Lead Sheet Transcription

Pierre Lardet

4th Year Project Report
Computer Science and Mathematics
School of Informatics
University of Edinburgh

2025

Abstract

This skeleton demonstrates how to use the `infthesis` style for undergraduate dissertations in the School of Informatics. It also emphasises the page limit, and that you must not deviate from the required style. The file `skeleton.tex` generates this document and should be used as a starting point for your thesis. Replace this abstract text with a concise summary of your report.

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Pierre Lardet*)

Acknowledgements

Table of Contents

Chapter 1

Introduction

1.1 Motivation

Lead sheets are a common way of representing music in a simple and concise format. They consist of an aligned melody and harmony through chord symbols. Lead sheets are widely used in the music industry as they provide a quick and easy way to learn and perform music, especially in jazz music where improvisation is common. Lead sheets are also used in music education, as they provide a simple way to introduce students to music theory and notation. However, creating lead sheets can be a time-consuming and error-prone process for complex pieces of music. This project aims to develop a model that can automatically generate lead sheets from audio recordings of music, which could be used to help musicians, educators, and students create lead sheets more easily and accurately.

However, there is no good source of lead sheets for many songs. Popular websites such as Ultimate Guitar [XX] and HookTheory [XX] provide user-submitted chord annotations for almost any song, but the quality of these annotations can vary significantly [XX]. Tools like Chordify [XX] can automatically generate chord annotations from audio recordings, but it is paid and designed for popo music. Online forums where a variety of leads can be found do exist, such as MusicNotes [XX] and SheetMusicDirect [XX], but are expensive, have limited selections and are also user submitted, meaning their quality cannot be guaranteed.

To this end, we investigate the use of machine learning models for lead sheet generation, requiring the model to perform automatic chord recognition and melody transcription.

1.2 Aims

The aims of this project are:

- To implement a state-of-the-art model for automatic chord recognition.
- Investigate methods of improving on this baseline model for music

- To investigate the use of synthetic data generation for improving the performance of the model.

1.3 Outline

The report is structured as follows:

- **Chapter 2** provides background information on chord transcription and related work.
- **Chapter 3** describes the datasets and evaluation metrics used in this project.
- **Chapter 4** describes the baseline model, investigates how it can be improved and compares it to other similar models.
- **Chapter 5** extends this work with synthetic data generation and compares results on a new dataset.
- **Chapter 6** concludes the report and provides suggestions for future work.

Chapter 2

Background & Related Work

2.1 Background

2.1.1 Lead Sheets

Lead sheets are a form of musical notation that contain both the melody and harmony. Only the dominant melody is present which is often the vocal line and is written in standard musical notation on a stave. The harmony is represented by chord symbols, which are written above the stave at the relevant time. A lead sheet also contains the key, time signature and, optionally, the lyrics of a song. An example of a lead sheet is shown in Figure 2.1.

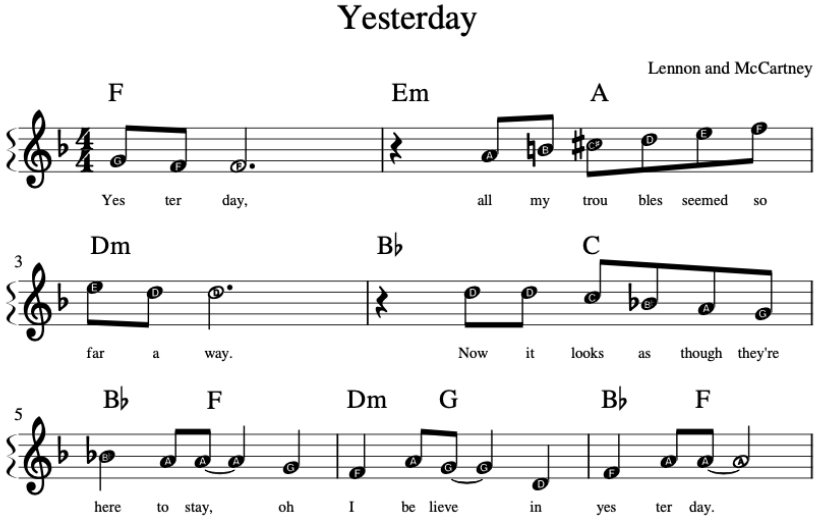

Figure 2.1: An example of a lead sheet for ‘Yesterday’ by the Beatles.

Lead sheets are most notably used in jazz music where their origins lie. They date back to the mid 20th century, and were originally called ‘fake sheets’ because they were used by musicians to ‘fake’ their way through a song [21]. Any jazz musician worth their salt owns a ‘real book’, so called to distinguish it from the fake books that were used in

the past. Real books contain lead sheets for hundreds of jazz standards, and are still an essential tool for any seasoned jazz musician.

More recently they have served as a useful tool for musicians in other genres too, such as pop and rock, who want to learn and perform songs quickly and easily. They allow efficient communication of the important elements of a song, without the need for a full score, encouraging further improvisation and personalisation of the song.

Lead sheets do not contain all the information of a full score, such as dynamics, articulation, or specific voicings of chords. Furthermore, they are not suitable for all types of music, such as classical music, where a full score is necessary to convey the composer's intentions, or rap music where the lyrics and beat are normally the most important elements. However, they are a useful tool for many musicians, and are a common way of representing music for both learning and performing.

2.1.2 Automatic Music Transcription

Automatic Music Transcription (AMT) is a field within Music Information Retrieval (MIR) that aims to construct models that convert musical audio into symbolic representations. [2] provide a comprehensive review of different forms of transcription. For lead sheet transcription, we are interested in their highest level of transcription: notation-level. We want to be able to take an audio recording of a song and generate a lead sheet that contains the melody and harmony of the song. This is a challenging task, as it requires the model to isolate and transcribe a melody, transcribe the chordal information, and then combine these two elements into a coherent lead sheet where the melody and harmony are aligned in time/beat.

There are three sub-fields of musical transcription that are of interest to lead sheet generation: Automatic Chord Recognition (ACR), Melody Transcription (previously called F0 estimation) and Beat Detection.

2.1.3 Music Features

Recorded music can be represented in a variety of ways as input to a machine learning model. The simplest way is to leave the data as a waveform: a time-series vector of amplitudes. Data in the raw audio domain has seen successful use in generative models such as Jukebox [12] and RAVE [5].

A very commonly used representation of general audio data, and specifically musical data, is the spectrogram. A spectrogram is a conversion of the time-series data into the time-frequency domain, calculated by a SFTF (short-time Fourier Transform) Spectrograms are commonly used in many audio processing tasks, such as speech recognition, music recognition [36] and music transcription, specifically polyphonic transcription [34]. However, as noted by Pauwels et al. [30], only logarithmic spectrograms have been used in ACR tasks, and linear spectrograms have been used in melody transcription tasks.

A common version of the spectrogram used in music transcription is the Constant-Q Transform (CQT), originally proposed by Brown [3]. The CQT is a version of a

spectrogram with frequency bins that are logarithmically spaced and bin widths that are proportional to the frequency. This is motivated by the logarithmic nature of how humans perceive pitch: a sine wave that is perceived as one octave higher than another has double the frequency. The CQT is used in many music transcription tasks, such as automatic chord recognition [17], and a popular alternative, Melspectrograms, have found use in melody transcription [34].

Chroma vectors are a 12-dimensional time-series representation, where each dimension corresponds to a pitch class. Each element represents the presence of each pitch class in the Western chromatic scale in a given time frame. Such features have been generated by deep learning methods [29] or by hand-crafted methods [26].

More recently, features extracted generative models have been used as input. The proposed benefit is that the vast quantities of data used to train these models allows for rich representations of the music. Chris Donahue and Liang [10] use features from JukeBox [12] to train a transformer [35] for both melody transcription and chord recognition.

2.2 Related Work

2.2.1 Lead Sheet Transcription

Older work has attempted to automatically produce lead sheets [32, 37]. They use hand-crafted feature extractions and probabilistic models for melody and chord classification. AMT has developed dramatically since this work which leaves much room for improvement.

[10] is the only recent work we found that produces a full lead sheet generation model. They propose the use of features extracted from middle layers if Jukebox [12] which were found to lead to the best performance in downstream tasks [7]. These features are used to train a transformer, with a primary focus on melody transcription. The model and code is available[1]. They use the same methodology to train an ACR model, and combine these with beat detection and engraving software to produce a lead sheet.

The authors found the use of these generative pre-trained features to be beneficial for melody transcription. However, they use community-submitted annotations from an online forum, presumably for lack of a better alternative, whose quality and variety may be lacking. The nature of this data means that the model is only trained on 24 second audio clips, which limits performance on longer segments of audio. The authors claim that the overall model performs well, especially in the verses and choruses of pop music where vocal lines are loud and clear, and that performance across genres is surprisingly good given the pop-centric dataset. However, further analysis is ommitted and if the dataset contains music with few songs with more complex harmonic structures such as jazz music, then the model will produce overly simplistic representations of the chords. Crucially, the work does not explore ACR models beyond simply copying their method and dataset used in melody transcription and there is no quantitative evaluation of

[1] https://github.com/chrisdonahue/sheetsage/tree/main

performance in ACR. Furthermore, the system can struggle with melody lines shifting between instruments, quiet vocal lines

2.2.2 Automatic Chord Recognition

[30] provides an overview of ACR since its inception in 1999 with the work of Fujishima [14] up to 2019, and provide suggestions for future avenues of research. Among them are the lack of exploration of feature and chord representations and the imbalance with chords classes present in chord datasets.

Datasets that have seen common use in ACR over the last decade include:

- *Mcgill Billboard*: 890 chord annotations of songs randomly selected from the Billboard 'Hot 100' Chart bnetween 1958 anmd 1991. [4]
- *Isophonics*: 300 annotations of songs from albums by The Beatles, Carole King and Zweieck. [6]
- *RWC-Pop*: 100 pop songs with annotations available[2] for chords. [15]
- *USPop*: 195 annotations of a larger dataset with artists chosen for popularity. [1]
- *JAAH*: 113 annotations of a collection of jazz recordings. [13]

Other, small datasets, also exist, and have been compiled together into the *Chord Corpus* by de Berardinis et al. [11], with standardised annotation formats.

A variety of machine learning methods have been applied to such datasets. Older methods such as [24] use HMMs, but more recent methods use deep learning. Often, CNNs and RNNs are used in combination [38, 22, 27] with CNNs performing feature extraction from a spectrogram feature, and an RNN (normally Bi-LSTM or Bi-GRU) predicts chord sequences. More recently, transformers have been applied to the entire process Chris Donahue and Liang [10] and Chen and Su [8, 9].

Evaluation is typically done using accuracy and recall of correct chord predictions, or more music-aware measures such as the correct root note, 3rd or the MIREX metric which measures that chords have at least 3 notes in common. These are all implemented by Raffel et al. [31] in the `mir_eval` library [3]. Qualitative evaluation is also often carried out.

A big problem frequently encountered in ACR is the lack of labelled data. This work uses 1200 labelled songs with audio. This is due to the difficulty and time associated with labelling data aligned in time and the legal sensitivity of the data involved.Data has been scaled up using augmentation and semi-supervised learning [20] with some success. Research has been done into the use of synthetic data [23, 33] and supervised learning [25] for MIR tasks, but not for ACR.

Another problem is that the existing data is often imbalanced, with a large number of common chords like major and minor chords and fewer chords like diminished and

[2] `https://github.com/tmc323/Chord-Annotations`
[3] `https://mir-evaluation.github.io/mir_eval/`

augmented chords, or chords with upper extensions and inversions. This can lead to models that are biased towards predicting major and minor chords. Attempts to address this have been made by re-weighting classes [22] or adjusting the sampling of training examples to balance chord classes [29].

Chapter 3

Experimental Setup

This chapter outlines the datasets used in this work, the preprocessing applied to the audio and chord annotations, the evaluation metrics used to compare the models and details of the training process used throughout.

3.1 Datasets

Two ACT datasets are used in this work. The first dataset is simply referred to as the '*Pop* dataset, as much of the music in the dataset comes from the Billboard Hot 100 charts, or other popular bands. The second dataset is the *JAAH* (Jazz Annotations and Analysis of Harmony) dataset mentioned in Section 2.2.2. While the chords annotations are publicly available on Github[1], the audio was kindly given to me by Andrea Poltronieri, a PhD student at the University of Bologna and the author of 'ChoCo' (the Chord Corpus).

This rest of this chapter explains processing applied to the audio and chord annotations common to both datasets, before discussing the details of both the *Pop* and *JAAH* datasets in detail.

3.1.1 Preprocessing

3.1.1.1 Audio to CQT

The audio was first converted to a Constant-Q Transform (CQT) representation explained in Section 2.1.3. The CQT is computed using the `librosa` library [28], using the `.cqt` function. A sampling rate of 44100Hz was used, with a hop size of 4096, and 36 bins per octave, 3 octaves and a fundamental frequency corresponding to the note `C1`. This returns a complex-valued matrix containing phase, frequency and amplitude information. Phase information is discarded by taking the absolute value, before being converted from amplitude to dB, equivalent to taking the log. These default parameters were chosen to be consistent with previous works [27]. However, the hop size was varied in order to investigate the effect of different frame lengths on performance.

[1] https://github.com/smashub/choco

To save on computational time, the CQT was pre-computed into a cached dataset rather than re-computing each CQT on the fly on every run. This was done for each hop size used.

3.1.1.2 Chord Annotations

The chord annotations are represented as a sorted list of observations, each containing the chord, start time and duration. The chord itself is represented as a string in Harte notation [16]. For example, C major 7 is `C:maj7` and A half diminished 7 in its second inversion `A:hdim7/5`. However, chords in the dataset are not this complicated. The notation also includes `N` representing no chord.

This flexible annotation however is far too flexible. This would lead to thousands of chords. Instead, we define two chord vocabularites. The first is a simple chord vocabulary, which contains only major, minor for each root and a no chord symbol `N`. Then this vocabulary has 25 labels. Chords outside the vocabulary are mapped to `N`. For example, `C:maj7` would be mapped to `C:maj`, while `A:hdim7/5` would be mapped to `N`. The second is a more complex vocabulary, which contains 14 qualities for each root: major, minor, diminished, augmented, minor 6, major 6, minor 7, minor-major 7, major 7, dominant 7, diminished 7, half diminished 7, suspended 2, suspended 4. Additionally there is a no chord `N` and a dedicated out-of-vocabulary chords `X`, leading to 170 labels. This vocabulary or very similar vocabularies have been used by the literature [27, 18, 22**?**].

Chords are mapped by first converting to a set of pitch classes, tranposing these such that 0 is the root, and matching pitch classes to a list of templates identifying the quality of the chord, as decribed in [27]. However, for some chords this was not sufficient. For example, a `C:maj6(9)` chord would not fit perfectly with any of these templates due to the added 9th. Therefore, the chord is also passed through Music21's chord quality function [**?**] which matches chords such as the one above to major. This function would work alone however as its list of qualities is not as rich as the one defined above. If the chord is still not matched, it is mapped to `X`.

3.1.2 Pop

The *Pop* dataset consists of songs from the *Mcgill Billboard*, *Isophonics*, *RWC-Pop* and *USPop* datasets mentioned in Section 2.2.2. This collection was originally proposed in work by Humphrey et. al [18] in order to bring together most of the known datasets for chord recognition. The dataset consists of 1,217 songs, filtered for duplicates and selected for those with annotations available. The dataset was provided with obfuscated filenames and audio as `.mp3` files, and annotations as `.jams` files [19].

3.1.2.1 Data Integrity

Several possible sources of error in the dataset were investigated.

Duplicates: Files were renamed using provided metadata identifying them by artist and song title. This was done to identify duplicates in the dataset. The dataset was

first filtered for duplicates, of which there was one - Blondie's 'One Way or Another', which had two different versions. The duplicate was removed from the dataset. Further duplicates may exist under different names but the songs were listened to throughout the project and no othe rduplicates were found. Automatic analysis of the audio could help, although cannot be guaranteed to find different versions of the same song, as above.

Chord-Audio Alignment: 10 songs were manually investigated for alignment issues. This was done by listening to the audio and comparing it to the annotations directly at various points in the song, with a focus on the beginning. It became apparent that precise timings of chord changes are ambiguous. The annotations aired on the side of being slightly early but were all certainly good annotations, with detailed chord labelings including inversions and upper extensions. This observation was borne in mind later when analysing performance of the models at transition frames in Section [XX].

Automatic analysis of the alignment of the audio and chord annotations was also done using cross-correlation of the derivative of the CQT features of the audio over time and the chord annotations, varying a time lag. A maximum correlation at a lag of zero would indicate good alignment as the audio changes at the same time as the annotation changes. First, the CQT of the audio was computed following the procedure defined in Section 3.1.1.1. The derivative of the CQT in the time dimension was then estimated using `librosa`'s `librosa.feature.delta` function. The chord annotations were converted to a binary vector, where each element corresponds to a frame in the CQT, and is 1 if a chord change occurs at that frame, and 0 otherwise. Both the CQT derivatives and binary vectors were normalised by subtracting the mean and dividing by the standard deviation. Finally, cross-correlation was computed using `numpy`'s `numpy.correlate` function. A typical cross-correlation for a song is shown in Figure **??**.

We can see that the cross-correlation repeats every 100 frames or so. Listening to the song, we can interpret this as 4 bars repeating. In particular, the transients of the drum beat are likely to be very similar every 4 bars. To check alignment across the dataset, we can plot the lag of the maximum cross-correlations as a histogram. This is shown in Figure **??**. If we assume that the annotations are not incorrect by more than approximately 10 seconds, this would mean we could restrict our maximum correlation search to a window of 100 frames either side of 0. A second histogram, with the maximums reduced to a 100 frame window around 0 is shown in Figure **??**. XX We see that...

A final simple check was done by looking at the lengths of the audio and chord annotations. A histogram of difference in length is shown in Figure **??**. The majority of songs have a difference of 0, with a few having a difference of 1. There are some outliers, but not enough to warrant further investigation.

Incorrect and Subjective Annotations: Throughout manual listening, no obviously wrong annotations were found. However, looking at songs which the first trained models perform the worst on, 'Lovely Rita' by the Beatles sticks out. The model consistently guessed chords one semitone off, as if it thought the song was in a different key. Upon listening, it became clear that recording's tuning was not in standard A440 and so the song was removed. No other songs were found to have such issues.

Chord annotations are inherently subjective to some extent. Detailed examples in this dataset are given by Humphrey et. al [18]. They also note that there are several songs in the dataset with questionable relevance, as the music itself is not well-explained by chord annotations. However, these are kept in for consistency with other works as this dataset is often used in the literature. Some works [27] decide to use the median as opposed to the mean accuracy in their evaluations in order to counteract the effect of such songs.

3.1.2.2 Chord Distribution

Much of the recent literature has focused on the long tail of the chord distribution [XX], using a variety of methods to attempt to address the issue. It is first helpful to understand the distrbution of chords in the datasets, shown in Figure **??**. The distribution is broken down both by root and by quality, only for the larger chord vocabualary as the smaller vocabualary does not have the same long-tailed distribution. We can see that [XX]..

3.1.3 JAAH Dataset

I was warned by Andrea that the JAAH dataset has not been as commonly used as dataset the Billboard dataset. Therefore he could not guarantee that the audio was aligned for this dataset.

- As yet, JAAH is unused in this work - Data was received as .flac files which were first converted to .mp3 to be in line with the Billboard dataset - Comparison of the two datasets - Description of the JAAH dataset and its use in this work - Intended to be used as a test set to test the synthetic data generation.

3.2 Evaluation

3.3 Training

Chapter 4

Model Comparison

In this section, we compare various model in the *Pop* dataset. We start by describing

4.1 Baseline

- Simple logistic regression, single-layer NN.

4.2 CRNN

- Describe simple model from CQT through CNNs and RNNs - Transition frame analysis
- Incorrect region analysis - Hop length - Re-weighting loss

4.3 Structured Training

4.4 Transformer

- Transformer expansion

4.5 Using Generative Features

- Using generative features

4.6 Comparison

Chapter 5

Synthetic Data Generation

5.1 Motivation

- The need for more data

5.2 Generation method

- Generation method

5.3 Experiments

- Brief description of the experiments and metrics I'm looking at

5.4 Results

- Comparison on normal data - Comparison on JAAH

Chapter 6

Conclusions

Bibliography

[1] Adam Berenzweig, Beth Logan, Daniel P. Ellis, and Brian Whitman. A large-scale evaluation of acoustic and subjective music-similarity measures. *Computer Music Journal*, 28(2):63–76, 2004.

[2] Bhuwan Bhattarai and Joonwhoan Lee. A comprehensive review on music transcription. *Applied Sciences*, 13:11882, 10 2023. doi: 10.3390/app132111882.

[3] Judith Brown. Calculation of a constant q spectral transform. *Journal of the Acoustical Society of America*, 89:425–, 01 1991. doi: 10.1121/1.400476.

[4] John Burgoyne, Jonathan Wild, and Ichiro Fujinaga. An expert ground truth set for audio chord recognition and music analysis. pages 633–638, 01 2011.

[5] Antoine Caillon and Philippe Esling. Rave: A variational autoencoder for fast and high-quality neural audio synthesis, 2021. URL https://arxiv.org/abs/2111.05011.

[6] Chris Cannam, Craig Landone, and Mark Sandler. Omras2 metadata project. In *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR)*, pages 309–310, 2009.

[7] Rodrigo Castellon, Chris Donahue, and Percy Liang. Codified audio language modeling learns useful representations for music information retrieval, 2021. URL https://arxiv.org/abs/2107.05677.

[8] Tsung-Ping Chen and Li Su. Harmony transformer: Incorporating chord segmentation into harmony recognition. In *International Society for Music Information Retrieval Conference*, 2019. URL https://api.semanticscholar.org/CorpusID:208334896.

[9] Tsung-Ping Chen and Li Su. Attend to chords: Improving harmonic analysis of symbolic music using transformer-based models. *Trans. Int. Soc. Music. Inf. Retr.*, 4:1–13, 2021. URL https://api.semanticscholar.org/CorpusID:232051159.

[10] John Thickstun Chris Donahue and Percy Liang. Melody transcription via generative pre-training, 2022. URL https://arxiv.org/abs/2212.01884.

[11] Jacopo de Berardinis, Albert Meroño-Peñuela, Andrea Poltronieri, and Valentina Presutti. Choco: a chord corpus and a data transformation workflow for

musical harmony knowledge graphs. *Scientific Data*, 10, 09 2023. doi: 10.1038/s41597-023-02410-w.

[12] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music, 2020. URL https://arxiv.org/abs/2005.00341.

[13] Gabriel Durán and Patricio de la Cuadra. Transcribing Lead Sheet-Like Chord Progressions of Jazz Recordings. *Computer Music Journal*, 44(4):26–42, 12 2020. ISSN 0148-9267. doi: 10.1162/comj_a_00579. URL https://doi.org/10.1162/comj_a_00579.

[14] Takuya Fujishima. Realtime chord recognition of musical sound: a system using common lisp music. In *International Conference on Mathematics and Computing*, 1999. URL https://api.semanticscholar.org/CorpusID:38716842.

[15] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka. Rwc music database: Popular, classical, and jazz music databases. 01 2002.

[16] Christopher Harte, Mark Sandler, Samer Abdallah, and Emilia Gómez. Symbolic representation of musical chords: A proposed syntax for text annotations. pages 66–71, 01 2005.

[17] 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*, volume 2, pages 357–362, 2012. doi: 10.1109/ICMLA.2012.220.

[18] Eric J. Humphrey and Juan Pablo Bello. Four timely insights on automatic chord estimation. In *International Society for Music Information Retrieval Conference*, 2015. URL https://api.semanticscholar.org/CorpusID:18774190.

[19] Eric J. Humphrey, Justin Salamon, Oriol Nieto, Jon Forsyth, Rachel M. Bittner, and Juan P. Bello. Jams: A json annotated music specification for reproducible mir research. pages 591–596, 2014. 15th International Society for Music Information Retrieval Conference, ISMIR 2014 ; Conference date: 27-10-2014 Through 31-10-2014.

[20] Yun-Ning Hung, Ju-Chiang Wang, Minz Won, and Duc Le. Scaling up music information retrieval training with semi-supervised learning, 2023. URL https://arxiv.org/abs/2310.01353.

[21] 99 The real book, 2021. URL https://podcasts.apple.com/us/podcast/the-real-book/id394775318?i=1000519252462.

[22] Junyan Jiang, K. Chen, Wei Li, and Gus G. Xia. Large-vocabulary chord transcription via chord structure decomposition. In *International Society for Music Information Retrieval Conference*, 2019. URL https://api.semanticscholar.org/CorpusID:208334209.

[23] Nadine Kroher, Helena Cuesta, and Aggelos Pikrakis. Can musicgen create training data for mir tasks?, 2023. URL https://arxiv.org/abs/2311.09094.

[24] Kyogu Lee and Malcolm Slaney. Automatic chord recognition from audio using a hmm with supervised learning. pages 133–137, 01 2006.

[25] Yizhi Li, Ruibin Yuan, Ge Zhang, Yinghao Ma, Xingran Chen, Hanzhi Yin, Chenghao Xiao, Chenghua Lin, Anton Ragni, Emmanouil Benetos, Norbert Gyenge, Roger Dannenberg, Ruibo Liu, Wenhu Chen, Gus Xia, Yemin Shi, Wenhao Huang, Zili Wang, Yike Guo, and Jie Fu. Mert: Acoustic music understanding model with large-scale self-supervised training, 2024. URL https://arxiv.org/abs/2306.00107.

[26] Matthias Mauch and Simon Dixon. Approximate note transcription for the improved identification of difficult chords. pages 135–140, 01 2010.

[27] Brian McFee and Juan Pablo Bello. Structured training for large-vocabulary chord recognition. In *International Society for Music Information Retrieval Conference*, 2017. URL https://api.semanticscholar.org/CorpusID:3072806.

[28] Brian McFee, Colin Raffel, Dawen Liang, Daniel Ellis, Matt Mcvicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. pages 18–24, 01 2015. doi: 10.25080/Majora-7b98e3ed-003.

[29] Jeff Miller, Ken O'Hanlon, and Mark B. Sandler. Improving balance in automatic chord recognition with random forests. In *2022 30th European Signal Processing Conference (EUSIPCO)*, pages 244–248, 2022. doi: 10.23919/EUSIPCO55093.2022.9909558.

[30] Johan Pauwels, Ken O'Hanlon, Emilia Gómez, and Mark B. Sandler. 20 years of automatic chord recognition from audio. In *International Society for Music Information Retrieval Conference*, 2019. URL https://api.semanticscholar.org/CorpusID:208334309.

[31] Colin Raffel, Brian McFee, Eric J. Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, and Daniel P. W. Ellis. Mir_eval: A transparent implementation of common mir metrics. In *International Society for Music Information Retrieval Conference*, 2014. URL https://api.semanticscholar.org/CorpusID:17163281.

[32] Matti P. Ryynänen and Anssi P. Klapuri. Automatic transcription of melody, bass line, and chords in polyphonic music. *Computer Music Journal*, 32(3):72–86, 2008. doi: 10.1162/comj.2008.32.3.72.

[33] Gakusei Sato and Taketo Akama. Annotation-free automatic music transcription with scalable synthetic data and adversarial domain confusion, 2024. URL https://arxiv.org/abs/2312.10402.

[34] Keisuke Toyama, Taketo Akama, Yukara Ikemiya, Yuhta Takida, Wei-Hsiang Liao, and Yuki Mitsufuji. Automatic piano transcription with hierarchical frequency-time transformer, 2023. URL https://arxiv.org/abs/2307.04305.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

[36] Avery Wang. An industrial strength audio search algorithm. 01 2003.

[37] Jan Weil, Thomas Sikora, Jean-Louis Durrieu, and Gaël Richard. Automatic generation of lead sheets from polyphonic music signals. pages 603–608, 01 2009.

[38] Yiming Wu, Tristan Carsault, and Kazuyoshi Yoshii. Automatic chord estimation based on a frame-wise convolutional recurrent neural network with non-aligned annotations. In *2019 27th European Signal Processing Conference (EUSIPCO)*, pages 1–5, 2019. doi: 10.23919/EUSIPCO.2019.8902741.

Appendix A

First appendix

A.1 First section