International Journal on Document Analysis and Recognition (IJDAR) (2022) 25:339–350
https://doi.org/10.1007/s10032-022-00409-4

SPECIAL ISSUE PAPER

Combination of explicit segmentation with Seq2Seq recognition for fine analysis of children handwriting

Omar Krichen[2] · Simon Corbillé[1] · Éric Anquetil[2] · Nathalie Girard[1] · Élisa Fromont[1] · Pauline Nerdeux[2]

Received: 15 March 2022 / Revised: 15 June 2022 / Accepted: 22 August 2022 / Published online: 7 September 2022

Abstract
We consider the task of analyzing children handwriting in the context of a dictation task. The objective is to detect orthographic and phonological errors. To achieve this goal, we extend an existing handwriting analysis engine, based on an explicit segmentation of the handwritten input, originally developed for children copying exercises. We present a new approach, based on the combination of this analysis engine with a deep learning word recognition approach in order to improve both the recognition and segmentation performance. Explicit segmentation needs prior knowledge, and the deep network recognition predictions are a reliable approximation of the ground truth which can guide the analysis process. We propose to combine multiple prior knowledge strategies to further improve the analysis performance. Furthermore, we exploit the deep network approximate implicit segmentation to optimize the existing analysis process in terms of complexity.

Keywords Online handwriting recognition · Segmentation · Digital learning · Degraded handwriting · Sequence-to-sequence · e-Education

1 Introduction

This work aims at designing an educational system targeted toward primary school children, in order to help them master handwriting and spelling skills. More specifically, we deal with online interpretation of children handwritten **French cursive words**. The interpretation task in hand is a *word analysis task*, which differs from the *word recognition task*. Figure 1 illustrates these differences. In a recognition task, the objective of the system is to predict the correct character sequence, whereas the objective of the analysis task is to provide a qualitative evaluation. Consequently, the segmentation quality is instrumental, to enable the system to perform a fine-grained analysis of the pupil handwriting, such as highlighting in red the spelling mistakes directly on the ink (*c.f.* Fig. 1). Therefore, the educational system needs both an accurate recognition of the child's word but also a good segmentation at character level to precisely locate the spelling mistakes. To achieve this goal, we build on previous works on children handwriting analysis for cursive French words [1]. This approach is based on an explicit segmentation of the input word. A segmentation graph representing all possible segmentations of the word into letters is created. For each node of the graph, letters hypotheses are computed using a letter recognition and analysis system. The analysis result is a set of *n* best possible pseudo-word hypotheses. In order to be efficient, the explicit segmentation needs to be driven by *prior knowledge*, especially to deal with degraded children handwriting. Since the instruction to copy was displayed to the child, it served as prior knowledge to guide the letter hypotheses computation phase. This "base system" is discussed in more detail in Sect. 3. Our new targeted dictation task introduces new challenges, as illustrated in Fig. 2. The instruction is heard, not seen, by the pupil. This may induce a lot more spelling mistakes. In the figure, the written word "mai" is a homophone of the dictated instruction "mes." In this dictation context, the instruction is not directly exploitable to guide the analysis of the handwritten word. To provide a relevant and real-time analysis for this dictation task, new prior knowledge generation strategies are needed. We propose to combine the aforementioned engine,

Omar Krichen and Simon Corbillé have contributed equally to this work.

✉ Omar Krichen
omar.krichen@irisa.fr

Simon Corbillé
simon.corbille@irisa.fr

[1] IRISA Lab, Univ Rennes 1, 35000 Rennes, France

[2] IRISA Lab, INSA Rennes, 35000 Rennes, France

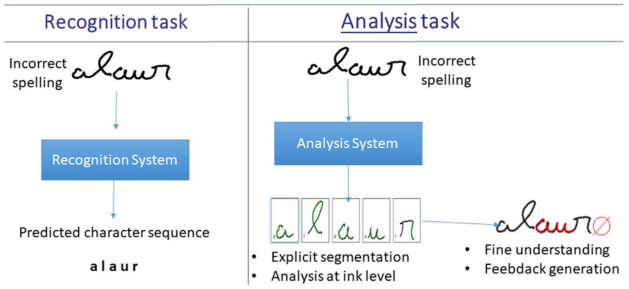


Fig. 1 Context: analysis of children handwriting: the dictated instruction is **“alors”** (“then” in French)

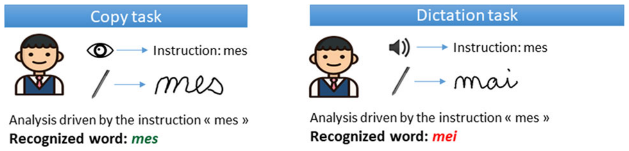


Fig. 2 Difference between a copy and a dictation task

with a deep learning word recognition approach, namely a Seq2Seq architecture. Our contributions consist in exploiting this hybridization in three different manners: (1) We define the Seq2Seq network recognition process as a new prior knowledge generation strategy, which will drive the analysis process; (2) We combine different prior knowledge strategies to further improve the system’s performance; (3) We exploit the Seq2Seq implicit segmentation to prune the explicit segmentation graph and optimize analysis complexity. This paper is organized as follows. Section 2 presents related works about handwriting recognition and segmentation. Section 3 provides a detailed account of the existing engine, while Sect. 4 describes the deep learning model used for our task. Section 5 presents the approaches combination and our listed contributions. Experiments are presented in Sect. 6. Conclusion and future works are given in Sect. 7.

2 Related works

This section presents the latest online and offline methods concerning handwriting recognition and segmentation. Handwriting can be represented **offline**, through an image, or **online** through a sequence of points. IAM datasets (offline [2] and online [3] versions) are composed of English adult-written sentences, **labeled at line level**. They are open and widely used to compare pure **recognition methods**. To the best of our knowledge, there are no available words datasets with character-level annotation.

2.1 Handwriting text recognition

Deep learning models outperform the previous methods [4,5] on handwriting text recognition (HTR) task. These traditional methods were based on a **bottom-up strategy**, i.e., by using expert knowledge to segment input data, then recognizing the character in each segmented element. A great advantage of deep learning models lies in the fact that there are end-to-end trainable. There is no need to segment the data, and the feature extraction is learned by the model. The two main deep learning approaches that tackle HTR are Connectionist Temporal Classification (CTC) [6] and Sequence to Sequence (Seq2Seq). The CTC approach divides input into frames for symbol prediction and computes a probability distribution over all possible outputs alignments, while the Seq2Seq approach translates an input sequence represented by an image into a sequence of characters. The CTC-based architectures designed for online recognition use **Bidirectional Long-Short Term Memory** [7] (BLSTM). The authors of [8] show that this type of architecture outperforms a traditional method based on **Hidden Markov Models**, whereas the authors of [9] use BLSTM with Bézier curvers encoding of online data to achieve state-of-the-art performances for online recognition on IAM-OnDB [3]. The CTC-based architectures designed for offline recognition are slightly different due to the nature of the input data. **Convolutional recurrent neural networks** [10,11] are based on a convolutional neural network coupled with a recurrent network with LSTM cell. The authors of [12] use a Seq2Seq method based on an encode-r-decoder model with an **attention module** to do offline recognition. More recently, [13] and [14] use **transformers**, which need a lot of synthetic data to perform well, for offline recognition. For our work, we use a Seq2seq model since this architecture gets state of the art performances when no synthetic data are used. The next part present methods which focus on handwriting segmentation.

2.2 Handwriting segmentation

The authors of [15] propose regularization methods on the CTC loss based on entropy and spacing to increase recognition performance and segmentation quality. They present a quantitative analysis on recognition performance and **qualitative analysis on segmentation** performance. The authors of [16] use a **convolutional prototype network** and most aligned frame based CTC training for handwriting recognition. They evaluate the recognition performance of their model on IAM [2] dataset whereas the segmentation is evaluated on a synthetic dataset representing a sequence of digits from MNIST [17] dataset. In this work, we choose to combine the Seq2Seq model good recognition performances with the explicit segmentation-based existing engine [1] presented in the introduction. The next Sect. 3 presents the existing system.

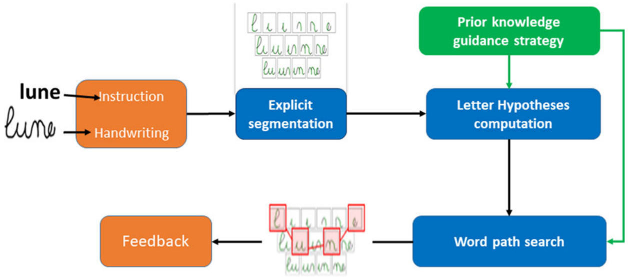


Fig. 3 Existing analysis engine, here the instruction serves as prior knowledge to guide the analysis

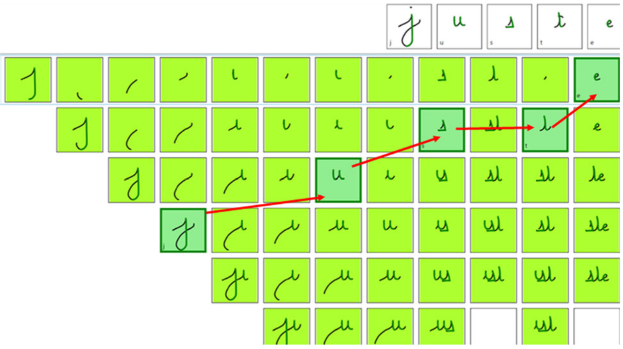


Fig. 4 Segmentation graph for the word "juste"

3 Existing analysis engine

In this section, we present the existing analysis engine (for more details, see [1]). Figure 3 illustrates its global principles.

Given the handwritten input and the instruction, the first step of the analysis is the explicit segmentation process. A segmentation graph is constructed based on the extraction of all possible cutting points around descending zones [18] and represents a partition of all possible segmentations given the extracted cutting points. Figure 4 illustrates the segmentation graph for the French handwritten word "juste."

Every node of the graph represents a possible letter hypothesis. The objective is to find the best path in the graph corresponding to the correct segmentation. For each node, confidence-based classifiers [19] compute letters hypotheses. The analysis process is generic and relies on *prior knowledge generation strategies*. Here, prior knowledge is instrumental, especially in the context of degraded handwriting to avoid recognition confusion at the letter level.

In a copying context, the prior knowledge strategy is straightforward. The instruction drives the letter computation process by filtering the computed hypotheses that belong to the instruction. The best segmentation path is the one **which minimizes the edit distance with the instruction**. This strategy is best suited when the child correctly reproduces the instruction. A first adaptation of this engine to the dictation context was proposed in [20]. Two prior knowledge generation strategies were defined to deal with the fact that driving the analysis by the instruction, in a dictation context, becomes obsolete. The first strategy consisted in asking the child to type was he/she has written on the keyboard. This *childtyping* drives the analysis, since it is a pretty reliable estimation of the ground truth. However, the objective is to be free from user input and to rely solely on the system capacities. The second prior knowledge generation strategy was to generate, for every instruction, a set of phonetically similar pseudo-words. For example, if the instruction is "alors" (*then* in French), the generated hypotheses would be "alaur, alor, alord, alort." This generation is based on the *Phonetisaurus engine* [21], a grapheme-to-phoneme WFST (Weighted Finite State Transducer). A Recurrent Neural Network Language Model (RNNLM) is used to extract the best phonetic hypotheses for a given word. This prior knowledge generation strategy enables to cover potential orthographic errors that sound similar to the instruction. The limit of this strategy resides in the fact that it could not cope with written words that were not phonetically similar to the dictated instruction. It is in order to overcome these limits that we choose to combine the existing analysis engine with the outputs of a Seq2Seq model, namely the predicted word and the correspondent implicit segmentation. The new prior knowledge generation strategy will therefore rely on the Seq2Seq predicted word to drive the generic analysis process. Section 4 describes the Seq2Seq architecture used, whereas Sect. 5 presents the combination of the approaches and its impact.

4 Deep learning model for handwriting recognition

Our **Seq2Seq model** is derived from [12] for the **encoder decoder architecture with hybrid Bahdanau attention** mechanism [10,11,14,22,23] for the encoder architecture. The encoder's parameters result of an ablation study where the number of convolutional, pooling, blstm layers and dropout are tested.

The authors of [12] demonstrate that using a joint training between encoder and decoder improves recognition performance. The encoder is trained with CTC loss [6] and the decoder with a cross-entropy. Thus, **the model makes one prediction with the encoder and one prediction with the decoder**. The final loss is defined as follows:

$$Loss = \lambda * Loss_{ctc} + (1 - \lambda)$$
$$* Loss_{crossentropy}, with \ \lambda \in [0, 1]$$

Figure 5 illustrates the connection with its three main parts: (1) The encoder performs the feature extraction of the input image into a feature vector. This vector is used by the encoder to make a word prediction; (2) The attention module focuses

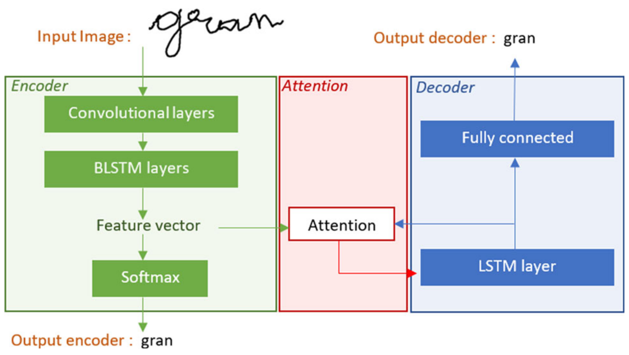


Fig. 5 Global architecture of sequence to sequence model

Table 1 Configuration of encoder: k is for kernel size, s for stride, p for padding and d for dropout

Type	Configuration
Input	height 128 * width
Convolution	#filters:8, k:3*3, s:1, p:0
Max pooling	k:2*2, s:2, d:0.2
Convolution	#filters:16, k:3*3, s:1, p:0
Max pooling	k:2*2, s:2, d:0.2
Convolution	#filters:32, k:3*3, s:1, p:0
Max pooling	k:2*2, s:2, d:0.2
Convolution	#filters:64, k:3*3, s:1, p:0, d:0.2
Convolution	#filters:128, k:4*2, s:1, p:0, d:0.2
Collapse convolution	#filters:128, k:9*1, s:1, p:0
Batch normalization	
BLSTM	4 layers, 128 units, d:0.5
Fully connection	Size alphabet + 1

All convolution layers are followed by the Leaky ReLU activation function, then a layer normalization

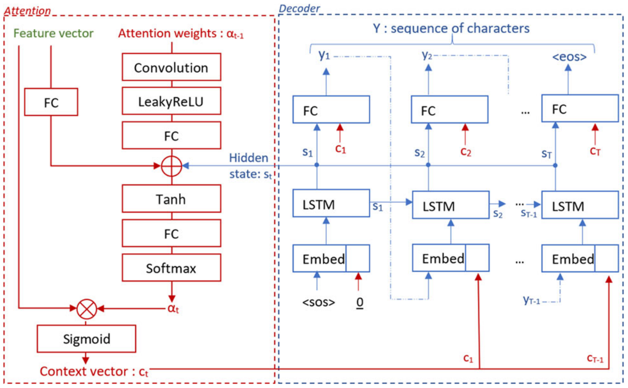


Fig. 6 Details attention module and decoder: the input is the feature vector produced by the encoder. The decoder produces one character at a time. It starts with the special character < sos > (start of sequence) and ends with < eos > (end of sequence). FC stands for fully connected layer, Tanh the tangent hyperbolic function and Embed the embedding of one prediction

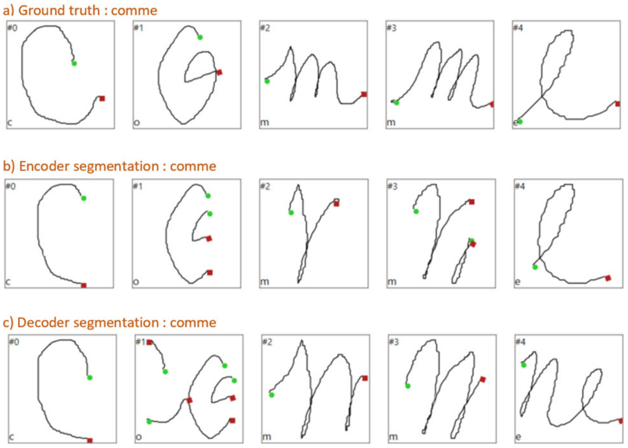


Fig. 7 Example of segmentation for the encoder/decoder

the decoder on a specific area in the feature vector; (3) The decoder decodes the feature vector and produces a word prediction.

The model takes as input a grayscale image resized proportionally to have a height of 128 pixels. The encoder first extracts spatial features with convolutional layers, then temporal features with recurrent layers, into a feature vector. This feature vector is used by the encoder to make a prediction and by the decoder through the attention module. Table 1 details the encoder's parameters.

Figure 6 illustrates the attention mechanism. The idea is to focus the decoder on a specific part of the feature vector, and thus ideally use features associated with a sub image representing one letter. The attention module produces at each time a context vector c_t from the feature vector emitted by the encoder and uses the hidden state of the decoder s_t. At each time, the decoder uses an embedding of the precedent prediction and the precedent context vector to update the hidden state s_t of the LSTM layer, then uses the hidden state to concatenate with the current context vector to produce the symbol prediction at the time t. The decoder's alphabet uses two extra symbols for the start and the end of the characters sequence (<sos> and <eos>).

For the character segmentation aspect, an approximation can be computed from the encoder or decoder prediction. For the encoder, we compute the receptive fields used to predict a character and extract the associated part of the image to get the segmentation. For the decoder, we re-use the attention map used by the decoder to predict a character and find its position in the associated input image. Figure 7 illustrates an example of segmentation of the French word "comme" by the Seq2Seq model. The segmentation quality is average due to the fact the network is trained on the recognition task. For the encoder, the letter "o" and "m" are incomplete. The decoder segmentation contains a lot of overlap between the letters "o" and "e." Section 6 details quantitative results for the segmentation and recognition evaluation.

This motivates our choice to combine a deep learning model, which does well recognition-wise, with the existing analysis engine, which does well segmentation-wise. Furthermore, even if the segmentation of deep model is approximate, it can be exploited to prune the explicit segmentation graph. The next section describes the hybridization between the two systems.

5 Combining deep recognition and explicit segmentation

In this section, we present the integration of the Seq2Seq recognition results into the explicit segmentation-based analysis process, the new prior knowledge generation strategies, as well as the pruning of the explicit segmentation graph.

5.1 Seq2Seq prediction as prior knowledge strategy

Figure 8 illustrates the defined prior knowledge generation strategy, which consists in coupling the explicit segmentation-based analysis approach with the Seq2Seq recognition outputs. The predicted sequence for each written word drives the generic analysis process, especially in the letter hypotheses computation phase, and the word paths search phase. Being a better approximation of the ground truth in a dictation context, this *deep prediction strategy* improves the engine performance, as we will see in Sect. 6.

A valid interrogation would be to question the fact that our system now has two recognition processes. A recognition process for each letter hypothesis (with Evolve classifier [24]), and a Seq2Seq recognition on the whole word. Shouldn't we rely on one or the other? The final goal is to provide feedback to the pupils at the ink level; therefore, the segmentation process is as important as the recognition process in our task. The fact that the existing analysis system relies on an explicit segmentation process, with a recognition at letter level, ensures that the predicted result is coherent in terms of letters localization. However, since we are faced with degraded children handwriting, the system needs some prior knowledge to prioritize the relevant letters hypotheses, hence the guidance of the analysis by the deep predicted sequence.

5.1.1 Deep prediction added value

Figure 9 illustrates the analysis of the written word “zme”, given the dictated instruction “cent” (hundred), with the three strategies: (a) Instruction strategy with result=“cent”; (b) Phonetic strategy with result=“cent”; (c) Deep recognition strategy with result=“zme.”

The instruction strategy is well suited when there are no errors, but can't cope with the analysis of children mistakes. As for the phonetic strategy, it is not well adapted to this situation either, since the written word “zme” does not sound similar to the dictated instruction “cent.” As for the third strategy, since the network was able to predict the correct word, the injection of this prior knowledge enabled the engine to correctly recognize and segment the word.

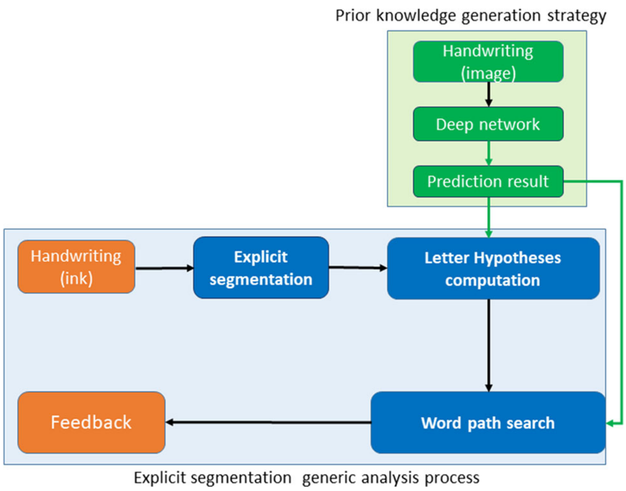

Prior knowledge generation strategy
Handwriting (image)
Deep network
Prediction result
Handwriting (ink)
Explicit segmentation
Letter Hypotheses computation
Feedback
Word path search
Explicit segmentation generic analysis process

Fig. 8 Deep prediction as prior knowledge strategy

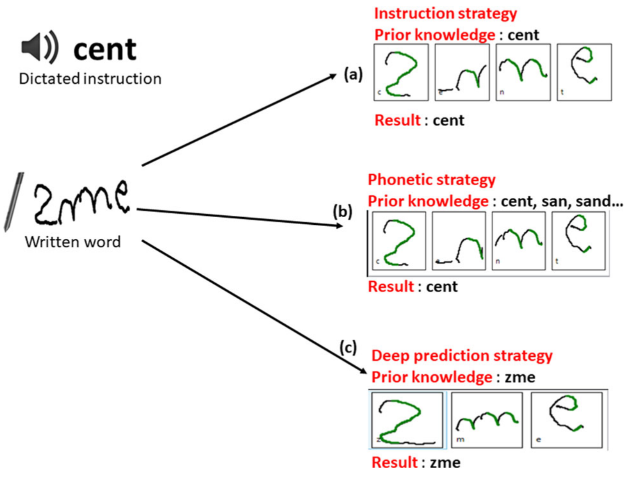


Fig. 9 Results of analysis strategies for the word “zme”, given the instruction “cent”

5.2 Strategies combination

Until now, we have studied the case where the Seq2Seq model is able to predict the correct sequence, and therefore have a positive impact as prior knowledge on the analysis engine. However, there are cases where it is not able to correctly interpret the input, such as in Fig. 10, which illustrates the analysis results of the written word *“biin”*, given the dictated instruction *“bien”*. We can see that the first two strategies ((a) and (b)) were only able to predict the first written letter “b,” which is also the first letter of the dictated instruction, whereas the third strategy (c) was only able to predict the lat-

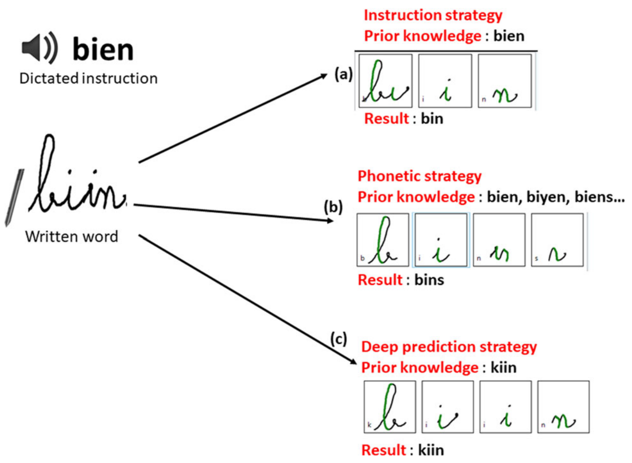


Fig. 10 Results of analysis strategies for the word “biin,” given the instruction “bien”

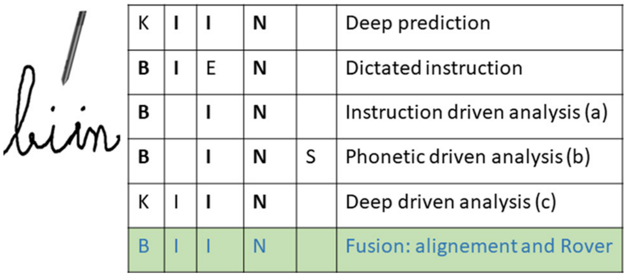
ter part of the word “iin.” Intuitively, since every strategy is best suited to a specific scenario, it is fair to assume that they could be complementary. We propose therefore to combine these strategies into a fourth one, named *fusion and competition*. The latter represents two ways of combining strategies, first a conjunction by merging these prior knowledge, then a dis-junction by introducing a notion of competition between the strategies prediction. We present now in detail the two steps of this fourth strategy.

5.2.1 Fusion

We propose the fusion of the results of the three mentioned strategies to generate an alternative approximation of the ground truth, which will serve as another prior knowledge source driving the analysis. This fusion is done in two steps: first by aligning the resulting character sequences using dynamic programming techniques, and second by introducing a voting algorithm called Rover [25], which chooses to most occurring character in the alignment. Figure 11 illustrates the alignment and fusion of the above-cited strategies, with the addition of the instruction and the deep model prediction. The fusion result corresponds to the ground truth “biin.” Therefore, if used as prior knowledge, it will enable the analysis engine to predict the correct word.

5.2.2 Competition

After the fusion step, which adds pertinent prior knowledge information, we introduce the competition step, which enables the system to choose the best strategy, depending on child production. Figure 12 illustrates this process. To choose the best prediction between instruction strategy, phonetic strategy, deep prediction strategy, and the fusion, we exploit metrics that are already present in the existing analysis engine. As explained in Sect. 3, the result of each analysis process is the segmentation path, which minimizes the edition distance with the prior knowledge that guides the instruction. This edition score consists of a Damerau–Leveinshtein [26] distance computed between the word hypothesis and the prior knowledge (e.g. the instruction). In addition, optimized costs are learned by the analyzer [1]. Another indication is the handwriting quality, represented by the analysis score. The analysis score S$_a$ of a path of length n P$_n$ is defined as follows, where S$_a$(i) is the analysis score of the *ith* element of the path:



K	I	I	N		Deep prediction
B	I	E	N		Dictated instruction
B		I	N		Instruction driven analysis (a)
B		I	N	S	Phonetic driven analysis (b)
K	I	I	N		Deep driven analysis (c)
B	I	I	N		Fusion: alignement and Rover

Fig. 11 Alignment and fusion of multiple prior knowledge for the written word “biin”

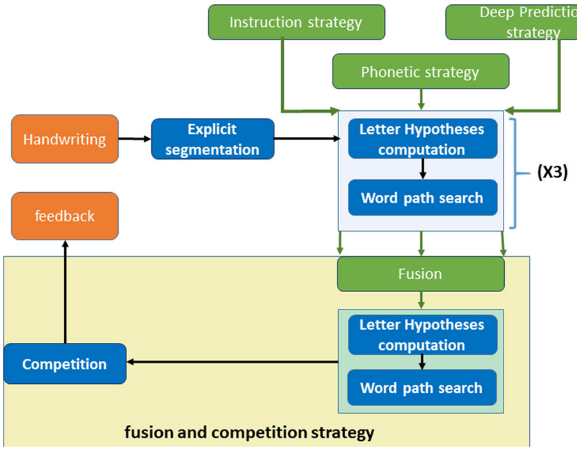


Fig. 12 Fusion and competition strategy

$$S_a(P_n) = \sqrt{\prod_{i=0}^{n} S_a(i)}\ [1].$$

Given these two metrics, we define a phonetic score that combines edition score pertinence and handwriting quality. The phonetic score is defined as follows:

$$PhoneticScore(P) = S_a(P) * 0.7 + \frac{1}{1 + |EditScore(P)|} * 0.3$$

The strategy chosen is the one where the predicted segmentation path has the best phonetic score. These parameters

(0.7, 0.3) are chosen empirically to give more weight to the analysis score of each strategy. We will see in detail the impact of fusion and competition strategy in Sect. 6. In this section, we have presented the integration of the Seq2Seq recognition results in the existing analysis chain and the proposed strategies to optimize the analysis process. Another output of the Seq2Seq model is the result of the implicit segmentation. We choose to use this segmentation result in order to prune the existing analysis process segmentation graph, which would enable to diminish the complexity of the process. Since we are in the context of real-time user interaction, the response time of the system has to be acceptable to the user. However, for long words, the analysis time can be fastidious. Moreover, the fusion and competition strategy increases the analysis complexity. We present in the next section this segmentation graph pruning strategy.

5.3 Segmentation graph pruning

The word path search step of the analysis (*c.f.* Fig. 8 in Sect. 5) generates all the possible segmentation paths from the graph. From all the paths generated, the one minimizing the edit distance with the prior knowledge is chosen as the prediction of the written word. We exploit the approximate implicit segmentation of the Seq2Seq model to prune the segmentation graph. The implicit segmentation is not directly exploitable to provide feedback, but can help optimize the analysis process. The objective is to have a nice trade-off between the analysis process performance and complexity. Figure 13 illustrates the word paths search process for the written word “alors.” For each node of the first level of the graph (highlighted in blue rectangles), all possible segmentation nodes paths are recursively constructed. Each node having at most four letter hypotheses with their analysis score, all segmentation paths (or word hypotheses) resulting from each segmentation node path are then generated. The Seq2Seq segmentation of the written word “alors” is framed in red in Fig. 14. Each rectangle represents the predicted letters as well as the points used by the attention mechanism to recognize it. This is used to prune the segmentation graph. First, a *deep matching score* (which is in fact an IoU score between the points in a graph segmentation node and the points in a deep segmentation node) is computed for each node of the graph relatively to the deep segmentation, to find the best corresponding deep predicted letter. The deep matching score is defined as follows:

$$DMScore(n_{graph}, n_{deep}) = \frac{\|points_{ngraph} \cap points_{ndeep}\|}{\|points_{ngraph} \cup points_{ndeep}\|}$$

The best deep matching node for a graph segmentation node is defined as follows:

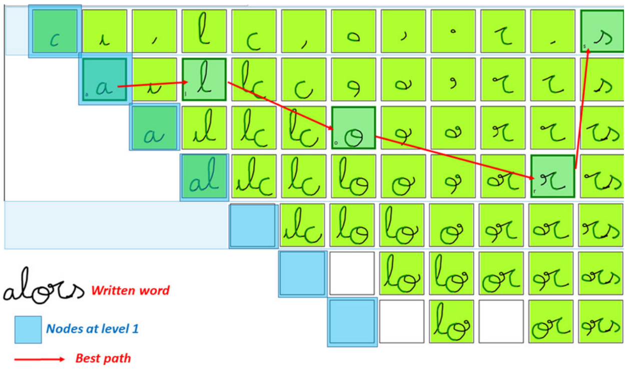


Fig. 13 Segmentation graph of written word “alors”

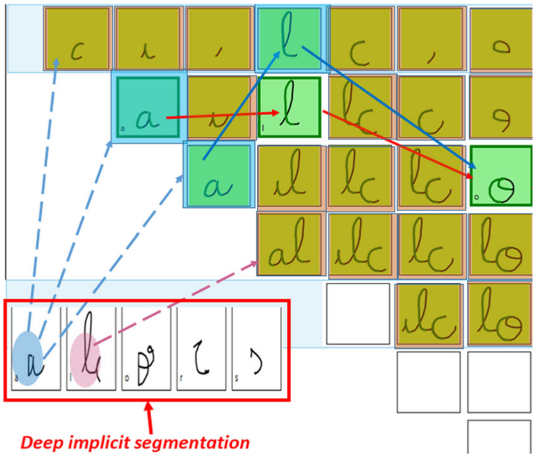


Fig. 14 Pruning process for part of the graph

$$DeepMatch(n_{Graph}) = \max_{nDeep \in Deep} DMScore(n_{Graph}, n_{Deep}).$$

Given the computed deep matching scores, the new segmentation paths search process consists in **selecting recursively, at each level, only the nodes whose analysis hypotheses contain the matching deep node predicted letter**, formalized as follows:

$$SelectedNodes(level_i) = n_{Graph} \in level_i,$$
$$\text{such as } DeepMatch(n_{Graph}) \in AnalysisHypotheses(n_{Graph}).$$

Figure 14 illustrates this pruning process for part of the segmentation graph. Dotted arrows represent the matching process at the first level. Nodes highlighted in red represent the discarded nodes, since their analysis hypotheses do not contain the predicted letter from the matched deep node. We can see that at the first level of the graph, only the relevant nodes have been selected. This is due to the fact that the implicit segmentation of the deep network was relatively consistent with the explicit segmentation.

In the example in Fig. 14, without the pruning strategy, the number of processed paths is 301, and goes down to only 18

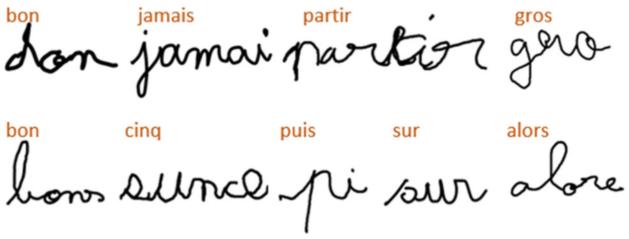


Fig. 15 Examples of cursive words written by children

paths when the pruning is activated. In both cases, the correct word and segmentation are predicted. We will see more in detail its impact, as well as the performance of the analysis engine in the next section.

6 Experiments

6.1 Dataset

This work needs **data annotated at character level** to evaluate the system on **recognition** and **segmentation** aspects. To our knowledge, open datasets of children handwriting with character annotation for words do not exist. For our experiments, we use a private dataset, composed of French cursive words written by children. The data were collected in classrooms on pen-based tablets and were recorded as **multivariate time series**. Each word is a sequence of points represented by their coordinates (x and y), their pressure and their time. Unfortunately, these children data are not publicly available due to RGPD laws.[1] Figure 15 illustrates examples of words in the database (the instruction is in orange). We can see that the handwriting is degraded because children are still learning writing, and naturally they do some mistakes. Another interesting aspect is the diversity of misspelling errors.

Our dataset is split into 6812 words written by more than 500 children for the training set and 1242 words written by more than 300 children for the test set. Train and test datasets come from different data acquisition campaigns (and different classroom). There are no children data present both in train and test set, this enables us to verify the ability of the system to generalize on unseen writing styles.

6.2 Deep learning model evaluation

For each experiment, λ of hybrid loss is set to 0.5 as suggested in [12]. We evaluate our deep learning model on the IAM-OnDB dataset [3] which is composed of adult handwritten English text. We train the model on a combination of train set and validation set with RMS prop optimizer during 200 epochs, then evaluate it on a test set. We set the learning rate at 0.001 and the batch size at 16. We evaluate the encoder and the decoder of our Seq2Seq model. Table 2 report the error rate on the test set. We can see that the encoder performs better than the decoder and outperforms the state of the art without the use of language model.

The deep learning model performs poorly with only children data. We use the model trained on IAM-OnDB then continue the training on the children handwriting.

Cross-validation with k folds equal to 10 is performed on **the training set** to evaluate the robustness of the system. The training set is split into 10 chunks. A fold is composed of a training part which represent 8 chunks, a validation part of 1 chunk and a test part of 1 chunk. Each fold results in a different splitting of the training set, thus all training set data are used for training and testing. For each fold, the validation set is used to choose the best model. A fold is evaluated on the test fold for the recognition task and the whole test set for the recognition and segmentation task. The recognition is evaluated with a recognition rate (100 - Word error rate) and the **intersection over union to evaluate the segmentation** (qualitative results are presented in Sect. 4). The Table 3 reports the results. We use the encoder prediction (label and segmentation) for the next experiments because its recognition rate are better on test fold. The recognition rate is better in fold test set because the data in the whole test set are from words written by unseen written styles. The Seq2Seq model has a greater recognition rate than the existing analysis engine (see more details on results in Sect. 6.4) while the segmentation rate is too approximate to make a precise feedback to the children. Combining the Seq2Seq model with the existing analysis engine makes it possible to have a model both efficient in recognition and segmentation. The next section presents the results of the different combination strategies.

6.3 Segmentation evaluation

To study the segmentation from a qualitative viewpoint, Fig. 16 illustrates the analysis results of the written word "gust." We can see that the raw deep segmentation (e) is

Table 2 Error rates on the IAM-OnDB test set in comparison with the best of state of the art

System	CER (%)	WER (%)
Without model language [9]	5.9	18.6
With model language [9]	4.0	10.6
Our Seq2Seq encoder	5.0	18.3
Our Seq2Seq decoder	5.5	20.2

CER character error rate, *WER* word error rate

[1] https://ec.europa.eu/info/law/law-topic/data-protection/data-protection-eu_fr.

Table 3 Mean and standard deviation for the recognition and segmentation (IoU) evaluation on children handwriting

	Encoder	Decoder
Fold recognition rate	86.65 ± 1.17	86.32 ± 1.25
Test recognition rate	75.08 ± 1.16	69.20 ± 2.17
Segmentation rate (IoU)	51.14 ± 7.06	45.91 ± 3.19

Recognition is evaluated on fold test set and whole test set. Encoder and decoder from Seq2Seq are evaluated in %

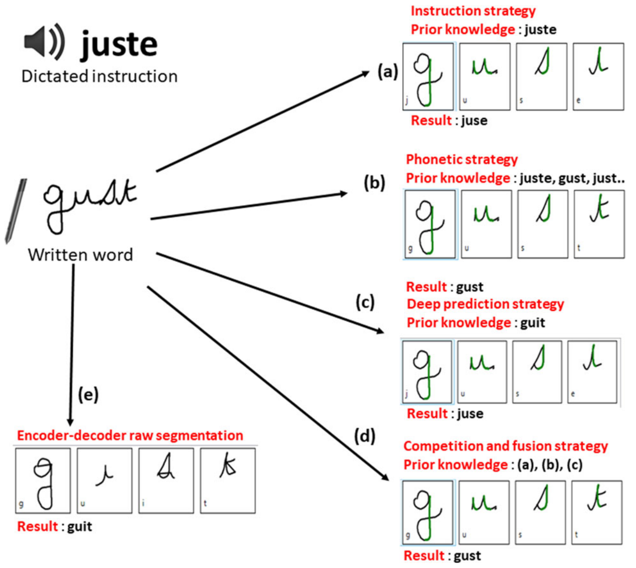


Fig. 16 Segmentation results for the word "gust", given the instruction "juste"

Table 4 Segmentation (IoU) performance of each strategy

Strategy	Segmentation rate (IoU)
Raw Seq2Seq	51.14 ± 7.06 %
Childtying strategy	**93.67**%
Instruction strategy	88.66 %
Phonetic strategy	88.72%
Deep prediction strategy	90.4% ± 0.54%
Fusion and competition	92.82% ± 0.28 %

Best approach and performance is indicated in bold

Table 5 Recognition performance of each strategy

Strategy	Recognition rate
Childtying strategy	78.98%
Instruction strategy	64.09 %
Phonetic strategy	66.42%
Deep prediction strategy	72.18% ± 0.73 %
Fusion and competition	**83.28%** ± 0.51 %

Best approach and performance is indicated in bold

approximate, compared to the explicit segmentation driven by the defined strategies (a, b, c, d). In this example, the phonetic strategy performed the best in terms of edition and analysis score, and therefore was chosen within the fusion and competition strategy. Correct segmentation and ground truth detection were performed.

We can observe the same results on the whole dataset, in terms of quality of segmentation. As the ground truth is annotated at the character level, we can therefore study how well the test set was segmented using the IoU metric. Table 4 illustrates the quality of segmentation for each strategy, from a quantitative viewpoint. Deep prediction and fusion/competition strategies are tested on the 10 models generated from the cross-validation. Mean and standard deviation results are reported.

As we have seen, the raw Seq2Seq segmentation rate is very approximate (51.14%). When we integrate the deep recognition results into the existing analysis engine, the segmentation performance improves with an IOU of 90.4% (better than instruction and phonetic strategies). This demonstrates the merits of combining explicit segmentation with the deep network recognition in the analysis process. Finally, the fusion and competition strategy (92.82%) comes a close second to the Childtyping strategy, which refers to the analysis being guided by the keyboard user input (93.67%). We can consider childtyping analysis performance as a sort of objective to reach for the system, without the aid of the user.

6.4 Recognition evaluation

Table 5 presents the recognition performance of each strategy, without the graph pruning, on the test set. We can see that in a dictation context, the instruction can't guide the analysis effectively, with a recognition rate of 64.09%. The phonetic analysis approach deals well with phonetically coherent misspellings, but fails to reach the ceiling of childtyping recognition performance (66.42%). Even if childtyping is a reliable approximation of the ground truth, the combination of degraded handwriting and in some cases, typing errors, explains the ceiling of 78.98%. The deep prediction strategy achieves better results than the phonetic strategy (72.18%). It is interesting to note that this strategy fails to achieve the recognition performance of the raw Seq2Seq; however, this is explained by the explicit segmentation aspect of the analysis engine. While the implicit segmentation is quite approximate, the explicit segmentation driven by the deep prediction is significantly better (*c.f.* Table 4). Finally, the fusion and competition strategy has better performances than the childtyping one (83.28%).

6.5 Impact of the pruning strategy

The deep learning model takes an average of 73 milliseconds per word to make a prediction. This computation is very fast; therefore, it is not included in the following time analysis. Table 6 presents the recognition and segmentation performance of the proposed strategies, as well as their average analysis time per word. In this table, we do not discuss the pruning with childtyping, instruction, or phonetic strategies, since they do not exploit the Seq2Seq results, contrary to the other two strategies. As we have seen, the fusion and competition strategy provides the best recognition and segmentation results (barring the childtyping strategy for segmentation); however, the analysis time (4.74 s per word) is more than three times bigger than the deep prediction guidance strategy. This is due to the fact that there are more segmentation paths that are processed for this strategy. Integrating the pruning enables to decrease the analysis time of the fusion strategy to an acceptable 0.67s on average, while loosing about 2% of recognition performance (80.87%), **which is still better than the childtyping strategy**. The pruning results also in loosing about 1% of segmentation precision. This is due to the approximate nature of the implicit segmentation. The same goes for pruning with the deep prediction guidance strategy. We can therefore conclude that the pruning constitutes an acceptable trade-off between analysis time and performance.

6.6 Feedback typology

This section presents the pedagogical output of our system, providing visual feedbacks on the children mistakes. Since we are in an educational context, we have to minimize the analysis system errors. Therefore, the degree of visual feedback precision and detail displayed to the child depends on the analysis confidence. When the analysis confidence is low, we generate more generic feedbacks, i.e. a warning on a zone of incertitude, or even no feedback at all. The feedback typology is illustrated in Fig. 17 and decomposed into three different levels: (1) *High confidence*: when the predicted word path corresponds to the prior knowledge strategy (e.g., the deep prediction) ⟹ precise feedback is given; (2) *Medium confidence*: when one letter distinguishes between the predicted word and the strategy ⟹ a warning is generated on an uncertain zone; (3) *Reject*: when the aforementioned conditions are not met ⟹ no feedback is given to the child. More details on feedback generation can be seen in [20].

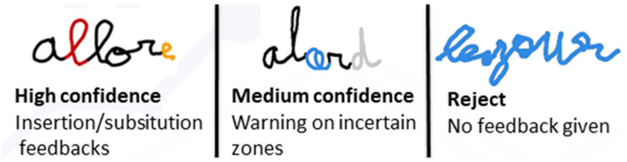


Fig. 17 Segmentation graph of written word “alors”

Table 7 Feedback generation pertinence

Confidence	Ratio (avg)	Errors rate (avg)
High	1103.9 ± 11 (88.8%)	15.2%
Medium	55.4 ± 4.4 (4.5%)	0%
Reject	82.7 ± 9.48 (6.7%)	0%
Total feedback	1146 (93.34%)	14.7%

Table 7 presents the feedback pertinence results on the fusion competition strategy with pruning. On one hand, the system has a high confidence feedback degree of 88.88% on the test set with an error rate of 15.2% on this type of feedback. On the other hand, the system has a low degree of medium and reject feedback (4.5 and 6.7% respectively). Putting high and medium confidence feedbacks altogether, we can see that the system minimizes its error rate from 21.13% (*c.f.* Table 6) to 14.7%, which is positive. However, since we are in an educational context, further work is needed to improve this feedback error ratio.

7 Conclusion

In this paper, we present an approach for the fine analysis, i.e., recognition and segmentation, of children handwritten words in a dictation context. This context introduces new challenges, since the handwriting is more degraded than adult handwriting, and the children are prone to misspelling mistakes, which makes the analysis task much harder than in a copying context. An explicit segmentation process is needed to provide precise feedback on the child’s mistakes. This explicit segmentation needs to be driven by prior knowledge. We propose to combine an existing explicit segmentation-

Table 6 Impact of pruning strategy

Strategy	Recognition rate	Segmentation rate (IoU)	Average time (s)
Deep prediction	72.18% ± 0.73%	90.4% ± 0.54%	1.34
Fusion competition	83.28% ± 0.511%	92.8% ± 0.28%	4.74
Deep prediction (pruning)	70.87% ± 2.3%	89.3% ± 1.12%	0.37
Fusion competition (pruning)	79.87% ± 0.94%	91.44% ± 0.98%	0.67

based analysis engine with a Seq2Seq architecture to generate relevant prior knowledge and adapt the system to the dictation context. Using the deep predicted character sequence as prior knowledge compensates for the fact that the dictated instruction cannot drive the analysis, as it has done for the copying context. We then propose to combine multiple strategies, the instruction, phonetically similar pseudo-words, and the deep prediction, in order to further improve analysis performances. Another contribution of this work is to use the implicit segmentation of the Seq2Seq to prune the analysis engine segmentation graph, which resulted in optimizing analysis complexity and time, while retaining good analysis performances, in fact outperforming the childtyping strategy, which constituted a "high ceiling baseline" for our task in terms of recognition performances. Our future works consist in further experimenting the system in pilot French schools. Another objective is to improve the Seq2Seq performances, in terms of recognition and segmentation, which will consequently improve the explicit segmentation based analysis engine. We could rely on synthetic data to further improve the network performances. Finally, we could explore the extension of this approach to languages other than French.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10032-022-00409-4.

Funding Partial funding was received from the P2IA project (French government) for this study.

Data availability The datasets generated during and/or analyzed during the current study are not publicly available due to privacy laws (RGDP) in France.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

References

1. Simonnet, D., Girard, N., Anquetil, E., Renault, M., Thomas, S.: Evaluation of children cursive handwritten words for e-education. Pattern Recognit. Lett. **121**, 133–139 (2019)
2. Marti, U.-V., Bunke, H.: A full english sentence database for off-line handwriting recognition. In: Fifth International Conference on Document Analysis and Recognition, ICDAR 1999, 20–22 September, 1999, Bangalore, India, pp. 705–708. IEEE Computer Society (1999)
3. Liwicki, M., Bunke, H.: Iam-ondb—an on-line English sentence database acquired from handwritten text on a whiteboard. In: Eighth International Conference on Document Analysis and Recognition (ICDAR 2005), 29 August–1 September 2005, Seoul, Korea, pp. 956–961. IEEE Computer Society (2005)
4. Tappert, C.C., Suen, C.Y., Wakahara, T.: The state of the art in online handwriting recognition. IEEE Trans. Pattern Anal. Mach. Intell. **12**(8), 787–808 (1990)
5. Plamondon, R., Srihari, S.N.: Online and off-line handwriting recognition: a comprehensive survey. IEEE Trans. Pattern Anal. Mach. Intell. **22**(1), 63–84 (2000)
6. Graves, A., Fernández, S., Gomez, F. J., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25–29, 2006, Volume 148 of ACM International Conference Proceeding Series, pp. 369–376. ACM (2006)
7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
8. Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., Schmidhuber, J.: A novel connectionist system for unconstrained handwriting recognition. IEEE Trans. Pattern Anal. Mach. Intell. **31**(5), 855–868 (2009)
9. Carbune, V., Gonnet, P., Deselaers, T., Rowley, H.A., Daryin, A.N., Calvo, M., Wang, L.-L., Keysers, D., Feuz, S., Gervais, P.: Fast multi-language LSTM-based online handwriting recognition. Int. J. Doc. Anal. Recognit. **23**(2), 89–102 (2020)
10. Shi, B., Bai, X., Yao, C.: An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Trans. Pattern Anal. Mach. Intell. **39**(11), 2298–2304 (2017)
11. Puigcerver, J.: Are multidimensional recurrent layers really necessary for handwritten text recognition? In: 14th IAPR International Conference on Document Analysis and Recognition, ICDAR 2017, Kyoto, Japan, November 9–15, 2017, pp. 67–72. IEEE (2017)
12. Michael, J., Labahn, R., Grüning, T., Zöllner, J.: Evaluating sequence-to-sequence models for handwritten text recognition. In: 2019 International Conference on Document Analysis and Recognition, ICDAR 2019, Sydney, Australia, September 20–25, 2019, 1286–1293. IEEE (2019)
13. Kang, L., Riba, P., Rusiñol, M., Fornés, A., Villegas, M.: Pay attention to what you read: non-recurrent handwritten text-line recognition. *CoRR*, abs/2005.13044 (2020)
14. Barrere, K., Soullard Y., Lemaitre A., Coüasnon, B.: Transformers for historical handwritten text recognition. In: Doctoral Consortium: ICDAR 2021, Lausanne, Switzerland, September. Nibal Nayef and Jean-Christophe Burie (2021)
15. Liu, H., Jin, S., Zhang, C.: Connectionist temporal classification with maximum entropy regularization. In: Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3–8, 2018, Montréal, Canada, pp. 839–849 (2018)
16. Gao, L., Zhang, H. Liu, C.-L.: Handwritten text recognition with convolutional prototype network and most aligned frame based CTC training. In: 16th International Conference on Document Analysis and Recognition, ICDAR 2021, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part I, Volume 12821 of Lecture Notes in Computer Science, pp. 205–220. Springer (2021)
17. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
18. Anquetil, E., Lorette, G.: On-line handwriting character recognition system based on hierarchical qualitative fuzzy modelling. In: Progress in Handwriting Recognition, pp. 109–116 (1997)
19. Simonnet, D., Anquetil, E., Bouillon, M.: Multi-criteria handwriting quality analysis with online fuzzy models. Pattern Recognit. **69**, 310–324 (2017)
20. Krichen, O., Corbillé, S., Anquetil, E., Girard, N., Nerdeux, P.: Online analysis of children handwritten words in dictation context.

In: 14th International Workshop on Graphics Recognition, Lausanne, Switzerland, September (2021)

21. Novak, J. R., Minematsu, N., Hirose, K., Hori, C., Kashioka, H., Dixon, P. R.: Improving wfst-based G2P conversion with alignment constraints and RNNLM n-best rescoring. In: INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association, Portland, Oregon, USA, September 9–13, 2012, pp. 2526–2529. ISCA (2012)
22. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings
23. Bluche, T., Messina, R.O.: Gated convolutional recurrent neural networks for multilingual handwriting recognition. In: 14th IAPR International Conference on Document Analysis and Recognition, ICDAR 2017, Kyoto, Japan, November 9–15, 2017, pp. 646–651. IEEE (2017)
24. Almaksour, A., Anquetil, É.: Improving premise structure in evolving Takagi–Sugeno neuro-fuzzy classifiers. Evol. Syst. **2**(1), 25–33 (2011)
25. Schwenk, H., Gauvain, J.-L.: Improved rover using language model information. 11 (2000)
26. Damerau, F.: A technique for computer detection and correction of spelling errors. Commun. ACM **7**(3), 171–176 (1964)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.