SBERT-based Chord Progression Estimation from Lyrics Trained with Imbalanced Data

Mastuti Puspitasari[1], Takuya Takahashi[1], Gen Hori[2],
Shigeki Sagayama[1], Toru Nakashika[1], ⋆

[1] Department of Computer and Network Engineering
The University of Electro-Communications, Tokyo 182-8585, Japan
[2] Department of Data Science, Faculty of Business Administration
Asia University, Tokyo 180-8629, Japan
m2131179@gl.cc.uec.ac.jp

Abstract. In this research, we developed a model that can estimate appropriate chord progression based on lyrics input. It outputs a sequence of chord that can be used to compose the corresponding lyrics input. By training the model with different datasets, it is also possible to estimate other musical components that are correlated with lyrics, for example rhythm pattern, instrument, tempo, and drum pattern. Using this set of musical components as a setup recommendation for composition can potentially automate the configuration process on AI-based composition tools. We sourced our training data from “Orpheus”, a web-based automatic composition system, resulting in more than 6,000 paired data of lyrics and musical components chosen by users who published their songs in the platform. Lyrics are pre-processed into semantics embedding using Sentence-BERT before being fed as training data into the multi-layer perceptron model as a classifier to estimate chord progression. Evaluation of this model is done objectively with ROC and F1 score, and subjectively through a survey.

Keywords: chord progression estimation, lyrics pre-processing, musical components, automatic composition, Orpheus, semantics embeddings, Sentence-BERT, multi-layer perceptron

1 Introduction

Following the recent trend in AI research, there have been tools (eg: soundraw.io, Orpheus [1]) developed to automate music composition. They depend on user input to generate music, some by asking users to select genre or mood, while others expect more detailed input such as lyrics and chord progression. The simpler a tool is, the more attractive it is to new users, but unfortunately, the output will never be as personal as the input is limited. On the other hand, while a more complex tool can result in more personalized music, it can be overwhelming for new users.

⋆ This research was funded by Grant-in-Aid for Scientific Research (B) No. 21H03462 from Japan Society for the Promotion of Science (JSPS) and a scholarship from The Ministry of Education, Culture, Sports, Science and Technology (MEXT) Japan with the cooperation of fellow members of Nakashika Laboratory, The University of Electro-Communications (UEC).

Ideally, such tedious process should be presented with an offer of automated assistance. The easiest approach to this would be by recommending randomly selected setup during configuration. However, this can possibly result in music that does not match the lyrics. The system may end up recommending an upbeat musical configurations when a user inputs sad lyrics, for example. As mood and nuance can be inferred from text, we argue that it should be possible to estimate appropriate musical compositions based on lyrics input. To achieve this, we decided to experiment with a number of classifier models and train them on relevant training data. We use semantics embeddings of lyrics as input and estimate the appropriate chord progression and other musical components based on what they learned from the training data.

Fortunately, such data can be extracted from existing compositions, as long as the necessary musical components data are also accessible. Even with appropriate training data, however, estimating appropriate musical components is not that straightforward. Since music is not strictly derivable from lyrics, there will never be one true exact match of a composition setup for a specific lyrics input. In fact, we cannot say that any setup is wrong at all, considering that one lyrics input can potentially result in various compositions that can equally be considered as good matches. For this reason, subjective approach is also necessary to evaluate the model performance.

By automating the selection process based on lyrics input, we offer a solution that can leverage a tedious process to be more user-friendly, and thus, encourage existing or potential users to use the system to compose more music. The data of future compositions can also be used to further train the system and improve its performance, allowing the system to evolve over time.

2 Related Works

Our work was initially inspired by [2] in which Turkish lyrics are used to estimate the meta-data of the song, which includes: genre, authors, and year of publication. Similar studies had also been done on genre classification for lyrics in different languages. In [3] for example, an approach similar to [2] is applied on Nordic lyrics. These works were done with conventional approach using feature-based text pre-processing.

In [4], word2vec [5] is used to pre-process the lyrics. Their goal was to estimate chord progression based on lyrics using the data extracted from Orpheus, which then made it the base of this research. It is unfortunate that their model was of a low accuracy, but we argue that it is expected as they included all chord progressions available on Orpheus regardless the number of samples. It is not ideal to train a model to classify a class with insufficient number of samples as it will result in overfitting. To ensure that each class has enough samples for training, we decided to focus our research on the top 10 chord progression available on Orpheus.

Another problem with this approach is that using word2vec to pre-process lyrics means the semantics of the sentence is not considered, as it is meant to be used for word pre-processing. Different lyrics that consist of the same words will result in the same embedding despite the order, for example ”king likes queen” shares the same embedding as ”queen likes king”. To consider the semantics of the lyrics, we decided to take a more state-of-the-art approach for the lyrics pre-processing by utilizing a language model that is able to directly derive embedding from sentences.

Fortunately, many language models have been developed in recent years. An example of this would be BERT [6], which is designed to pre-train deep bidirectional representations from unlabeled text. Several task specific modifications have also been done on BERT, including Sentence-BERT [7], which can be used to quickly measure similarity between two or more sentences, which would originally take hours for BERT to compute. By using SBERT, we convert our lyrics data into their semantics embeddings, which can then be paired with chord progression or other relevant musical components data and used to train our multi-layer perceptron models. Considering that it has been proven possible to infer genre from lyrics by [2], [3], and other studies, we argue that it should also be possible to infer specific musical components based on lyrics input.

3 Dataset

To train our models, we extracted composition data of the published songs in Orpheus [1], a Japanese automatic composition system with over 700,000 pieces composition generated by their users. As shown in table 1, this data consists of lyrics, musical components, and several statistics in regard to the composition. According to [8], chord progression can be used to infer music emotion, which has been proven by [10] to be derivable from lyrics. In Orpheus, there are over 1500 variations of chord progression to choose from, available for view on page[1].

Table 1: Raw Data Sample of a Published Composition in Orpheus

Lyrics	Chord	Rhythm	Instr.	Tempo	Drum	#Likes	#Bms
からまつの林を過ぎて、 からまつをしみじみと見き。 からまつはさびしかりけり。 たびゆくはさびしかりけり。	Pachelbel-Kanon	sync-auf-3-8sf	48	100	perc-hirata-rocknroll2	114	3

We extracted paired data of lyrics and chord progression for our main experiment and truncated our dataset by only taking samples of the top 10 chord progression to avoid overfitting. Experiment results of the other musical components will be included as ablation studies to consider potential future research.

4 Proposed Model

Conceptually, our system takes lyrics as input and outputs a recommendation of chord progression. To achieve this, we convert lyrics into semantics embedding with SBERT models pre-trained with Japanese corpus before feeding them into a multi-layer perceptron model as training data. The conversion from lyrics into numerical embedding is necessary because computers do not understand the meaning of words. This conversion allows computers to assign values to lyrics and understand which lyrics are similar or different based on their numerical representations.

For comparison purpose, we also rebuilt the word2vec model as proposed in [4] with the Japanese corpus used by the SBERT model. The pre-processing results of these models differ in terms of dimension, with a size of 768 for the SBERT model and 50 for the word2vec model which affects the input layer size of the multi-layer perceptron model used for chord progression estimation as shown in Fig. 1:

[1] `https://www.orpheus-music.org/Orpheus-lib-harmony.php`

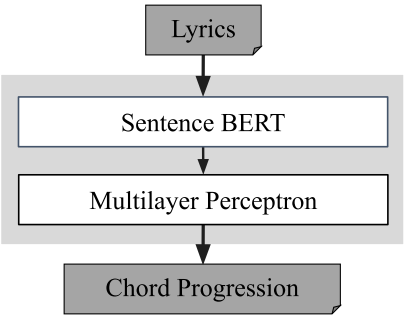


Fig. 1: Estimation Model Architecture

In [4], there was no mention of using a specific loss function on the training phase. For this reason, we used categorical cross-entropy to rebuild the word2vec mode, which is defined as follows, where p_i is the softmax probability of the i^{th} class:

$$L_{CE} = -\sum_{i=1}^{n} \log(p_i) \tag{1}$$

Unfortunately, applying this loss function to train an estimation model with imbalanced training data will likely result in overfitting. To mitigate this issue, we attempted a different approach that is based on [9], which claimed that applying focal factor $(1 - p_t)^\gamma$ can help to balance the weight of easy and hard samples and thus minimize the overfitting problem. Focal loss is calculated as follows:

$$L_{FCE} = -\sum_{i=1}^{n} (1 - p_i)^\gamma \log(p_i) \tag{2}$$

5 Experiments

5.1 Training with the Top 10 Chord Progression Dataset

In this section, we will discuss the result of our experiments on top 10 chord progression in terms of having the highest the number of samples. This was extracted from published Orpheus data with number of samples as shown in Table 2.

Table 2: Top 10 chord progression in published Orpheus Data

Label	#Samples
pattern O	1121
pattern FF	947
pattern Q	606
pattern P	570
pattern H	567
pattern E	539
pattern W	508
pattern R	402
Pachelbel Kanon Ending	394
User Harmony zkrxx7	388

Looking at the table, it is clear that there is a big difference in number of samples between the labels, showing an imbalance in data. Note that these labels represent different sequence of chords and not the chord progression itself. Refer to the link provided in section 3 for the full list of chord progression available on Orpheus.

5.2 Lyrics Pre-Processing and Loss Function

We experimented with the pre-processing using Japanese SBERT model and compare it with the word2vec model. The multi-layer perceptron models are also trained with two different cross-entropy (CE) loss functions, resulting in a total of four model variations: Word2Vec Categorical CE (WC), Word2Vec Focal CE (WF), Japanese SBERT Categorical CE (JC), and Japanese SBERT Categorical Focal CE (JF). They were trained with the top 10 chord progression dataset in 1000 epochs, with the ratio of 8:1:1, for training, validation, and test data respectively.

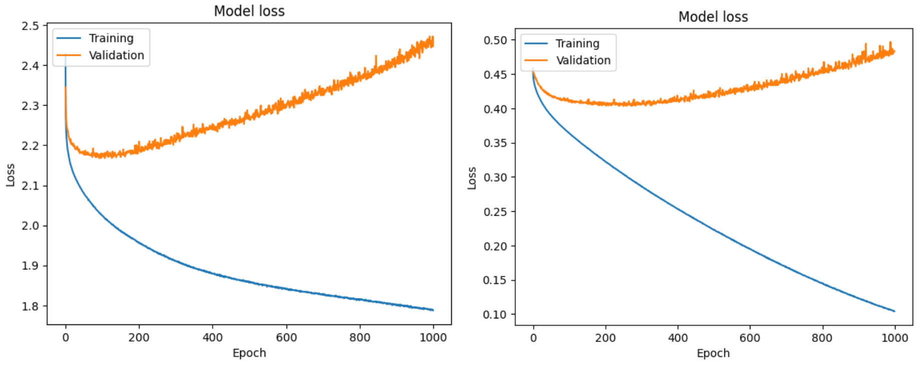
5.3 Final Accuracy and Overfitting of the Models on Chord Progression

We have compiled the final accuracy on both training (T) and validation (V) of each model in Table 3. We can see that using SBERT model pre-trained with the Japanese corpus results in higher accuracy (JC and JF) compared to those of word2vec (WC and WF). Note that due to the dataset unavailability, the word2vec model was pre-trained with a newer version of the Japanese corpus, and despite having this advantage, it was not able to achieve comparable accuracy values.

Table 3: Final training (T) and validation (V) accuracy of the 4 models

Model	T. Acc.(%)↑	V. Acc.(%)↑
WC	37.3	21.4
JC	**96.0**	**31.2**
WF	32.7	23.4
JF	**80.7**	**31.1**

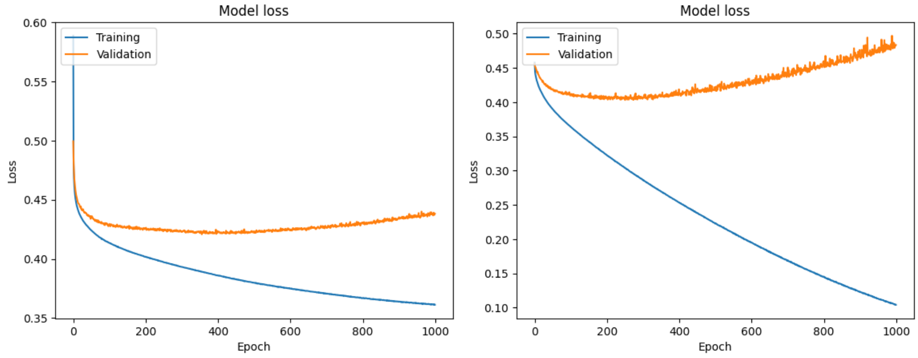
In Fig. 2, we can see that the models with categorical CE (WC and JC) are overfitting, and this can be minimized by applying focal CE during training (WF and JF) as shown in Fig. 3. Note that while JF seems to overfit badly based on the graph, the loss value is still below 0.5, which is still not to far off of WF.



(a) WF

(b) JF

Fig. 2: Loss over time of the 2 models trained with CE



(a) WF

(b) JF

Fig. 3: Loss over time of the 2 models trained with focal CE

Judging from both the accuracy and overfitting, it is safe to say that JF can replicate human preference in chord progression based on lyrics better than the other models, according to the data taken from composition published in Orpheus.

5.4 Objective and Subjective Evaluation

We evaluate the proposed model JF objectively by comparing its ROC AUC and F1 score with the word2vec approach WC and subjectively through survey to evaluate the quality of the generated music, in which a random composition published in Orpheus is recomposed with the chord progression recommended by JF and WC. We asked 10 respondents to score them based on how well the music matches the lyrics on a Likert scale from 1 (bad) to 5 (good). Note that the original composition is used as the ground truth in objective evaluations, and thus the lack of scores in Table 4.

Table 4: ROC AUC, F1, and Likert scores on chord progression

Model	ROC AUC (%)	F1 (%)	Likert (ave.±dev.)
Original	-	-	**3.5**(±1.08)
WC	64.2	23.4	2.7(±1.16)
JF	**69.6**	**32.1**	2.8(±**1.03**)

JF managed to get higher ROC AUC and F1 scores compared to WC. The Likert score of JF is also higher than WC with the lowest deviation. It can be concluded that using semantics instead of word embedding and changing the loss function to minimize overfitting result in better performance of the models in terms of recreating human preference and selecting the proper chord progression based on lyrics input.

5.5 Ablation Studies

To see the potential of applying this approach on other musical components, we considered four other subjects: rhythm pattern, instrument, tempo, and drum pattern. In [11], rhythm patterns were used to generate lyrics, which led us to believe that the opposite can also be done. Instruments are generally chosen by a composer according to the genre of music they are trying to produce. There is a typical tendency in tempo according to the genre of music as mentioned in [12]. Lastly, drum patterns in Orpheus were created with regards to musical genre with some variations.

As they are correlated with genre which is derivable from lyrics, it may be possible to derive them straight from lyrics. We experimented on top 10 dataset of these subjects with WC and JF and have compiled the evaluation result in Table 5. More details on these experiments, including the number of samples of each class in the top 10 datasets are available on their respective sheet in this spreadsheet [2].

Table 5: Models evaluation on the other 4 musical components

Musical Component	Model	ROC AUC (%)	F1 (%)	Likert (ave.±dev.)
Rhythm Pattern	Original	-	-	2.8(±**1.14**)
	WC	59.8	35.8	3.0(±1.41)
	JF	**67.4**	**38.2**	**3.0**(±1.15)
Instrument	Original	-	-	2.3(±1.25)
	WC	60.2	25.1	2.0(±**0.94**)
	JF	**65.5**	**28.8**	**3.0**(±1.56)
Tempo	Original	-	-	**3.6**(±1.17)
	WC	58.7	23.3	3.2(±**1.03**)
	JF	**66.2**	**28.9**	3.4(±1.35)
Drum Pattern	Original	-	-	**2.9**(±0.88)
	WC	56.0	21.8	2.7(±**0.82**)
	JF	**61.5**	**24.5**	2.7(±1.06)

Table 5 shows that JF is consistently superior than WC in terms of ROC AUC and F1 score. In the survey, it is also generally better in terms of performance compared to WC, with the exception on drum pattern. However, the drum pattern survey data shows that respondents were unsure of the sample difference and not confident in their answers. It can be concluded that JF performs better than WC when there is clear differences between the samples and respondents are confident. Another interesting point that is worth mentioning here is that on rhythm pattern, both WC and JF scored higher than the original composition, which shows the potential of these models in recommending appropriate musical components based on lyrics input.

6 Discussion

The proposed model JF managed to achieve higher ROC AUC, F1, and Likert score on chord progression estimation in comparison to the model WC as proposed in [4], and it is interesting that with similar approach, similar results are also reflected on other musical components, although with some degrees of deviation. However, the ROC AUC and F1 scores of the proposed model JF are still considerably low and mixed results can be seen on the survey. As the training is done by labelling to represent each class, similarities between each class are not considered.

By considering the feature similarities that are unique to each musical component, we argue that it is possible to achieve higher ROC AUC, F1, and Likert score of the classifier model. Chord sequence, for example, may be processed better with seq2seq approach instead of considering each sequence of chord as an entirely different class, rhythm pattern can be labeled with their individual notes, instrument can be grouped according to their similarities in terms of timbre, and so on.

[2] `https://docs.google.com/spreadsheets/d/16-MMdycFS2SN44hR5kFLeRmyNNquK3hHwhs4Lou3kY8/edit?usp=sharing`

7 Conclusion and Future Works

In this paper, we proposed a an approach to estimate chord progression, and potentially other specific musical components based on lyrics input by using SBERT model for lyrics pre-processing instead of word2vec as proposed in [4]. We also consider the imbalance in data and limit our scope by using top 10 dataset as training data. During training, focal cross-entropy is applied instead of cross-entropy loss function to mitigate the overfitting caused by the difference in number of samples between the classes.

The proposed model achieved higher ROC AUC and F1 score in comparison to the model proposed in [4]. Through a survey that compares audio samples configured with the two models and the original composition, it can be concluded that the proposed model generally performs better than the previous model, and can potentially generate music better than the original work in terms of how well they match the lyrics input. The proposed model can also be potentially improved by considering similarities between each class and features that are unique to each musical component.

References

1. Sagayama, S.: Orpheus : An Automatic Music Composition System. In: The journal of the Institute of Electronics, Information and Communication Engineers, pp.214–220. The Institute of Electronics, Information and Communication Engineers (2019)
2. Oğul, H., Kırmacı, B.: Lyrics Mining for Music Meta-Data Estimation. In: 12th IFIP International Conference on Artificial Intelligence Applications and Innovations, pp.528–539. HAL open science, Greece (2016)
3. de Lima, A., Nunes, Rodrigo M., Ribeiro, Rafael P., Silla, Carlos N.: Nordic Music Genre Classification Using Song Lyrics. In: Natural Language Processing and Information Systems, pp.89–100. Springer International Publishing, Cham (2014)
4. Shinohara, K.: Automatic Chord Progression Setting Considering the Meaning of Japanese Lyrics in Automatic Composition. Meiji University Graduation Thesis, Meiji University (2018)
5. Mikolov, T., Chen K., Corrado, G., Dean, J.: “Efficient Estimation of Word Representations in Vector Space”, arXiv (2013).
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Computing Research Repository (CoRR), arXiv (2018)
7. Reimers, N., Gurevych, I.: Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In: Computing Research Repository (CoRR), arXiv (2019)
8. Cho, Y.H., Lim, H., Kim D.W., Lee, I.K.: Music emotion recognition using chord progressions. In: IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp.2588–2593 IEEE (2016)
9. Lin, T.Y., Goyal, P., Girshick ,R., He, K., Dollar, P.: Focal Loss for Dense Object Detection. In: Facebook AI Research (FAIR), arXiv (2018)
10. Edmonds, D., Sedoc J.: Multi-Emotion Classification for Song Lyrics. In: Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pp.221–235. Association for Computational Linguistics (2021)
11. Oliveira, H.R.G., Cardoso, F.A., Pereira, F.C.: Tra-la-Lyrics: An approach to generate text based on rhythm. In: Computational Creativity 2007, pp.47–54. University of London, Goldsmiths (2005)
12. Wolf, T.: Genre Classification of Electronic Dance Music Using Spotify’s Audio Analysis. Towards Data Science (2020)