ASHESI UNIVERSITY COLLEGE

AN ALGORITHM FOR MULTI TEMPO MUSIC LYRIC TRANSCRIPTION

UNDERGRADUATE THESIS

B.Sc. Computer Science

Hector Amoah

2018

ASHESI UNIVERSITY COLLEGE

An Algorithm for Multi Tempo Music Lyric Transcription

UNDERGRADUATE THESIS

Undergraduate Thesis submitted to the Department of Computer Science, Ashesi University College in partial fulfilment of the requirements for the award of Bachelor of Science degree in Computer Science

Hector Amoah

April 2018

DECLARATION

I hereby declare that this undergraduate thesis is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

……

Candidate's Name:

……

Date:

……

I hereby declare that preparation and presentation of this undergraduate thesis were supervised in accordance with the guidelines on supervision of undergraduate thesis laid down by Ashesi University College.

Supervisor's Signature:

……

Supervisor's Name:

……

Date:

……

Acknowledgement

I would like to thank God for seeing me through this project. I would also like to thank my mother Crispina Crankson as well as my sister Amanda Amoah for their constant support and encouragement throughout this endeavour. I would like to thank my friends for all of their support, and finally, I would like to thank all of the students and faculty at Ashesi University that aided me in my research.

Abstract

This paper documents an attempt to create an algorithm for multi tempo music lyric transcription. This paper reviews music information retrieval as a field of study and identifies music lyric transcription as a subset of the music information retrieval field. The difficulties of music lyric transcription are highlighted and a gap in knowledge in the field is identified. There are no algorithms for music transcription that are applicable to all forms of music, they are usually specialised by instrument or by genre. The author attempts to fill this gap by creating a method for multi tempo music lyric transcription. The methodology used to achieve this goal is a three-step process of taking audio as input, processing it using the REPET separation technique, and transcribing the separated audio file. The result of this paper was a relative success, with the music being separated successfully and the lyrics being transcribed but with accuracy lost.

Table of Contents

Chapter 1: Introduction

1.1 Background

To understand what this paper means by Multi Tempo Music Lyric Transcription, a brief explanation must be given to the concepts contained within the phrase, starting with Music Lyric Transcription (MLT). There is an entire field of study focused on attaining information from music. Whether it is the study of the complex interactions of the various facets that make up music, or the study of the musical notes and arrangements within the music, there is an umbrella term given to the field, Music Information Retrieval (MIR) (Downie, 2003).

1.1.1 Music Information Retrieval

Music Information Retrieval is an upcoming field that seeks to make human interaction with and understanding of music better. The scenario described by Downie at the beginning of his paper on Music Information Retrieval is farfetched but ultimately possible with advancements in the field (2003). He describes a person singing out lyrics that are stuck in his or her head to the hearing of their computer, which then cleans up the rough adaptation of the song and manages to find information about the original (Downie, 2003). The name of the song, its author, and multiple versions of it than can be found online and downloaded to the listening pleasure of the user (Downie, 2003). This sums up the face value of Music Information Retrieval, but beneath the surface there are layers of dizzying complexity to be considered (Downie, 2003).

The diversity and vastness of music and the musical form make Music Information Retrieval a daunting field. The field concerns itself with the complex interactions of various facets of music, namely pitch, temporal, harmonic, timbral and textual, to name a few (Downie, 2003). Besides that, it also concerns itself with the choice of musical representation (Downie, 2003). Whether to use symbol-based representation, audio-based representation, or both (Downie, 2003). Under each these representations, there is the matter of the

implications on your computer system. How much computational power and storage is needed to process and display the music information? With over 10,000 albums and 100,000 works registered to copyright every year, as of 1999 (Downie, 2003). As if these were not enough variables to consider, the multicultural, multiexperiential and multidisciplinary aspects of music must also be considered (Downie, 2003).

1.1.2 Music Lyric Transcription

There is a branch of study underneath Music Information Retrieval focused on Music Lyric Transcription. This field, put simply, is the study Music Lyric Transcription is generally difficult to research, because of reasons I will put forth. Music Lyric Transcription takes all the difficulty of speech transcription and puts an additional layer of difficulty on top of it in the form of musical melody from musical instruments and various music generating tools. The music adds a pitch to the audio which is different from the pitch of the vocalist's voice. This is known as the Harmonic facet of Music Information (Downie, 2003). It obscures the voice from conventional voice recognition software, meaning any hope of voice recognition and ultimately transcription has to come after the vocals are separated from the music.

In light of these difficulties, Lyric Transcription Technology has been split into multiple techniques. Informed Music Transcription (IMT), Instrument Specific Music Transcription (ISMT), Genre specific music transcription (GSMT) (Gowrishankar & Bhajantri, 2016). Informed Music Transcription is done when there is possibly information from the user about the music, if there is not a default method will perform the transcription; this method can be referred to as a semi-automatic mechanism for transcription (Gowrishankar & Bhajantri, 2016). The instrument specific method focuses on developing a general method of transcribing based on the instruments used in the music (Gowrishankar & Bhajantri, 2016) and the genre specific method works using the knowledge of the genre of the music in a

similar way (Gowrishankar & Bhajantri, 2016). Under each of these techniques there are various approaches and algorithms that have been developed by scholars in the field (Gowrishankar & Bhajantri, 2016).

1.2 Problem Statement

The problem that becomes evident from the current breakdown of Music Lyric Transcription is the lack of a single unified algorithm or technique that can handle all types of music. Techniques either need extra information from the offset; focus on only one set of instruments; or focuses on one type of genre of music (Gowrishankar & Bhajantri, 2016). Furthermore, most of these approaches are focused on retrieving the musical score and musical information instead of the lyrics. Therefore, the problem statement for this thesis is; Despite the extensive research being conducted into Music Transcription, there is a gap in knowledge in the form of a comprehensive algorithm that covers all forms of music. Additionally, there is a lack of focus on algorithms that extract lyrics or more specifically vocals. Instead, most algorithms currently focus on retrieving the musical melody.

1.3 Objective

This research paper seeks to properly review the current methods of music transcription and identify the approaches that can be modified or implemented to fill in the gaps in research and technology identified in the problem statement. At the end of this paper, I will propose an algorithm based on prior music transcription algorithms and technologies, which will be able to transcribe lyrics from music. Insights from this research can be used to develop a comprehensive system for transcribing lyrics from music, taking into consideration language and intonations and adjusting for them.

1.4 Topic Relevance

Academically, we have already identified that this research will fill a gap in the information and technology in the field of Music Information Retrieval. There are only a few or no people researching an algorithm that retrieves lyrics from all types of music, regardless of pitch and tempo. However, besides its academic relevance, this topic also has relevance to the global music industry. The music industry, an important subset of the entertainment industry, is a multi-billion-dollar industry globally (International Federation of the Phonographic Industry, 2017). Similar to all other industries in the digital age we find ourselves in, technology is rapidly being integrated into this industry. Music streaming services like Spotify, Tidal, SoundCloud and Apple Music are the primary sources of music for many consumers in the industry, growing in revenue by 60.4% since the beginning of 2016 and accounting for 50% of the total income in the music industry (International Federation of the Phonographic Industry, 2017).

These streaming services all have similar features; a vast library of streamable music content and categorical grouping of music based on genre, artiste and album. Finding music on these streaming platforms is an easy enough task if you know the title of the song, or at the very least the artist who sang it. In cases where consumers do not know the artist or title but have simply heard the song playing and want to find out the information, the music identification application Shazam is a convenient tool at the disposal of the consumer.

Shazam is an easy to use application. Simply tap the button in the Shazam app when a song is playing, this works with any portion of the song, and Shazam tells you the information on record of the song ("Company-Shazam," n.d.). Shazam has information such as the title of the song; the name and sometimes biography of the artist or band who sang it; what album the song belongs to; links to the song on the various streaming services I listed

above, which Shazam has partnerships with; the music video of the song, if it happens to have one; and even the lyrics of the song (“Company-Shazam," n.d.).

Shazam uses proprietary software that recognizes audio samples and locates audio files that closely match the sample provided, in the Shazam database, matching the sample with the stored file using landmarks and fingerprints within the audio files (US9401154 B2, 2016). With the streaming services it collaborates with, you can quickly get access to the actual song and play it immediately.

Shazam is therefore a great source for a very comprehensive amount of information on music. The only shortcomings of Shazam are that unfortunately it does not have every single song in the world catalogued, and therefore in rare cases you might ask it to identify a song it has no knowledge of. I would like to clarify that this is an extremely rare occurrence. Shazam has a database of millions of songs which it can readily identify (“Company-Shazam," n.d.). Cases where it does not identify a song occur when this song is not available commercially, or in any official sense, that is, if the song is not an original recorded song available for purchase or streaming on any official music outlet.

Another shortcoming of Shazam is it does not provide the lyrics to every song in its database. This is by no fault of the Shazam team. Services like Apple Music have artist submitted lyrics for most of their music, so in most cases finding the song will ultimately find you the lyrics to the song. Artist submitted may however be an inaccurate definition for these lyrics, to be completely accurate, the lyrics are submitted by whoever is in charge of managing the digital content of the artist. This distinction is important because sometimes the lyrics provided are not accurate and end up misrepresenting the artists words. Besides these inaccurate lyrics, there is also the problem of a complete lack of lyrics. Some artist accounts do not submit any lyrics to the streaming sites on which they are hosted.

At this point you may be asking yourself if music lyrics are important enough to warrant any attention. Does it matter if some songs do not come with lyrics? Does it matter that some lyrics found on streaming services are inaccurate? To answer this question, I will present some information for your scrutinization.

The website lyrics.com, a member of the STANDS4 Network is free comprehensive online lyric resource ("About The STANDS4 Network," n.d.). This website has a database of hundreds of thousands of lyrics to songs and album information on multiple artists ("About The STANDS4 Network," n.d.). This website is mainly crowd sourced, and the STANDS4 Network collectively serves millions of uses worldwide ("About The STANDS4 Network," n.d.).

The website genius.com is another provider of music knowledge that focuses on music lyrics ("Genius – About Genius," n.d.). Genius serves over a hundred million people each month, all searching for song lyrics and having discussions about them or contributing to the database of music lyrics ("Genius – About Genius," n.d.). These are websites with a primary focus on providing lyrics to the public, and their scale and level of patronage is massive. This shows that a considerable amount of the general public is very interested in music lyrics. Personally, I am a living testament to this fact, since it was my interest in music lyrics that led to my choice of topic. With this, I believe the importance of digital music technology and music lyrics have been established.

This research will provide a contribution to the society of music lyric enthusiasts and to the digital music industry, ultimately remedying the need for crowd sourcing lyrics and placing the ability of generating your own lyrics to any song on the spot in the hands of the general public.

1.5 Research Questions

The research questions for this project are:

1. Is an algorithm for multi tempo music lyric transcription plausible?
2. Are there any music transcription algorithms currently in existence that are generalizable to all types of music?
3. How can current music transcription technologies be altered to create a multi tempo music lyric system?

1.6 Hypothesis

There exist a few music transcription technologies, or at least one, out of the multitude in existence, that can be used for transcribing lyrics from any song given minor modifications or perhaps once combined with other algorithms or technologies and then further modified.

Chapter 2: Literature Review

To properly conduct research into an algorithm for multi tempo music lyric transcription, a comprehensive review of all the methods of music transcription available today is necessary. Gowrishankar and Bhajantri performed such a review in their paper "An Exhaustive Review of Automatic Music Transcription Techniques" (2016).

This paper by Gowrishankar and Bhajantri is a comprehensive review of all the current models and technologies used for Automatic Music Transcription (AMT) (Gowrishankar & Bhajantri, 2016). The review included the strengths and weaknesses of current AMTs and possible ways of improving them (Gowrishankar & Bhajantri, 2016). To conclude the paper, they stated possible future enhancements to AMT, that could take care of the shortcomings of the systems currently in place (Gowrishankar & Bhajantri, 2016).

My literature review will have a similar structure and will be based on the information gathered in their paper, as well as information from similar scholarly works in the field. The structure to which I am referring is a review of music transcription techniques based on the various categories identified, Informed Music Transcription (IMT), Instrument Specific Music Transcription (ISMT) and Genre Specific Music Transcription (GSMT).

2.1 Informed Music Transcription

As stated before, Informed Music Transcription is a technique in which some extra information is presented about the music that is being transcribed (Gowrishankar & Bhajantri, 2016). It may either be a semi-automatic system in which there is partial automation and partial human interference, or a score informed system, in which the information is coming from a prior score of the music being transcribed (Gowrishankar & Bhajantri, 2016). There have been multiple music transcription algorithms developed with this technique.

Ye Wang and Bingjun Zhang modelled a system that uses Human and Computer Interaction to inform their model (Wang & Zhang, 2008). They let the user enter the type of instrument they are going to use to play the music, which then allows the algorithm to select an instrument model to use in transcribing (Wang & Zhang, 2008). The approach also takes in both audio and video of the user, allowing a video and audio analysis in a process of multimedia fusion (Wang & Zhang, 2008).

Shlomo Dubnov proposed a system of analyzing the structure of music given information about its signal's statistical properties (Dubnov, 2008). The scholar used this statistical data to model to visualize the music, which allowed for easier interpretation and transcription (Dubnov, 2008). This visualization presents structural features that are used to detect high points in music that are of interest and necessary for information retrieval (Dubnov, 2008).

J. J. Carabias-Orti et al. proposed an unsupervised process for multi-scene adaptive MIDI (Musical Instrument Digital Interface) (Carabias-Orti et al., 2010). Information about the instrument is obtained directly from the song to be transcribed (Carabias-Orti et al., 2010). They analyze spectral smoothness, perceptual significance and other properties at different frames of the same MIDI note (Carabias-Orti et al., 2010). After this analysis, harmonic spectral envelopes were created from the data, which were then applied to matching pursuits to produce harmonic decompositions (Carabias-Orti et al., 2010).

These decompositions are the end product that the algorithm seeks to create (Carabias-Orti et al., 2010). They have the significant part of the audio in one decomposition and any unwanted parts in a separate decomposition (Carabias-Orti et al., 2010). Testing and implementation of this technique resulted in 40.5% accuracy of music transcribed, which is not very accurate (Carabias-Orti et al., 2010).

Emmanouil Benetos and Simon Dixon came up with a method for automatic music transcription using joint multiple-F0 estimation (Benetos & Dixon, 2011). Simply put, they

attempted to use multiple pitch estimation to analyze music signals (Benetos & Dixon, 2011). They represented the music using a time-frequency image and suppressed noise during preprocessing using pink noise assumption (Benetos & Dixon, 2011). A salience function was used to select the pitch candidates for the multiple-F0 estimation (Benetos & Dixon, 2011).

For the optimal pitch candidate combination, they proposed a score functions with combines the spectral and temporal characteristics and suppresses the harmonic errors of the candidates (Benetos & Dixon, 2011). In the post processing stage, they used Conditional Random Fields (CRF) and Hidden Markov Models (HMM) trained on MIDI data (Benetos & Dixon, 2011). After the process was tested on classical piano and jazz music, they recorded a 60.5% level of accuracy of transcribed data, which is still not ideal.

These are a few of the Informed Music Transcription techniques reviewed by Gowrishankar and Bhajantri. The authors reviewed more techniques than these, each different from the next in methodology and results (Gowrishankar & Bhajantri, 2016). Next, I will address Instrument Specific Music Transcription.

2.2 Instrument Specific Music Transcription

Instrument Specific Music Transcription, as I explained earlier, is a technique in Automatic Music Transcription that requires information about the instrument used in the music to be known beforehand, or to be inferred from the recordings being transcribed (Gowrishankar & Bhajantri, 2016). Like IMT, ISMT also has multiple algorithms and approaches under it.

Barbancho et al. proposed a system for extracting the fingering configurations automatically from a recorded guitar performance (Barbancho et al., 2012). They used a set of 330 different fingering configurations corresponding to different musical chords played on the guitar fretboard (Barbancho et al., 2012). The transcription framework was formulated

using a Hidden Markov Model in which the hidden states were different fingering configurations and the observed acoustic features were obtained from a salience of a range of candidate note pitches measured by a multiple fundamental frequency estimator within individual frames (Barbancho et al., 2012). They trained their model on 22 recordings and tested it on 14 recordings, resulting in accuracy as high as 88% (Barbancho et al., 2012).

Anssi Klapuri proposed another computationally efficient algorithm along with Tuomas Virtanen for modelling and representing time carrying sounds (Klapuri & Virtanen, 2010). They encoded the individual sounds instead of statistics from multiple sounds representing a specific class (Klapuri & Virtanen, 2010). The model was generic and was used to represent any multidimensional data sequence (Klapuri & Virtanen, 2010). They applied the model to represent various musical instrument sounds compactly and accurately, with a Signal to Noise Ratio of 33 decibels (Klapuri & Virtanen, 2010).

Mohammad Akbari and Howard Cheng attempted to create a real time piano music transcription method based on a computer vision tool named claVision (Akbari & Cheng, 2015). The transcription was performed analyzing video performances with the computer vision tool alone, the sound was not taken into consideration (Akbari & Cheng, 2015). The claVision system was very accurate, detecting keys with a 97.4% precision (Akbari & Cheng, 2015). Sadly, this system is in no way transferrable to music lyric transcription, since it does not use the sound information and will therefore be unable to detect the words of the songs in any way.

Vipul Arora and Laxmidhar Behera proposed a method for clustering and identification of music in polyphonic audio using both semi-supervised and unsupervised algorithms (Arora & Behera, 2014). The algorithms used auditory scene analysis theory, which dealt with the perception and segregation of simultaneous acoustic streams (Arora & Behera, 2014). Their proposed clustering had three levels, pitched event decomposition, group object

formation and source streaming (Arora & Behera, 2014). The unsupervised model had an accuracy of 79.3% and the supervised approach had an accuracy of 75.2% (Arora & Behera, 2014).

Finally, from the review by Gowrishankar and Bhajantri I will look at Genre Specific Music Transcription. This is the last of the Automatic Music Transcription techniques reviewed but eh pair of scholars.

2.3 Genre Specific Music Transcription

Genre Specific Music Transcription, the final technique for Automatic Music Transcription addressed by Gowrishankar & Bhajantri, requires that general information about the music genre be known beforehand (2016). Below are a few algorithms and techniques under GSMT.

Olivier Gillet and Gaël Richard proposed a method for transcription and separation of drum signals from polyphonic signals (Gillet & Richard, 2008). The method combined the music signal from the original source and enhanced drum tracks obtained through source separation (Gillet & Richard, 2008). The resulting system was a complete and accurate drum transcription system which integrated many features and allowed for optima feature selection per track (Gillet & Richard, 2008). The system had a 79.8% level of accuracy (Gillet & Richard, 2008).

Anglade et al. designed a new genre classification system using low level signal-based feature and high-level harmony-based features (Anglade et al., 2010). They did this using a first order logic random forest based on transitions and built using The Inductive Logic Programming Algorithm (TILDE) (Anglade et al., 2010). Their testing revealed that when SVM classifiers were used, there was a statistically significant improvement over the tests conducted with the standard classifier (Anglade et al., 2010).

Vishweshwara Rao and Preeti Rao created a system for voice pitch contour extraction in polyphonic music using a dynamic programming algorithm and main-lobe matching (Rao & Rao, 2010). Their focus was on improving pitch accuracy in the presence of strong pitched accompaniments in music (Rao & Rao, 2010). Their system performed the F0 candidate selection and salience computation in two individual steps (Rao & Rao, 2010). The main-lobe matching method was used for identification of sinusoids from the resulting short-time magnitude spectrum (Rao & Rao, 2010). The accuracy of their pitch improvement on a test sample was 84.1% (Rao & Rao, 2010).

Vishweshwara Rao et al. also came up with a signal-driven window-length adaptation of sinusoid identification on music signals (Rao et al., 2012). They investigated adaptation of analysis window lengths using signal sparsity (Rao et al., 2012). From this research they realised that the main-lobe matching sinusoid detection method outperformed the amplitude envelope and phase-based methods (Rao et al., 2012).

This concludes the review of the three techniques of Automatic Music Transcription conducted by Gowrishankar and Bhajantri. There was a multitude of information to be taken into consideration, but the major take home form this paper and the papers it reviewed was the sheer multitudes of techniques out there under each discipline. There was also an extensive use of Hidden Markov Models and multiple pitch estimation, which is similar to what I am trying to achieve (Gowrishankar & Bhajantri, 2016). The key information I obtained from this paper was the use of separation techniques to isolate vocals and musical melodies, which led me to the next topic I reviewed.

2.4 Repeating Pattern Extraction

When looking for algorithms for separating music and vocals, I came across a paper by Rafii and Pardo on the Repeating Pattern Extraction Technique (REPET) (Rafii & Pardo, 2013). Conceptually it is a simple method, it uses the assertion that the basis of music as an art is

repetition (Rafii & Pardo, 2013). Using this assertion, the authors attempt to find the repeating pattern in a music file and separate that portion from the rest of the sound (Rafii & Pardo, 2013). The repeating pattern is referred to as the background, and the remainder of the sound is referred to as the foreground (Rafii & Pardo, 2013). The background consists of the music itself whilst the foreground consists of the vocal aspect and any noise that happens to be in the audio file (Rafii & Pardo, 2013). Amongst all of the separation algorithms I came across, this one was the most successful, as shown by tests on a 1000 audio song clips which ended in successful separation of background and foreground (Rafii & Pardo, 2013). The only drawback of this method is it takes a noticeable amount of time to complete the process of separation (Rafii & Pardo, 2013). However, once the method is applied to a song it eventually successfully separates the foreground and the background, and being interested in transcribing the vocals, I can then use speech recognition on the foreground.

2.5 Voice / Speech Recognition

I proceeded to conduct a review of various speech recognition scholarly articles, implementations and reviews. Deciding between implementing my own speech recognition system and using an already existing one. I also conducted a review of Natural Language Processing, considering the fact that my method would eventually be used on songs with different languages, accents and intonations. I ultimately decided to use an existing system for speech recognition, based on accuracy and processing power constraints. Below is the review of speech recognition technology and natural language processing I conducted, still with music lyric transcription in mind.

The article “Navigation-Orientated Natural Spoken Language Understanding for Intelligent Vehicle Dialogue” by Zheng, Liu & Hansen sought to test the viability of a two-stage framework for voice-based human-machine interfaces, as seen in next generation

intelligent vehicles (2017). The two steps in the framework are Automatic Speech Recognition (ASR), which converts the speech to text, and Natural Language Processing (NLP) (Zheng, Liu, & Hansen, 2017).

The NLP is implemented using a Deep Neural Network (DNN) framework that analyses sentiment and context on a sentence level and word/ phrase level (Zheng, Liu, & Hansen, 2017). The author concluded by suggesting the use of a Recurrent Neural Network (RNN) framework for implementing the NLP to make up for errors caused by ASR which are not catered for in the DNN framework (Zheng, Liu, & Hansen, 2017).

"Performance Analysis and Optimization of Automatic Speech Recognition" by Tabani, Arnau, Tubella and González looks at Automatic Speech Recognition (ASR) in the mobile world (2017). Due to the heavy processing power required for ASR, it is not easily delivered on mobile devices and embedded systems (Tabani, Arnau, Tubella, & González, 2017). This paper seeks to test a performance and energy characterization of Pocketsphinx, a toolset used for ASR (Tabani, Arnau, Tubella, & González, 2017). The authors optimize the performance and energy consumption of the toolset at the end of the endeavor, without any loss to the accuracy of the ASR system (Tabani, Arnau, Tubella, & González, 2017).

"Natural Language Processing" is a chapter out of the Annual Review of Information Science and Technology which is dedicated to Natural Language Processing (NLP) (Chowdhury, 2003). The chapter outlines the core concept of NLP, defining it, outlining its scope, tools and techniques used in NLP, its applications and the general procedure involved in performing NLP itself (Chowdhury, 2003). It is an indispensable source of information on NLP for anyone seeking to understand and work with it.

Chapter 3: Methodology

3.1 Description of Research Design

My research methodology will consist entirely of experimentation. I will outline the various stages in the music lyric recognition process I seek to create, namely, sound recording; sound processing, where the music will be separated from the vocals; speech recognition; and finally, transcription. Each of these processes will have its own module in my methodology that I will implement.

For the sound recording step, I will record audio straight from whatever device is being used. Pre-recorded audio files can be used, especially during testing, but for the desired end product, recording is ideal. For sound processing, the music has to be split from the vocals, using the REPET algorithm I identified in my literature review. This separation will result in a background file with the music and a foreground file with the vocals. The background of each separation will be kept and can be used for something else in the future if need be. The vocals, however, move on to the speech recognition step. At the speech recognition step, using Google speech recognition technology, the vocals from the song will be transcribed.

I will need data to test my proposed solution against. I will obtain this data, namely music files, from the open source music repository SoundCloud. Besides non-proprietary posts, all music uploaded by the artists on SoundCloud is free and not subject to any form of copyright that will be infringed upon by its use in this research.

3.2 Research Sample

As stated in the general description, I will be using a sample of open source, free to play music from Soundcloud, an online repository of user submitted music. Artistes who submit their music to Soundcloud are automatically giving anyone who comes across their music complete access to the music. This access includes the right to remix or make covers of this music. The only restriction with Soundcloud, is the same basic restriction that most

intellectual property has, it cannot be claimed by anyone else, and it must to cited or referenced when used in any form of work.

There are some songs on Soundcloud that were not uploaded by their original creators, but instead just by fans who chose to share music that would originally be pay to play. This music can be considered as bootlegged and therefore I will not use any music that falls into this category.

The reason I am using Soundcloud as the source of my sample is that I will need legal access to different types of music. Soundcloud is one of the only place where music this diverse is available for free and is completely legally at the disposal of its patrons.

3.3 Research Method

The method for my research was a simple python script that had a total of four methods or functions. Three of them were to represent each stage of the process and the last was to collectively run all the methods. I used python because it's simple syntax makes it ideal for prototyping and its vast library of modules simplify a number of tasks that I needed to perform. For instance, the "nussl" module which I used, has an implementation of the REPET algorithm necessary for my research. This is an explanation of each method and the role it plays in my methodology.

The first method is named "record". It can be seen in the appendix under code listing 1. This method uses the wave and pyaudio modules available in python to record audio from the default microphone of whatever device the code is run on and store the audio in a wave file. In this code I am recording one minute (60 seconds) of audio and saving that recording into a file named "song.wav". When the program is ready to record, it prints "recording audio…" onto the console, and when it finishes, it prints "Done Recording", after which it begins to process the recording and convert the audio to a wave file.

The next method is named “repet_song” and takes in an audio file as a parameter. It can be seen in the appendix under code listing 2. This function uses the “nussl” python module, created by the North-western University Source Separation Library (NUSSL). The function inside of nussl being utilised is an implementation of the Repeating Pattern Extraction Technique (REPET) using python’s scipy and numpy library. The method takes in an audio file, runs the REPET function on it, and produces a foreground and background audio file. It saves both files, but only returns the foreground, which contains the vocals.

The third method is named “transcribe” and it accepts an audio file as a parameter. This function can be found in the appendix under code listing 3. This method is the last is the three-step process of lyric transcription. It uses Google’s Speech Recognition API to transcribe the audio file passed to it and it returns the transcription in the form of a string. To use Google speech recognition, one must sign up for Google’s cloud services, since the speech recognition is run on a server in Google’s cloud. Once you are signed up, it is easy to integrate Google services into your code using the various necessary python modules.

Finally, there is a “test” function that simply connects the three functions together. This function can be found under the appendix under code listing 4, along with the import statements for all python modules used and the execution default statement.

The method for testing is the python script described above. For testing I passed a sample of 20 songs through the script to attempt to transcribe the lyrics, all songs were transcribed successfully but the accuracy of the transcribed lyrics was low, indicating noise or interference in the audio being transcribed.

Chapter 4: Analysis

Reviewing the results based on the modules, we can tell that the first and second module of the methodology were successful. In the third module, music with predominant vocals passed through transcription with a majority of the lyrics being properly transcribed. Songs with weak vocals were however not transcribed accurately. An analysis of this outcomes shows that there is noise in the foreground produced by the separation by the REPET method, and there is a need for further processing of the foreground produced to either boost the vocals or remove any auditory noise present.

Chapter 5: Conclusion

In conclusion, this research paper detailed the process of attempting to create an algorithm for multi tempo music lyric transcription. The method used in this research successfully achieves the goal of transcribing lyrics for multiple songs of multiple tempos despite a slight inaccuracy in transcribed lyrics due to audio noise, however, it relies heavily on already established modules for music separation and transcription. Due to this, there is little room editing the method to eliminate the noise being encountered. Furthermore, there is no room for the desired expansion of scope of the research to accommodate music in different languages and with multiple intonations accents.

For further study on this topic, the separation and transcription can be implemented by the researcher, allowing for more control over the quality of output vocals and lyrics. Natural language processing can also be incorporated to allow for the identifications of sentiment, and intonations as well as the identification and transcription of multiple languages.

References

About The STANDS4 Network. (n.d.). Retrieved March 15, 2018, from https://www.lyrics.com/about.php

Akbari, M., & Cheng, H. (2015). Real-Time Piano Music Transcription Based on Computer Vision. *IEEE Transactions on Multimedia, 17*(12), 2113-2121. doi:10.1109/tmm.2015.2473702

Anglade, A., Benetos, E., Mauch, M., & Dixon, S. (2010). Improving Music Genre Classification Using Automatically Induced Harmony Rules. *Journal of New Music Research, 39*(4), 349-361. doi:10.1080/09298215.2010.525654

Arora, V., & Behera, L. (2014). Musical Source Clustering and Identification in Polyphonic Audio. *IEEE/ACM Transactions on Audio, Speech, and Language Processing, 22*(6), 1003-1012. doi:10.1109/taslp.2014.2313404

Barbancho, A. M., Klapuri, A., Tardon, L. J., & Barbancho, I. (2012). Automatic Transcription of Guitar Chords and Fingering From Audio. *IEEE Transactions on Audio, Speech, and Language Processing, 20*(3), 915-921. doi:10.1109/tasl.2011.2174227

Benetos, E., & Dixon, S. (2011). Joint Multi-Pitch Detection Using Harmonic Envelope Estimation for Polyphonic Music Transcription. *IEEE Journal of Selected Topics in Signal Processing, 5*(6), 1111-1123. doi:10.1109/jstsp.2011.2162394

Carabias-Orti, J., Vera-Candeas, P., Caadas-Quesada, F., & Ruiz-Reyes, N. (2010). Music Scene-Adaptive Harmonic Dictionary for Unsupervised Note-Event Detection. *IEEE Transactions on Audio, Speech, and Language Processing, 18*(3), 473-486. doi:10.1109/tasl.2009.2038824

Chowdhury, G. G. (2003). Natural language processing. *Annual Review of Information Science and Technology*, 37(1), 51–89. https://doi.org/10.1002/aris.1440370103

Company. (n.d.). Retrieved April 14, 2018, from https://www.shazam.com/company

Downie, S. J. (2003). Music Information Retrieval. *Annual Review of Information Science and Technology,* 295-340. Retrieved March 15, 2018, from http://music-ir.org/downie_mir_arist37.pdf

Dubnov, S. (2008). Unified View of Prediction and Repetition Structure in Audio Signals With Application to Interest Point Detection. *IEEE Transactions on Audio, Speech, and Language Processing, 16*(2), 327-337. doi:10.1109/tasl.2007.912378

Genius – About Genius. (n.d.). Retrieved April 18, 2018, from https://genius.com/Genius-about-genius-annotated

Gillet, O., & Richard, G. (2008). Transcription and Separation of Drum Signals From Polyphonic Music. *IEEE Transactions on Audio, Speech, and Language Processing, 16*(3), 529-540. doi:10.1109/tasl.2007.914120

Gowrishankar, B. S., & Bhajantri, N. U. (2016). An exhaustive review of automatic music transcription techniques: Survey of music transcription techniques. In 2016 *International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)* (pp. 140–152). https://doi.org/10.1109/SCOPES.2016.7955698

International Federation of the Phonographic Industry. (2017, April 25). IFPI Global Music Report 2017. Retrieved from http://www.ifpi.org/downloads/GMR2017.pdf

Klapuri, A., & Virtanen, T. (2010). Representing Musical Sounds With an Interpolating State Model. *IEEE Transactions on Audio, Speech, and Language Processing, 18*(3), 613-624. doi:10.1109/tasl.2010.2040781

Mahedero, J. P. G., Martínez, Á., Cano, P., Koppenberger, M., & Gouyon, F. (2005). Natural Language Processing of Lyrics. In *Proceedings of the 13th Annual ACM International Conference on Multimedia* (pp. 475–478). New York, NY, USA: ACM. https://doi.org/10.1145/1101149.1101255

McVicar, M., Ellis, D. P. W., & Goto, M. (2014). Leveraging repetition for improved automatic lyric transcription in popular music. In 2014 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 3117– 3121). https://doi.org/10.1109/ICASSP.2014.6854174

Rafii, Z., & Pardo, B. (2013). REpeating Pattern Extraction Technique (REPET): A Simple Method for Music/Voice Separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(1), 73–84. https://doi.org/10.1109/TASL.2012.2213249

Rao, V., Gaddipati, P., & Rao, P. (2012). Signal-Driven Window-Length Adaptation for Sinusoid Detection in Polyphonic Music. *IEEE Transactions on Audio, Speech, and Language Processing, 20*(1), 342-348. doi:10.1109/tasl.2011.2162319

Rao, V., & Rao, P. (2010). Vocal Melody Extraction in the Presence of Pitched Accompaniment in Polyphonic Music. *IEEE Transactions on Audio, Speech, and Language Processing, 18*(8), 2145-2154. doi:10.1109/tasl.2010.2042124

Tabani, H., Arnau, J. M., Tubella, J., & González, A. (2017). Performance Analysis and Optimization of Automatic Speech Recognition. *IEEE Transactions on Multi- Scale Computing Systems*, PP(99), 1–1. https://doi.org/10.1109/TMSCS.2017.2739158

Wang, A. L.-C., & III, J. O. S. (2016, July 26). US9401154 B2. Retrieved from http://www.google.com/patents/US9401154

Wang, Y., & Zhang, B. (2008). Application-Specific Music Transcription for Tutoring. *IEEE Multimedia, 15*(3), 70-74. doi:10.1109/mmul.2008.49

Zheng, Y., Liu, Y., & Hansen, J. H. L. (2017). Navigation-orientated natural spoken language understanding for intelligent vehicle dialogue. In 2017 *IEEE Intelligent Vehicles Symposium* (IV) (pp. 559–564). https://doi.org/10.1109/IVS.2017.7995777

Appendix

Code Listings

1.

```
#Function for record module of methodology.
#Takes in a minute of audio, saves it into a wav file
#Returns the wav file

def record():
    chunk = 1024
    format = pyaudio.paInt16
    channels = 1
    rate = 44100
    record_seconds = 60
    wave_output_filename = "song.wav"

    p = pyaudio.PyAudio()


    stream = p.open(format=format,
                    channels=channels,
                    rate=rate,
                    input=True,
                    frames_per_buffer=chunk)

    print("\nRecording Audio...")

    frames = []

    for i in range(0, int(rate / chunk * record_seconds)):
        data = stream.read(chunk)
        frames.append(data)

    print("Done Recording")
```

```
    stream.stop_stream()
    stream.close()
    p.terminate()

    wavefile = wave.open(wave_output_filename, 'wb')
    wavefile.setnchannels(channels)
    wavefile.setsampwidth(p.get_sample_size(format))
    wavefile.setframerate(rate)
    wavefile.writeframes(b''.join(frames))
    wavefile.close()

    return wave_output_filename
```

2.

```
#Function for the processing module using REPET
#Function uses nussl package with in built REPET separation tool
#Takes in an audio and separates foreground from background using REPET algorithm
#Returns the foreground wav file created
def repet_song(song):
     # Set up audio signal
     signal = nussl.AudioSignal(song)

     # Set up a REPET object
     repet = nussl.Repet(signal)
     repet.run()

     #Retrieve Audio signals from REPET separation
     background, foreground = repet.make_audio_signals()

     #Store background and foreground into wav files
     background.write_audio_to_file('signal_background.wav')
```

```
    processed_audio =
    foreground.write_audio_to_file('signal_foreground.wav'
    )


    return processed_audio
```

3.

```
def transcribe(song):
    # Instantiates a client
    client = speech.SpeechClient()

    #  The name of the audio file to transcribe
    file_name = os.path.join(
        os.path.dirname(__file__),
        song)

    # Loads the audio into memory
    with io.open(file_name, 'rb') as audio_file:
        content = audio_file.read()
        audio = types.RecognitionAudio(content=content)

    config = types.RecognitionConfig(

encoding=enums.RecognitionConfig.AudioEncoding.LINEAR16,
        sample_rate_hertz=44100,
        language_code='en-US')

    # Detects speech in the audio file
    response = client.recognize(config, audio)

    result_string = ""

    #Concatenates speech from audio into a string
    for result in response.results:
```

```
        result_string
+=('{}'.format(result.alternatives[0].transcript))

    return result_string
```

4.

```
"""
     Methodology for Undergraduate Computer Science Thesis
     Author: Hector Amoah
"""
#import necessary modules
import pyaudio, wave, sys, time, nussl

# Import the Google Cloud Client Library
from google.cloud import speech
from google.cloud.speech import enums
from google.cloud.speech import types

#Test function to run all the modules of the methodology
def test():
    song = record()
    processed_song = repet_song(song)
    lyrics = transcribe(processed_song)
    print(lyrics)

if __name__ == '__main__':
    # Run the test when the file is executed
    test()
```