve the recognition performance. In the future, we aim to investigate an approach utilizing both string and tree-structured decoders to further improve the recognition performance.

Acknowledgement. This work was supported in part by the MOE-Microsoft Key Laboratory of USTC, and Youtu Lab of Tencent.

References

1. 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 Recogn. Lett. **35**, 58–67 (2014)
2. Alvaro, F., Sánchez, J.A., Benedí, J.M.: An integrated grammar-based approach for mathematical expression recognition. Pattern Recogn. **51**, 135–147 (2016)
3. Anderson, R.H.: Syntax-directed recognition of hand-printed two-dimensional mathematics. In: Symposium on Interactive Systems for Experimental Applied Mathematics: Proceedings of the Association for Computing Machinery Inc., Symposium, pp. 436–459 (1967)
4. Awal, A.M., Mouchère, H., Viard-Gaudin, C.: A global learning approach for an online handwritten mathematical expression recognition system. Pattern Recogn. Lett. **35**, 68–77 (2014)
5. Bahdanau, D., Chorowski, J., Serdyuk, D., Brakel, P., Bengio, Y.: End-to-end attention-based large vocabulary speech recognition. In: International Conference on Acoustics, Speech and Signal Processing, pp. 4945–4949 (2016)

6. Belaid, A., Haton, J.P.: A syntactic approach for handwritten mathematical formula recognition. IEEE Trans. Pattern Anal. Mach. Intell. **1**, 105–111 (1984)
7. Chan, K.F., Yeung, D.Y.: Mathematical expression recognition: a survey. Int. J. Doc. Anal. Recogn. **3**(1), 3–15 (2000)
8. Chan, W., Jaitly, N., Le, Q., Vinyals, O.: Listen, attend and spell: a neural network for large vocabulary conversational speech recognition. In: International Conference on Acoustics, Speech and Signal Processing, pp. 4960–4964 (2016)
9. Cho, K.: Natural language understanding with distributed representation. arXiv preprint arXiv:1511.07916 (2015)
10. Deng, Y., Kanervisto, A., Ling, J., Rush, A.M.: Image-to-markup generation with coarse-to-fine attention. In: International Conference on Machine Learning, pp. 980–989 (2017)
11. He, T., et al.: Layer-wise coordination between encoder and decoder for neural machine translation. In: Advances in Neural Information Processing Systems, pp. 7944–7954 (2018)
12. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017)
13. Huang, P.Y., Liu, F., Shiang, S.R., Oh, J., Dyer, C.: Attention-based multimodal neural machine translation. In: Conference on Machine Translation, vol. 2, pp. 639–645 (2016)
14. Mahdavi, M., Zanibbi, R., Mouchere, H., Viard-Gaudin, C., Garain, U.: ICDAR 2019 CROHME+ TFD: competition on recognition of handwritten mathematical expressions and typeset formula detection. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 1533–1538. IEEE (2019)
15. Miller, E.G., Viola, P.A.: Ambiguity and constraint in mathematical expression recognition. In: AAAI, pp. 784–791 (1998)
16. Mouchere, H., Viard-Gaudin, C., Zanibbi, R., Garain, U.: ICFHR 2014 competition on recognition of on-line handwritten mathematical expressions (CROHME 2014). In: International Conference on Frontiers in Handwriting Recognition, pp. 791–796 (2014)
17. Mouchère, H., Viard-Gaudin, C., Zanibbi, R., Garain, U.: ICFHR 2016 CROHME: competition on recognition of online handwritten mathematical expressions. In: 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 607–612. IEEE (2016)
18. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
19. Wang, J., Du, J., Zhang, J.: Stroke constrained attention network for online handwritten mathematical expression recognition. arXiv preprint arXiv:2002.08670 (2020)
20. Wang, J., Du, J., Zhang, J., Wang, Z.R.: Multi-modal attention network for handwritten mathematical expression recognition. In: International Conference on Document Analysis and Recognition, pp. 1181–1186 (2019)
21. Wu, J.-W., Yin, F., Zhang, Y.-M., Zhang, X.-Y., Liu, C.-L.: Image-to-markup generation via paired adversarial learning. In: Berlingerio, M., Bonchi, F., Gärtner, T., Hurley, N., Ifrim, G. (eds.) ECML PKDD 2018. LNCS (LNAI), vol. 11051, pp. 18–34. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-10925-7_2
22. Zanibbi, R., Blostein, D., Cordy, J.R.: Recognizing mathematical expressions using tree transformation. IEEE Trans. Pattern Anal. Mach. Intell. **24**(11), 1455–1467 (2002)

23. Zhang, J., Du, J., Dai, L.: Track, Attend and Parse (TAP): an end-to-end framework for online handwritten mathematical expression recognition. IEEE Trans. Multimedia **21**(1), 221–233 (2019)
24. Zhang, J., Du, J., Yang, Y., Song, Y.Z., Dai, L.: SRD: a tree structure based decoder for online handwritten mathematical expression recognition. IEEE Trans. Multimedia (2020)
25. Zhang, J., Du, J., Yang, Y., Song, Y.Z., Wei, S., Dai, L.: A tree-structured decoder for image-to-markup generation. In: International Conference on Machine Learning, pp. 11076–11085. PMLR (2020)
26. Zhang, J., et al.: Watch, attend and parse: an end-to-end neural network based approach to handwritten mathematical expression recognition. Pattern Recogn. **71**, 196–206 (2017)