Automated Accompaniment Generation for Vocal Input

M. J. N. Jayaweera
2018

Automated Accompaniment Generation for Vocal Input

A dissertation submitted for the Degree of Master of Computer Science

M. J. N. Jayaweera
University of Colombo School of Computing
2018

Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: M. J. N. Jayaweera

Registration Number: 2015/MCS/036

Index Number: 15440365

Signature: Date:

This is to certify that this thesis is based on the work of

Mr./Ms.

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name:

Signature: Date:

Abstract

Music composition has become a popular area in the present. Melody of music is considered as the key element and forms the theme of the music. Therefore usually, a composer initially comes up with an idea for a melody, and then the music accompaniment is created in order to form up a piece of music or a song. However, creating appropriate accompaniment with proper chord progressions is definitely a complex task for non-musicians. Thus, automated accompaniment generation for vocal melodies can be very useful. A system which is capable of generating accompaniment for vocal melodies is presented in this thesis. The vocal input is taken as the input to the system via a microphone and it undergoes several modules in order to detect the proper chord progression for the melody. Initially, the vocal melody is divided into chunks based on the tempo in beats per minute selected by the user. A tone detection algorithm is used in order to detect the notes being sung and the associated energies of those notes. A dataset is prepared by analyzing popular songs that use simple chord progressions. The data set is trained by using a multi target learning and evaluation module which is used to predict the proper chord progression. Once the chord progression is identified, a simple accompaniment is assigned at each chord detected area. The experiments are carried out by obtaining vocal melodies from different users for a specific song. The results of experiments are evaluated by comparing the original simple chord progression of the selected song, with the chord progression generated by the system for each vocal melody. These findings are useful to develop an accurate system for accompaniment generation and to enhance the functionalities of existing systems related to the music industry.

Acknowledgements

I would like to express my deepest appreciation to all those who provided me the possibility to complete this report.

My deepest appreciation to the project supervisor, Dr. K. L. Jayaratne whose contribution in stimulating suggestions and encouragement, providing crucial information and for evaluating us.

A special gratitude for all the lecturers for providing me with the knowledge to complete this project successfully.

My deepest indebtedness to my parents who supported me in countless means over these years.

Finally my thanks to all the colleagues and the people who supported me with their finest abilities over the duration of the project.

Table of Contents

List of figures

List of tables

List of Abbreviations

FLWT - Fast Lifting Wavelet Transform

FFT - Fast Fourier Transform

CPU - Central Processing Unit

BPM - Beats Per Minute

GUI - Graphical User Interface

CR - Class Relevance

1. Chapter 1: Introduction

Currently, music industry has become a very popular area among everyone. People are interested and involved in the music industry as an occupation, as they are inspired by music and as a hobby. In addition, some involve in music in order to deliver a message through music, as it can be said that "music speaks louder than words". People who are inspired by music are rather interested in composing new songs. This is due to the unique identity attached to person who compose the song. However, engaging in this trend to compose music has different barriers.

Therefore, this chapter will briefly discuss the motivation, the constraints or problems arise when composing music and the automated system developed as a mean of providing a solution to overcome those key problems. In addition, this chapter also illustrates the aim, objectives and the scope of the research.

1.2. Motivation

Music composition is the process of forming a piece of music by combining the elementary parts of music. The basic elements of music are pitch, rhythm melody, dynamics, timbre and texture.

- Pitch – A series of notes that forms the scale of the music.
- Rhythm – The time element of the music.
- Melody – Series of notes which forms the theme of the music.
- Dynamics – The loudness and softness of the music and how they are organized.
- Timbre – Tone color or sound quality.
- Texture - Indicates the number of individual musical melodies.

Since the melody of music is the key element and forms the theme of the music, a composer often comes up with an idea for a melody, and then the accompaniment patterns and chord progressions are formed in order to convert it into a piece of music. However, this requires a vast knowledge of musical theory and practical experience as well. The melody is often experienced by using a musical instrument in order to form the proper chord progressions for the music. Therefore, creation of appropriate instrumental accompaniment for a melody line is definitely a difficult task for people who are unable to play a musical instrument and who are lacking in knowledge in the area of music theory. Since instrumental accompaniment plays a significant role in forming a melody, automated accompaniment generation thus can be very useful for those who are interested in singing and music composition.

1.3. Statement of Problem

The system mainly targets the audience who are willing to sing but have a limited knowledge on the subject of composing music. Although these people have a thorough idea on the melody in their mind and possess the talent to sing, they would have to interact with a musician in order to compose a song for them. The inability to perform such tasks and the cost that would be incurred if performed by other musicians demotivates such talented individuals.

Different barriers existing to compose unique melodies for individuals who possess the inspiration as well as the talent can be listed as follows;

- Knowledge barriers on musical theory and practical experience.
- Knowledge barriers on how chords are formed.
- Time barriers as it takes a long time to learn an instrument.
- Cost barriers to purchase a musical instrument.

Therefore, the above mentioned barriers will restrain talented individuals from composing songs. This will cause a negative impact in the music industry as the industry is unable to attract talented individuals.

As a solution to the above problem, a system which is capable of composing music by generating accompaniment automatically based on the characteristics of vocal input will be very useful especially for these individuals who are restricted with the above mentioned barriers. Hence, this solution will surely be a driving force to speed up the attraction of talented individuals in the industry.

1.4. Aim

The aim of the project is to build a system which is capable of generating musical accompaniment automatically, based on vocal melodies recorded via microphone.

1.5. Objectives

- Study different pitch detection algorithms in order to determine the best algorithm for detecting the note frequencies.
- Analyze different multi target classification modules for determining the best module in order to predict chords.

1.6. Scope

The proposed solution contains many sub modules. Each module is interconnected and the output from one model is used as the input of the other.

- Sampling the audio signal
- Pitch Detection
- Chord Estimation
- Accompaniment Assignment

The system allows the user to sing or hum a melody through a microphone along with a selected rhythm which is used to extract pitch notes. Base on the pitch notes the most probable chords are identified by using a statistical model and finally a simple piano accompaniment is assigned to the identified chords.

1.7. Structure of the Thesis

The thesis first illustrates the related work carried out on the selected topic. Next, the design of the system that is followed in order come up with the solution for the defined problem in the introduction chapter is described. The implementation of the system that follows the design is illustrated next. After the implementation, the evaluation and the test results are presented. Finally the conclusion and the improvements that can be done in order to develop a more accurate system is discussed.

2. Chapter 2: Background / Literature Review

A piece of music is usually is a combination of a vocal melody and an instrumental accompaniment. The melody of a music piece consists of a series of notes and considered as the core of the music piece. Therefore, a composer often starts with the melody part of the music in order to compose the music piece. The melody is decorated with series of chords which will form the accompaniment. A chord consists of multiple notes played at the same time by a musical instrument. However, composing a music piece is definitely a difficult task for the people who are unable to play a musical instrument and who are lacking of musical knowledge. Thus, automated accompaniment generation for a vocal melodies can be useful as musical accompaniment plays an important role when it comes to music composition. Several researches have been conducted in order to detect chords and to automate the accompaniment generation for a melody. This section describes the related works carried out by several parties and the evaluation on how the proposed system differs from those work.

The research conducted by Ziheng Chen, Jie Qi and Yifei Zhou focuses on machine learning in automatic music chords generation. The aim of the research was to learn the relationship between notes, chords and adjacent chords and assign chords to a melody. In here an algorithm is proposed that takes a music piece with several measures as the input and several models such as Logistic regression, Naive Bayes, Support Vector machine, Random Forest, Boosting and Hidden Markov Model are used to predict chords which is then analyzed and compared [1].

A dataset of forty three lead sheets are collected with several properties. Each measure in the data set contains a single chord and these chords are basic common chords which are used in music pieces. In addition, the data are in Music XML format which is a digital sheet music representation in western music. Next, the dataset is sent through a preprocessing stage. Initially in the preprocessing stage, all forty three songs are transposed in to C major. In addition, the chord types are restricted to simple chords in C major. Next, the measures in the dataset which didn’t have chords or notes are deleted. Finally, if there are two continuous chords assigned in the measures of the dataset, the second chord is considered as the chord of the measure and the first one is deleted. After the preprocessing stage the dataset is used to extract features for computation. The first feature which is taken into consideration is whether a note is presented in the measure. Next, the beat with the notes are considered in order to identify which notes tally with the beat in the measure as the chords should accompany the beat tones. Finally, the duration with the notes are considered in order to identify how long that a chord should be presented.

The results show that out of the models which are been used, the Random forest and Boosting perform the best results. However, Hidden Markov Model also shows a good result if more information is included in the measure. But the highest accuracy which can be achieved is only about 70% [1]. One of the major limitation of this research is the accompaniment is always generated in C major scale. In addition, a music piece with several measures are used as the input to each model initially.

University of Passau Faculty of Computer Science and Mathematics have conducted a research to illustrate how mathematic signal processing and music theory can be combined together in order recognize chords. In here, the theory of Fourier analysis is taken into consideration in order to identify how it can be used to extract important frequencies from audio signals, how these frequencies are related to the pitch and how the chords are generated from the pitch information. Finally a simple and accurate algorithm is proposed for chord detection emphasizing on the detection quality.

The proposed chord recognition algorithm was introduced, elaborating on a number of design decisions that were made such as the focus on a system that requires no training and a restriction to 24 major/minor chords [2]. The algorithm initially takes uncompressed mono PCM audio data from an entire song as the input and outputs estimated chords and their durations. In here, 24 major and minor chords are considered as these chords are commonly used in popular music. The input to the algorithm firstly is sampled at a rate of 48000 Hz and the input data is computed with a window size of 16384 and overlap factor of 8 by using Blackman-Harris window. Next, the frequencies are used to determine which notes are played and extract the harmonic content of the input. In addition, each detected note is assigned to a pitch class. Next, the chord probabilities are derived and finally the chord sequence is estimated. One limitation is the learning phase is removed from the system in order to improve the detection quality.

Work of Matthias Mauch, Katy Noland and Simon Dixon has shown that the chord recognition can be improved by using the knowledge of repeating structure in a song. In here, a novel Dynamic Bayesian Network (DBN) is presented that integrates metric position, chord, key, base note and two beat synchronous audio features into a single music context model. The most probable sequence of keys, metric positions, base notes and chords are inferred simultaneously via Viterbi inference. The chord confusion is improved by modifying the front end processing of the audio. A comparison is done between the effect of learning chord profiles as Gaussian mixtures and the effect of using chromagrams generated from an approximate pitch transcription method. Next, the chromagram information is distributed between the repeated structural segments such as verses in a song using a novel structural segmentation algorithm. The proposed, most complex method transcribes chords with a state-of-the-art accuracy of 73% on the song collection used for the 2009 MIREX Chord Detection tasks [3]. In addition, the use of the chromagrams from approximate transcription has shown a great accuracy.

Work of Roger B Dannenberg focuses on an on-line algorithm for real time accompaniment based on a score fed into the system containing the parts of a soloist. The system follows a soloist and melody is matched against the expected score fed into the system. The matching is tolerant with the mistakes and errors of the performance by the soloist, and the best match between the performance inputs and the expected score is presented. In addition, the information related to timing is generated in order to synchronize the accompaniment with the soloist performance.

The system can be divided into three sub problems. They are, detection of the inputs from a soloist, matching process between the inputs and the expected score eliminating the errors and generation of the accompaniment. However the research more focuses on the matching process between the inputs and the expected score fed into the system and hence, an efficient dynamic programming algorithm is presented [4]. In order to find the best match, an integer matrix is computed where each row corresponds to an event in the score and each column corresponds to a detected performance event. In here, the previous values of the row and the column are taken into consideration while computing the best match in order to eliminate performance mistakes. The algorithm gives length of the best match by incrementally producing the results as the input becomes available. The position in the score of the current performance event is detected by keeping track of the length of the best match up to the current event. It is the largest value in the matrix yet computed. The performance of the algorithm is achieved, by considering a small window centered around the current score location. The matching process is then used to report the temporal location of the performer with respect to the score and this information is used to generate the accompaniment. The system was experimented using an AGO keyboard for input and using a digital synthesizer for the output. The scores for the soloist and the relevant accompaniment are read from a file into the system.

However, one limitation of the proposed algorithm is it tends to jump in ahead in the score when a wrong event is detected. It is not capable of identifying whether the soloist produced a wrong event skipped one or more events. Instead it jumps ahead in the score matching the future event. Another limitation of the system is the accompaniment and the score have to be fed into the system prior to the actual input. However, the algorithm tracks and matches the events performed by the soloist even when notes are omitted and when additional notes are played.

Ching-Hua Chuan and Elanie Chew have conducted a research on a hybrid system for style-specific accompaniment generation by constructing chord progression list from a MIDI melody based on neo-Riemannian transforms. The system first identifies which notes of a given melodic composition are most suitable for chords and then the relevant triads are assigned to each identified chord tone. These triads act as checkpoints at each bar of the melody. Next, using neo-Riemannian transforms the system constructs possible chord progressions between checkpoints and represents the alternate paths in a tree structure. Finally, the final chord progressions are generated by a Markov chain with learned probabilities for the neo-Riemannian transforms between checkpoints.

The system consists of three major sections. They are, chord tone determination, triad construction and check point setup and chord progression generation. The chord tone determination module is used to classify the given melodic composition into chord tones and none chord tones. Machine learning techniques are used to determine chord tones. The module consists of a Support Vector Machine (SVM) which is used to identify the chord tones for each bar of the melody given [5]. Machine learning is carried out considering seventeen attributes for each note in a bar in order to classify chord tones and none chord tones. The most important attributes are the pitch class, duration, upper neighbor, lower neighbor, third and fifth. Once the chord tones are identified, the triads are assigned to harmonize each bar. The triads are assigned by examining the notes in each bar which strongly support the triad of the chord. For example the notes C E and G are used to form the C major chord where E and G notes are considered as third and fifth notes of the triad. If the bar contains the third or fifth notes or both notes of the triad it forms the chord by combining the root note of the triad. If the chord tones cannot be harmonizes by any triad a seventh chord is assigned. In addition if the chord tone cannot be covered by any triad a compound chord is assigned. This is done for each chord tone identified for every checkpoint. Then a suitable path of chord progression is generated by examining the each pair of adjacent checkpoints by using neo-Riemannian transforms. It is a modal music analysis which represent each chord in terms of roman numerals. A tree structure is used in order to represent the possible chord progression between two checkpoints. Each node of the tress represents a chord. Child node represent the next chord resulted by neo-Riemannian transforms for the next adjacent bar. The height of the tree represents the number of bars between two check points. Once the tree structure is constructed all successful paths of the tree is examined and the likelihood of a path is calculated from the conditional probabilities in the Markov chain.

The results were experimented by considering four similar songs by Radiohead. The results were much efficient as it the chords generated were much similar to the original melody of each song. However the experiments were carried out only for the songs which are played in major scale. In addition, in certain cases the system has generated different chord progressions for the same melody. However, the major limitation of this system is a melodic composition has to be fed into the system initially.

Research	**Author**	**Aim**	**Findings**
Machine Learning in Automatic Music Chords Generation	Ziheng Chen, Jie Qi, Yifei	To learn the relationship between notes, chords and adjacent chords and assign chords to a melody and predict the most accurate model out of Logistic regression, Naive Bayes, Support Vector machine, Random Forest, Boosting and Hidden Markov Model for chord recognition.	Random forest and Boosting perform the best results. Limitation of this research is the accompaniment is always generated in constant (C major) scale. In addition, a music piece with several measures are used as the input to each model initially.
Design and Evaluation of a Simple Chord Detection Algorithm.	University of Passau Faculty of Computer Science and	To illustrate how mathematic signal processing and music theory can be combined together in order recognize chords using Fourier analysis.	The theory of Fourier analysis is taken into consideration in order to identify how it can be used to extract important frequencies from audio signals

	Mathematics	To propose a simple and accurate algorithm for chord detection emphasizing on the detection quality.	An algorithm is proposed for chord detection emphasizing on the detection quality The proposed algorithms supports limited number chords. The learning phase is removed from the system in order to improve the detection quality.
Using Musical Structure To Enhance Automatic Chord Transcription	Matthias Mauch, Katy Noland, Simon Dixon	To improve chord recognition using the knowledge of repeating structure in a song.	Results show that the method transcribes chords with a state of the art accuracy of 73%.
An On-line Algorithm for Real-Time Accompaniment	Dannenberg, R.	To propose an on-line dynamic programming algorithm for real time accompaniment based on a score fed into the system containing the parts of a soloist.	The algorithm tracks and matches the events performed by the soloist even when notes are omitted and when additional notes are played. One limitation of the proposed algorithm is it tends to jump in ahead in

			the score when a wrong event is detected. In addition, the algorithm is not capable of identifying whether the soloist produced a wrong event skipped one or more events. Instead it jumps ahead in the score matching the future event. The accompaniment and the score have to be fed into the system prior to the actual input.
A Hybrid System for Automatic Generation of Style Specific Accompaniment	Chuan, C.-H., Chew, E	To propose a hybrid system for style-specific accompaniment generation by constructing chord progression list from a MIDI melody based on neo-Riemannian transforms.	The results were much efficient as it the chords generated were much similar to the original melody of each song. The experiments were carried out only for the songs which are played in major scale. In certain cases the system has generated different chord progressions for the same melody. A melodic composition has to be fed into the system initially.

Table 1: Literature Review Analysis

The proposed system attempts to address the limitations of the related works carried out. The proposed approach differs from these related works by emphasizing on the popular music and aiming at non-musician users and entertainment purposes. Unlike, the work carried by Dannenberg listed in Table 1, the system does not follow a soloist by initially feeding the score into the system as it targets only the vocal inputs. Furthermore, in most cases of related work conducted, undergoes preprocessing stages like feeding scores and melodic compositions whereas the proposed system only requires sampling of the melody sung by the user. Initially, a pitch detection algorithm is used in order to detect the actual notes sung by the user before the chord assignment process. In here, the pitch detection algorithm falls under FFT category and is efficient than the one which is used as part of the work carried by University of Passau Faculty of Computer Science and Mathematics. Once the notes are detected, the scale of the sung melody is detected based on the notes and the associated energies of those notes. This improves the relevance between the chords and notes. Unlike the work carried out by Ziheng Chen, Jie Qi and Yifei, the proposed system considers every major scale in order to generate the proper chord progressions. Next the notes and the scale information is used in order to predict a proper chord progression. The proposed system consists of a learning phase unlike the work carried out by University of Passau Faculty of Computer Science and Mathematics. In here, a private training data set that consist of notes and the associated chords being played in popular songs is used. Therefore, this makes the system more coherent and relevant to the real input.

3. Chapter 3: Design and Analysis

3.1. System Description

This section provides the brief introduction and the design of the entire automatic accompaniment generation system for vocal input. Figure 1 illustrates the high level architecture along with the data flow of the system. The system consist of four main sub systems which are integrated to form the entire system. They are Sampler, Pitch Detector, Chord Estimator and the Accompaniment Assigner.

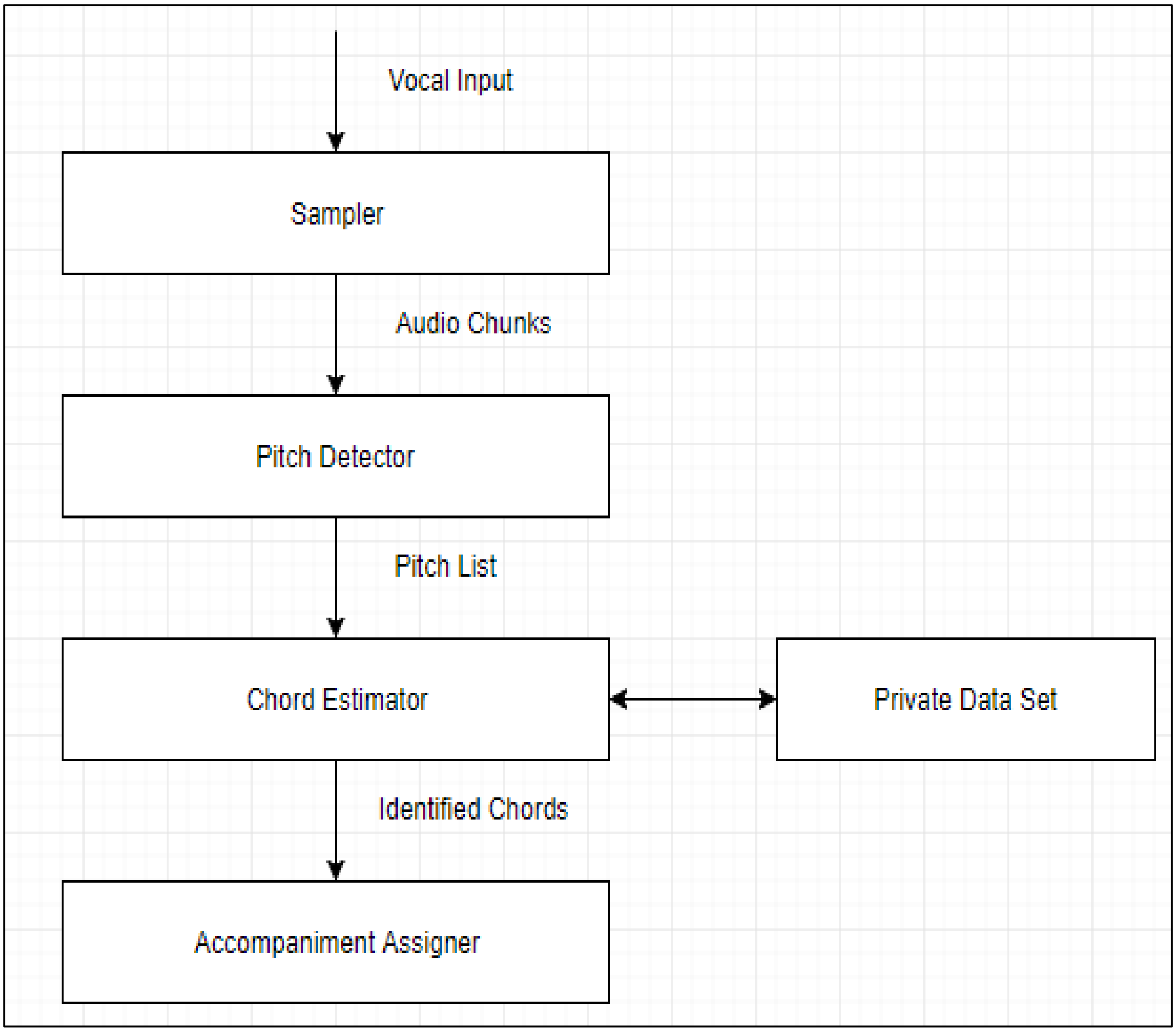


Figure 1: High Level Architecture of System

3.2. Sampler

Sampling is the process of converting the audio input into a digital representation. The numbers in the digital representation exhibit the magnitude of the audio signal at specific points in time and these are called samples. Initially the system allows the user to sing or hum a melody by using a microphone. In here, a metronome is played based on the tempo value selected by the user. This helps to synchronize the voice of the user with the metronome beat. Therefore the system can assume that the user is capable of singing into the correct tempo. During this process the input melody is sampled at a rate of 44.1 kHz and stored as a WAV file. The reason for selecting the 44.1 kHz as the sampling rate is because it is the audacity default setting and most recommended one for audio processing. Next the recorded WAV file is divided into several chunks of WAV files based on the tempo selected by the user. This is because the chords are normally assigned to a melody based on the beat bars in most of the popular songs that users simple chord progressions. Each chunk WAV file is used to detect the notes being sung by the user. Figure 2 illustrates a graphical view of how the WAV file is divided into chunk WAV files based on the tempo selected in BPM by the user.

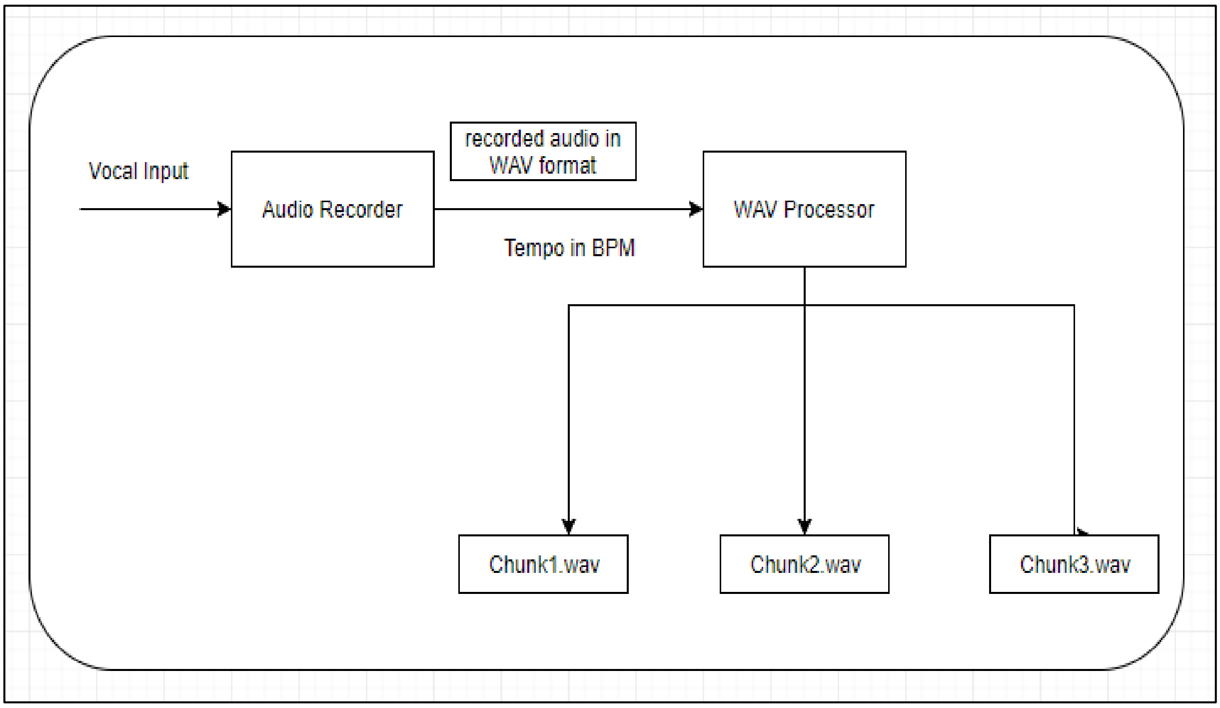


Figure 2: Wav File Chunking

3.3. Pitch Detector

This module is developed in order to identify the notes being sung by the user. In here, each chunk WAV file is used as an input to this module in order to detect the notes for every chunk. In this section, several pitch detection algorithms are taken into consideration. They are;

- YIN
- Dynamic Wavelet
- McLeod Pitch Method
- Goertzel Algorithm

3.3.1. YIN

This algorithm is presented for the estimation of the fundamental frequency (F0) of speech or musical sounds. It is based on the well-known autocorrelation method with a number of modifications that combine to prevent errors. [6]. YIN analyses the error mechanisms presented in the classic autocorrelation algorithm and provides certain improvements to reduce these errors. The results have shown that the error rates are three times lower than the best competing methods. However, YIN produces the pitch values mixed with noise frequencies and hence, a low pass filter is required initially in order to eliminate noise.

3.3.2. Dynamic Wavelet

Dynamic Wavelet is a real time pitch detection algorithm that emphasizes on low latency, high time resolution and accuracy [7]. This uses the implementation of Fast Lifting Wavelet Transform (FLWT) that splits the signal into high-pass and low-pass components and generating approximation and details respectively. The pitch of the vocal samples is detected by examining the peaks of the signal. Although this algorithm is useful in real time applications and shows some considerable amount of pitch detection accuracy still this tends to produce unexpected and errored results.

3.3.3. McLeod Pitch Method

The McLeod Pitch Method is a fast, accurate and robust method for finding the continuous pitch in monophonic musical sounds [8]. It makes use of a normalized version of the Squared Difference Function. It does not require low pass filtering for removing noise and can be used for real time pitch detection with a sampling rate of 44.1 kHz. In addition the algorithm does not require any post processing which is a common feature in other pitch detection mechanisms. Furthermore, the algorithm has introduced a mechanism of normalization that deals with edge problems in the signal unlike in algorithms that uses autocorrelation method. Although this algorithm is said to be accurate it takes considerable amount of time to produce results.

All of the above mentioned algorithms are suitable for real time pitch detection and can be best use for detecting a single defined frequency. Since it is developed to detect frequencies in real time fashion it is not much suitable for detecting frequencies for chunk WAV files which consist of a range of notes.

3.3.4. Goertzel Algorithm

The Goertzel Algorithm is an efficient algorithm that can be used to detect one or more predefined frequencies presented in a signal. It can perform tone detection using much less CPU horsepower than the Fast Fourier Transform, but many engineers have never heard of it [9]. For real-valued input sequences, it applies a single real-valued coefficient by using real-valued arithmetic at each iteration. The Goertzel Algorithm falls under the Fast Fourier Transform (FFT) algorithm category and it has a higher order of complexity than other FFT algorithms, but it efficiently computes a small number of selected frequency components as well as the magnitude presented for each frequency. The main advantage of Goertzel algorithm is that it is no longer required to round the detected frequencies unlike other pitch detection algorithms in order to detect the corresponding notes, thus obtaining more accurate results. It requires some preliminary parameters. They are;

1. The Sample rate
2. The block size
3. Cosine and Sine terms
4. Coefficient

In order to calculate the frequency presented in the signal it uses a simple threshold test of the magnitude. If the magnitude of a particular frequency is higher than the defined threshold, the frequency is said to be presented in the signal. The system uses a threshold of 35db.

By the analysis of above mentioned pitch detection methods, the Goertzel Algorithm is selected as it produces faster and accurate results compared to other methods. In addition, Goertzel Algorithm is best usable when finding predefined frequencies presented in the signal. In here, list of predefined frequencies, the audio buffer of each chunk and the sample rate are given as the inputs to the algorithm. It produces a series of frequencies and the energies of each frequency by taking the input parameters of each chunk into consideration. The identified list of frequencies are filtered based on a low pass and high pass filters. Each frequency has its corresponding music note. All possible notes that are used to determine the chords are identified by the use of table presented in Figure 3 [10].

Note	Hz	Note	Hz	Note	Hz	Note	Hz	Note	Hz	Note	Hz	Note	Hz
C1	32.7	C2	65.4	C3	130.8	C4	261.6	C5	523.3	C6	1046.5	C7	2093.0
C#1	34.6	C#2	69.3	C#3	138.6	C#4	277.2	C#5	554.4	C#6	1108.7	C#7	2217.5
D1	36.7	D2	73.4	D3	146.8	D4	293.7	D5	587.3	D6	1174.7	D7	2349.3
D#1	38.9	D#2	77.8	D#3	155.6	D#4	311.1	D#5	622.3	D#6	1244.5	D#7	2489.0
E1	41.2	E2	82.4	E3	164.8	E4	329.6	E5	659.3	E6	1318.5	E7	2637.0
F1	43.7	F2	87.3	F3	174.6	F4	349.2	F5	698.5	F6	1396.9	F7	2793.8
F#1	46.2	F#2	92.5	F#3	185.0	F#4	370.0	F#5	740.0	F#6	1480.0	F#7	2960.0
G1	49.0	G2	98.0	G3	196.0	G4	392.0	G5	784.0	G6	1568.0	G7	3136.0
G#1	51.9	G#2	103.8	G#3	207.7	G#4	415.3	G#5	830.6	G#6	1661.2	G#7	3322.4
A1	55.0	A2	110.0	A3	220.0	A4	440.0	A5	880.0	A6	1760.0	A7	3520.0
A#1	58.3	A#2	116.5	A#3	233.1	A#4	466.2	A#5	932.3	A#6	1864.7	A#7	3729.3
B1	61.7	B2	123.5	B3	246.9	B4	493.9	B5	987.8	B6	1975.5	B7	3951.1

Figure 3: Music Note Frequencies

3.4. Chord Estimator

The core module of the system is the chord estimator. A chord can be classified mainly as major or minor. A chord can be formed based on two considerations. They are the root and the harmony. There are twelve possible root notes. C, C#, D, D#, E, F, F#, G, G#, A, A# and B. Harmony of the root note is the rest of the notes which is being played in order to form the chord. For Example C has the harmonies of E and G. The pitch values for each chunk is used as the input and base on a statistical model the most probable chord combination is estimated. Before estimating the chord progressions, it is necessary to detect the scale of the recorded melody. In music theory, a scale can be defined as a set of musical notes ordered by fundamental frequency or pitch. In many cases the scale consists of seven notes and they are called major and minor scales. Most of the songs are sung either in major or minor scales. Initially, the scale or the key of the melody is obtained based on the notes and the energies generated by Goertzel algorithm. In music theory each scale has its own series of notes. Table 2 shows the notes associated with each major scale.

Scale	Notes
C major	C, D, E, F, G, A, B
C# major	C#, D#, F, F#, G#, A#, C
D major	D, E, F#, G, A, B, C#
D# major	D#, F, G, G#, A#, C, D
E major	E, F#, G#, A, B, C#, D#
F major	F, G, A, A#(B♭), C, D, E
F# major	F#, G#, A#, B, C#, D#, F
G major	G, A, B, C, D, E, F#
G# major	G#, A#, C, C#, D#, F, G
A major	A, B, C#, D, E, F#, G#
A# major	A#, C, D, D#, F, G, A
B major	B, C#, D#, E, F#, G#, A#

Table 2: Notes Associated With Scales

The notes and the magnitudes of each note detected by the Goertzel algorithm are compared with the notes of each scale in order to accurately identify the scale. Figure 4 represents a graphical view of notes and associated energies for detecting the scale.

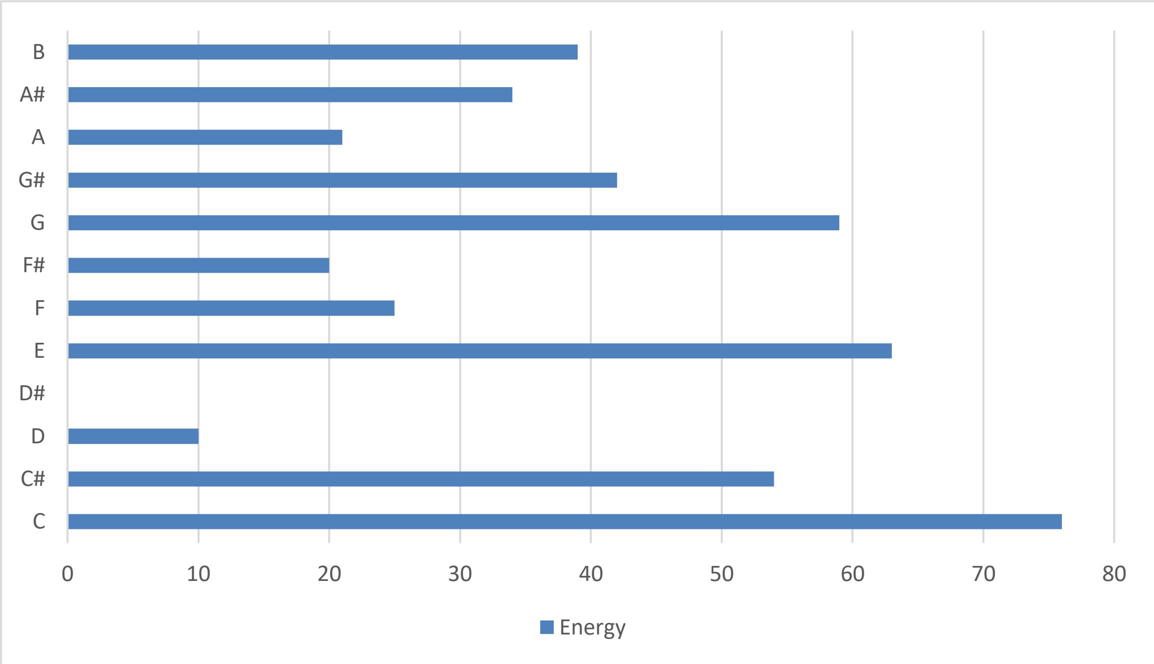


Figure 4: Scale Detection

According to Figure 3, the scale of the melody will be determined as C major as the notes C, E and G have higher energies.

A private data set is used in order to determine a proper chord progression for the vocal melody given to the system. A set of songs which has basic chords are considered in order to prepare the data set. For example the C major scale can contain the basic chords such as C, F, G, Dm, Am and etc. The data set is prepared based on the song database specified in https://www.hooktheory.com/site. In here, the chord progression patterns used in each song are taken into consideration. Therefore the dataset will contain a list of chord progression patterns for each song with the notes being played for each chord type which is used to predict the correct and accurate chord progression patterns. Below is an example on how the chord progression is selected for the dataset for songs.

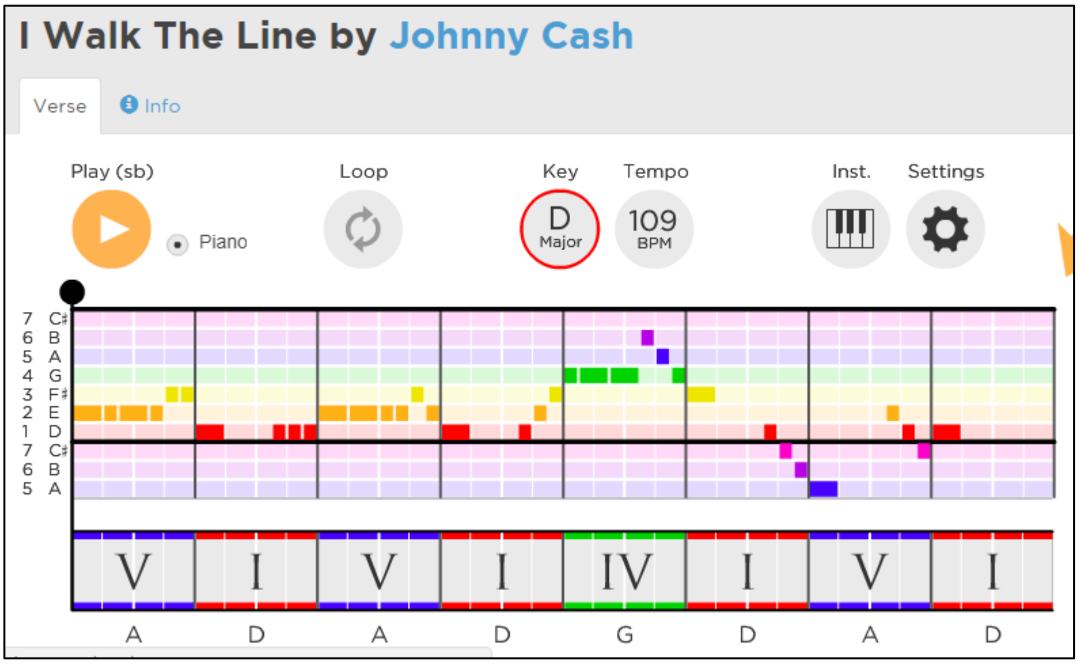


Figure 5: Chord Progression for the Data Set

According to Figure 5, the scale or the key of the song is D major. The chord progression for the song "I Walk The Line" is A, D, A, D, G, D, A, D and the associated notes being played are E, D, E, D, G, F#, A, D respectively.

The data set is prepared by considering 5 songs that falls under each scale which will sum to 60 songs as there are 12 major scales in music. Each song which is selected is transposed to each and every scale in order to make the dataset accurate and consistent. Therefore, total number of songs in the dataset will be 60 into 12 which is 720. The dataset consists of 4 class labels as in most cases of simple popular songs, the chords are assigned to 4 bars based on the notes. A multi target module Class Relevance Method (CR) is used as the classifier module which shows high accuracy per class label compared to other multi target modules. The reason for selecting a multi target classification module is to properly determine the chord transition patterns for a particular melody. The class labels show more than 50% accuracy and the Hamming Loss is low for CR compared to other modules. Table 3 illustrates the comparisons of the multi target modules.

Multi Target Module	Class Label 0	Class Label 1	Class Label 2	Class Label 3	Hamming Score	Hamming Loss
Bayesian Classifier Chains (BCC)	0.714	0.540	0.588	0.634	0.619	0.381
Classifier Chains (CC)	0.714	0.540	0.600	0.608	0.616	0.384
Classifier Chains with probabilistic output (CCp)	0.714	0.540	0.600	0.608	0.616	0.384
Class Relevance (CR)	0.714	0.534	0.610	0.637	0.624	0.376

Table 3: Comparison of Multi Target Classification Modules

Based on the above comparison among multi target classification modules the CR is selected. The pitch list of each chunk, the detected scale and the tempo selected by the user are processed along with the private data set which uses CR is then used to generate the chord progression patterns.

3.5. Accompaniment Assignment

Once the most suitable chord combination is identified by the chord estimator, a musical accompaniment is assigned to each chunk WAV. The chunk WAV files are combined along with the chords in order to form the song. In here, a simple piano chord is assigned as the accompaniment.

4. Chapter 4: Implementation

This chapter explains how the system is implemented and the technologies used.

The system, Automatic Accompaniment Generation for vocal input is entirely developed using JAVA platform. Figure 6 shows the graphical user interface of the system.

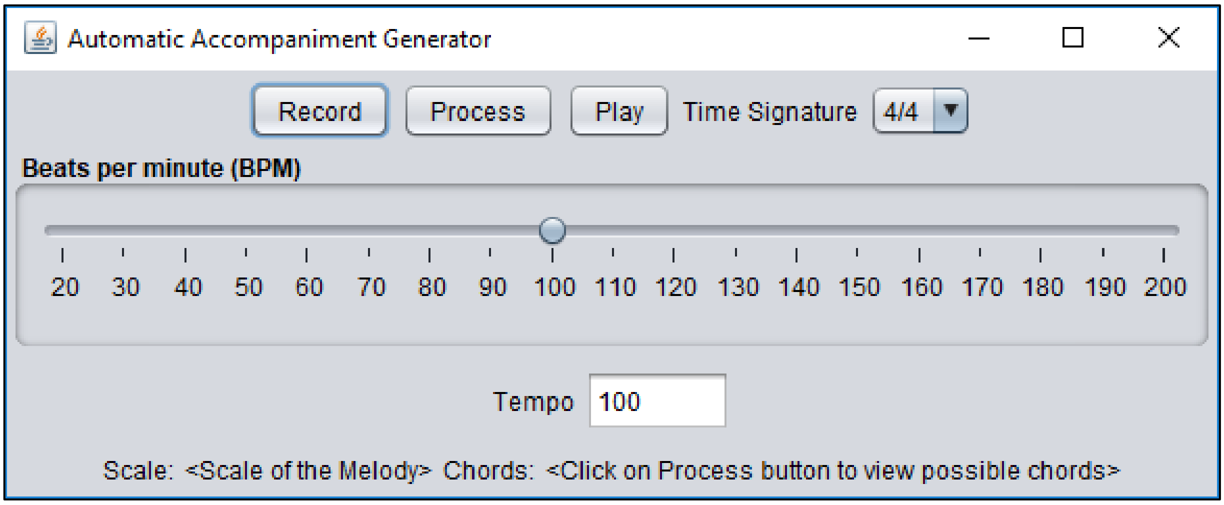


Figure 6: Initial Step GUI

Initially the user has to select or enter the tempo and click on the "Record" button in order to record the melody in WAV format. Once the "Record" button is clicked the system starts recording the vocal melody. The "Stop" button will appear as soon as the "Record" button is clicked so that anytime user is able to complete the recording process. Figure 7 shows the 2nd step of the GUI.

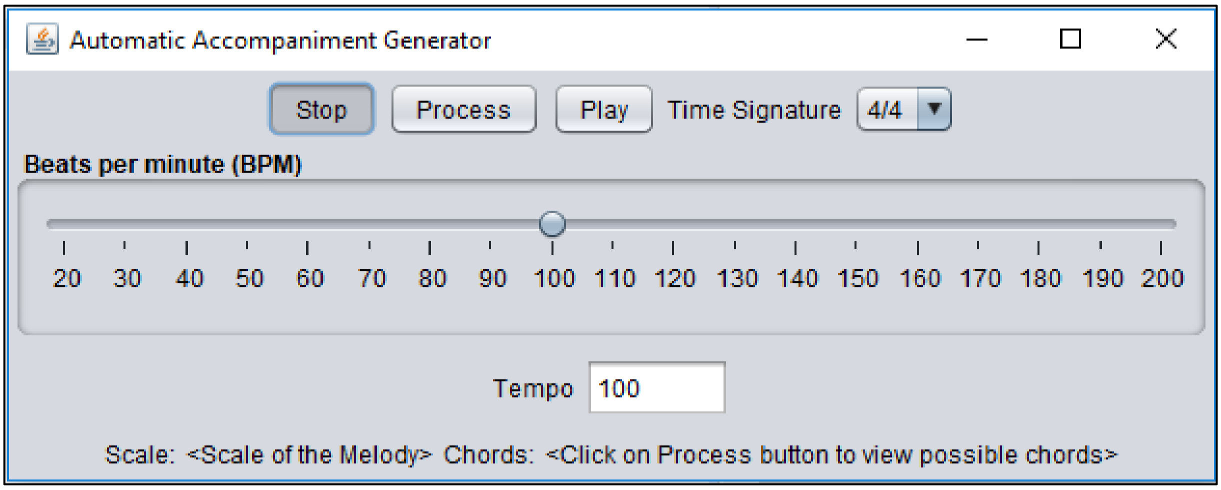


Figure 7: Step 2 GUI

Once the recording is completed a directly called “Recorded_Audio_aagfvi” will be created in the user’s home directory, including the vocal track in WAV format, chunk WAV files created based on the tempo bpm selected by the user and a properties file that contains important properties of the audio such as bpm, sample rate, sample size in bits, encoding and etc. The system analyzes each chunk WAV file and generates a list of frequencies and magnitudes for those frequencies using Goertzel algorithm. These frequencies will be used to initially detect the scale and then both the pitch list and scale is passed to estimate the chords. Once the chords are identified using the CR method which falls under multi target classification modules category, both the scale and identified chords are returned to the user. Figure 8 shows how the user sees the detected scale and chords identified by the system.

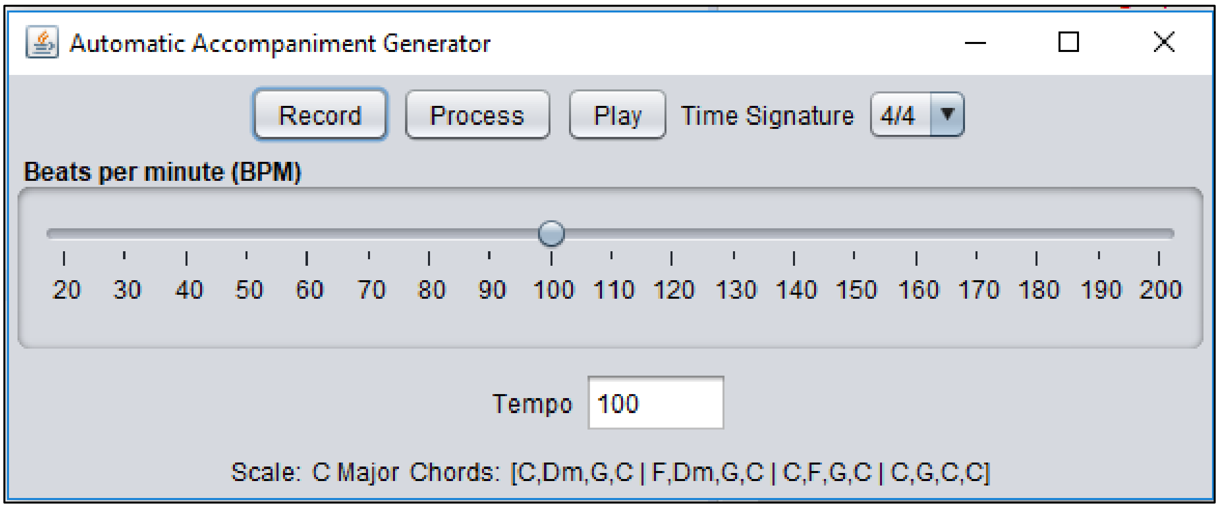


Figure 8: Step 3 GUI

Finally, the user has to click on the “Play” button to listen to the melody that the user has given into the system combined a piano accompaniment.

The tool MEKA which is an open source implementation of methods for multi label learning and evaluation is used in order to experiment the results. In most cases the machine learning techniques involves predicting a single target variable. Since the chord progressions can be best determined based on the notes being sung and the previous chords being played the multi label learning tool MEKA is used.

5. Chapter 5: Evaluation and Testing

Multiple vocal melody tracks with equivalent length from different users who are capable of singing to the pitch and tempo are considered in order to experiment the results. In here, both male and female singers are taken into consideration. The main problem of assigning chords for a melody section is that it differs from user to user based on their knowledge in music, lyrics of the song and their imagination of the melody. Therefore we cannot exactly comment on a chord progression saying that it is the correct one for a particular melody. For example, a melody contains only note C, what are the possible chords for it? The answer could be C Major, F Major, A minor, etc. There is no best answer for this question. Hence it's very hard to decide which chord is the best option without considering the context.

Therefore, experiments are established by taking a song with a simple chord progression and comparing the number of correctly identified chords for the recorded vocal melody against the actual chord progression of the song. A percentage of correctly identified chords for the recorded melody are evaluated.

For the evaluation of results from the above mentioned experiment, a randomly selected sample of 20 vocalists and the song "Hey Jude" was taken into consideration. The group selected were male and female individuals comprising of different vocal frequencies, pitches and scales. Also, this randomly selected sample of individuals performed different mistakes such as tempo mistakes, pitching mistakes etc. Therefore, this experiment takes into account all the above mentioned vocal characteristics and mistakes of the individuals.

The original song "Hey Jude" is played in F major scale which has the below simple chord progression for the chorus section included in the song database specified in https://www.hooktheory.com/site.

F, C, C, F, A#, F, C, F

The user might sing "Hey Jude" in a different scale. Therefore, the generated chord progression for the song "Hey Jude" is transposed to its original scale F major, in order to compare the detected chord progression against the actual chord progression.

The transposition of chords is done based on the chromatic circle as shown in Figure 9 [11].

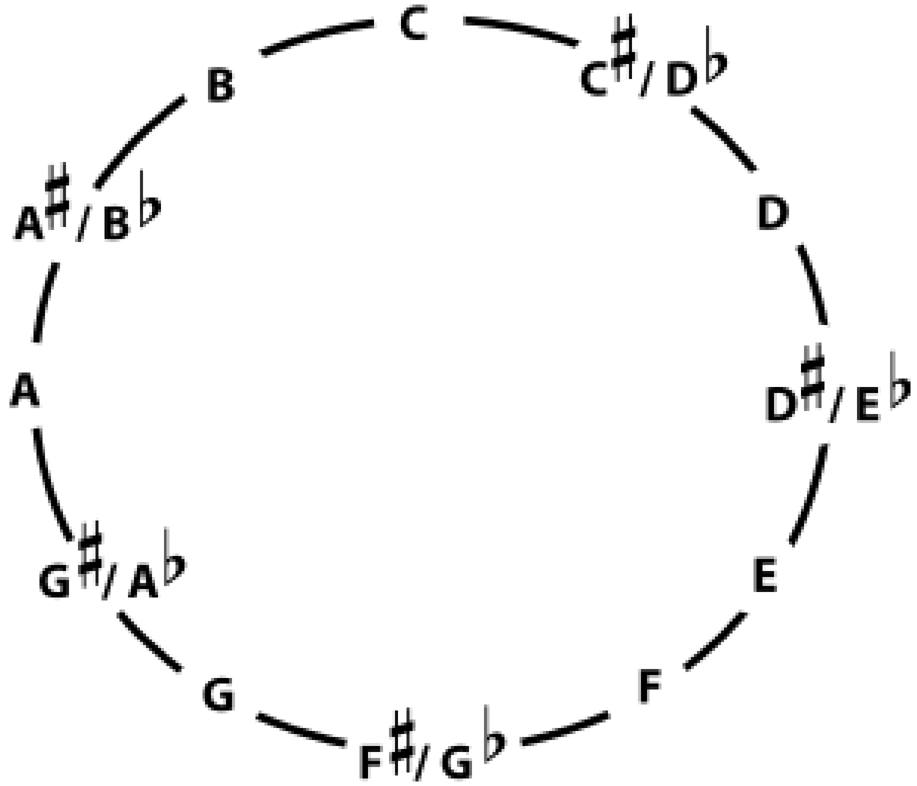


Figure 9: Chromatic Circle

Assume the original scale of a song is “F major” and the detected scale for the vocal input is “A major”. In addition assume that the original song has the chord progression of F, C, C, F, A#, F, C, F and the detected chord progression for the vocal input is A, E, E, E, A, E, A, A.

As each scale has its own chords, in order to compare and evaluate the chords with the original chord progression, the detected chords in “A major” scale need to be transposed into “F major” scale. As the initial step, the number steps that passes by from scale “A major” to scale “F major” needs to be calculated either in clockwise or anticlockwise directions in the chromatic circle. If we chose the anticlockwise direction, based on chromatic circle in Figure 9, moving from “A” to “F” requires 4 steps (A -> G# -> G -> F# -> F). Therefore, for each chord detected for “A major” scale, 4 steps needs to be calculated in the anticlockwise direction in order to identify the transposed chord.

Therefore, after calculating 4 steps in the anticlockwise direction for each chord detected in “A major scale”, the result will be as follows.

A, E, E, E, A, E, A, A $\longrightarrow$ F, C, C, C, F, C, F, F

An example on how the comparison is done is shown below.

Bar No	1	2	3	4	5	6	7	8
Original chord Progression in F major scale	F	C	C	F	A#	F	C	F
Detected chord progression in A major scale	A	E	E	E	A	E	A	A
Detected chord progression transposed into F major scale	F	C	C	C	F	C	F	F
	√	√	√	X	X	X	X	√

Table 4: Comparison of Results

According to the details shown in Table 4, there are 4 chords that do not match the original chords in 4th, 5th, 6th, and 7th bar sections out of total 8 chords. Therefore, this shows 50% of correctly identified chords.

Below table illustrates the results of the experiment for the song "Hey Jude" which has the original chord progression F, C, C, F, A#, F, C, F.

	Detected scale	Detected chord progression	Transposed chord progression to F major	Percentage of correct chords
Person 1	A Major	A, E, E, E, A, E, A, A	F, C, C, C, F, C, F, F	50%
Person 2	D# Major	D#, A#, G#, G#, D#, A#, A#, G#	F, C, A#, A#, F, C, C, A#	50%
Person 3	A# Major	A#, F, A#, A#, A#, F, A#, A#	F, C, F, F, F, C, F, F	50%
Person 4	A# Major	A#, F, A#, A#, D#, Gm, A#, A#	F, C, F, F, A#, Dm, F, F	62.5%
Person 5	B Major	B, F#, E, B, E, B, B, F#	F, C, A#, F, A#, F, F, C	62.5%
Person 6	A Major	A, E, A, A, C#m, E, D, A	F, C, F, F, Am, C, A#, F	50%

Person 7	C# Major	C#, C#, F#, F#, C#, F#, C#, C#	F, F, A#, A#, F, A#, F, F	25%
Person 8	C# Major	C#, A#m, C#, C#, F#, C#, C#, C#	F, Dm, F, F, A#, F, F, F	62.5%
Person 9	C# Major	C#, G#, C#, C#, C#, A#m, F#, G#	F, C, F, F, F, Dm, A#, C	37.5%
Person 10	C Major	C, G, C, C, C, G, C, C	F, C, F, F, F, C, F, F	50%
Person 11	F# Major	F#, C#, F#, C#, F#, C#, F#, F#	F, C, F, C, F, C, F, F	37.5%
Person 12	G# Major	G#, Fm, C#, C#, Cm, Fm, C#, G#	F, Dm, A#, A#, Am, Dm, A#, F	25%
Person 13	C Major	C, C, F, F, F, C, C, C	F, F, A#, A#, A#, F, F, F	50%
Person 14	C# Major	C#, C#, C#, G#, C#, G#, G#, C#	F, F, F, C, F, C, C, F	37.5%
Person 15	D# Major	D#, D#, D#, D#, D#, D#, D#, D#	F, F, F, F, F, F, F, F	50%
Person 16	G# Major	G#, Fm, C#, D#, G#, D#, G#, G#	F, Dm, A#, C, F, C, F, F	25%
Person 17	A Major	A, E, C#m, D, A, G, A, A	F, C, Am, A#, F, D#, F, F	37.5%
Person 18	A Major	A, E, E, E, A, E, A, A	F, C, C, C, F, C, F, F	50%
Person 19	C# Major	C#, G#, C#, C#, C#, G#, C#, C#	F, C, F, F, F, C, F, F	50%
Person 20	C# Major	C#, C#, F#, F#, F#, C#, C#, C#	F, F, A#, A#, A#, F, F, F	50%

Table 5: Results of Experiments

Therefore, based on the test results the system shows total 46% of correctly identified chords for 20 vocal tracks collected by different users.

6. Chapter 6: Conclusion and Future Work

A system for automated accompaniment generation is presented in this thesis. Initially the melody that the user sings into the microphone is stored as WAV format with a sample rate of 44.1 kHz which is the recommended standard sampling rate for audio processing. The stored WAV file is divided into chunk WAV files based on the tempo that the user selects initially. A metronome is played while the user sings in order synchronize the beat with the melody. Each chunk is then processed via a tone detection algorithm called Goertzel in order to detect pitch values. These pitch values is used to detect the scale of the melody assuming that the scale is constant throughout the melody. Next the most probable chord sequence is calculated based on the scale and the notes detected. In here a private data set is used in order to predict the most probable chord sequence. This dataset consist of the chord progressions used in most of the popular songs. The dataset is trained by a multi target classification module CR which shows more than 50% accuracy for each label and less hamming error compared to other multi target classification modules. As the chord progressions for a song differ from user to user based on certain factors, there is no best way of evaluating the correct progressions. Therefore, the testing and evaluation is done by comparing the original simple chord progression of "Hey Jude" song with the detected chord progression. In here, 20 vocal tracks which consist of both female and male singers are collected in order to evaluate the results. The system shows 46% of correctly identified chords for 20 collected vocal tracks.

Below are the assumptions which are taken into consideration due to several constraints.

- Chord Estimation module assumes that the scale or the key of the melody is constant. In some of the modern music the key changes are involved and this is called as 'Modulation' in terms of music theory. In this type of situation, dynamic scale detection algorithms have to be considered. Therefore, key changes are not taken into consideration and hence, the user has to sing the melody in a constant scale.

- Chord Estimation module assumes that the tempo of the melody that the user sings in to the microphone is constant. In some melodies, changes in the tempo can be included and in order to detect the tempo of the melody, dynamic beat detection algorithms have to be considered. Therefore, changes in the tempo are not considered.

- Simple chord combinations with major and minor triads are used in order to estimate the chords. Complex chords such as 7th, 9th, augmented, diminished chords are not supportable.
- The system assumes that the user sings to the accurate tempo and pitch.
- Currently the system supports on beat chords with each melody bar. In some music, chord changes happens rapidly within each bar of the melody.
- Currently the system is designed to support songs that plays in Major scales.

As part of future work the above assumptions can be taken into consideration in order to develop a more accurate system. Currently the system detects the scale of the melody based on the notes being sung and the energies generated for each note by Goertzel algorithm. Therefore it is also better to go through an accurate well known scale detection algorithm as part of the future work. In addition, when it comes to music industry there are chord progression patterns which consist of complex chords based on the genre of the music. As per now the system generates only simple chords. Therefore as part of the future work, the chord patterns can be generated considering the genre of music as well. Moreover, currently the system identifies the chunk WAV files to assign chords based on the tempo selected in BPM. But when it comes to modern music there can be off beat chords which appear in the middle of chunks as well. Therefore, it is better to identify the silences of the vocal audio inside the chunks itself in order to assign chords, which will indicate that the user has paused singing for inhaling. Furthermore, dynamic beat detection algorithms and accompaniment with more music instruments can be embedded into the system in order to form an accurate accompaniment to vocal inputs.

References

[1] Z. Chen, J. Qi and Y. Zhou, Machine Learning in Automatic Music Chords Generation, 2015.

[2] C. Hausner, "Design and Evaluation of a Simple Chord Detection Algorithm," University of Passau, 2014.

[3] M. Mauch, K. Noland and S. Dixon, "USING MUSICAL STRUCTURE TO ENHANCE AUTOMATIC CHORD TRANSCRIPTION," in 10th International Society for Music Information Retrieval Conference (ISMIR 2009), Kobe, 2009.

[4] R. B. Dannenberg, "An On-line Algorithm for Real-Time Accompaniment," in Proceedings of the 1984 International Computer Music Conference, 1985.

[5] C.-H. Chuan and E. Chew, "A Hybrid System for Automatic Generation of Style Specific Accompaniment," in 4th International Joint Workshop on Computational Creativity, London, 2007.

[6] A. de Cheveigne and H. Kawahara, "YIN, a fundamental frequency estimator for speech and musica)," Acoustical Society of America 111(4):1917-30 May 2002, 2002.

[7] E. Larson and R. Maddox, Real-Time Time-Domain Pitch Tracking Using Wavelets, 2014.

[8] P. McLeod and G. Wyvill, "A SMARTER WAY TO FIND PITCH," in International Computer Music Conference, 2005.

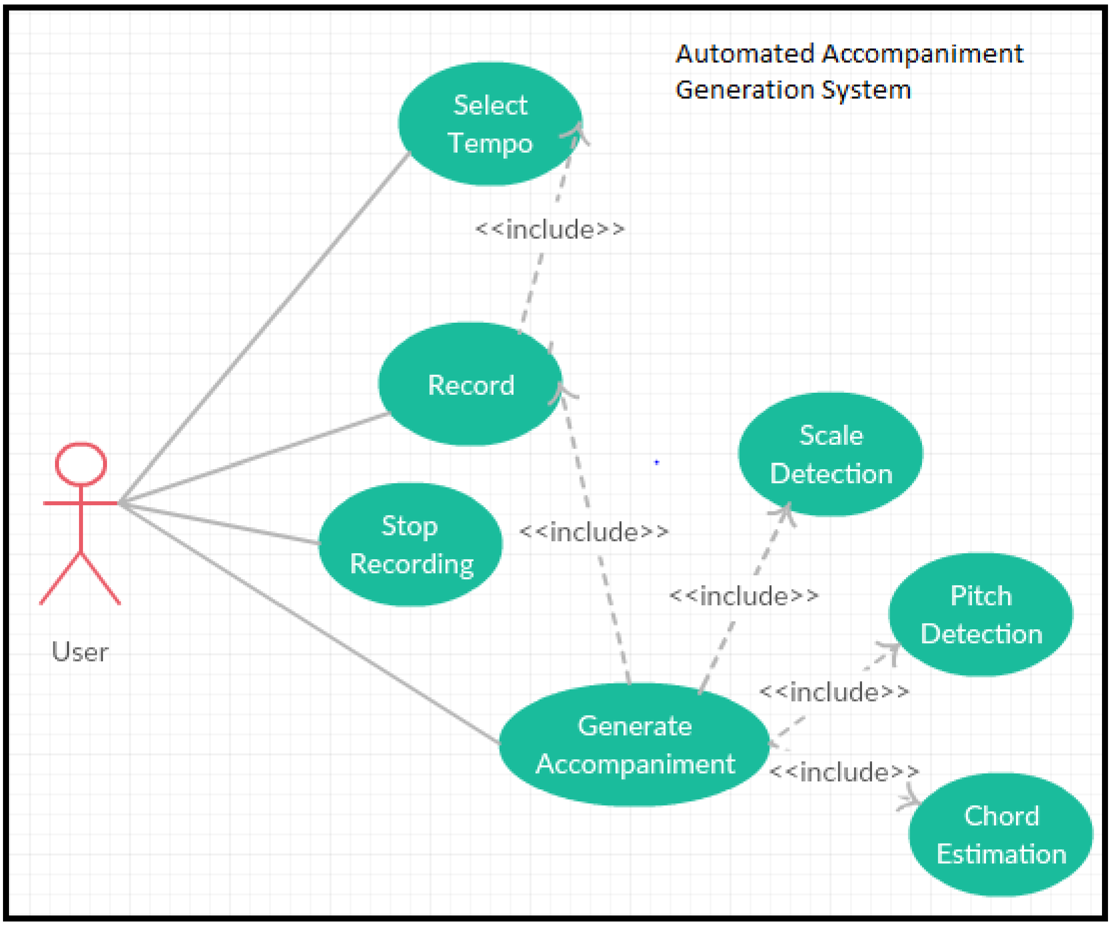
[9] K. Banks, The Goertzel Algorithm, 2002.

[10] F. Grondin, "Guitar Pitch Shifter," [Online]. Available: http://www.guitarpitchshifter.com/pitchshifting.html.

[11] EarMaster, "EarMaster," [Online]. Available: https://www.earmaster.com/music-theory-online/ch06/chapter-6-4.html.

Appendix A

Use Case Diagram



Appendix B

Class Diagram