FUNCTIONAL HARMONY RECOGNITION OF SYMBOLIC MUSIC DATA WITH MULTI-TASK RECURRENT NEURAL NETWORKS

Tsung-Ping Chen and Li Su
Institute of Information Science, Academia Sinica, Taiwan
{tearfulcanon, lisu}@iis.sinica.edu.tw

ABSTRACT

Previous works on chord recognition mainly focus on chord symbols but overlook other essential features that matter in musical harmony. To tackle the functional harmony recognition problem, we compile a new professionally annotated dataset of symbolic music encompassing not only chord symbols, but also various interrelated chord functions such as key modulation, chord inversion, secondary chords, and chord quality. We further present a novel holistic system in functional harmony recognition; a multi-task learning (MTL) architecture is implemented with the recurrent neural network (RNN) to jointly model chord functions in an end-to-end scenario. Experimental results highlight the capability of the proposed recognition system, and a promising improvement of the system by employing multi-task learning instead of single-task learning. This is one attempt to challenge the end-to-end chord recognition task from the perspective of functional harmony so as to uncover the grand structure ruling the flow of musical sound. The dataset and the source code of the proposed system is announced at `https://github.com/Tsung-Ping/functional-harmony`.

1. INTRODUCTION

.

Harmony and tonality represent the essence of Western tonal music. A complete analysis of the *functional harmony* in a musical piece needs one to utilize several interrelated concepts, such as chord progression, diatonic function, chord inversion, key modulation, to name but a few. These concepts are of fundamental importance in music theory, as they provide a systematic guide for one to understand how a phrase starts and how it ends, how one chord is related to another, how a chord is related to the key of the music, and more generally, how music works.

Computational approaches to analyzing musical harmony have gained wide attention in the past decades. Many works related to this topic, such as chord recognition [2, 6, 12, 18, 21, 23, 35], key detection [3, 9, 17, 27], and chord sequence modeling and generation [5, 10, 14, 28, 31, 32], as the *sub-problems* of the complete *functional harmony recognition* problem, have been extensively studied. Among these sub-problems, chord recognition is arguably the most widely-investigated one.

Chord recognition focuses on the identification of *chord symbol*, i.e., symbols which indicate the root note, the chord quality (e.g., Major), and occasionally an extra interval number (e.g., seventh) of a chord. [1] Such a notation system provides direct instructions on chord construction, and therefore becomes prevalent in jazz and pop music. However, this notation system is insufficient for a more holistic analysis as it provides no information about *chord functions*. [2] For example, the *secondary chord* [3] that plays an important role in the analysis of the hierarchical structure in a chord sequence is rarely discussed in the literature. Little efforts at such data annotation are due to it requires musicology expertise. As a result, there is no systematic studies on a more holistic recognition system based on all the above-mentioned concepts of functional harmony analysis, to the best of our knowledge. Although this topic has been extensively studied in the field of music information retrieval (MIR), the computers' ability of harmonic analysis is still quite limited.

In this paper, we discuss the *functional harmony recognition* problem. To tackle this problem, we first build a new dataset comprising five different chord functions, namely the key, primary degree, secondary degree, quality, and inversion. Since there is no unique and exact definition on functional harmony analysis of music, we alternatively consider the functional harmony recognition problem as the recognition of the above-mentioned five aspects, in order to facilitate the discussion in an engineering sense. We formulate this problem with the perspective of multi-task learning (MTL), and implement the system using the recurrent neural networks (RNN) with long short term memory (LSTM) units, a network structure that has been found useful in the audio chord recognition problem [6]. Experiments on the dataset show that the chord functions can be better resolved within the multi-task learning scenario

[1] For example, a chord played with notes C-E-G-B is notated as CM7.

[2] In the strict sense, the term *chord function* refers to the *diatonic function*, namely the Roman numeral annotation and the functions like tonic (T), dominant (D) and sub-dominant (S). In this paper we opt to choose a rather loose definition by regarding key, degree, and inversion also as some generalized 'functions' of a chord.

[3] In this paper, the term *secondary chord* refers to the chord that does not serve the key. The borrowed chords, altered chords and the secondary dominant belong to this category.

compared to a single RNN structure, marking a step toward a more advanced computational music analysis framework.

2. RELATED WORK

2.1 Chord recognition and key detection

The chord recognition problem has been widely investigated on both the audio and symbolic data. In recent years, various machine learning techniques have been applied in this problem. In audio data processing, RNN-based methods such as the LSTM-based networks have been adopt due to its potential to model the long-term dependency of a time series [6,12,30]. Besides, [26] proposes a word2vec neural network to model the *harmony tension*, which also represents another perspective of chord function modeling. In symbolic data processing, early studies based on handcrafted rules have considered the chord recognition of Roman numeral notations (i.e., chord symbol and tonality) [13]. [19] considered deep neural networks in chord recognition. Recent approaches based on machine learning, with evaluation performance include: [15] applies deep learning to identify non-cord tones in symbolic music data, and [21] uses a semi-Markov conditional random field (CRF) model for symbolic-level chord recognition.

Most of the studies on the key detection problem investigate the global key or home key detection [9, 17]. [17] proposes a global key finding algorithm with a convolutional neural network (CNN). The studies of key modulation detection are less seen, while there are still some related works such as local key detection [27].

2.2 Multi-task learning (MTL)

The MTL technique is proposed to fit one shared network to multiple related sets of labels, i.e., to learn multiple tasks at a time [20, 29]. If a primary task itself is difficult or is short of training data, its performance can be improved by introducing some auxiliary tasks by assuming these tasks share similar network structure.

MTL has exhibited great potential in MIR [11] since different attributes of music are often highly related. For example, in [34], the neural network is shared by the chord recognition task as well as the root note recognition task, and doing this can help to improve the accuracy of chord recognition. Similar ideas can also be seen in other models such as the multi-chain hidden Markov model (HMM) [22] and the dynamic Bayesian network [24]. Therefore, it suggests that the functional harmony problem itself is a multi-task learning problem, as determining one type of chord function usually needs the information of another.

2.3 Datasets for functional harmony recognition

Accurate annotation chord functions is hard to build in the audio domain, but rather feasible in the symbolic domain. There are a few datasets including annotation of some, if not all, chord functions: for example, the KSN dataset provides the annotation of chord and key modulation (i.e., the Roman numeral annotation) [16], the Theme And Variation Encodings with Roman Numerals (TAVERN) dataset has Roman number chord annotation [8], and the Yale Classical Archive Corpus (YCAC) dataset has local tonic label and chord [33].

3. DATA AND LABELS

We propose the Beethoven Piano Sonata with Function Harmony (BPS-FH) dataset, which contains the symbolic musical data and functional harmony annotations of the 1st movements of 23 of Beethoven's Piano Sonatas. [4] BPS-FH dataset provides a more consistent corpus in terms of musical form and genre with concise annotations for the analysis of harmony. As an ongoing work, the annotation will be extended to all the 32 piano sonatas.

3.1 Annotation process: harmonic analysis [5]

The BPS-FH dataset is annotated by an expert musicologist with a basic *harmonic analysis* process step-by-step. As opposed to the chord symbol annotation, the traditional harmonic analysis in music theory and musicology adopts a relative representation for chords to emphasize the interaction between chords in a given context. To perform harmonic analysis, there are several implicit processes:

- Key identification: the first step of harmonic analysis is to identify the *local key* according to context. Note that in many classical musical pieces, there is no exact analysis on the local key, for key modulation usually occurs, making it hard to find the local key in a certain excerpt. [6] When the ambiguity occurs, finding a later cadence which is in a key-steady context, and then analyzing chords backwards might give a solution.

- Segmentation: since music itself is not represented originally as a sequence of chords, it is important to identify reasonable segments for labeling chords. A convincing segmentation should take the temporal rhythm and the harmonic rhythm (i.e., the rate at which the chords change) into consideration.

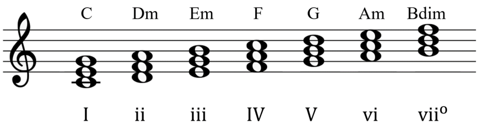
- Harmonic reduction: after determining the segments, each segment is reduced to a chord symbol (including chord root and chord quality) according to the tones within it. Harmonic reduction is a non-trivial and complicated process; there are many confusing factors, such as the non-chord tones, or the absence of harmonic tones in the segment.

- Inversion recognition: the inversion of a chord is determined by which of the notes is the bottom note, or bass note, of the chord. Typically, the lowest note in

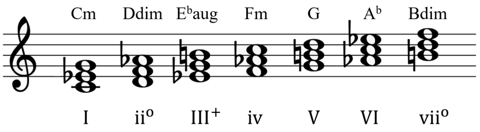
[4] The 23 pieces are: No. 1, 3, 5, 6, 8, 11, 12, 13, 14, 16, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 31, and 32. And all the repetitions in the sonatas are unfold.

[5] In this paper, harmonic analysis refers to Roman numeral analysis.

[6] In music, modulation is the act of changing from one key (tonic, or tonal center) to another. Generally speaking, the key of a musical piece refers to the global key which identifies the global tonic note and the final point of rest for the piece, while a modulation conducts the piece temporarily to another key, that is, a local key, which replaces the global tonic with a temporary tonic in a local area.



(a) Diatonic function in C major.



(b) Diatonic function in C minor (harmonic minor scale).

Figure 1: Illustration of diatonic functions in relation to the diatonic chords of the given keys. Note that in minor key, the superscript $^+$ is added to the *mediant* because it is an augmented chord.

a segment would be considered as the bass of the reduced chord. However, the lowest note is not always regarded as the bass notes; the pedal point is one of such examples.

- Labeling diatonic functions: after determining the key and the chord symbol, a *function* is assigned to the chord. In a major key, the following Roman numerals are used to represent the functions of diatonic chords: I (tonic), ii (super-tonic), iii (mediant), IV (sub-dominant), V (dominant), vi (submediant), and vii^o (leading). The capital numerals denote major chords, the lowercase numerals denote minor chords, and the superscript ^{o} denotes diminished chords. Figure 1 shows the details of diatonic functions in both major and minor keys.

Figure 3b exhibits a brief example of harmonic analysis for the excerpt in Figure 3a. It is worth mentioning some possible confusions when analyzing harmony on this example: at measure 83, there are two non-chord tones, G at the 1st beat, and E^b at the second half of the 2nd beat, both of which might be confusing for harmonic reduction. Especially, the existence of the the non-chord tone G prevents the note E (the last note of measure 82) from directly resolving to F, and blurs the boundary between F-minor key and E^b-major key. Hence, the key modulation might occur at measure 83 as labeled, but might also occur at measure 84 or even 85. It should be acknowledged that harmonic analysis is inherently subjective, and the confounding effect of subjectivity may affect the performance of a chord recognition system in many ways [25]. Details about the harmonic analysis techniques and labeling paradigms can be found in [1] and [4] .

3.2 Annotations in the BPS-FH Dataset

A fundamental harmonic analysis provides the information of key, degree, quality and inversion. Therefore, the BPS-FH dataset has the corresponding annotations as follows:

- Key: the key to which a chord belongs in a local area. Since key modulation is essential in piano sonata, we trace the change of key, that is, we specify the *local key*, or temporary tonic, so as to show that how a key deviates from the global one during the course of the movement.

- Primary degree and secondary degree: degree refers to the position of a chord's root on the diatonic scale of a key. [7] There are seven possible degrees on a diatonic scale, that is, 1, 2, ..., 7. We use a pair of degrees, *primary degree* and *secondary degree*, for both diatonic chords and secondary chords. Primary degree indicates the position of the temporary tonic on the scale, while secondary degree denotes the position of the chord's root based on the temporary tonic; the couple of degrees is represented as secondary degree/primary degree. In the case of diatonic chord, the primary degree is always 1. That is, the temporary tonic is the same as that of the current key. As for the secondary chord, both the primary degree and the secondary degree can be any possible degree. For example, the diatonic chord V is represented as 5/1, while the secondary chord V/IV is represented as 5/4.

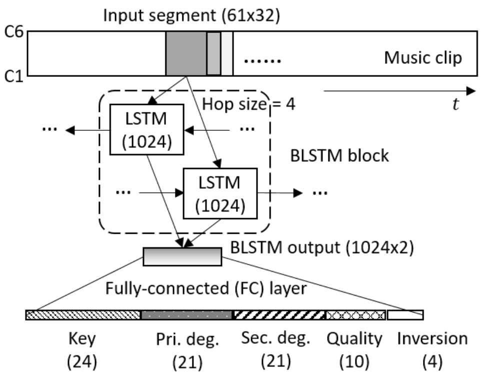
- Quality: chord quality is defined by the intervals within a chord. For instance, a major triad has a major third and a perfect fifth above its root. 10 types of chord quality are identified in the dataset, which are major triad (M), minor triad (m), augmented triad (a), diminished triad (d), major seventh (M7), minor seventh (m7), dominant seventh (D7), diminished seventh (d7), half-diminished seventh (h7), and augmented sixth (a6).

- Inversion: inversion of a chord describes which of the tones in a chord is the bass note. For example, the C-major triad has three candidates, C, E and G, as its bass, and thus has three possible inversions (root position is regarded as one inversion in the context). For triads and seventh chords, there are totally four possible inversions: the 0th inversion (root position), 1st inversion (^{6}or $^6_3$ for triad, and $^6_5$ for seventh chord), 2nd inversion ($^6_4$ for triad, and $^4_3$ for seventh chord), and 3rd inversion ($^4_2$ for seventh chord). Note that only seventh chords have 3rd inversion.

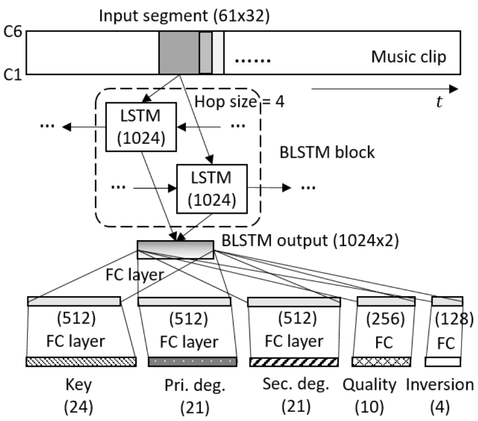
In summary, the BPS-FH dataset contains 86,950 note events, 29 different keys, 531 key modulations, and 7,394 chord labels. [8]

[7] For example, the chord C major triad has the degree 1 in C major key, while has the degree 4 in G major key.

[8] Among all the chords, 3,438 are inverted; 839 are secondary chords; 2,951 are major triads; 1,356 are minor triads; 25 are augmented triads; 286 are diminished triads; 30 are major seventh chords; 86 are minor seventh chords; 2,037 are dominant seventh chords; 453 are diminished seventh chords; 104 are half diminished seventh chords; 66 are augmented sixth chords.



(a) MTL-BLSTM-RNN with 1 task-specific layer



(b) MTL-BLSTM-RNN with 2 task-specific layers

Figure 2: Illustration of the MTL-based functional harmony recognition system, with a BLSTM-RNN model taking a data stream as input.

3.3 Data representation

The input data is represented in the format of a 61-key piano-roll, with the pitch range from C1 to C6 (middle C = C4); the duration of each note is measured in crotchet beats. For time resolution, we define a 32th note as the minimal time step. All the note events out of this pitch range are transposed to fit in, while the durations of the note events whose lengths are shorter than the minimal time resolution are set to be the same as the time resolution. A piano-roll at a time instance is called a *frame*.

As shown in Figure 2, the input of the LSTM cell is a segment of data with 32 frames. That is, for a musical piece with 4/4 meter, the length of a segment is 4 beats (or equivalently 1 bar). And a musical *clip* containing 64 segments is fed to the neural networks. The *hop size* for the neural networks is 4 frames (or half a beat.)

4. MODEL

We employ recurrent neural networks (RNN) with bidirectional long-short-term memory (BLSTM) units (denoted as BLSTM-RNN hereafter) to model sequences of functional harmony, by using the above-mentioned data representation as input. Such kind of model has been widely used in audio chord symbol recognition problems [6, 7, 12], and has been found capable in learning long-term information such as music structure. Specifically, we consider the following two types of networks:

Label	Dim	Content
Key	24	24 major and minor keys
Pri. deg.	21	7 Roman numerals by 3 (neutral, ♯, ♭)
Sec. deg.	21	7 Roman numerals by 3 (neutral, ♯, ♭)
Quality	10	M, m, a, d, M7, m7, D7, d7, h7, a6
Inversion	4	0th, 1st, 2nd, 3rd

Table 1: Chord function labels in the BPS-FH dataset, including key, primary degree (pri. deg.), secondary degree (sec. deg.), chord quality, and chord inversion.

Set	Piece No.
Training	1, 3, 5, 11, 16, 19 20, 22, 25, 26, 32
Validation	6, 13, 14, 21, 23, 31
Testing	8, 12, 18, 24, 27, 28

Table 2: The pieces in training, validation, and testing sets.

- MTL-BLSTM-RNN with 1 task-specific layer: as shown in Figure 2a, we adopt a simple BLSTM architecture with 1024 hidden units for multi-task leaning. The outputs of the forward and the backward cells are concatenated and form a 1024-by-2 matrix. This matrix is flattened and is connected to the output layer through a fully-connected layer. The output layer is a 80-D vector containing the classes for the five tasks listed in Table 1. Each class is one-hot encoding, and the Softmax function is used for the output vector.

- MTL-BLSTM-RNN with 2 task-specific layers: as shown in Figure 2b, the architecture is the same as the above, but with an additional task-specific layer before the output layer, in order to further increase the model capacity.

Moreover, to verify the advantage of MTL, we also consider the single-task learning (STL) as a baseline approach, where the same BLSTM-RNN is used. As a result, there are five networks in the STL-BLSTM-RNN model, each for one chord function recognition task respectively, and are trained individually in the experiment.

5. EXPERIMENT

5.1 Experimental settings

In the training stage, we divide the 23 pieces in the dataset into three parts, namely the training set, the validation set, and the testing set. Each part contains overlapped clips which are the input instances of the BLSTM networks. Each clip contains 64 segments, and the overlap between two consecutive clips is 32 segments. To balance the data

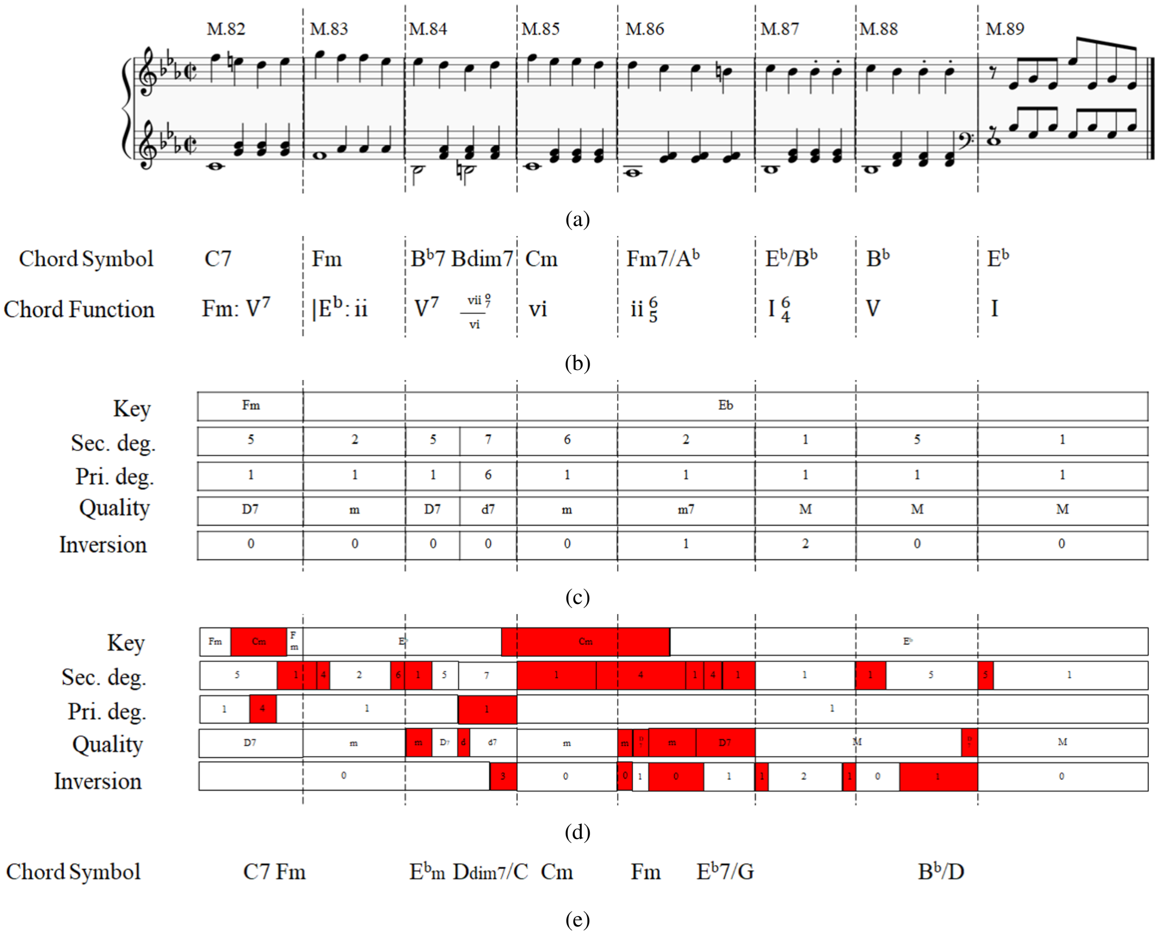


Figure 3: (a) An excerpt from the 1st movement of Beethoven's Piano Sonata No. 8, MM. 82-89. (b) The harmonic analysis of this excerpt represented in both chord symbol and chord function. Note that the slash used in chord symbol stands for an inversion, and the note behind the slash denotes the bass of the chord. In the analysis, this expert starts from F minor, modulates to E^b major at measure 83, and finally ends with an authentic cadence. (c) 5 types of annotations representing the functions in (b). (d) The testing result of chord function recognition of the excerpt. Wrong predictions are marked in red. (e) The translation of the result in (d) to chord symbol. For the sake of concision, only the wrong predictions lasting at least one quarter note are translated.

distribution among all possible keys, We perform data augmentation by transposing all the clips into 12 keys. As a result, there are 7,320 clips for training, 3,672 clips for validation, and 3,636 clips for testing. Table 2 shows the musical pieces used in each set. In the experiment, we compare the following two tasks:

- Chord symbol recognition: with the symbolic data of music as inputs, the model outputs chord symbol predictions in a segment-wise manner. We used 25 chord classes for the output layer, that is, 24 classes for 12 major triads and 12 minor triads, and an 'other' class for chords not belonging to either major triads or minor triads.
- Chord function recognition: similar as the chord symbol recognition, but the outputs of the model are chord functions containing five components.

Both the MTL and STL schemes are tested on the chord function recognition task, while the chord symbol recognition is tested with STL. For the chord function recognition task with MTL scheme, the outputs of the five chord functions are translated to chord symbol to evaluate the performance in terms of chord symbol recognition. And for the chord function recognition task with STL scheme , five different networks are trained individually for the evaluation of chord function recognition.

All networks are implemented with TensorFlow, and are trained using stochastic gradient descent with the Adam optimization method. For training objective, we compute categorical cross-entropy between targets labels and network outputs, and include a L2 regularization term. Moreover, to prevent over-fitting and to speed up training convergence, recurrent batch normalization is applied, and the dropout rate at the input and the output of the LSTM cell is set to be 0.5.

Task	Model	Key	Degree	Secondary	Quality	Inversion	Overall	Translation
Chord Symbol	STL-BLSTM-RNN	–	–	–	–	–	72.71	-
Chord Function	STL-BLSTM-RNN	67.06	48.31	9.38	61.87	57.95	23.57	56.05
	MTL-BLSTM-RNN with 1 task-specific layer	68.48	50.49	10.96	62.31	60.04	25.53	56.91
	MTL-BLSTM-RNN with 2 task-specific layers	66.65	51.79	3.97	60.59	59.10	25.69	56.25

Table 3: Accuracy (in %) of functional harmony recognition and comparison between multi-task BLSTM and single-task BLSTM. In the table, *Degree* stands for the accuracy of correctly predicting both the primary and secondary degrees of all chords; while *Secondary* indicates the accuracy of correctly predicting the degrees of secondary chords.

5.2 Evaluation metrics

We compute the segment-level *accuracy*, the ratio between the number of correct detection and the number of total segments in the testing set, for each category. Only one accuracy value is computed in the case of chord symbol recognition, while six types of accuracies are computed in the case of chord function recognition, namely the accuracies of key, degree, secondary chord, quality, inversion, and finally, the overall accuracy. Note that the accuracy of secondary chord is computed when a secondary chord does exist. The overall accuracy counts the segments in which the five chord function detections are all correct. An extra translation accuracy is computed to examine the performance of chord function recognition in terms of chord symbol recognition.

5.3 Results

Table 3 shows the results of chord symbol recognition and chord function recognition. In the task of chord symbol recognition, the STL-BLSTM-RNN-based model gives an accuracy of 72.71%. In comparison to other existing works which also estimate chord symbols on classical music datasets such as [12,21], this result is acceptable while also reveals the room for improvement in recognizing chords in western classical music.

In comparison with the chord symbol recognition task, performing the chord function recognition task is much more challenging. Specifically, the best overall accuracy among all chord function recognition tasks is only 25.69%, which is far from that of chord symbol recognition. This is partly because there are as many as 10 chord qualities for the model to predict, and partly because tonal harmony itself is complicated and equivocal. On the other hand, MTL-BLSTM-RNN model with 1 task-specific layer outperforms the single-task one for all chord functions. This indicates that employing multi-task learning results in a promising improvement. Among all chord functions, the improvements of predicting degree and inversion are the most significant, with 2.18% and 2.09% increases in accuracy respectively. This consequence may result from the fact that identifying the degree and identifying the inversion of a chord are relatively difficult in classical music, and thus benefit more from multi-task learning. Moreover, the accuracies of secondary chord are very low for all experiment settings; adding one more task-specific layer even degrades its performance. This displays the difficulty of learning the chord representation consisting of semantic information. Finally, we translate the predictions of chord function recognition tasks into chord symbol to examine the performance in terms of chord symbol recognition. It comes as no surprise that the all the translation accuracies are lower than that of chord symbol recognition. This again marks the challenge of chord function recognition, as it needs to consider not only the elements constructing a chord symbol, but also more high-level semantic information such as local key and degree.

An example of the chord function recognition result is shown in Figure 3d. Because the prediction is segment-wise, there are numbers of discontinuities in the predicted sequences. This issue can be addressed by further incorporating temporal smoothing models such as the CRF [21] in the future. A close examination of this result shows that although the model gives 'wrong' predictions, part of the predictions does match the ground truth on the level of chord symbol. For instance, as demonstrated in Figure 3d & 3e, there are whole-bar error predictions in key and secondary degree at measure 85; however, these detections become correct if we translate them into chord symbol: they are both C minor triads, albeit in different keys. In fact, further analysis points out that the prediction of the modulation to C minor at measure 85 is also meaningful: there does exist a potential modulation for there is a *tonicization* of vi constructed by the previous chord vii^{o}_{7}/vi at the second half of the measure 84. From this point of view, the model does provide more insight into the analysis of tonal structure in this excerpt, as an expert analyzer can do.

6. CONCLUSION AND FUTURE WORK

We have given a systematic investigation on the problem of functional harmony recognition of symbolic data based on deep learning techniques. Experiments on the proposed Beethoven Piano Sonata with Functional Harmony dataset indicate that functional harmony recognition is a task much more challenging than the chord symbol recognition, and a multi-task learning framework provides a promising solution better than a single-task one. Detailed analysis results not only give insightful interpretation, and also pose further challenging problems on recognizing key modulation, secondary degree, etc., all with its semantic level higher than chord symbols. This work marks a preliminary step towards a holistic approach of modeling functional harmony, and also provide the potential for one to analyze interpretable and meaningful music patterns from music, or to explore some alternative interpretation of music in the study of computational music analysis.

7. REFERENCES

[1] Edward Aldwell and Carl Schachter. *Harmony and Voice Leading*. Belmont: Thomsom Schirmer, 3 edition, 2003.

[2] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. Audio chord recognition with recurrent neural networks. In *Proceedings of the 14th International Conference on Music Information Retrieval (ISMIR)*, pages 335–340, 2013.

[3] Wei Chai and Barry Vercoe. Detection of key change in classical piano music. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, pages 468–473, 2005.

[4] Nicholas Cook. *A guide to musical analysis*. London: J. M. Dent & Sons, 1987.

[5] W. Bas de Haas, José Pedro Rodrigues Magalhães, Frans Wiering, and Remco C. Veltkamp. Automatic functional harmonic analysis. *Computer Music Journal*, 37(4):37–53, 2013.

[6] Junqi Deng and Yu-Kwong Kwok. Large vocabulary automatic chord estimation using deep neural nets: Design framework, system variations and limitations. *arXiv preprint arXiv:1709.07153*, 2017.

[7] Junqi Deng and Yu-Kwong Kwok. Large vocabulary automatic chord estimation using bidirectional long short-term memory recurrent neural network with even chance training. *Journal of New Music Research*, 47(1):53–67, 2018.

[8] Johanna Devaney, Claire Arthur, Nathaniel Condit-Schultz, and Kirsten Nisula. Theme and variation encodings with roman numerals (TAVERN): A new data set for symbolic music analysis. In *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, 2015.

[9] Emilia Gómez. *Tonal Description of Music Audio Signals*. PhD thesis, Universitat Pompeu Fabra, 2006.

[10] Mark Thomas Granroth-Wilding. *Harmonic Analysis of Music Using Combinatory Categorial Grammar*. PhD thesis, The University of Edinburgh, 2013.

[11] Philippe Hamel, Matthew E. P. Davies, Kazuyoshi Yoshii, and Masataka Goto. Transfer learning in mir: Sharing learned latent representations for music audio classification and similarity. In *Proceedings of the 14th International Conference on Music Information Retrieval (ISMIR)*, pages 9–14, 2013.

[12] Takeshi Hori, Kazuyuki Nakamura, and Shigeki Sagayama. Music chord recognition from audio data using bidirectional encoder-decoder LSTMs. In *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2017*, pages 1312–1315. IEEE, 2017.

[13] Plácido R Illescas, David Rizo, and José Manuel Iñesta Quereda. Harmonic, melodic, and functional automatic analysis. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 165–168, 2007.

[14] Nori Jacoby, Naftali Tishby, and Dmitri Tymoczko. An information theoretic approach to chord categorization and functional harmony. *Journal of New Music Research*, 44(3):219–244, 2015.

[15] Yaolong Ju, Nathaniel Condit-Schultz, Claire Arthur, and Ichiro Fujinaga. Non-chord tone identification using deep neural networks. In *Proceedings of the 4th International Workshop on Digital Libraries for Musicology*, pages 13–16. ACM, 2017.

[16] Hitomi Kaneko, Daisuke Kawakami, and Shigeki Sagayama. Functional harmony annotation database for statistical music analysis. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR): Late Breaking session*, 2010.

[17] Filip Korzeniowski and Gerhard Widmer. End-to-end musical key estimation using a convolutional neural network. In *Proceedings of the 25th European Signal Processing Conference (EUSIPCO)*, pages 966–970, 2017.

[18] Filip Korzeniowski and Gerhard Widmer. On the futility of learning complex frame-level language models for chord recognition. *arXiv preprint arXiv:1702.00178*, 2017.

[19] Pedro Kröger, Alexandre Passos, Marcos Sampaio, and Givaldo de Cidra. Rameau: A system for automatic harmonic analysis. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 273–281, 2008.

[20] Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 912–921, 2015.

[21] Kristen Masada and Razvan Bunescu. Chord recognition in symbolic music using semi-markov conditional random fields. In *Proceedings of the 18th International Society for Music Information Retrieval Conference*, pages 23–27, 2017.

[22] Matthias Mauch. *Automatic chord transcription from audio using computational models of musical context*. PhD thesis, Queen Mary University of London, 2010.

[23] Brian McFee and Juan Pablo Bello. Structured training for large-vocabulary chord recognition. In *Proceedings of the 18th International Conference on Music Information Retrieval (ISMIR)*, pages 188–194, 2017.

[24] Yizhao Ni, Matt McVicar, Raúl Santos-Rodriguez, and Tijl De Bie. An end-to-end machine learning system for harmonic analysis of music. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(6):1771–1783, 2012.

[25] Yizhao Ni, Matt McVicar, Raúl Santos-Rodriguez, and Tijl De Bie. Understanding effects of subjectivity in measuring chord estimation accuracy. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(12):2607–2615, 2013.

[26] Ali Nikrang, David R. W. Sears, and Gerhard Widmer. Automatic estimation of harmonic tension by distributed representation of chords. *arXiv preprint arXiv:1707.00972*, 2017.

[27] Hélène Papadopoulos and Geoffroy Peeters. Local key estimation based on harmonic and metric structures. In *Proceedings of the 12th International Conference on Digital Audio Effects (DAFx)*, pages 1–8, 2009.

[28] Christopher Raphael and Joshua Stoddard. Functional harmonic analysis using probabilistic models. *Computer Music Journal*, 28(3):45–52, 2004.

[29] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[30] Siddharth Sigtia, Nicolas Boulanger-Lewandowski, and Simon Dixon. Audio chord recognition with a hybrid recurrent neural network. In *Proceedings of the 16th International Conference on Music Information Retrieval (ISMIR)*, pages 127–133, 2015.

[31] David Temperley and Daniel Sleator. Modeling meter and harmony: A preference-rule approach. *Computer Music Journal*, 23(1):10–27, 1999.

[32] Christopher William White. A corpus-sensitive algorithm for automated tonal analysis. In *Mathematics and Computation in Music*, pages 115–121. Springer, 2015.

[33] Christopher William White and Ian Quinn. The yale-classical archives corpus. *Empirical Musicology Review*, 11(1), 2016.

[34] Mu-Heng Yang, Li Su, and Yi-Hsuan Yang. Highlighting root notes in chord recognition using cepstral features and multi-task learning. In *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1–8. IEEE, 2016.

[35] Xinquan Zhou and Alexander Lerch. Chord detection using deep learning. In *Proceedings of the 16th International Conference on Music Information Retrieval (ISMIR)*, pages 52–58, 2015.