Transformer-Based Chord Recognition with Unsupervised Pre-training of Input Embeddings

Maral Ebrahimzadeh[1], Valerie Krug[1], Sebastian Stober[1]
[1] *Artificial Intelligence Lab, Otto von Guericke University Magdeburg, Germany*
Email: {maral.ebrahimzadeh, valerie.krug, stober}@ovgu.de

Abstract
Automatic chord recognition (ACR) is a popular task in the field of music information retrieval. The available research for ACR tasks indicates that there is less tendency to work on symbolic data rather than audio data. One of the main reasons for this underrepresentation is that there are few symbolic music datasets with adequate annotations available. To tackle this issue, it is possible to use unsupervised techniques on datasets without chord labels for pre-training generalized input embeddings.

In this paper, we use the Harmony Transformer (HT) architecture by Chen and Su in its recent version from 2021. We propose to exploit skip-grams of pitches as an unsupervised embedding technique instead of learning the input embedding as part of the network. This improves the HT such that it can make use of the large amount of unlabeled data. We do our experiments on Lakh MIDI dataset and also on BPS-FH dataset which was used in the Harmony Transformer related paper to compare the results. We also propose to use Explainable Artificial Intelligence (XAI) techniques to interpret how the model performs the chord recognition task, for example, by identifying prediction-relevant features in the input data.

Introduction
According to the fact that it is a time-consuming task to define chord labels for music pieces and that it requires deep understanding of music theory, Automatic Chord Recognition (ACR) is widely investigated in music information retrieval. ACR is a type of sequence labeling task. It gets a music sequence as an input and assigns a chord label to each segment of the music piece. But it is not specified which part of a segment can be accepted as a single complete chord. Therefore, it is also important to define a precise chord segment. In ACR, there are more research for audio data than for symbolic data. One of the reason is that there are only a few symbolic music dataset available which also include chord labels [1]. Whereas there are some large datasets available without any chord annotation.

It is shown in [4] that word2vec [12, 13] methods are able to learn harmonic structure of music. Hence, it is used in creating chord embeddings for different tasks like music generation. However, they only use the small datasets with chord annotations.

In this paper, we introduce our method for computing chord embeddings based on word2vec methods. We use the skip-gram [12] model and we train our model on datasets with and without chord labels. We also propose a visualisation method to evaluate our chord embeddings and we show that our pre-trained embeddings perform better in representing harmonic content for unlabeled data. For ACR task, we train the improved version of Harmony Transformer (HT) [1], in which we replace its embedding layer with our pre-trained embedding.

Related work
Prior to the emergence of deep learning, in the early research into the task of ACR, the main emphasis was on extracting precise and robust harmonic features like 12 dimensional chroma feature vector [5]. To obtain information about the type of the chord, researchers extended the feature vector to incorporate bass note [11]. Moreover, Hidden Markov Models (HMMs) were mostly employed as a generative model to obtain the label sequence [10].

After the popularity of Deep Neural Networks (DNNs) increased, different DNN architectures have been used for ACR task, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). In one of the earliest studies, a CNN was trained to perform a major-minor chord classification [7]. An RNN is used in [8] and predicts chord sequence without their duration by using a language model with chord progression data. To complete ACR task, they combined their model with a chord duration model in [9].

By increasing usage of attention mechanisms in different domains, three different transformer-based chord recognition have been proposed in recent years: Bi-directional Transformer for Chord Recognition (BTC) [14], Harmony Transformer (HT) [3] and the improved version of HT [1]. BTC only uses the encoder part of the transformer to obtain the long-term dependency in musical sequences. HT utilizes both encoder and decoder part of a transformer. Then, the encoder is responsible for chord segmentation and gives the segments as an input to the decoder. Based on the segmentation of the music sequence, decoder recognises the chord labels. The HT uses an intra-block intra-Multi Head Attention in the input of the models for learning localized harmonic features. It also obtains the adjacent information of inputs by using the convolutional Feed Forward Networks (FFN) instead of the fully-connected FFNs.

Method
According to the promising performance of HT in ACR task for chord annotated symbolic data, we use the same

model as our baseline. To make it possible to use large unlabeled symbolic music datasets, we replace the embedding layer with an unsupervised pre-trained embeddings.

Chord Embedding

As word2vec methods have the capability to learn harmonic concepts of music, we implement a skip-gram model to create pre-trained chord embeddings as an input for our chord recognition model. We consider each chord as a word by assigning a chord value to it. As the first step, we extract the chord track by using MIDI miner[6] because we only need the chord track of music pieces. Then, we convert the MIDI file to its piano roll representation. We convert all pitches to one octave resulting in a chroma vector. Hence, we find all available pitch classes in a chord. If we consider V_i as the chroma vector, we propose to compute chord values as above:

$$\texttt{chord value}_i = \sum 2^i (\texttt{if } V_i = 1)$$

Finally, we have a progression of chord values for each chord track. After training a skip-gram on this set of chord values, we have an embedding matrix for chords, in which vocab size is equal to the number of unique chord values.

Chord Recognition

We use the HT architecture which incorporates an encoder and a decoder as shown in Figure 1. The encoder performs the segmentation task, so that the defined segment is harmonically complete. The output of the encoder is a binary sequence in which 1 means that the chord changes and 0 means the segment keeps the same chord as the previous segment. The decoder performs the recognition task. Apart from the music sequence, it also takes the chord segments as an input. At the end, it predicts a chord label for each segment as an output.

In our approach, instead of giving the music sequence as an input and train the whole HT, we first train a skip-gram model and use the pre-trained chord embeddings as an input.

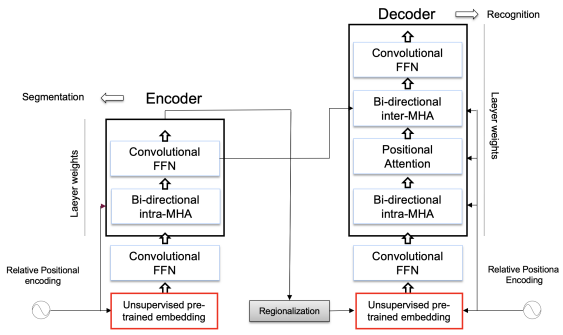

Figure 1: Modified Harmony Transformer by replacing the embeddings with pre-trained input embeddings.

Visualization

To visualize the pre-trained embedding, we use the concept of circle of fifth in music theory. This circle represents a sequence of keys and their root chords in which neighbour chords are seven semitones far from each other. Therefore, the closer the two keys are, the more pitch classes they share in common. To give an example, C major and G major are neighbors with eleven common pitch classes. They are only different in one pitch class which is F for C major instead of $F^{\#}$ for G minor. Moreover, each major key has a relative minor key with the same number of sharps and flats. Accordingly, major and minor keys can be displayed in a circle of fifth, such that the relative keys are paired. To illustrate, C major and A minor are relative major and minor keys with zero sharps and flats. In Figure 2 we can see the circle of fifth for major and minor keys.

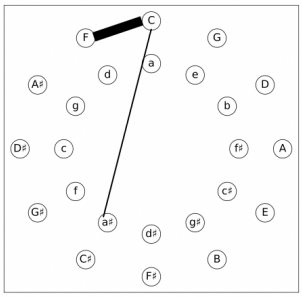

Figure 2: Similar chords in circle of fifth. The width of the line shows how harmonically similar the chords are.

We project the similarities of our pre-trained embeddings to the circle of fifth, to evaluate if it is learning the harmonic concept of music.

For this purpose, we compute an embedding matrix for chord labels by using the embedding matrix of chord values and its assigned chord labels. There are twelve different roots including $C, C^{\#}, D, D^{\#}, E, F, F^{\#}, G, G^{\#}, A, A^{\#}, B$ and two different qualities which are major and minor. Therefore, we have 24 different chord labels but a large size of unique chord values. Correspondingly, there is more than one embedding for each chord label. To get a unique embedding value for each chord label, we compute the average over the embeddings that are most frequently associated with the respective chord label.

Finally, we find the two nearest neighbours for each chord by using Euclidean distance and map them into the circle of fifth like indicated in Figure 2. Therefore, similar chords are connected to each other with a line. The wider the line is, the more similar the chords are in the embedding space.

Experiments

We evaluate our embeddings by training our model on two different symbolic music datasets, one with and one without chord annotation. The BPS-FH dataset [2] includes the first movement of Beethoven's 32 piano sonatas with chord annotation. And the Lakh Midi Dataset (LMD) [15] which dose not include any chord label.

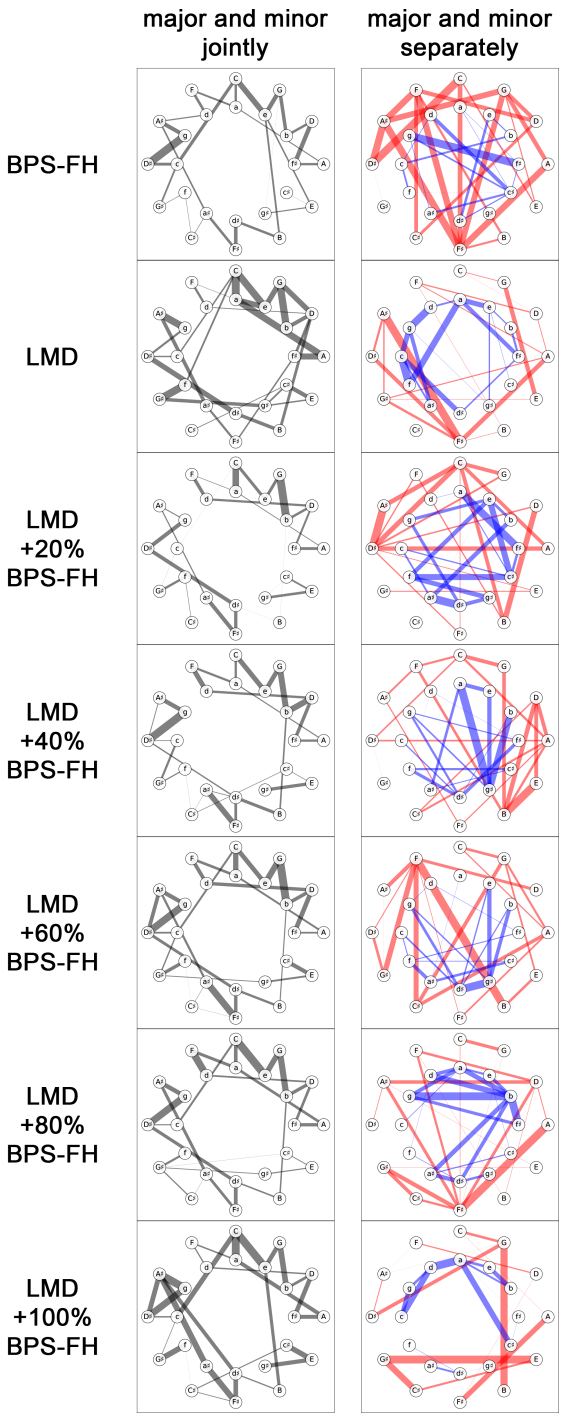

Figure 3: Visualization of similar major and minor chords where black shows similar chords between all major and minor chords, red shows similar major chords and blue shows similar minor chords

Embedding visualization

We compute the embeddings for BPS-FH dataset and we visualize the similar major and minor chords in the circle of fifth. We also perform the experiments for LMD and combined datasets that add 20 percent of BPS-FH to LMD at each time. As we can see in Figure 3, our proposed embedding can learn harmonic concepts better when using unlabeled datasets.

We evaluate our experiments once more by visualizing similar major and minor chords separately, because it is a common practice like how it is done in [4]. In Figure 4, similar major chords are shown in red and similar minor chords are presented in blue.

We also analyse our embeddings with computing the average path distance between chords. For this purpose, we define the shortest path between each similar chords in the circle of fifth visualization as it is shown in Figure 5. Then we compute an average over the whole shortest path distances between paired similar chords. We do this analysis for both BPS-FH, LMD datasets and LMD added by 20 percent of BPS-FH each time. Moreover, we perform this evaluation for similar major and minor chords separately, similar major and minor chords jointly and similar major and minor chords jointly and separately. As similar chords are close to each other in the circle of fifth, we expect that a good embedding has a shorter average path distance.

We can see that LMD embeddings have the shortest averaged path distance in similar major and minor chords separately, and similar major and minor chords jointly. In the similar major and minor chords jointly, BPS-FH and LMD have a similar value.

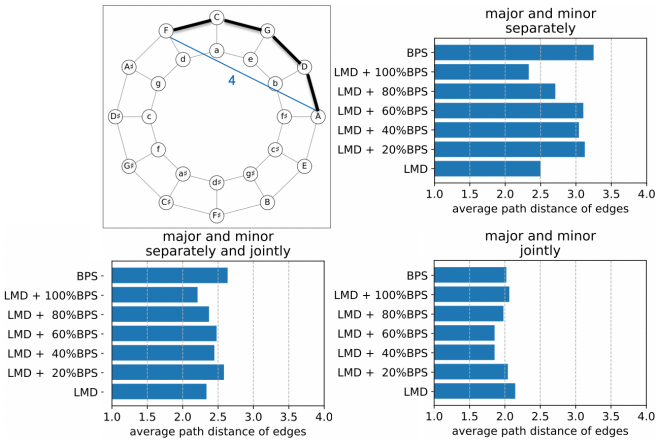

Figure 4: Average path distance among chords.

After we trained our skip-gram model, we use the pre-trained embeddings as inputs for the HT. We perform this experiment for BPS-FH but it is not working better than the baseline. The test accuracy for the baseline and our approach is 85.3% and 72.83% respectively. As our embeddings learn the similar chords better for unlabeled dataset, we expect that it will have a better performance for unlabeled datasets like LMD.

Conclusion and Future work

In this paper, we proposed a method to compute chord embeddings using skip-gram. We trained our model with two different datasets and also with combinaton of them. Moreover, we introduced a method to visualize embedding similarity of chords in the circle of fifth. The visualization shows that our embeddings learn better the similarity between chords for unlabeled datasets.

We used HT as a baseline for ACR task, but instead of using their suggested embedding layer, we proposed to use our pre-trained embeddings. We performed an ex-

periment for unlabeled data which could not outperform the baseline. This is a surprising finding. Although the embeddings appear to be better according to harmonic similarity it is not yet improving the performance of the baseline. This opens questions for further research.

We will continue our experiment with large unlabeled datasets and we expect that for unlabeled data, our approach will improve the chord prediction accuracy. Also, we expect that we will need less labeled data for training the HT.

References

[1] Tsung-Ping Chen and Li Su. "Attend to chords: Improving harmonic analysis of symbolic music using transformer-based models". In: *Transactions of the International Society for Music Information Retrieval* 4.1 (2021).

[2] Tsung-Ping Chen, Li Su, et al. "Functional Harmony Recognition of Symbolic Music Data with Multi-task Recurrent Neural Networks." In: *ISMIR*. 2018, pp. 90–97.

[3] Tsung-Ping Chen and Li Su. "Harmony Transformer: Incorporating chord segmentation into harmony recognition". In: *Neural Netw* 12 (2019), p. 15.

[4] Ching-Hua Chuan, Kat Agres, and Dorien Herremans. "From context to concept: exploring semantic relationships in music with word2vec". In: *Neural Computing and Applications* 32 (2020), pp. 1023–1036.

[5] Takuya Fujishima. "Realtime chord recognition of musical sound: Asystem using common lisp music". In: *Proceedings of the International Computer Music Conference 1999, Beijing*. 1999.

[6] Rui Guo, Dorien Herremans, and Thor Magnusson. "Midi Miner–A Python library for tonal tension and track classification". In: *arXiv preprint arXiv:1910.02049* (2019).

[7] Eric J Humphrey and Juan P Bello. "Rethinking automatic chord recognition with convolutional neural networks". In: *2012 11th International Conference on Machine Learning and Applications*. Vol. 2. IEEE. 2012, pp. 357–362.

[8] Filip Korzeniowski, David RW Sears, and Gerhard Widmer. "A large-scale study of language models for chord prediction". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2018, pp. 91–95.

[9] Filip Korzeniowski and Gerhard Widmer. "Improved chord recognition by combining duration and harmonic language models". In: *arXiv preprint arXiv:1808.05335* (2018).

[10] Kyogu Lee. "Automatic chord recognition using an HMM with supervised learning". In: *ISMIR-2006* (2006).

[11] Matthias Mauch and Simon Dixon. "Approximate Note Transcription for the Improved Identification of Difficult Chords." In: *ISMIR*. 2010, pp. 135–140.

[12] Tomas Mikolov et al. "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems* 26 (2013).

[13] Tomas Mikolov et al. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013).

[14] Jonggwon Park et al. "A bi-directional transformer for musical chord recognition". In: *arXiv preprint arXiv:1907.02698* (2019).

[15] C Raffel. "Learning-based methods for comparing sequences, with ap- plications to audio-to-midi alignment and matching (Unpublished doctoral dissertation). Columbia University." In: (2016).