Technical Report: Experimenting on improving the architecture of PKSpell

Paul Knoll
JKU Linz
[344.076] Practical Work in AI

1 Introduction

PKSpell is a deep learning system for joint pitch spelling and key signature estimation given a sequence of midi notes [1]. Its model architecture, which is based on two RNNs, is described in more detail in Chapter 2. This technical report describes a set of experiments conducted on alternating the architecture of PKSpell to improve prediction results. One of the focus points in finding a better architecture was to leverage the principal of hierarchical attention [3] to find representations of measures, thereby enabling the network to work with the data on a higher level of abstraction. Furthermore, in designing the new architectures it was taken into account that the model should always predict the same key signatures for all notes of the same measure, as different ones are illogical and incorrect, which was not considered in the original approach.

2 PKSpell architecture

To understand the alternations made in Chapter 3, we first take a quick look at the original PKSpell architecture which is depicted in Figure 1. Foremost, the input of the model, which is the pitch-class of each note and the corresponding duration, are fed into a BiGRU. The embeddings produced by this first BiGRU are then used twofold, according to the principle of multi-task learning. One the one hand, with the help of one final linear layer they are directly used for tonal-pitch-class estimation. On the other hand, they are fed into a second BiGRU, whose embeddings are then used for key signature estimation after passing through one final linear layer. [1]

3 Experiments

In this chapter the different attempts at improving the architecture of PKSpell are described. The resulting pitch spelling and key signature validation accuracies can bee seen in Table 1. The corresponding code can be found in [2]. Because it is a crucial part of many of these attempts, first a structure that is hereafter called "hierarchical RNN" is explained. As mentioned before, the central idea in finding a better architecture was to incorporate benefits that result from hierarchical attention into the model. The obvious choice for splitting up the data to generate high level representations was to use the bars that split up the notes into measures, analogously to how Yang et al. [3] used the periods that split up a document into sentences. To this end, another input called "eom" (end of measure) is given into the model, which is 1 if the given note is the last one of its measure and 0 if not. For the hierarchical RNN the input notes are split up according to the eom information. This way, when creating the new representations, the RNN is only considering information from the notes of the given measure. Unfortunately, in the given time it was not possible to properly parallelize this process, so only one piece at a time can be processed. Therefore a loop has been implemented that

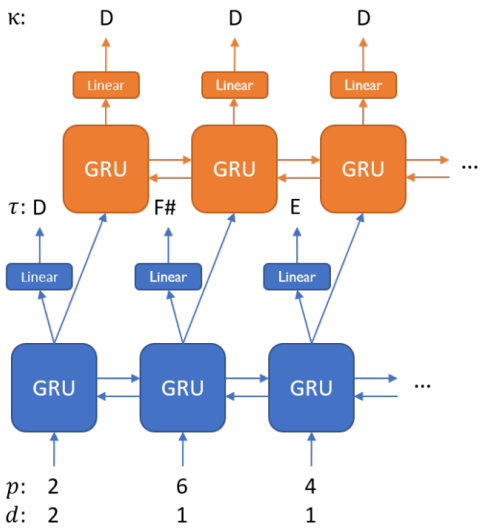

Figure 1: The original model architecture of PKSpell. For each pitch-class p and duration d in input, a tonal-pitch-class τ and a key signature κ are produced. [1]

	Pitch Spelling	Key Signature
Original PKSpell paper [1]	96.50%	90.30%
Approach 1	95.44%	86.47%
Approach 2	95.04%	49.34%
Approach 3	95.60%	86.38%
Approach 4	93.15%	66.31%
Approach 5	96.77%	87.26%
Approach 6	95.89%	88.75%
Approach 7	96.31%	88.68%

Table 1: Validation accuracies for the different attempts and the original model.

iterates through the batch, which is also why those experiments take longer to train than the original architecture. If not mentioned otherwise, the same parameters have been used for the experiments like in the original PKSpell approach. For the hierarchical RNN a hidden dimensionality of 256 has been chosen.

3.1 Hierarchical Attention

As mentioned, the first idea was to implement hierarchical attention into the model architecture. To achieve this, an attention layer was put on top of the previously described hierarchical RNN. As attention layers Dot-Product Attention, Additive Attention, and Multi-head Attention have been considered. There was no mayor difference noticeable when exchanging the attention layer. In the end a context vector that contains the weighted sum of the values according to the attention weights was obtained. This way the dimensionality has been reduced by the lengths of the measures, and only one context vector for each measure is left. These context vectors then have been repeated according to the lengths of its original measures, so that it can be used together with the embeddings of the first RNN for the final prediction. For the pitch prediction, the hidden dimension of the first RNN and the hierarchical RNN have been stacked on top of each other. This way, the final linear layer for pitch prediction can leverage the higher level measure information, while also getting the note wise information from the first RNN as it needs to predict an individual pitch for each note. For the

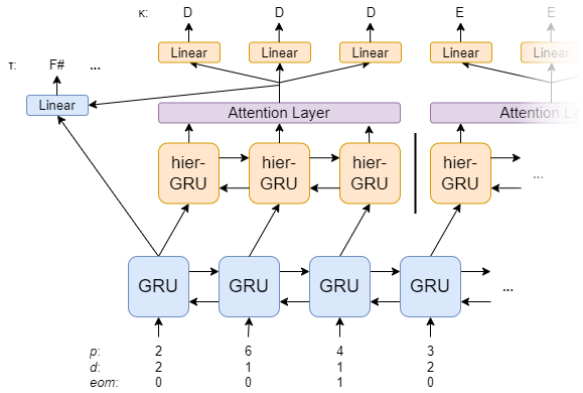

Figure 2: The model architecture of approach 1. For each pitch-class p, duration d and measure information eom in input, a tonal-pitch-class τ and a key signature κ are produced.

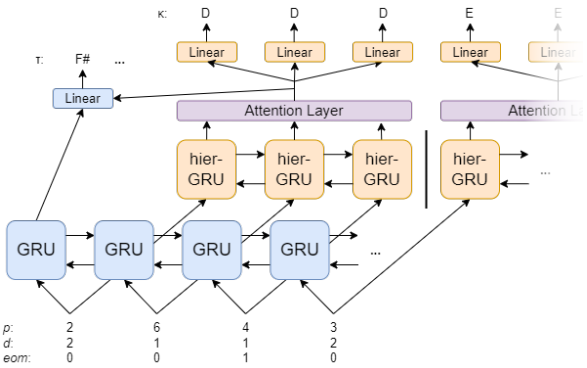

Figure 3: The model architecture of approach 2. For each pitch-class p, duration d and measure information eom in input, a tonal-pitch-class τ and a key signature κ are produced.

key signature estimation, only the repeated context vectors from the hierarchical RNN have been given into the final linear layer. In theory, they should contain all the information necessary for the prediction and this way, the same key signature gets estimated for all notes of the same measure. Because of this structure, the final linear layer for key signature prediction has to compute a value for every note separately, even if in theory only one per measure is necessary as the inputs are the same. This structure has been chosen as an additional linear layer inside the inefficient loop would increase computation time.

Two variants of the described procedure have been tested, where the input of the hierarchical RNN has been exchanged. In one test, the hierarchical RNN was fed with the "raw" notes, so it got the same input as the first RNN of the original approach. This architecture can be seen in Figure 3. In another test, it got the embeddings of the first RNN as input, which is depicted in Figure 2. When having the raw notes as input for the hierarchical RNN, the results were considerably worse with regard to key spelling prediction compared to the other approaches. While the hierarchical RNN was able to extract some information from the raw notes only, this architecture only reached a key signature validation accuracy of 49.34%, as can be seen in Table 1. When getting the embeddings of the first RNN as input for the hierarchical RNN, the results were much better but still didn't reach the performance of the original PKSpell approach in terms of pitch and key signature validation accuracy. This difference when exchanging the input of the hierarchical RNN could be a sign that for accurate key signature estimation, information outside of the concerned measure is necessary, which can not be obtained when feeding the hierarchical RNN with the raw notes.

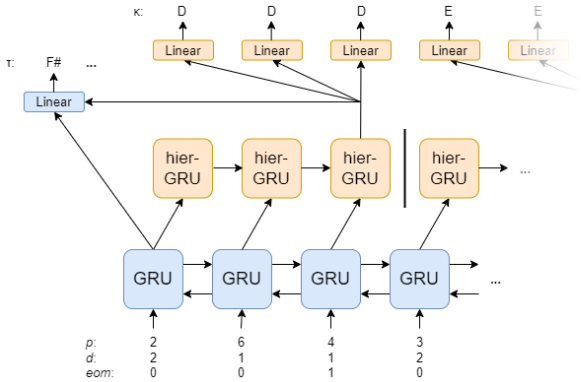

Figure 4: The model architecture of approach 3. For each pitch-class p, duration d and measure information eom in input, a tonal-pitch-class τ and a key signature κ are produced.

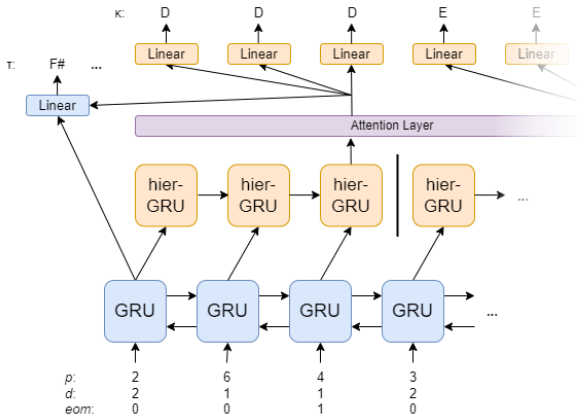

Figure 5: The model architecture of approach 4. For each pitch-class p, duration d and measure information eom in input, a tonal-pitch-class τ and a key signature κ are produced.

3.2 Other approaches

As the hierarchical attention approach didn't deliver the desired results, some different ideas were tested. In those tests the input for the hierarchical RNN is always the output of the first RNN. To have a different way of getting a representation of every measure an idea was to just take the last element of the hierarchical RNN. For this a unidirectional GRU was used, so that all the information of the measure is summed up in the last element. Then as in the hierarchical attention approach, this element gets repeated according to the length of the measure, the rest of the procedure is also identical. The architecture of this attempt is depicted in Figure 4. Unfortunately this approach did also not surpass the performance of PKSpell.

Another idea was to use the last elements of a unidirectional GRU as representations for the measures as described before, but implementing an attention layer before repeating the elements again. For this, Multi-Head attention with 4 heads was used, to compute attention between different measures. The rest of the architecture is the same as described above. A visualization of the architecture can be seen in Figure 5. This approach did slightly worse compared to the others in terms of pitch spelling prediction, reaching an accuracy of only 93.15% and thereby beeing the worst in this series of tests. The key signature prediction is noticeably worse than with the other architectures, yielding the second worst validation accuracy of 66.31%.

Another tested idea was to discard the hierarchical RNN and only use the first RNN of the original paper. This is depicted in Figure 6. Thereby the RNN should learn to encode the measure information

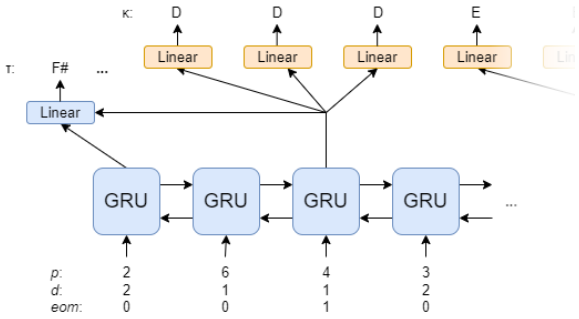

Figure 6: The model architecture of approach 5. For each pitch-class p, duration d and measure information eom in input, a tonal-pitch-class τ and a key signature κ are produced.

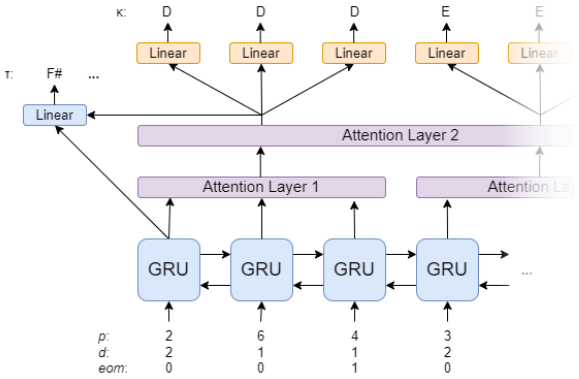

Figure 7: The model architecture of approach 6. For each pitch-class p, duration d and measure information eom in input, a tonal-pitch-class τ and a key signature κ are produced.

to the last element of each measure, while processing the whole piece at once. These last elements where then extracted with the eom information and repeated and processed as in the other approaches. This architecture did also not surpass the performance of the original one on the validation set. This approach was to one that yielded the best results in terms of pitch validation accuracy with 96.77% (compared to 96.50% with the original PKSpell architecture). Thereby this model is showing a slight improvement in pitch spelling prediction. Because this improvement is only very slight and also the architecture for the pitch spelling prediction is basically the same because it consists of only one RNN, this improvement is considered as random.

The last tested approach was to use the embeddings of first RNN, and only apply attention after that to obtain the measure representations. For this, the produces embeddings got split up like for the hierarchical RNN, but then not processed by a RNN, but fed into an attention layer. This first attention layer is summing up the context vector at the end, to reduce the dimensionality by the lengths of the measures and obtain one representation for every measure. Then in a second attention layer, attention between the different measure representations is applied, without summing up the weighted values at the end as to obtain the dimensionality and have one representation per measure. The rest of the processing pipeline is then identical to the approaches described before. This architecture, which is depicted in Figure 7, produced good results but could still not quite reach the performance of the original architecture. Without the second attention layer (Figure 8), the pitch spelling validation accuracy decreased slightly and the key signature accuracy improved slightly, while still staying below the performance of the original approach.

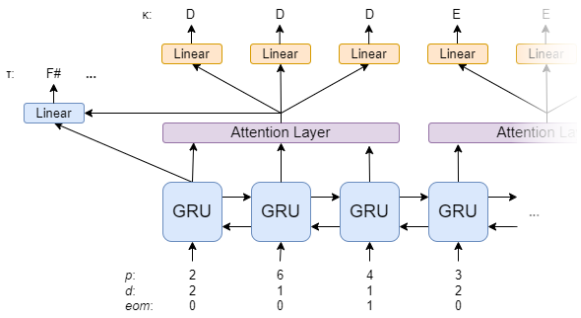

Figure 8: The model architecture of approach 7. For each pitch-class p, duration d and measure information eom in input, a tonal-pitch-class τ and a key signature κ are produced.

4 Conclusions

In all the tested approaches there was no significant improvement visible, compared to the original architecture. One reason that could cause this is that the information outside of the measure could actually be important for the key signature of the measure. As the approaches focus on summarizing the information inside of one measure, this would explain why they are not as effective as expected. Another reason for the bad performances could be the lack of optimization of hyperparameters. Because of the inefficient loop in the hierarchical RNN, the resulting long training times did not allow for many training runs with different parameters. So it could be that some of the approaches actually improve the original performance with the right choice of parameters. The parallelization of this process was not in focus here and is left for future work.

References

[1] Francesco Foscarin, Nicolas Audebert, and Raphaël Fournier S'niehotta. PKSpell: Data-driven pitch spelling and key signature estimation. In *International Society for Music Information Retrieval Conference (ISMIR)*, 2021.

[2] Paul Knoll. Pkspell hierarchical. `https://github.com/paulbzm/pkspell_hierarchical`, 2022. GitHub repository.

[3] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1174. URL `https://aclanthology.org/N16-1174`.