ON THE INTEGRATION OF LANGUAGE MODELS INTO SEQUENCE TO SEQUENCE ARCHITECTURES FOR HANDWRITTEN MUSIC RECOGNITION

Pau Torras **Arnau Baró** **Lei Kang** **Alicia Fornés**
Computer Vision Center, Computer Science Department
Universitat Autònoma de Barcelona
pau.torras@e-campus.uab.cat
{abaro, lkang, afornes}@cvc.uab.cat

ABSTRACT

Despite the latest advances in Deep Learning, the recognition of handwritten music scores is still a challenging endeavour. Even though the recent Sequence to Sequence (Seq2Seq) architectures have demonstrated its capacity to reliably recognise handwritten text, their performance is still far from satisfactory when applied to historical handwritten scores. Indeed, the ambiguous nature of handwriting, the non-standard musical notation employed by composers of the time and the decaying state of old paper make these scores remarkably difficult to read, sometimes even by trained humans. Thus, in this work we explore the incorporation of language models into a Seq2Seq-based architecture to try to improve transcriptions where the aforementioned unclear writing produces statistically unsound mistakes, which as far as we know, has never been attempted for this field of research on this architecture. After studying various Language Model integration techniques, the experimental evaluation on historical handwritten music scores shows a significant improvement over the state of the art, showing that this is a promising research direction for dealing with such difficult manuscripts.

1. INTRODUCTION

Optical Music Recognition (OMR) [1] is devoted to the automated transcription of musical documents. As in most document analysis subfields, OMR has gone through a revolution [2] during the last decade, spearheaded by the many advances in Deep Learning. In fact, the latest deep learning architectures are raising the bar of the state of the art, boosting the performance on many different topics of research. In particular, Sequence to Sequence (Seq2Seq) is a Deep Learning architecture that has been quite successful [3]. It was originally conceived for Natural Language Processing and applied to neural machine translation and related subjects, but it has seen adoption in plenty of other fields, including Handwritten Text Recognition [4]. Recently, this architecture has also shown its potential for OMR, outperforming the well-known Long Short Term Memory Neural Networks with Connectionist Temporal Classification (BLSTM+CTC). [5, 6].

As in BLSTM+CTC, Seq2Seq models have the advantage that they do not require symbol-level bounding boxes for training. Instead, the network can learn to identify symbols in an image from the ground-truth token sequence alone. This might not be especially relevant when working with typeset scores, since this information can be provided with relative ease, but it becomes crucial when no such information is available or it is very costly to obtain. This is the current situation for handwritten music recognition in general [7], but more remarkably so in historical music scores [8, 9].

Historical handwritten scores are particularly interesting to recognise because there are many of them stored in archives, churches and libraries throughout. Most of them have never been transcribed, which makes it important to devote efforts towards their conservation, transcription, study and dissemination. However, aside from the aforementioned lack of detailed-annotated data, these scores are much harder to recognise than regular typeset ones because of hundreds of years worth of paper degradation, the evolution of music notation conventions and the irregular nature of handwriting, which leads to many ambiguities and hard-to-read passages even for trained humans.

As expected, even the recent Seq2Seq architectures fail in such scenario. Nevertheless, in the handwritten text recognition literature, we have found that the incorporation of a Language Model (LM) can tackle most of these ambiguities. This technique consists on the application of a statistical LM trained to identify probable sequences of tokens, which can then be used to assess the likelihood of the recognised sequence and perform due corrections in the case of an unreasonably unexpected output [10, 11]. As in n-grams, it regulates what sequences are considered most likely.

Inspired by this idea, in this work we explore the integration of LMs into a Seq2Seq architecture to minimise the ambiguities when recognising historical handwritten scores. Concretely, we integrate a LM through three different techniques: Shallow, Deep [10] and Candidate Fusion [11]. From the exhaustive evaluation of their performance on historical manuscripts, we discuss the advantages and disadvantages of these models, concluding that they are capable to significantly boost the performance in the aforementioned domain.

The structure for this document is the following. An overview of current trends in music recognition is provided in Section 2. Section 3 is devoted to describing the architecture. Section 4 describes the adaptation of the input data for music score recognition, and the datasets employed to train the LMs. Section 5 summarises experiments performed to evaluate the performance of the various models. Section 6 is a discussion of the results and section 7 addresses the conclusions and closing words.

2. PREVIOUS WORK

Prior to the Deep Learning "revolution" of the last decades there was mainly one typical pipeline for OMR, which consisted on a set of well-established steps [1]: image preprocessing and staff segmentation, music symbol recognition, music notation reconstruction and final representation construction. This, however, changed with the advances made in Deep Learning, which led to two distinct kind of approaches.

On the one hand, there has been a "continuist" approach, where the aforementioned pipeline is more or less preserved, but one or more steps are implemented with Deep Neural Models. Examples of these systems can be found in plenty of works. For example, Calvo-Zaragoza *et al.* proposed a new method for staff line detection [12] through the use of Convolutional Neural Networks; Calvo-Zaragoza also presented work regarding pixel-level document binarization with Convolutional Neural Networks alongside Fujinaga and Vigliensoni [13]; Hajic *et al.* [14] proposed a way to segment musical symbols and classify them in a single step using U-Nets; Pacha *et al.* [15] proposed a method to reconstruct the relationships between segmented symbols through the use of Convolutional Neural Networks together with a novel graph-based system to represent them.

On the other hand, there have been attempts to perform the full OMR pipeline using a single neural-based end-to-end architecture. The work of van der Wel *et al.* [5] is interesting because it is the first precedent of Seq2Seq for OMR, although it was exclusively designed for typeset scores. Newer models such as Huang *et al.* [16] YOLO darknet53-based architecture seem to have dropped the recurrent aspect while improving on the state of the art in this context of typeset scores.

In terms of handwritten scores, RNNs are still being used with good results. Baró *et al.* [17] used a CRNN model on handwritten scores, which was the first single-step baseline that was established for this domain. For handwritten old scores, Calvo-Zaragoza [9] proposed a CRNN + CTC model with an n-gram LM for recognising a specific set of scores in Mensural notation. Lately, Baró et al. [6] presented a single-step system based on a Seq2Seq model with an attention mechanism for recognising handwritten scores in common western notation.

The earliest instance of Language Modelling subject to a recognition task is [18], which used n-grams in order to make OCR machines context-aware and therefore more robust. Since n-grams are fairly easy to implement and give reasonably good results, they have been used quite consistently even in recent times [9], although with the rise of DNN technology other approaches based on neural LMs have emerged. Indeed, the integration of LMs into Seq2Seq architectures has also been studied through various methods that take advantage of RNN-based LMs. These were introduced for fields within or related to Natural Language Processing like neural machine translation [10], handwritten text recognition [11], or speech recognition [19], although the core idea is equally valid whenever the final target is any ordered sequence of tokens.

In summary, Seq2Seq-based recognisers are promising architectures that have shown to benefit from the integration of LMs. However, while LMs have been applied to music recognition through n-grams [9], no precedents of RNN-based LMs along with Seq2Seq OMR architectures exist. Therefore, we hypothesise that such integration has the potential to improve the current state-of-the-art results in OMR, as it has already been observed in other related fields [10, 11].

3. SEQUENCE TO SEQUENCE-BASED OMR

This section describes the core Seq2Seq system for OMR, the three LM models and their integration into the architecture.

As stated before, our architecture is inspired in the Seq2Seq OMR model described in [5, 6]. The whole architecture is depicted in Figure 1, with a reference to the LM integration step (see the dashed lines). Next, we describe its properties and its inference process.

3.1 Sequence to Sequence model

Seq2Seq models [3] are architectures capable of converting arbitrary-length input sequences into arbitrary-length output sequences. They are an Encoder-Decoder architecture: the input sequence is transformed by the Encoder into an intermediate representation that the Decoder will use to generate the output sequence.

A score image, which is treated as a sequence of column vectors, is fed into a Convolutional Neural Network based on a VGG19 [20] with its last max pooling layer removed. Then, the Encoder, a bidirectional stack of Gated Recurrent Units (GRU) [21], generates an intermediate representation comprised of as many feature vectors as the convolutional output. The idea behind this bidirectionality is that, by processing the input image from both ends of the sequence, the model has the information of the full image for all inference steps and is therefore much more context-aware.

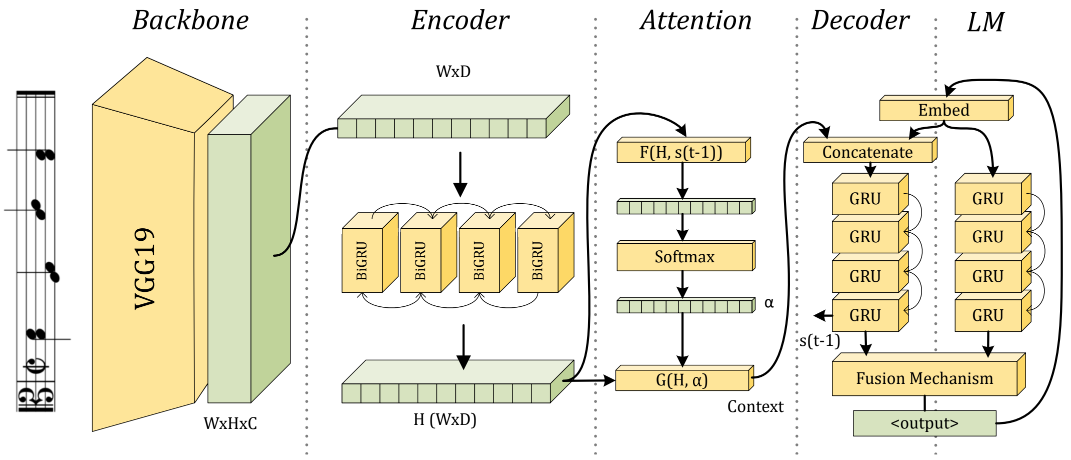

Figure 1. Summary of the Seq2Seq model used in this work.

When the Encoder has processed the input image completely, the Decoder iteratively receives the generated hidden state alongside the last predicted token in the sequence, which produces the next output token until a special "end" token is produced. In order to assess the relevance of each of the hidden state's vectors, a location attention mechanism (Chorowsky *et al.* [22]) weights each vector in the hidden state, with the idea of making the model capable of "focusing" on specific regions of the input image.

3.2 Language Model Integration

LMs are systems that model the probability distribution of possible tokens at time step t conditioned by predictions at time steps 1 to $t-1$. Many language modelling techniques exist throughout such as n-grams [9], but RNNs are known to be a superior choice overall [23], thus this work focuses on a single LM architecture consisting on four stacked GRUs.

LM integration with Seq2Seq models has been explored through various approaches aiming at improving recognition performance. Three of such approaches have been explored in this work: Shallow, Deep [10], which are among the most used methods, and Candidate Fusion [11], which showed good performance on handwritten text recognition.

Figure 2 shows a depiction of these methods and the following paragraphs are devoted to describing them in detail.

3.2.1 Shallow Fusion (Gulcehre et al. *[10])*

This technique was devised in the context of neural machine translation. It is a very intuitive system in which the final output is obtained by summing log probabilities from the LM and the Seq2Seq model. Let P, P_{CL} and P_{LM} be the probability distribution of tokens predicted by the full model, the Seq2Seq component and the LM respectively, and let λ be an arbitrary hyperparameter set on training, Shallow Fusion is implemented as

$$\log P\left(y_t|y_1 \dots y_{t-1}\right) = \log P_{CL}\left(y_t|y_1 \dots y_{t-1}\right) \quad (1)$$
$$+ \lambda \log P_{LM}\left(y_t|y_1 \dots y_{t-1}\right).$$

3.2.2 Deep Fusion (Gulcehre et al. *[10])*

This method comes from the same context as Shallow Fusion and builds further on its idea by merging both LM and Seq2Seq's outputs in a more fine-grained manner. Essentially, the λ parameter is substituted by a coarse gating mechanism and the final output is obtained using more information from across the model. Let σ be the sigmoid activation function and W_{DF} and b_{DF} be learnable parameters, Deep Fusion is implemented as

$$P(y_t|y_1 \dots y_{t-1}) = \mathrm{softmax}(W_{DF}h_t^{DF} + b_{DF}). \quad (2)$$

The Deep Fusion hidden state h_t^{DF} is obtained concatenating the Seq2Seq context vector c_t, the Classifier's hidden state h_t^{CL} and a gated version of the LM's hidden state, as seen in

$$h_t^{DF} = \left[c_t; h_t^{CL}; g_t h_t^{LM}\right]. \quad (3)$$

The coarse gate mechanism g_t is in its turn computed as

$$g_t = \sigma(v_g^T h_t^{LM} + b_g) \quad (4)$$

where v_g and b_g are learnable parameter vectors. We use the implementation seen in [24], which does not feed the previously inferred character in equation 3.

3.2.3 Candidate Fusion (Kang et al. *[11])*

This method was shown to be more suitable than Deep and Shallow fusion in the context of Handwritten Text Recognition. The core idea behind it is to reinforce the decision process of the Seq2Seq Decoder at each output time step by feeding it the output of the LM, so that both pipelines can be leveraged accordingly. It can be defined as

$$h_t = \mathrm{Decoder}(\left[c_t, y_{t-1}, p_{t-1}^{lm}\right], h_{t-1}) \quad (5)$$

where c_t is the current context vector, y_{t-1} is the previous prediction and p_{t-1}^{lm} is the probability distribution obtained by the LM with the output at the previous time step.

Some comparisons can be drawn among all three methods both from their literature and their architectures. The main selling point for Shallow Fusion is that it adds very little complexity into the model, which is compensated by

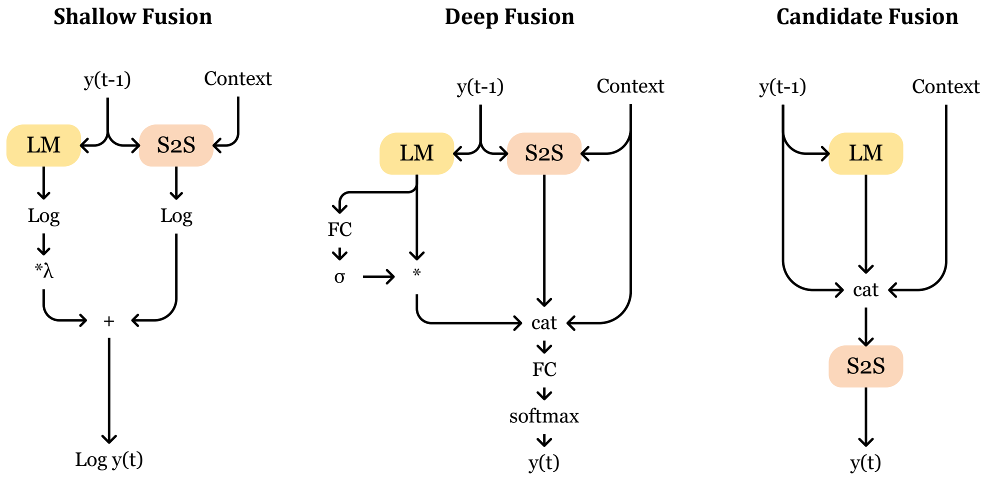

Figure 2. Dataflow graph depicting every integration method that was implemented

the fact that it requires hyperparameter tuning for its λ value and the impossibility to modify said value depending on the LM output. Deep Fusion poses as a more flexible model that can learn to weight the importance of the output of the LM, at the cost of incorporating further layers into the model. Finally, Candidate fusion boosts the communication between the LM and the Seq2Seq component and produces an output obtained not by linearly combining both outputs at the final inference step, but rather by letting the Seq2Seq combine the criteria of visual features and Language Probabilities. However, this might involve more training for the model to become acquainted with the output of the LM.

All these methods require both the classifier and the LM to be properly pretrained for successful integration. More detail is provided in section 5.

4. DATASETS

This section describes the adaptation of the data for music score recognition using the Seq2Seq architecture.

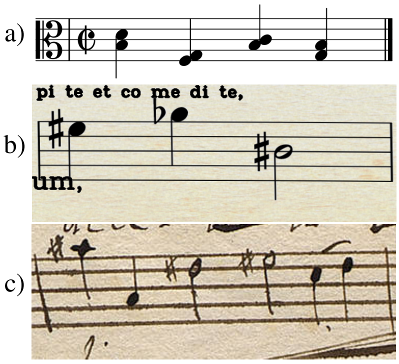

Figure 3. Sample measures from the SM, SO and HW datasets respectively.

4.1 Serialising Input Data

Input music measures are annotated at a *musical primitive* level. This means that notes are not full tokens by themselves, but are instead divided into their core elements: noteheads with their pitch and type (black or white), stems with their orientation, flags, beams and so on. There are also some tokens which are atomic, such as time signatures, dots, accidentals and rests, and some twin tokens that require opening and closing, such as beginning and end segments of a slur or a beam.

The `epsilon` token is a special one used to separate groups of primitives belonging to different symbols placed in adjacent columns. Thanks to this, 2D music notation can be serialised into a flat one-dimensional array of tokens that Seq2Seq can work with. An example of this format is given in Figure 4.

4.2 Training Datasets

Various datasets of differing characteristics were used to train the models, each of them for a specific task (more detail on section 5). Their description is shown below along with some examples (See Figure 3). Note also that, when referring to synthetic datasets, we imply the musical content of these scores is randomly generated (thus we assume that these datasets are, except for some trivial examples, disjoint).

Synthetic Modern (SM): Dataset comprised of polyphonic measures of synthetic typeset scores. Most usual music symbols can be found: G, C and F clefs, accidentals, note components, time signatures and barlines, to name a few. An example is shown in Figure 3a.

Synthetic Old (SO): A synthetic dataset with monophonic measures distorted with typical paper degradation effects. Similar to SM in terms of the range of tokens present. An example is shown in Figure 3b.

Handwritten (HW): A compilation of measures of real handwritten scores from a church in Barcelona called Santa María del Pi. They were composed by its Kapellmeister Pau Llinàs back in the 18th century for choral interpretation during liturgical events. An example is shown in Figure 3c.

Adjusted Synthetic Modern (ASM): A reduced version of the SM dataset (see section 5).

```
barline_light.noNote, epsilon, sharp.S4, epsilon,
noteheadBlack.S4, steamQuarterHalfDown.noNote, epsilon,
dot.noNote, epsilon, noteheadBlack.S4, flag8thDown.noNote,
epsilon, noteheadBlack.S3, steamQuarterHalfDown.noNote
epsilon, noteheadBlack.S3, steamQuarterHalfDown.noNote,
epsilon, barline_light.noNote
```

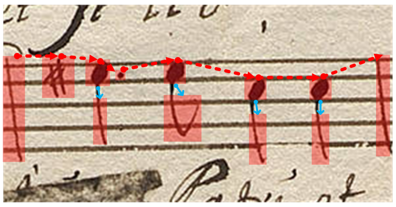

Figure 4. Sample measure from the HW dataset with its ground truth annotation. Bounding boxes indicate the boundaries of what each "atomic" token is, dotted arrows indicate epsilons in the transcription and small vertical arrows indicate symbols that are placed together between epsilons (or rather, primitives belonging to the same symbol).

5. EXPERIMENTS

The evaluation of all proposed LM integration methods was performed under two training strategies, characterised by the dataset which was used to pretrain the LM. Regardless of the LM dataset, training parameters and strategies were the same altogether. For the sake of reproducibility, Table 1 summarises these hyperparameters and characterises the various datasets employed throughout.

Since the goal for our work is to improve results on handwritten scores, a training strategy was conceived to gain the benefits of extra data from synthetic scores while preventing optimisation towards them. All integration methods tested hereby require both the LM and the recogniser to be properly pretrained. Thus, we first trained a LM with an unmodified version of the SM dataset. Since we were aware that this dataset had many tokens that were not present in the HW one, we created a version of the SM dataset comprised of the 66% of samples which contained a higher ratio of tokens also present in the HW one, which we will refer to as ASM, and we trained another LM with it. The idea was trying to "de-noise" the output of the LM in HW scores so that its predictions had a higher level of confidence.

In both cases, we trained the Seq2Seq classifier with the unmodified SM dataset until the model did not improve for 30 epochs. We then joined both models and trained them using a Curriculum Learning strategy: initially, 90% of samples in the training mix were from the SO dataset and the remaining 10% from the HW dataset. Every 10 epochs the proportion of SO scores decreased by 10% over the total, down to 10%. Since the number of SO samples is much higher than the number of HW samples, the latter were duplicated randomly to match the number of samples from the former. The incorporated image augmentation system for training was used to prevent overfitting on input images. Note also that experiments with homogeneous datasets were avoided since they were seen to decrease performance in earlier tests.

Validation and test were performed using HW dataset samples. Lastly, for Shallow Fusion we used a $\lambda = 0.1$ after testing three instances of the full architecture on the SM dataset and keeping the value that gave better output results.

6. EXPERIMENTAL RESULTS

This section is devoted to explaining the results obtained with the aforementioned training strategies. This is, Shallow, Deep and Candidate Fusion using a LM pretrained with the SM or the ASM Dataset. Numerical results are provided using the Symbol Error Rate (SER(%)) metric, which is defined as

$$SER(\%) = \frac{I + R + S}{T} \cdot 100 \qquad (6)$$

where I, R and S are the number of token insertions, removals and substitutions in order to obtain the ground truth sequence from the predicted sequence and T is the length of the ground truth sequence. Lower values mean better results.

6.1 Quantitative Results

Table 2 shows the results obtained from all of our experiments. Given the fact that Seq2Seq model pre-training on the SM gave results well below 1% SER(%), we believe it is not worth to experiment with the addition of a LM when transcribing synthetic samples. Instead, we show test results using the training strategy in 5 and two baseline models: the BLSTM + CTC model and the LM-less Seq2Seq model [6]. All results are obtained using the HW test partition as input.

Best baseline results are 56.20% and 31.79% of SER(%) for BLSTM + CTC and Seq2Seq respectively. However, authors comment in the paper that there might be overfitting in the best result of the former model because training was done only with handwritten samples. When training with a mix of synthetic and real data, the authors state an increase from 56.20 SER(%) to 74.40 SER(%).

Our proposed models obtained mostly better results than those from the Baseline. Candidate and Deep Fusion are the better performing architectures, with best results (in bold in Table 2) between 5 and 6 SER(%) points below the baseline. Shallow Fusion obtained best results on par with the baseline.

The general pattern is that earlier iterations perform worse than latter ones. There are a few exceptions, which are the SM version of Deep Fusion and the ASM version of Shallow Fusion, which obtain better results in intermediate phases. This might be caused by the fact that the model might be entering local minima, which it may leave after further epochs.

Table 1. Reproducibility table. The first segment is devoted to training hyperparameters. The second one to showing relevant information about the various datasets that have been employed.

Parameters	**All Training**	**Data**	**SM**	**SO**	**HW**
Optimiser	Adam	**Train Samples**	18,900	17,872	147
Learning Rate (LR)	$3 \cdot 10^{-4}$	**Valid Samples**	6,300	5,957	49
LR Checkpoints	@ 20, 40, 60, 80, 100 epoch	**Test Samples**	6,300	5,957	49
LR Sigma	0.5	**Avg. Line Length**	22	15	17
Loss Function	Cross-Entropy	**Classes**	109	123	62

Table 2. Summary of performed experiments and results in SER(%) (Lower is better). The table header indicates the proportion of Synthetic scores against Handwritten scores. The "Pre" column indicates the LM pretraining dataset.

Model	**Pre**	**90-10**	**80-20**	**70-30**	**60-40**	**50-50**	**40-60**	**30-70**	**20-80**	**10-90**	**0-100**
CNN + BLSTM [6]	-	-	-	-	-	-	-	-	-	-	**56.20**
Seq2Seq Baseline [6]	-	60.03	-	-	66.20	-	43.38	-	37.86	34.56	**31.79**
Seq2Seq + Deep LM	SM	31.30	28.52	29.87	29.37	28.05	**26.11**	27.74	27.37	28.32	-
Seq2Seq + Shallow LM	SM	36.79	32.91	33.27	33.36	31.76	32.75	30.87	30.72	**30.58**	-
Seq2Seq + Cand. LM	SM	33.50	28.93	28.64	28.08	27.48	26.82	27.23	26.61	**25.80**	-
Seq2Seq + Deep LM	ASM	28.24	29.53	27.82	27.36	25.95	27.21	25.63	**25.15**	25.54	-
Seq2Seq + Shallow LM	ASM	35.34	34.75	36.67	**32.42**	34.23	34.52	33.76	33.79	35.13	-
Seq2Seq + Cand. LM	ASM	32.07	28.61	28.71	27.55	27.71	27.20	27.77	28.04	**25.73**	-

Another general remark is that models pretrained with the ASM dataset seem to perform slightly better, with a 0.96 SER(%) improvement in Deep Fusion and a 0.07 one in Candidate Fusion, although this difference could be also attributed to optimisation since it is not substantial.

6.2 Discussion

Numerical proof is found that a LM does help improve recognition results in historical handwritten music scores, especially when using Candidate or Deep Fusion. However, we agree that it is not easy to assess their differences outside of a subjective qualitative study.

Expectedly, LM lowers the presence of certain syntactic mistakes (for instance, tokens that require a specific successor) or provides information on tokens that appear frequently. There is, however, a set of possible recognition mistakes that the LM was initially presumed to be able to correct which we found it unable to. The most relevant was enforcing the beat of the bar that is being recognised. It can be argued that at no point in the measures that comprise the dataset the time signature is indicated aside from its very beginning, but since the training dataset is written exclusively in a 4/4 time signature, the LM might have adapted to measures adding up to a beat value. Perhaps this is due to the purely statistical approach taken with the LM, so some postprocessing (based on music notation rules) may be needed for approaching such consistency checks.

Other "artistic" aspects of music cannot be corrected with the LM, such as the pitch and duration of notes, which can only be predicted up to a certain point based on its frequency of appearance. This was expected and, unsurprisingly, most noteheads have been predicted on the most common range within the original score.

A final remark is that we have observed that the adjustment strategy attempted with the ASM dataset showed no significant improvement. Instead, in order to better align training and test datasets without overfitting, more data should be used for training. A common issue when trying to collect data for this purpose is that most common transcriptions of old music adapt their notation style to current trends, which defeats the purpose of using such data for recognition.

7. CONCLUSION

This work successfully explored the integration of LMs into a Seq2Seq OMR architecture for recognising historical handwritten scores. An improvement of around 6 SER(%) points from the baseline was obtained when using a Deep Fusion mechanism, lowering it to 25.15 SER(%). This was achieved by reinforcing the model's capacity to keep consistency on predicted sequences. Thus, we can conclude that the integration of language models into OMR Seq2Seq architectures is a promising research direction worth exploring.

From the results we obtained, we propose some future work avenues. Since language models do not seem to enforce key global aspects like beat, a grammar-based parser might be implemented on top of the neural model in order to correct syntactical mistakes. This could use the probability distribution produced by the neural model to weight all possible corrections. Another improvement could be to use the extra information the LM provides in order to reinforce specific steps within the model, such as the attention mechanism. Perhaps this preemptive information might point the model where to look at in the score image.

8. ACKNOWLEDGEMENTS

This work has been partially supported by the Spanish project RTI2018-095645-B-C21, the CERCA Program/ Generalitat de Catalunya, and the FI fellowship AGAUR 2020 FI_B2 00149 (with the support of the Secretaria d'Universitats i Recerca of the Generalitat de Catalunya and the Fons Social Europeu). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

9. REFERENCES

[1] A. Rebelo, I. Fujinaga, F. Paszkiewicz, A. R. Marcal, C. Guedes, and J. S. Cardoso, "Optical music recognition: state-of-the-art and open issues," *IJMIR*, vol. 1, no. 3, pp. 173–190, 2012.

[2] J. Calvo-Zaragoza, J. H. Jr, and A. Pacha, "Understanding optical music recognition," *CSUR*, vol. 53, no. 4, pp. 1–35, 2020.

[3] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *NeurIPS*, 2014, pp. 3104–3112.

[4] J. Michael, R. Labahn, T. Grüning, and J. Zöllner, "Evaluating sequence-to-sequence models for handwritten text recognition," in *ICDAR*. IEEE, 2019, pp. 1286–1293.

[5] E. van der Wel and K. Ullrich, "Optical music recognition with convolutional sequence-to-sequence models," in *ISMIR*, 2017, pp. 731–737.

[6] A. Baró, C. Badal, and A. Fornés, "Handwritten historical music recognition by sequence-to-sequence with attention mechanism," in *ICFHR*, 2020, pp. 205–210.

[7] J. Hajič and P. Pecina, "The MUSCIMA++ dataset for handwritten optical music recognition," in *ICDAR*, vol. 1, 2017, pp. 39–46.

[8] A. Pacha and J. Calvo-Zaragoza, "Optical music recognition in mensural notation with region-based convolutional neural networks," in *ISMIR*, 2018, pp. 23–27.

[9] J. Calvo-Zaragoza, A. H. Toselli, and E. Vidal, "Handwritten music recognition for mensural notation with convolutional recurrent neural networks," *PRL*, vol. 128, pp. 115–121, 2019.

[10] C. Gulcehre, O. Firat, K. Xu, K. Cho, L. Barrault, H.-C. Lin, F. Bougares, H. Schwenk, and Y. Bengio, "On using monolingual corpora in neural machine translation," *arXiv preprint arXiv:1503.03535*, 2015.

[11] L. Kang, P. Riba, M. Villegas, A. Fornés, and M. Rusiñol, "Candidate fusion: Integrating language modelling into a sequence-to-sequence handwritten word recognition architecture," *PR*, vol. 112, p. 107790, 2021.

[12] J. Calvo-Zaragoza, A. Pertusa, and J. Oncina, "Staff-line detection and removal using a convolutional neural network," *MVA*, vol. 28, no. 5-6, pp. 665–674, 2017.

[13] J. Calvo-Zaragoza, G. Vigliensoni, and I. Fujinaga, "Pixel-wise binarization of musical documents with convolutional neural networks," in *MVA*, 2017, pp. 362–365.

[14] J. Hajic Jr, M. Dorfer, G. Widmer, and P. Pecina, "Towards full-pipeline handwritten omr with musical symbol detection by u-nets." in *ISMIR*, 2018, pp. 225–232.

[15] A. Pacha, J. Calvo-Zaragoza, and J. Hajic Jr, "Learning notation graph construction for full-pipeline optical music recognition." in *ISMIR*, 2019, pp. 75–82.

[16] Z. Huang, X. Jia, and Y. Guo, "State-of-the-art model for music object recognition with deep learning," *Appl. Sci.*, vol. 9, no. 13, p. 2645, 2019.

[17] A. Baró, P. Riba, J. Calvo-Zaragoza, and A. Fornés, "From optical music recognition to handwritten music recognition: A baseline," *PRL*, vol. 123, pp. 1–8, 2019.

[18] C. Y. Suen, "n-gram statistics for natural language understanding and text processing," *PAMI*, vol. 1, no. 2, pp. 164–172, 1979.

[19] T. Hori, J. Cho, and S. Watanabe, "End-to-end speech recognition with word-based rnn language models," in *SLT*, 2018, pp. 389–396.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[21] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *EMNLP*, 2014, pp. 1724–1734.

[22] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *NeurIPS*, vol. 28, 2015, pp. 577–585.

[23] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Eleventh annual conference of the international speech communication association*, 2010.

[24] S. Toshniwal, A. Kannan, C.-C. Chiu, Y. Wu, T. N. Sainath, and K. Livescu, "A comparison of techniques for language model integration in encoder-decoder speech recognition," in *SLT*, 2018, pp. 369–375.