Universiteit Leiden

Master Computer Science

VocalFlows: A co-creative AI to suggest vocal flows

Name:	Noel Rohan Vasanth
Student ID:	s2808595
Date:	31/07/2022
Specialisation:	Artificial Intelligence
1st supervisor:	Dr. Peter van der Putten
2nd supervisor:	Dr. Rob Saunders

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Universiteit Leiden

Leiden University

Master thesis

VocalFlows: A co-creative AI to suggest vocal flows

Author:

Noel Rohan Vasanth - s2808595

August 1, 2022

Universiteit Leiden

Contents

1 Introduction

Creativity is the ability to create that which does not exist [33]. Most times creative outcomes stem from those which already exist, but brings about a new linkage which generates new outcomes [7]. The multidisciplinary field of *Computational Creativity* explores the idea of how computational systems could be made to produce creative outcomes to unbiased observers in particular tasks [17].

A growing sub-field of Computational Creativity is the use of computational systems to generate music. Such systems are called *Music Generation Systems* (MGS) [10]. The answer to the question "can computers be creative?" is the origin of Computational Creativity and it is not directly answered through the course of this work, instead a domain of creativity within MGS is explored, where a computational system is setup to aid its user in being creative, thereby, bringing about creativity. The concept of an AI agent providing suggestive aid to its user is known as *co-creativity* and the AI agent is known as a co-creative AI [25]. This work explores the development and assessment of a co-creative AI in the MGS field of Computational Creativity.

The challenge faced with a computational creativity task is the evaluation of the outcome. Creativity is subjective to the unbiased observer, and mathematical evaluation of the outcome is not enough [59], thereby, human evaluation tasks are performed along with mathematical evaluation to give the needed assessment.

Singing Voice Synthesis (SVS) [45, 30, 57] is a sub-field of MGS concerned with the generation of singing voice, synthesizing vocals from lyrics and at times with acoustic features (such as pitch, timbre, etc), and is an equivalent task of text-to-speech (TTS) but for singing voice. Compared to speech, singing voices have a complex prosody requiring SVS to take control of the duration and pitch of the vocals [62] unlike in TTS. Lyrics alignment with vocal separation [57, 15, 69, 60], acoustic modeling using deep neural networks [14, 37, 45], and parametric synthesis with spectrogram features [6, 32], are previous works in the rapidly growing field of SVS.

These tasks utilize high quality audio data for training, and require data to be annotated for the specific purpose of the SVS system such as labelling lyrics with the timestamped audio for lyrics alignment, or annotating the fundamental frequencies of the audio for pitch manipulation. Acquiring these datasets are challenging because they require human annotation and are scarcely available to the public. Music datasets are generally comprised of instrumentals, vocals, silences, and noise all together [4, 19]. To generate the sung vocal parts of a song without manual recording along with its manual labelling incurs cost.

Several challenges arise with developing an SVS system, as well as difficulties from the subjective goal of creativity. These include:

- Datasets containing audio with aligned lyrics are not available and have to be created from mining data from the Web. Websites [1] containing files of lyrics in an *LRC format* (LyRiCs) giving timestamped sentences of the lyric text are available. But this is not usable considering the requirement for alignment of also the words in lyrics and not only sentences. *DeepSinger* [57], is a SVS system that mines data from the Web. But their work is currently unavailable to the public.

- A single database with both audio and lyrics in the English language are not available. Therefore audio files will have to be matched with lyric files using the title and artists

[1] https://www.megalobiz.com/

of the tracks, which is challenging due to the inconsistencies in labelling across different platforms.

- Music websites contain song files with both the instrumental and the vocal parts together. SVS systems cannot be trained on such data because of the large quantity of noise. Therefore the removal of noise by separation of the instrumentals for each audio is necessary.
- After separation of the noise from songs, the vocals may contain large duration of silences. On a phoneme-level, silences are sometimes part of a single phoneme which causes difficulties in structuring of sentences.
- The automated alignment of audio with lyric texts is challenging due to the variations in vocal duration, pitch and tone of similar audios. State-of-the-art deep neural networks to automate lyrics alignment are performed usually at the sentence and word-levels. To achieve accurate alignment on a phoneme-level with web crawled data requires careful signal processing of the audio files.

Due to the developments in the field of Computational Creativity and SVS, an idea of a co-creative AI to suggest vocals given lyrics is possible to explore. The idea is to develop a system that suggests multiple vocals as audio containing a given phrase or lyric, each with a different flow or melody. A "flow" or melody of the vocals is what differentiates two vocals with the same lyrics. Even songs that are covers of an original song sound different than the original because of the unique way it is sung.

This task takes a computational creative approach, different from previous SVS systems since the flow of the vocals is derived from a novel evolutionary method and is uncontrolled by any audio feature such as fundamental frequency, tone or duration. These features are an additional cost to annotate in audio data and an "overhead free" approach is desirable. Genetic Algorithms (GA) [29] have creative potential because of the crossover and mutation operators [31] providing the ability to combine significant solutions to create new linkage. Although its use is not standard in SVS, for the purpose of providing creative outcome, a novel genetic algorithm method is developed to generate original sounding vocal flows.

There are challenges that arise due to the creative solution of this task:

- Designing a creative model using genetic algorithms requires a good representation of the data for the specific task. This dictates the variables of the fitness function [31]. Due to the computational creative approach, there does not exist such a fitness function for this task. Developing aforesaid model is challenging.
- A method of mathematical evaluation for comparison of similar audio outputs is to be developed. Mathematical evaluation of the results requires signal processing of the variable lengths of the output files. When comparing similar audios, discrepancies in the duration of the vocals can influence their correlation.
- Evaluation of the outcome has to done with a human observation and not only mathematical estimation due to the subjectivity of the task. Development without a standard measure is often difficult.

This work introduces VocalFlows, a co-creative singing voice synthesis AI agent to suggest vocal flows from an input lyric developed from scratch using data mined from the Web. To overcome the aforementioned challenges, the following steps are included in the pipeline of VocalFlows:

- *Web crawling.* The most popular songs of the year are collected from the Web irrespective of the language along with their lyrics, both from different sources.
- *Vocal separation.* The songs are separated from their instrumental parts to extract their vocal parts, using a source separation model called Spleeter [2] [27].
- *Filtration and processing.* Only data with matching title and artist of the lyrics and songs are filtered. The retained vocals and corresponding lyrics are checked for English language using Polyglot [3] [2]. The lyrics are processed to phonemes using a grapheme to phoneme converter [49].
- *Alignment of lyrics.* The lyrics and corresponding audio are aligned with a pre-trained alignment model [60] at a word level and a phoneme level.
- *ALT Fitness function model.* An automatic lyrics transcription (ALT) model was developed based on the Deep Speech 2 [3] model used for speech recognition [65] leveraging the training on the collected dataset.
- *Genetic Algorithm.* A representation for the genetic algorithm (GA) was developed using an index dictionary for the phonemes in the English language according to the CMU pronouncing library [4] and the GA was run to produce the vocal suggestions.
- *Evaluation.* Both subjective and objective evaluation methods were developed to analyse the results of the VocalFlows AI.

Experiments were conducted on the ALT model, as well as the GA model to assess the effectiveness of VocalFlows AI. Different methods of generating results with VocalFlows in terms of words and sentences are also observed. Human evaluation was conducted through a questionnaire to assess the quality in terms of the recognition of the results. The study shows that VocalFlows can successfully generate suggestive flows to songwriters as a co-creative AI with data mined from the Web.

The contributions of this work are as follows:

- The first co-creative SVS system to suggest vocal flows developed from data collected without any human interaction and annotation.
- A new music dataset of the top 500 popular songs of the year as audio files along with lyrics in a *csv file* matched with title and artist.
- A new dataset with pairs of phonemes and their corresponding aligned audio files.
- A simple and efficient ALT model, that can transcribe the lyrics of a vocal input.
- A co-creative VocalFlows GA model the generates suggestions of vocal flows given an input lyric uncontrolled by pitch, timbre, and singer identity.

All the contributions are made available online[5], along with the code to replicate datasets with options to reduce its size for the sake of public usage.

The remainder of the thesis is structured as follows. In Section 2 the related work are discussed. The collection of the datasets and the pipeline of the VocalFlows AI is explained

[2] https://github.com/deezer/spleeter
[3] https://github.com/aboSamoor/polyglot
[4] http://www.speech.cs.cmu.edu/cgi-bin/cmudict
[5] https://github.com/noelvasanth/VocalFlows

in Section 3. The experiments performed using the VocalFlows AI are described in Section 4 and their results in Section 5. A reflection on the results, limitations and the future work is discussed in Section 6. Section 7 concludes this thesis.

2 Related Work

In this section, the key components of text-to-speech (TTS), lyrics alignment, singing voice synthesis (SVS), speech recognition, and genetic algorithms (GA) are introduced. The related works on these fields are discussed to give insight into the techniques used in this work.

In order to understand how different parts of the VocalFlows system work, their background is described. Singing voice synthesis systems are based on text-to-speech systems where the similar tasks performed on speech are leveraged for singing voice. The related work of TTS and SVS systems are given in Section 2.1 and Section 2.2 respectively. The recent developments of lyrics alignment methods discussed in Section 2.3 accurately align text at a phoneme-level allowing for the development of the automatic lyrics transcription (ALT) model, which is differentiated from previous ALT models in Section 2.4. Insight is shown into the choice of evolutionary methods for the VocalFlows AI and their progress over the years in Section 2.5.

2.1 Text-to-Speech

Text-to-speech systems [46, 47, 56] convert given text input to natural and intelligible speech output. TTS is a growing field, which brought about the growth of SVS systems. Due to the development of artificial intelligence and deep learning, TTS systems are nowadays mostly neural based systems [61, 56, 35]. The pipeline of neural TTS begins with the conversion of sequences of text to linguistic features by text analysis, which are then fed to an acoustic model to produce acoustic features such as mel-spectrograms, which vocoders transform into speech waveforms. Neural TTS have also developed from learning short sequences using vocoders [46, 46] extending to larger sentences with transformers [35]. TTS systems are the basis for SVS systems. SVS systems follow a similar pipeline containing components of TTS.

2.2 Singing Voice Synthesis systems

Previous SVS systems were initially made using hidden Markov Models (HMM) [1], but with the advancements in deep learning they make use of neural based synthesis techniques [6, 57, 37, 30, 14] for synthesizing high quality singing voice. The vocoder architecture of TTS [47, 56] pre-trained on speech is used as a foundation over which SVS vocoders are trained on music data [57, 30, 14]. Adversarial networks are also used as vocoders trained on spectrograms [12, 11]. The SVS systems and TTS systems take audio features as musical score to their models for singer-independent high fidelity.

Jukebox [22] creates themed singing voice and music by training on conditioned information such as artists or genre. Jukebox uses three vector-quantized variational auto-encoders (VQ-VAE) for three levels of coarseness on the audio information captured: (1) a coarse top-level encoding that is able to learn structure and patterns of long sequences, (2) an mid-level encoding that learns more detail within the audio input, and (3) a bottom-level encoding that learns granular and highly detailed information of the audio input. Using transformers, a combination of the three encodings are decoded sequentially with conditioned information such as artists and genres to produce original music in the style of the artist or genre. For lyrics conditioning the transformers were provided with the entire songs lyrics and were

trained to predict a lyrics position within the music thereby learning to predict alignment. This SVS system successfully produces entire songs with control on lyrics, artists and genre.

DeepSinger [57] uses data mined from the web for and the initial stages of data extraction such as separation of singing voice with the instrumentals are done similarly with VocalFlows. DeepSinger uses two transformer models: (1) Bi-directional LSTM encoder-attention-decoder model for alignment of the lyrics and the audio based on WaveNet architecture [47]. This model is trained with guided attention to learn the position of the text with the position of the audio using a diagonal mask to reduce the loss. With the accurate positioning of lyrics, the duration of each word in the song is also extracted. (2) a feed-forward transformer singing model based on FastSpeech [56] that takes the lyrics, duration and the pitch as an input along with the reference audio to generate singing voice audio that contains the input lyrics that is similarly sung according to the reference input. VocalFlows does not use controlling parameters such as pitch and a reference vocal input instead it uses an inspiration set to sample audio bringing about creative linkage via evolutionary algorithms.

With the exception of DeepSinger [57], human recordings and/or annotations are collected for high quality datasets. Publicly available SVS systems are not trained only on English languages [32, 30, 14]. A recent survey on neural SVS systems [16], discusses a lack of unified datasets unlike in TTS, but show the rate of growth of these systems. VocalFlows does not use a musical score to train data and unlike any previous standard work, uses a genetic algorithm model to synthesize vocals. Besides this, VocalFlows also differs from DeepSinger in web crawling and provision of the dataset for public use.

2.3 Lyrics alignment

Without the alignment of text and audio, TTS and SVS systems cannot be trained. This is a key component of such systems that require the annotation audio data with text and their corresponding timestamped occurrences.

The work of [34], use cumbersome human annotations for alignment. Previous TTS works use HMM based Montreal Forced Alignment (MFA) [40] leveraging speech recognition from the Kaldi toolkit [54] to observe the timing of recognised text from speech. Due to the complex prosody of sung vocals alignment is done differently with their lyrics, as speech recognition alone is unreliable for singing voice. In [14], TTS speech recognition models are trained over singing voice data for word-level and phoneme-level alignment using MFA. The scarcity of audio-lyrics alignment datasets have made it difficult to leverage this method. Data augmentation techniques augment singing voice datasets to bring about improvements in lyrics transcription on pre-trained speech recognition models [67].

The recently published DALI dataset [41] contains labelled data of audio spectrograms and corresponding text, which has led to SVS systems leveraging spectrograms for alignment [57, 60, 30]. These use encoder-attention-decoder based transformer models to align lyrics with spectrogram input.

The works of [60] uses dynamic time warping (DTW) and attention jointly with the separation of vocal and accompaniment making precise phoneme level alignment and is similarly used in VocalFlows. The difference is in VocalFlows, the accompaniment or instrumental parts of the audio file are not given to the alignment model.

2.4 Automatic Lyrics Transcription

Automatic lyrics transcription (ALT) is the similar task of Automatic Speech Recognition (ASR) for music audio instead of speech audio [20, 18, 21]. Previous works heavily rely on

the Kaldi ASR toolkit [20, 18] as it provides a language model and an acoustic model. The use of Time-Delay Neural Networks (TDNN) [51] is preferred for this task over RNNs due to the ability of modeling long-term context, to predict the lyric. With the development of self-attention [63], many language processing tasks have used attention based models, including ASR tasks [56, 65], which has catalysed its development. Similarly with ALT models, attention based TDNNs [18, 20] produce high-quality results.

These works are not simple and efficient to leverage, since they have the Kaldi dependency. The process flow of the task of ALT is to develop a pre-trained ASR model by leveraging the same over music data. ALT models were shown to produce quality results[6], and an ALT model based on the Deep Speech 2 [3] ASR model was developed and used in VocalFlows trained on the aligned vocal-audio data.

2.5 Computational creativity

The field of Music Generation Systems (MGS) [28, 10, 8], create musical content either autonomously (all by the machine), or as an assistance to human musicians. The results of a exhaustive survey [8] of MGS systems observe that the key components are representation of the data such as MIDI, waveform, spectrogram, symbolic, or textual, and the architecture of the networks such as CNNs, RNNs, LSTMs, GANs, Evolutionary algorithms such as Genetic algorithms (GA), or also Markov chains. Depending on the task, the representation is chosen, and the MGS model can be determined.

Genetic Algorithms [29], are used in MGS systems because they do not require the context of the task as they only require a fitness function and an input representation [10]. The earliest and most popular GA for MGS was GenJam [5]. It is an MGS system developed using GA for Jazz music assistance originally having a human intervening as the fitness function to discriminate whether the outcome was good or bad. Later, a fitness function developed as a set of rules for harmonization of melody was performed by a GA [52]. With development of probabilistic methods, the use of Markov chains were used in the fitness function [39], and with recent development of deep learning, neural networks are now used in the fitness functions [42, 68]. VocalFlows incorporates an ALT model as a fitness function with a combination of text and audio waveform representation in the genetic algorithm.

The VocalFlows AI is a neural SVS system that utilizes a computational creative MGS in the form of a GA. The ALT model used in the GA is trained on aligned lyrics data mined from the web. The background work demonstrates data mined from the web can produce quality singing voice with a reference vocal such as in DeepSinger [57]. To achieve understandable or recognisable singing voice without references and with end-to-end open source data is the aim of the VocalFlows AI. The methods incorporated are discussed in Section 3.

3 Methodology

This section is divided into two parts. Section 3.1 describes the methods used to collect and process the website data. Section 3.2 explains the pipeline of the VocalFlows AI detailing each step in the process.

[6] https://www.assemblyai.com/blog/how-well-does-ai-transcribe-song-lyrics/

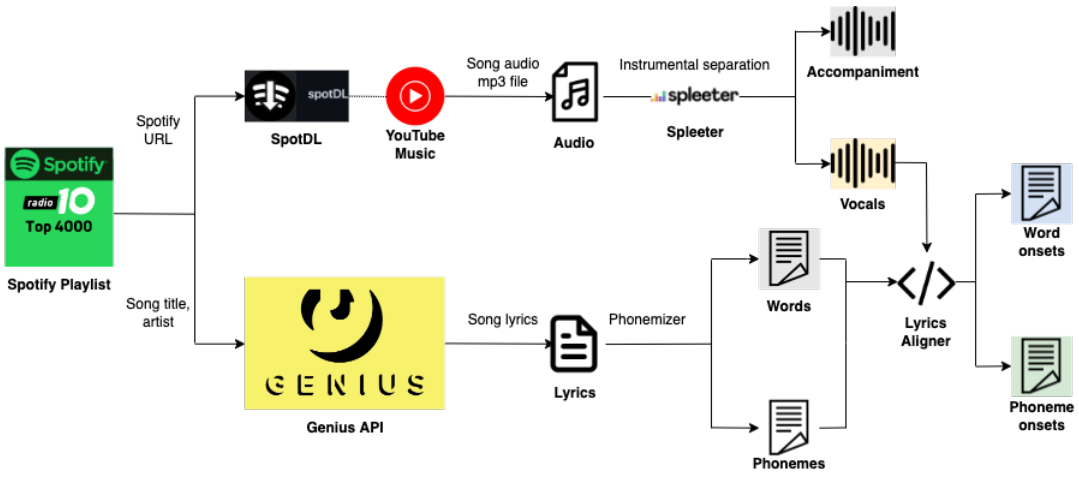


Figure 1: Pipeline for the acquisition of the Top 500 songs dataset used in VocalFlows

3.1 Dataset

For the collection of songs with metadata such as title, artist, and lyrics, websites with APIs such as Musixmatch[7] provide lyrics synchronised with the audio for companies like Spotify[8]. The goal is to obtain such a dataset with synchronised lyrics and corresponding audio, but the Musixmatch API is not entirely open source as only 30% of the song lyrics are provided with synchronisation for developer use. Music data crawling could not be achieved through music APIs for developers, but had to be done through websites. To the best of my knowledge MiniLyrics[9] was the only site that had synchronised lyrics, but the website went down on May 2021. Other websites such as Megalobiz[10], LyricsWikia[11], Songlyrics[12] provide LRC (LyRiCs) file format providing limited availability of synchronised lyrics for English songs and are not reliable. There do not exist reliable open source music databases or platforms that can provide both synchronised lyrics and audio files in English for any song of choice, and its creation was necessary.

Two types of datasets are formed through collection of audio and lyrics via Web crawling methods. These methods are discussed to give insight on the challenges of collection of music data for SVS systems. The two datasets created are:

1. Top 500 songs dataset - consisting of music separated vocal audio and corresponding lyrics.

2. Phoneme-audio pairs dataset - dictionary of 39 English language phonemes as keys and their corresponding vocal segments derived from the top 500 songs dataset.

3.1.1 Top 500 songs dataset

To begin with the collection of songs, it is necessary to know what songs to collect. For this purpose a list of the top 4000 popular songs of 2021 was observed in the form of a

[7] https://developer.musixmatch.com/
[8] https://open.spotify.com/
[9] https://minilyrics.com/
[10] https://www.megalobiz.com/
[11] http://lyrics.wikia.com/
[12] http://www.songlyrics.com/

No. of songs	Duration of songs	Memory
466	26.3 hrs	8.7GB

Table 1: Summary of the statistics of the top 500 songs dataset.

Spotify playlist[13] organised by a radio station in the Netherlands[14]. The playlist contained song metadata such as title and artist collected using the Spotify API[15] for developers. The API also provides the song URL for each song in the playlist. The playlist contained 3960 songs. With the title and artist metadata, the song could be found on the Genius API[16] for retrieving the song lyrics. Each song's lyrics were obtained using the Genius API provided the title and artist conditions were met.

Non-alphanumeric characters were stripped from each title and artist text of the Spotify playlist data to have the highest success of matching on both platforms due to inconsistencies in the names of the songs and names of the artists between platforms. After processing, 3556 out of 3960 songs could be matched and the lyrics were obtained for those songs. There were 404 songs with inconsistent metadata. Using the song lyrics, the language of the song could be determined. To the song metadata, the language was also added. It was found that 297 songs were not in English language, and the remaining 3259 songs were selected. The lyrics are processed to convert numbers into words, for example 1973 to 'nineteen seventy three'. All special characters were removed from the lyrics, to form a word corpus. The processed lyrics were also passed through a phonemizer[17] to acquire the phoneme for each word in the word corpus of each song. The phoneme corpus is collected to have its availability if required but was not used in VocalFlows.

Using the song URL metadata, the songs could be downloaded by using the spotDL API[18]. The spotDL API works by taking a song's spotify URL, and looking up the song's Spotify metadata on YouTube Music from where the song is downloaded. Only the first 500 songs of the 3259 English songs were downloaded of which 480 songs were successfully obtained without errors. The dataset occupied 8.9GB of memory for storage. Downloaded audio files contain both background instrumentals and vocals. Each audio file is separated into their respective vocals and accompaniment. Both are collected for the purpose of making a complete dataset. The dataset contains approximately 26.3 hours of English vocal audio data, double the amount of English vocal audio data used in DeepSinger [57]. If all the 3259 songs were downloaded the method has the potential to contain 162.95 hours of English vocal audio data, which is more than any dataset that is publicly available online.

With the possession of the words and the vocals, the synchronization of the two could be made possible. The lyrics aligner [60] model was trained on the MUSDB18 [55] dataset with the lyrics annotated manually. The model aligns vocals to the phoneme-level and supports 39 phonemes according to the ARPAbet notation given in the CMU pronouncing Dictionary[19]. The pre-trained model of the lyrics aligner was used in VocalFlows, with modified functions to suit the collected data. The model was not fed with the source audio data for internal separation, instead pre-separated vocals were used.

[13] https://open.spotify.com/playlist/0JSRbELaotklnZODkGRyHf
[14] https://www.radio10.nl/
[15] https://developer.spotify.com/
[16] https://genius.com/developers
[17] https://github.com/bootphon/phonemizer
[18] https://github.com/spotDL/spotify-downloader
[19] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

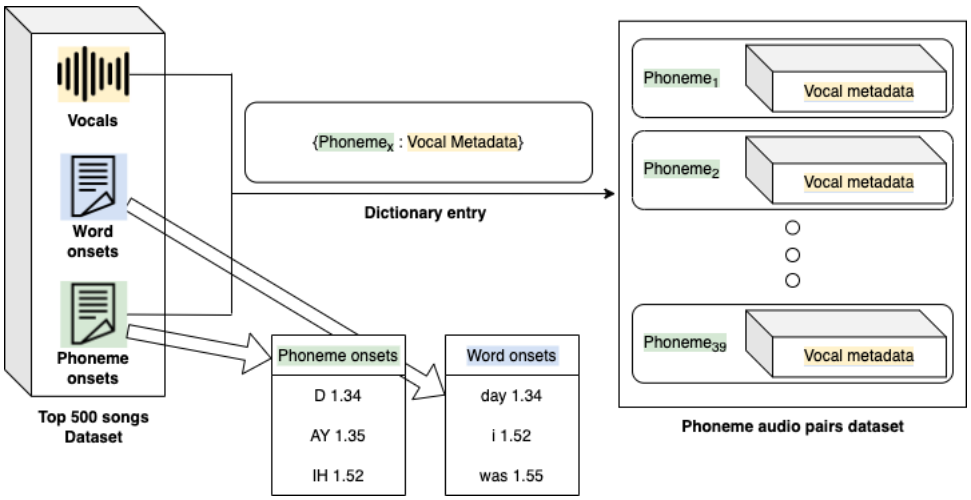


Figure 2: Process of the derivation of the Phoneme-audio pairs dataset used in VocalFlows

The model uses Dynamic Time Warping (DTW) combined with an attention mechanism to align lyrics. DTW helps with aligning different duration of audio belonging to the same word or phoneme by fitting the alignment path to a time sequence that matches the audio by conditioned trials. This is an important aspect that differentiates speech processing and singing voice processing. In speech processing the prosody and duration of spoken words are generally consistent between different instances of audio. The duration of singing voice for each word varies between different instances of singing audio, and DTW stretches or shrinks the vocals to a constant duration. The alignment model provided the necessary onsets of the word and phonemes in terms of the frequency frame, which was converted to the timestamp of the audio thereby observing the synchronisation of the lyrics and audio. A total of 466 were aligned without errors, and is the total number of aligned words and phones with vocal audio contained in the dataset. This method of obtaining the top 500 songs dataset is shown in Figure 1.

3.1.2 Phoneme-audio pairs dataset

The top 500 songs dataset described in Section 3.1.1, made available phoneme onsets for each of the 466 songs. There are only 39 songs in the English ARPAbet notation according to the CMU pronouncing dictionary as discussed previously. A new dataset which was used as the inspiration set [58] for VocalFlows was derived from the top 500 songs dataset, and is named Phoneme-audio pairs dataset. The 39 phonemes are used as keys in a dictionary, whose values are every occurrence of the corresponding phoneme in the 466 songs.

This process is shown in Figure 2. An example of the word ‘Day’ is represented with phonemes ‘D’ and ‘AY’ as can be seen in Figure 2. Here, the phonemes and words are along with their start time in the vocal file. The phoneme ‘D’, along with all vocal metadata such as the vocal track filename, filepath to find it in the top 500 songs dataset, start time, and duration of audio segment for that corresponding start time in that vocal file is recorded and stored as the value to the key ‘D’ in the phoneme audio pairs dictionary. The same is done for the next phoneme ‘AY’ and is repeated until all the phonemes in that vocal file is recorded into the dictionary. This is repeated for all vocal files. A total of $153,941$ audio segments for the 39 phonemes are observed. The distribution of the count of audio segments per phoneme is shown in Figure 3.

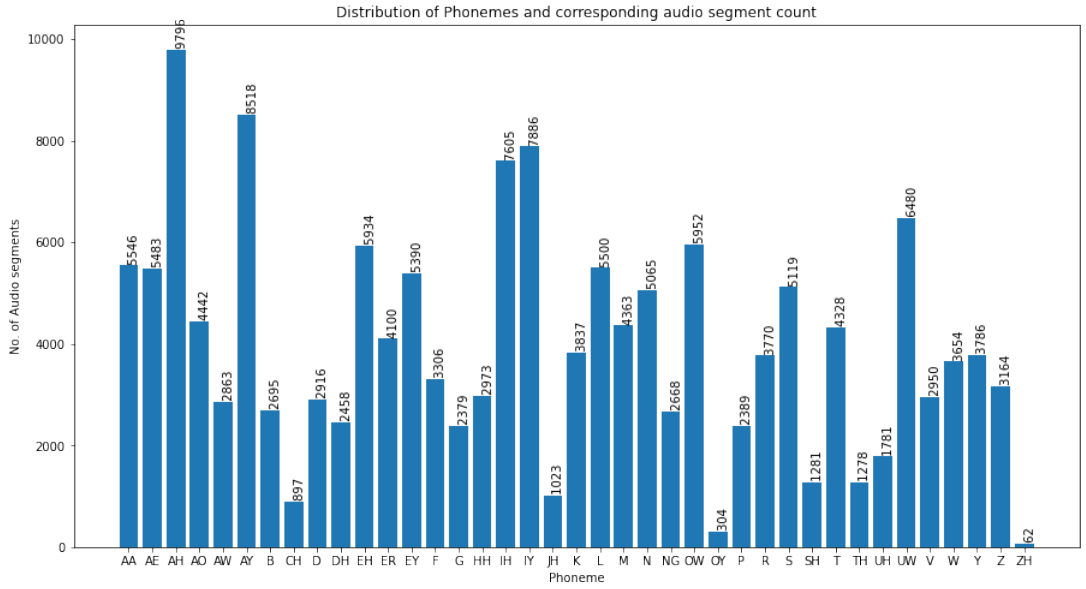


Figure 3: The statistics of the Phoneme-Audio Pairs dataset with respect to the count of audio segments per phoneme

It is observed that rarely occurring consonant phonemes have a lower amount of representation within the top 500 songs dataset than frequent vowel phonemes. The phoneme-audio pairs dataset is the pairing of the vocal files with their corresponding phonemes and is used as the inspiration set in the VocalFlows model.

3.2 VocalFlows Pipeline

The phoneme-audio pairs dataset described in Section 3.1.2, is the inspiration set that is used in the VocalFlows model. The most basic model that can be developed from this inspiration set is described under Section 3.2.1 as the baseline model. Description of the VocalFlows model is given in Section 3.2.2.

3.2.1 Baseline Model

A word is a sequence of phonemes. For example, the word “today” can be broken up into four phonemes: T, AH, D, and AY. From the inspiration set, each phoneme has a set of audio segments that can be used to audibly represent that phoneme in a given word or sentence. The baseline method includes the random selection of one audio segment corresponding to the given phoneme, and the combination of such audio segments to generate the word. Hence, in the example of the word “today”, four audio segments are randomly selected from the inspiration set, one for each phoneme and are combined to form the word “today”. The audio features such as duration, timbre and pitch, are inherent to the combined outcome and are not manipulated.

3.2.2 VocalFlows Model

After the previous steps of web crawling and obtaining the top 500 songs dataset as well as the inspiration set, a computational creative approach is used to design the SVS system of VocalFlows in the form of a genetic algorithm (GA). The model is given a phrase or

a lyric as a text input, the text input is processed and converted to words, each word is phonemized, and each phoneme is a key in the inspiration set from which an audio segment can be obtained. These segments represent the input to the VocalFlows GA. For example, for lyric "i love you", the words of the lyric are split up as "i", "love", and "you" and the phonemes are AY, $L\ AH\ V$, and $Y\ UW$. For each phoneme, an audio segment from the inspiration set is selected randomly, just as in the baseline model.

The genetic algorithm consists of the following components:

1. Inputs: A list of audio segments that are chosen randomly to represent the respective phonemes that make up the given word. The index of the audio segment in the list is the same as the index of the phonemes in the sequence with which they form the word. For example, for the word "love", the input is a list given below in (1).

$$[< L_{vocal} >, < AH_{vocal} >, < V_{vocal} >] \tag{1}$$

   This represents an individual in the population of the GA. The population can be of size N.

2. Output: Similar to the input, a list of audio segments that are processed by the GA. An output individual is represented the same as given in (1) above, for the given word "love". The size of the output population is also N, each individual having the highest fitness of the evolutionary process.

3. Fitness function: An automatic lyrics transcription (ALT) model that takes the population as input, combines it into one audio, and transcribes the audio into lyric. The transcribed lyric is checked against the ground truth lyric and the following are returned:

   (a) Character error rate (CER): The Levenshtein distance [66] between the hypothesized or transcribed text and the ground truth text on a character-level. The equation for character error rate is given in 2 below:

$$CER = \frac{S_c + D_c + I_c}{N_c} \tag{2}$$

   where S_c is the number of characters substituted, D_c is the number of characters deleted, I_c is the number of characters inserted, N_c is the number of characters in the ground truth text. This represents the edit distance computed by Levenshtein distance measure, divided by the count of the characters in the text. For example, if the ground truth text is "i love you", and the predicted text is "i lob yo", the CER will be $\frac{3}{10} = 0.3$. For a single word "love" and predicted "lob" the CER will be $\frac{2}{4} = 0.5$.

   (b) Word error rate (WER): The Levenshtein distance between hypothesized or transcribed text and ground truth text on a word-level. The equation for word error rate is given in 3 below:

$$WER = \frac{S_w + D_w + I_w}{N_w} \tag{3}$$

   where S_w is the number of words substituted, D_w is the number of words deleted, I_w is the number of words inserted, N_w is the number of words in the ground truth text. This represents the edit distance computed by Levenshtein distance measure, divided by the count of the words in the text. For example, if the ground

truth text is “i love you”, and the predicted text is “i lob yo”, the WER will be $\frac{2}{3} = 0.66$. For a single word “love” and predicted ”lob” the WER will be 1.

(c) Decoded prediction: The predicted word or lyric of the ALT model.

(d) Predicted phonemes: The phonemized decoded predictions.

Automatic Lyrics Transcription (ALT) Model was developed using the top 500 songs dataset audio spectrograms as input and lyrics as the ground truth. The architecture of the ALT model was based on the Deep Speech 2 [3] speech recognition model. The model prior to training on the top 500 songs dataset, was first trained on the LibriSpeech dataset [48] consisting of 100 hours of clean speech data for the speech recognition task. The architecture of the ALT model used in VocalFlows is shown in Figure 4. The model takes audio spectrogram as input and returns predicted word represented as a one hot encoded vector of 29 characters (26 English language characters and 3 special characters: apostrophe, comma, and full stop). The model calculates the Connectionist Temporal Classification (CTC) loss [24] between the continuous temporal input and the target sequence.

VocalFlows Genetic Algorithm consists of the selection, crossover and mutation functions that are essential to produce creative outcome towards minimising the CER and WER, thereby maximising the fitness. As discussed previously, the inputs and outputs of the GA are the population of individuals that are represented as a list of audio segments corresponding to the phoneme sequence of a given lyric. The fitness function calculates the CER and WER of the combined audio segments, and the generation of the lyric is evolved through this iterative process. A budget parameter of 100 iterations is used and the size of a population N consists of 10 individuals. During the process of the GA, N individuals are recombined and mutated $2 \times 20 \times N$ times to allow for a high chance of the best combination. This is explained further in the below steps. Therefore $N = 10$ parents, produces 400 children to select the fittest from. Other values of N were not further experimented with since $N = 10$ proved to be a reliable parameter. The core steps of the GA are as follows:

1. Selection: In this step, the top N individuals of the population are selected based on their fitness values. The WER and CER, and an index score are used as the fitness values to sort the population. The index score utilizes the decoded phonemes and the ground truth phonemes marking the index of those decoded phonemes that match the ground truth. The index score is the sum of all the matched phonemes. For example, if the decoded phoneme are $[TAH]$ to make the word “to”, and the ground truth phonemes are $[TAHDAY]$ to make the word “today”, then the index score is the sum of $[1, 1, 0, 0] = 2$, CER would be 0.4, and WER would be 1. The sorting order prioritizes WER, then CER and then the index score.

2. Crossover: In the crossover step, the input population is considered as the parent population, and the children are the recombined versions of the parents after crossover. During crossover, the number of crossovers are parameterized to be $10 \times N$, where N is the size of the parent population. For each of the $10 \times N$ iterations, two parents from N, are randomly selected and a single point crossover at a randomly chosen point with crossover probability of 0.5 is performed forming two children. Therefore, the total output children will be of size $2 \times 10 \times N$, and the output size of the population after this step will be $20 \times N + N$ as the parents are added back into the population with the children. The process can be seen in Figure 5. The selection of two parents for the purpose of crossover is not based on fitness of the parents to allow for variability in melody by random selection. The fitness function prioritizes understandability or

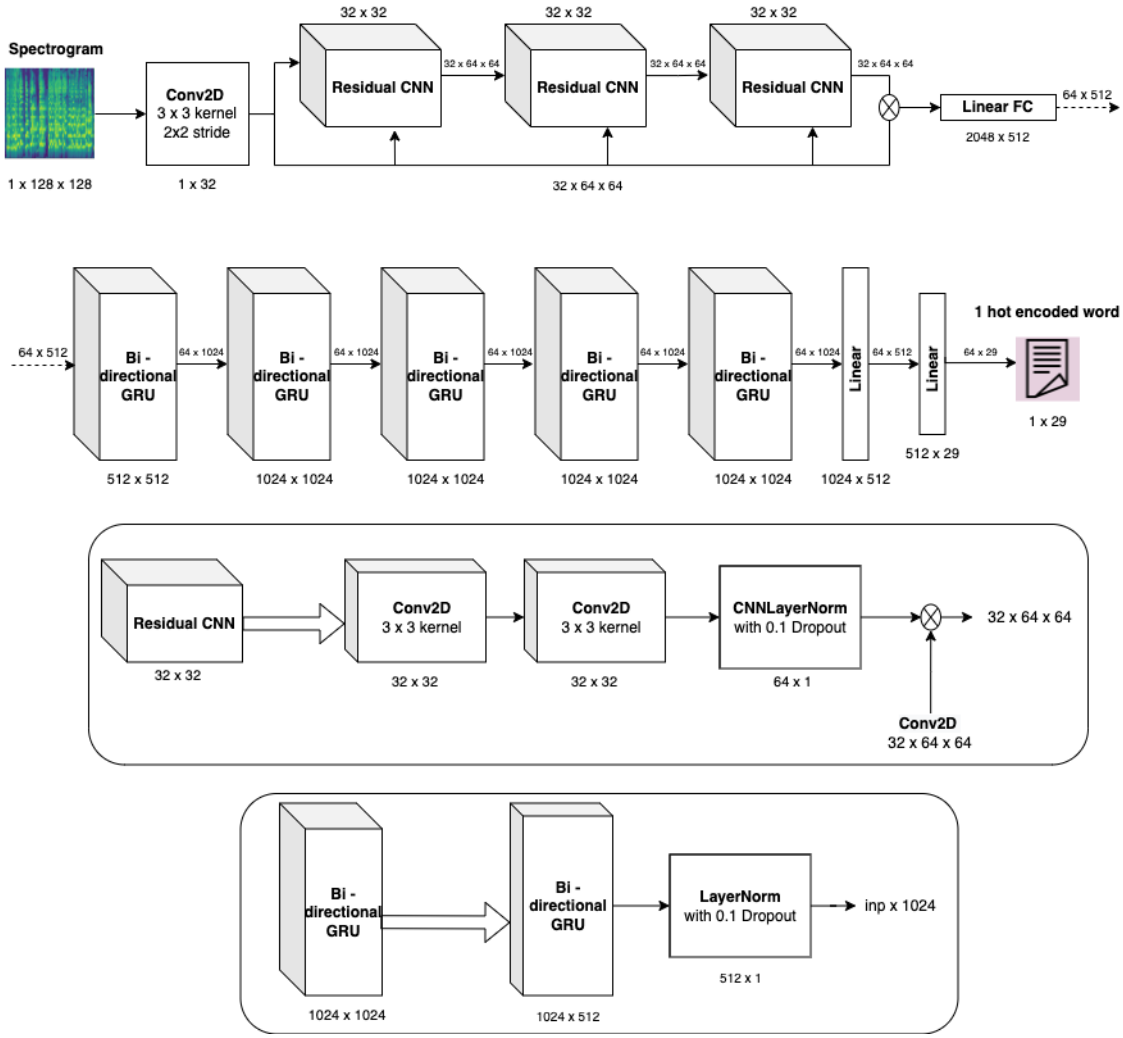


Figure 4: Shows the architecture of the ALT model used as a fitness function in VocalFlows trained on the top 500 songs dataset. The internal architectures of the Residual CNN layers and the Bidirectional GRU layers [3] are shown below the ALT architecture respectively. The model takes the spectrogram as an input and returns the one hot encoding of the hypothesized word.

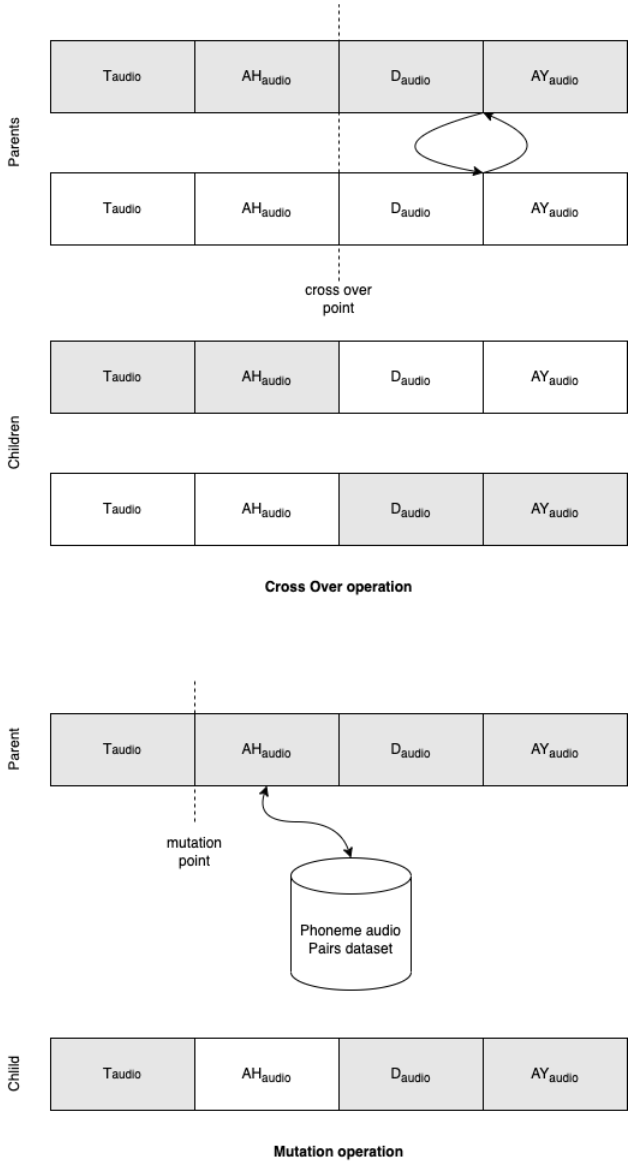


Figure 5: Crossover and mutation operations by the VocalFlows genetic algorithm for an example lyric "today". The phonemes for the lyric "today" are T AH D and AY. The audio segments for each phoneme is taken from the phoneme audio pairs dataset, and is represented as T_{audio} AH_{audio} D_{audio} and AY_{audio}. The grey and white colours are used to represent different individuals of a population in the crossover operation, and for a mutated audio segment of an individual in the mutation operation.

recognition of the lyrics due to its function to discriminate the collective audio as text. Selection of only the fittest parents for crossover results in the loss of melodic variability in the resultant samples.

3. Mutation: After crossover, the mutation step is performed on the $20 \times N$ size population with a mutation probability of 0.1. Mutation involves replacing an audio segment of the individual with another audio segment of the same phoneme from the phoneme-audio pairs dataset. The output of this stage is the same size as it's input, that it, $20 \times N$. The process is shown in Figure 5.

After mutation, the population is evaluated for fitness and the top N individuals of the population are selected through the selection step. Fitness conditions such as the $CER_{avg} \leq 0.2$ and $WER_{avg} \leq 0.3$ are checked for in every iteration of the GA. If these conditions are met, or if the budget constraints are exhausted, then the GA converges to an outcome.

4 Experiments

The top 500 songs dataset was used to train the Automatic Lyrics Transcription (ALT) model. For each song in the dataset, the words were used as the labels and the word onsets were used to extract the corresponding waveforms. The waveforms were converted into spectrograms and were used as the training data. The training data consisted of $134,960$ vocal waveforms for $6,140$ unique words, which was split into train and test sets by 70% and 30% respectively. The training loss, learning rate, CER, and WER per iteration is recorded and shown in Figure 6. For training the ALT model, the hyperparameters were set to 5×10^{-4} as the learning rate, 10 as the batch size, and 10 as the number of epochs. On evaluation on the test set, the final test set CTC loss [24] values were found to be 0.5595, average CER was 0.1578 characters per text, and average WER was 0.4669 words per text, that is a 46.7% word error rate, which is comparable with the state of the art. Training was performed on a Tesla K80 GPU with 12GB RAM and took 30 hours to converge.

The VocalFlows genetic algorithm (GA) was run to generate singing voice for the input lyrics. When the GA converges, the result is a suggestion of N vocal samples. The vocal samples can be produced by four possible methods mentioned below. These methods depend on the way in which the ALT model is trained and the type of outcome generated by the GA.

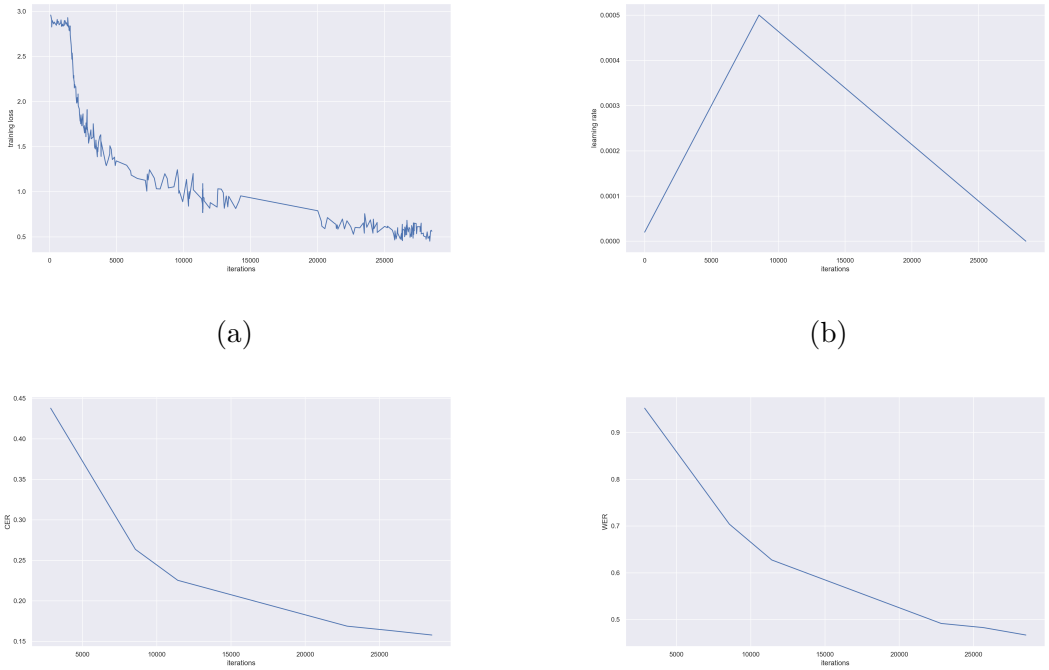
1. Phrase-to-Phrase: The ALT model is further trained on entire phrases and the GA generates phrases as one output.

2. Phrase-to-Word: The ALT model is further trained on entire phrases and the GA generates words from the same phrase one by one. The words generated are combined together to produce the output.

3. Word-to-Phrase: The ALT model is not trained further on phrases, and the GA generates entire phrases as out output.

4. Word-to-Word: The ALT model is not trained further on entire phrases and the GA generates words from the same phrase one by one. The words generated are combined together to produce the output.

The ALT model training explained in this Section, is not trained on sentences or phrases, but only on words. Therefore, the experimentation on the aforementioned methods, required



(a) (b)

(c) (d)

Figure 6: Shows the training process of the ALT model from a pre-trained Speech recognition model on the top 500 songs dataset. The graphs report the (a) CTC loss versus the no. of iterations, (b) the learning rate versus the no. of iterations, (c) the CER versus the no. of iterations, and (d) the WER versus the no. of iterations.

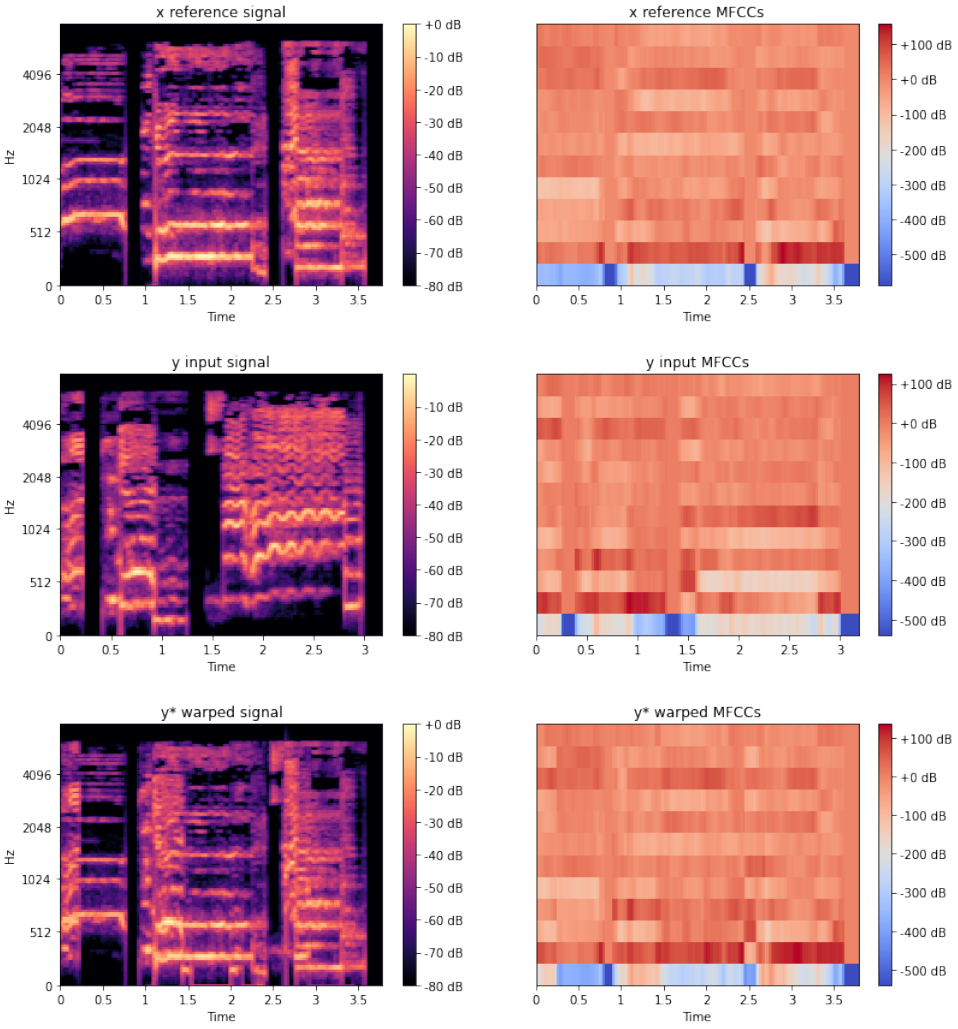


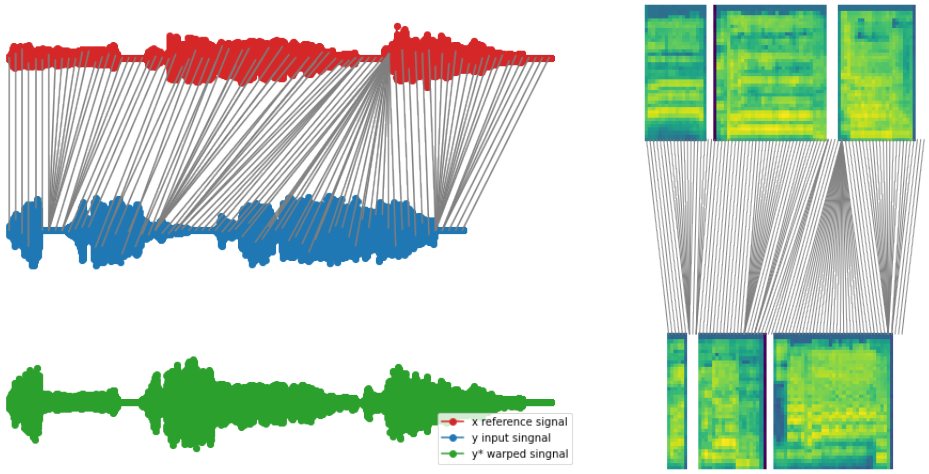
Figure 7: Shows an example of dynamic time warping of 2 vocal samples $x_{reference}$ and y_{input} for the same example phrase *a good time*. Column 1 (left) shows the mel-spectrogram of the sample and Column 2 (right) shows the MFCC features of the same corresponding sample. From top to bottom: $x_{reference}$ signal, y_{input} signal and the y^*_{warped} signal.

training the ALT model further on common phrases of word length with 3, 4 and 5. In order to avoid those phrases that are repeated in a song, the count of such phrases is limited to 3 times per song. All the training parameters for training on phrases were kept the same as the previous parameters when training for words.

4.1 Evaluation

The ALT model was compared to the state of the art (TDNN-F, MSTRE-Net) [18, 21] lyrics transcription models. The state of the art models have all been trained on different datasets. The previous work compares models to the average test set WER and hours of song audio in the train set. This comparison was similarly observed for the VocalFlows ALT model.

The results of the VocalFlows GA are singing voice audios that are suggestions to the user in terms of recommended melodies for the input lyric. Due to the subjective nature of the results they cannot be evaluated without human observation. For the purpose of doing



(a) (b)

Figure 8: Shows the dynamic time warping alignment performed for two signals of an example phrase *a good time* with respect to their (a) waveform and (b) mel-spectrograms.

so, human evaluation was conducted via a questionnaire in which 58 participants volunteered to take part. The questionnaire comprised of the following components:

1. Three audio clips that are suggestions made by the VocalFlows AI were presented to the participant. As they are recommendations, these audio clips all contain singing voices for the same input lyric. The participant is asked for the transcription of these audio clips. The transcription is observed for possible differences in the perception of the VocalFlows AI result.

2. For the same audio clips, the participants were asked for a recognition score between 1 and 5. The pronunciation of the lyric in the vocal outputs are made without a reference as seen in Section 3.2.2, therefore the score is used to determine the difficulty in recognizing the result of the VocalFlows AI.

3. For the same audio clips, the participants were asked for a melody score between 1 and 5. The melody of the vocals are also not made with a reference vocal, therefore the score is used to discern subjectively how participants value melody in the results of the VocalFlows AI.

4. An audio clip that is a suggestion made by the baseline model was presented to the participant for the same lyric as the previous questions. The participant was asked to mark a recognition score between 1 and 5 for this audio clip. This was done for the purpose of comparison on the same input given to a baseline model with random selection and the VocalFlows AI.

5. The same set of 4 questions were repeated for 4 different input lyrics.

6. The participant was asked for their music involvement in terms of playing an instrument or music production or singing. This question gives an idea of their knowledge about music and whether they find the results of the VocalFlows AI melodic and recognizable.

The results of the VocalFlows AI were also subject to objective evaluation by testing the similarity of the results with the top 500 songs dataset. The input lyrics used in the

experiments were used to extract the similar audio clips as an entire phrase from the dataset. This was possible since common phrases used in the dataset songs were used as inputs. Audio fingerprinting [9] is a technique that is used to recognize unlabelled songs by comparing their similarity against a database of songs or song metadata irrespective of the audio format. These are also known as Content Based audio Identification systems (CBID). CBID systems have matching algorithms that are used in linking unlabelled songs with the labelled match.

Mel-frequency cepstral coefficients (MFCC) [64] are audio features that are commonly used in CBID systems for audio fingerprinting [53, 36, 13, 23]. Techniques shown in voice recognition CBID systems [43] are similarly followed to develop a matching algorithm. This involves the following steps:

1. The unlabelled audio U and the labelled audio V are inputs to the matching algorithm.
2. Twenty six MFCC audio features [26] are extracted for both the audio. This is shown in equation 4. Here, X represents the audio U or V, x represents the MFCC features of shape $26 \times t_X$, where t_X represents the no. of time intervals or the duration of the audio X and 26 is the amount of MFCCs extracted.

$$MFCC(X) = x \in \mathbb{R}^{26 \times t_X}, \tag{4}$$

3. The MFCCs of the labelled audio $v \in \mathbb{R}^{26 \times t_V}$ are aligned with that of the unlabelled audio $u \in \mathbb{R}^{26 \times t_U}$ using dynamic time warping (DTW) [44]. The DTW process requires an additional parameter to use as a distance measure. For this the cosine distance was used. The DTW returns the cost matrix, and the alignment path. The alignment path can be used to stretch or shrink the labelled audio V at different time intervals producing $V*$ to match the unlabelled audio U.
4. The two audio signals $V*$ and U, both having time duration of t_U, can now be compared using cross correlation to find the similarity. The similarity score is given as the Pearson correlation coefficient [50].
5. With "thresholding" of the similarity score, a classification of similar audio can be made.

Using this method, audio fingerprinting or CBID system is developed to match similar audio together. The outputs of the VocalFlows AI and select audio segments from top 500 songs dataset are matched and a similarity score is measured. It can be seen in Figure 7 the process of dynamic time warping of an input signal y_{input} to be stretched to the size of a reference signal $x_{reference}$ before finding the similarity measure between the two. The signals are examples of phrase *a good time* generated by the VocalFlows AI. The resultant y^*_{warped} signal has the duration of the $x_{reference}$ signal allowing for cross correlation. The waveforms of the same example are shown in 8 (a) to show the dynamic time warping process. The grey lines in the image show the points of the signals that are to be aligned with one an another.

5 Results

On comparing the VocalFlows ALT model with the state of the art models: TDNN-F [18], and MSTRE-NET [21], it is observed that the VocalFlows ALT model performs similarly. This is shown in Table 2.

The task for which the VocalFlows ALT model was trained for differs from that of the state of the art models, as it was required to only discriminate words generated by the GA

Model	$\mathbf{WER}_{avg}$	Dataset	Hours	N-gram
TDNN-F	42.3%	DSing1	15.1	3
MSTRE-NET	42.1%	DALI	156.0	4
VocalFlows ALT	46.7%	Top 500 songs	18.4	1

Table 2: Shows the comparison between three ALT models: TDNN-F, MSTRE-NET and VocalFlows ALT on the basis of the average test set WER %, training dataset used, hours of song data used in training and the n-gram or number of words in the labelled text used in training.

and not short phrases. The state of the art models were trained on N-gram text corpus that were larger than 1, which implies that the model was required to transcribe lyrics of word lengths 3 and 4 instead of individual words. The WER_{avg} % is lower with 46.7% than the TDNN-F (42.3%) and MSTRE-NET (42.1%) models as can be seen in Table 2 but for different reasons in both cases.

The DSing1 and DALI datasets are human annotated and high quality of song recordings data, whereas the Top 500 songs dataset is obtained from websites and is aligned without manual effort. Compared to MSTRE-NET being trained on 156 hours of singing data, the VocalFlows ALT model is only trained on 18.4 hours. The TDNN-F model and MSTRE-NET model both train on labelled text that allows for contextual inference. By training only on words due to the specific requirement of the task, the context information between words is not available.

The four methods involving input to the ALT model and the generation of the vocal samples of the VocalFlows GA was experimented on. These methods are phase-to-phase, phase-to-word, word-to-phase and phase-to-word. The same lyric was used to test these four methods, the process of which is explained in Section 4. An objective evaluation of the same four methods were carried out and the results can be seen in Table 3 for the phrase "i love you". The experiment compares the average similarity of the audio waveforms of the commonly sung phrase "i love you" extracted from the top 500 songs dataset, produced by the VocalFlows AI and the combined average similarity of both the dataset and the model output.

It can be seen that the method that has the highest correlation with the dataset, as well as within the outputs of the model is the word to word method. The samples produced by the four methods were manually observed for subjective evaluation. The word to word method produced better enunciated words in the composition, but the remaining had no noticeable difference in recognition, or melody. The word-to-word method was found to converge faster, with clearer audio for the words in the composition and was used as the standard method. This method also allows for the combination of words that do not exist in the dataset.

The results of the objective evaluation for two sets of input phrases are shown in Table 3. Set (1) exists in the top 500 songs dataset and their average similarity scores are calculated in addition to the output of the VocalFlows AI. Set (2) consists of phrases that do not exist in the dataset and was produced only by the VocalFlows AI. The similarity score between two signals defined by their cross correlation coefficient, informs of the signals relation in terms of their waveform. The maximum score that can be achieved is 1 and is mostly the case if the two signals being compared are exactly the same. For this purpose the same signals were never compared.

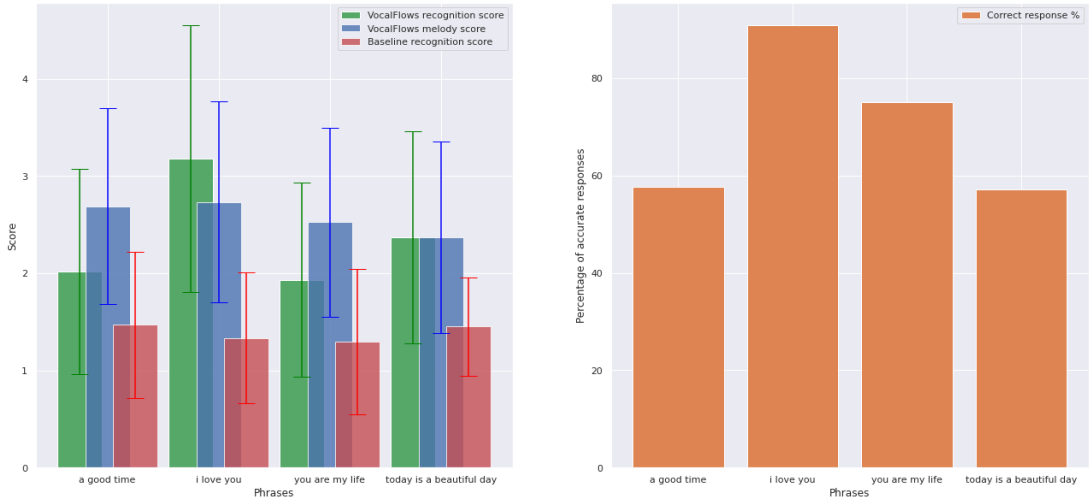
(Set) Phrase/Lyric		$\textbf{corr}_{dataset}$	$\textbf{corr}_{VocalFlows}$	$\textbf{corr}_{combined}$
(1) a good time		0.683 ± 0.103	0.735 ± 0.072	0.694 ± 0.105
(1) i love you	phrase to phrase	0.705 ± 0.076	0.723 ± 0.018	0.708 ± 0.072
	phrase to word		0.722 ± 0.001	0.707 ± 0.074
	word to phrase		0.708 ± 0.008	0.708 ± 0.072
	word to word		0.767 ± 0.056	0.716 ± 0.075
(2) you are my life		*N/A*	0.745 ± 0.028	*N/A*
(2) today is a beautiful day		*N/A*	0.739 ± 0.048	*N/A*

Table 3: Shows the cross correlation similarity scores (max similarity = 1) of different phrases sung by the VocalFlows AI. Two sets of phrases were chosen: (1) phrases that exist in the top 500 songs dataset and (2) phrases that do not exist in the top 500 songs dataset. The scores are calculated between audio found within the dataset under $\text{corr}_{dataset}$ column, between audio produced by the VocalFlows AI under $\text{corr}_{VocalFlows}$ column, and for the combined audio under the $\text{corr}_{combined}$ column. For the phrase "i love you", the results of the method objective experiments explained in Section 4 are also calculated.

The average similarity score gives an idea of how variation in melody and enunciation of the same phrase between multiple signals are correlated. If the phrase has a lower score, then it implies that the signals sound differently from each other, although they sing the same phrase. Therefore this score is not an ideal measurement for capturing differences in melody, but it provides information on recognition of the signals. If the average score is high, then their similarity in terms of recognition of the phrase is high, and it is assumed to have inherent melody variations. Table 3 shows that the comparison of the similarity scores. The dataset similarity scores are lower due to the higher variation in melody of the songs. Since the ALT model is trained to transcribe words, a higher fitness vocal is aimed at being more recognisable.

A questionnaire survey was conducted where 58 participants performed a subjective human evaluation of the results of the VocalFlows AI and the baseline model. This involved transcription of the vocals and scoring the outputs on recognition and melody. The results of which are shown in Figure 9. The participants found that the lyrics were consistently easier to transcribe on the VocalFlows output than on the baseline output for the same vocal. A 95.6% of participants found the VocalFlows AI to be more recognisable than the baseline model. Although the lyrics were recognisable, the participants found the vocal suggestions to be more melodic than recognisable. 73.9% of participants scored melody higher for the outputs than recognition. When transcribing any song there are always inconsistencies in results between participants and is not an easy task to obtain accurate results for. For example, the phrase "you are my life" was transcribed accurately by 75% of the participants, but was commonly mistaken for the phrase "you are my love" and "you are my high".

The results show that phrases that do not entirely exist in the top 500 songs dataset are equally recognisable and melodic compared to those that do. It was observed that phrases with multiple syllables are harder to generate accurately by the VocalFlows AI such as in the phrase "today is a beautiful day" wherein the word "beautiful" was the most inaccurately identified word. Despite the subjective difficulties, the consensus of the survey is that the



(a) (b)

Figure 9: Shows the (a) melody score and recognition score of the output of the VocalFlows AI compared with the recognition score of the baseline model on a scale of 1 to 5, and (b) percentage of correct responses for the four phrases used the in the questionnaire.

participants agree that the VocalFlows AI produces better outputs than the baseline model in terms of recognition, and agree that there exists inherent melody in the output samples of the VocalFlows AI. To assess the credibility of the participants in their evaluation it is noted that 58% of the participants are involved in making music either by singing, music production or playing instruments.

6 Discussion

The VocalFlows AI requires the training of the ALT model, which took 30 hours to converge on 12GB RAM. Networks that are used in Open AIs Jukebox [22] are heavy and require 1.2 million songs of training data with more than 10,000 epochs to train. Expensive machinery having GPUs with over 50GBs of RAM would take more than a week to train on. Although these networks are the state-of-the-art, they are cumbersome to train and they are difficult to bend to the needs of specific tasks. A big inhibition to the implementation of SVS systems is the large amounts of data required and the heavy machinery for training. DeepSinger [57] performs the crawling of web data and uses their dataset for singing voice synthesis similar to VocalFlows AI, but the dataset or the code to replicate this data is not made publicly available. Although this work has similarities with VocalFlows AI, the architectures used in their alignment model and singing model are also computationally expensive and require large overhead from previous model architectures such as FastSpeech [56] and WaveNet [47] that DeepSinger is based on.

The evaluation of the results is mainly done through human recognition and melodic inferences. These results give insight into the use case of the VocalFlows AI. The survey shows that majority of the participants are musically inclined, but the subjectivity of the task is more varied than that. Closed silences between words imply faster and shorter phrases, which is subjectively Hip-hop and Rap music whereas open silences imply longer and airy phrases, which is subjectively Rock, R&B, and Pop Music. These are the most

common genres in the top 500 songs dataset, but the results are not conditioned on genres. The faster vocals are harder to interpret than the longer vocals, which could influence human recognition, but silences between phonemes are not manipulated by the VocalFlows AI.

6.1 Limitations

There do exist limitations with VocalFlows AI and are as follows:

- The model currently only works with phonemes of the English Alphabet. From the data collection to the processing of the data, all of the songs are in English. But these can be substituted for any language. The VocalFlows AI is not language dependent and is flexible in this respect, provided the lyrics alignment model is also trained on the required language. However, this has not been experimented with in this work.
- The VocalFlows AI currently only uses a fitness function that is trained on word-level lyrics recognition and not sentence level. Works such as Jukebox [22] and DeepSinger [57] use lyrics of the entire song at once. Other ALT models [21, 20, 18] use N-grams of size 3 or 4 whereas VocalFlows is currently limited to 1. This brings about a loss in contextual information, which the VocalFlows ALT model has the potential to learn. Due to the requirements of the task and the word-to-word generative method used by the genetic algorithm approach, the 1-gram lyrics recognition ALT model sufficed.
- For the sake of being fully creative, the VocalFlows AI does not benefit from pitch, duration, timbre and tone audio features as it currently does not control any of these parameters, which most SVS systems do. The VocalFlows AI therefore has no conditioning on genre or artist and has no control on the melody of the outcome. The melody is a requirement that the VocalFlows AI wants to achieve but is left inherent to the generated outputs.
- With the noticeable overhead of large networks and datasets required for modelling singing voice using transformers or GANs, the recognisability aspect of the objective task was prioritised as human recognition of the lyrics of outputs was lacking. Due to this requirement and the availability of the inspiration set, the evolutionary approach was suitable and achieved the necessary recognition of lyrics as seen in the survey. The results however are not recognised by all the participants but simply by a majority. A reason for such could be the manipulation in resembling vocal structure in songs and the silences between them. This is currently limited in the VocalFlows AI and is manipulated only by using silences of 200ms in between words.

The gaps in quality caused by the aforementioned limitations could be bridged with advanced deep learning methods, but the attempt to achieve the outcome using evolutionary methods was performed and observed to bring about creative results. Genetic algorithms are versatile methods and can be used without training or without transfer learning any contextual conditioning that are often used in deep learning methods [57, 37, 14].

6.2 Future Work

The VocalFlows AI can be applied for inspiring creativity in song writing blocks opening up new melodies that are generated without the influence of an instrumental track. Currently only working in languages that have the English phonetic, but future work could be to improve the set of phonetic capabilities of the AI. The VocalFlows method achieves the goal

of suggesting vocal samples given an input lyric, but does not enhance the melodic aspect of the vocals. The melody contained in the outputs are a byproduct of the phonetic combination of the vocals done by the genetic algorithm. VocalFlows AI does not control audio features such as pitch, duration or tone. In music production, these audio features are controlled by expert sound engineers to provide further melodic suggestions to the user.

To use such features requires fundamental understanding in music theory, but many previous works have left the sound engineering to deep neural networks [22, 34, 57]. Advances in deep learning, transformers, and audio processing have shown neural networks to be capable of converting the pitch of an audio to a reference pitch without changing musical content [38], producing singing voice from speech audio features [12], and also producing entire songs with lyrics of popular artists such as Elvis and Sinatra [22]. These works combine the latent spaces of reference inputs with the latent space of generated outputs to produce audio effects and vocal samples of different melody, pitch or tone.

This differentiates the VocalFlows AI as it uses evolutionary algorithms and does not use traditional audio processing methods. The limitation is that these methods are heavy and come with large overheads. Memory efficient and lightweight methods such as those used in VocalFlows are more flexible and necessary to develop. However, few components of the heavier methods can be incorporated to improve melody and recognition of the results such as context-based learning and pitch information. The lyrical context of commonly used sequence of words can help leverage the recognisability of the lyrics. Similarly the pitch of each sung phoneme can be observed and the sequence-to-sequence pitch information can be learned to better predict what fundamental frequencies work together in providing aesthetic melodies.

7 Conclusion

The work contributes to the field of audio processing, and computational creativity in the following ways:

- A new open source dataset, the top 500 songs dataset, with the code to create similar datasets is made publicly available. This dataset allows for Web crawling of songs from online websites as well as the synchronisation of the lyrics to produce high quality vocal audio files with corresponding text and duration of the texts sung in the corresponding vocal audio files.
- A new dataset, the phoneme-audio pairs dataset is made publicly available. This dataset is used for representation of the inputs in the VocalFlows genetic algorithm. It is a unique dataset that allows for evolutionary approaches in audio processing.
- An automatic lyrics transcription model (ALT) trained on words for the specific task of word-to-word generation in the VocalFlows AI. The ALT model was also trained on n-gram sentences upto a length of 5 and thereby has the potential to fit tasks that require phrase-to-phrase or phrase-to-words transcription.
- A genetic algorithm that evolves phonemes into words with inherent melody for synthesising singing voice. This is a novel technique used in the VocalFlows AI to suggest vocal flows to users.
- An open source audio fingerprinting technique for the objective evaluation of the similarity between audio signals using MFCC features and dynamic time warping.

The objective and subjective results show that the VocalFlows AI produces outputs that have melodic suggestions to its user. The end-to-end singing voice synthesis is performed with original data that is Web crawled and aligned to be synchronised as required inputs to the VocalFlows AI. The novel computational creative approach using genetic algorithms in audio processing is a step in the direction of creative singing voice music generation systems.

Acknowledgement

I would like to thank Dr. Peter van der Putten for being my primary supervisor to oversee my work along with my secondary supervisor Dr. Rob Saunders. You both have guided me providing support and necessary mentoring to give shape to my thesis. I would also like to thank all the participants who took part in the survey for taking the time and effort to evaluate my work.

References

[1] *INTERSPEECH 2006 - ICSLP, Ninth International Conference on Spoken Language Processing, Pittsburgh, PA, USA, September 17-21, 2006*. ISCA, 2006.

[2] R. Al-Rfou, B. Perozzi, and S. Skiena. Polyglot: Distributed word representations for multilingual NLP. *CoRR*, abs/1307.1662, 2013.

[3] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, E. Elsen, J. Engel, L. Fan, C. Fougner, T. Han, A. Hannun, B. Jun, P. LeGresley, L. Lin, S. Narang, A. Ng, S. Ozair, R. Prenger, J. Raiman, S. Satheesh, D. Seetapun, S. Sengupta, Y. Wang, Z. Wang, C. Wang, B. Xiao, D. Yogatama, J. Zhan, and Z. Zhu. Deep Speech 2: End-to-end speech recognition in English and Mandarin, 2015.

[4] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere. The Million Song Dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.

[5] J. Biles et al. Genjam: A genetic algorithm for generating jazz solos. In *ICMC*, volume 94, pages 131–137. Ann Arbor, MI, 1994.

[6] M. Blaauw and J. Bonada. A neural parametric singing synthesizer. *arXiv preprint arXiv:1704.03809*, 2017.

[7] M. A. Boden. Creativity and artificial intelligence. *Artificial intelligence*, 103(1-2):347–356, 1998.

[8] J. Briot, G. Hadjeres, and F. Pachet. Deep learning techniques for music generation - A survey. *CoRR*, abs/1709.01620, 2017.

[9] P. Cano, E. Batle, T. Kalker, and J. Haitsma. A review of algorithms for audio fingerprinting. In *2002 IEEE Workshop on Multimedia Signal Processing.*, pages 169–173, 2002.

[10] F. Carnovalini and A. Rodà. Computational creativity and music generation systems: An introduction to the state of the art. *Frontiers in Artificial Intelligence*, 3, 2020.

[11] P. Chandna, M. Blaauw, J. Bonada, and E. Gómez. Wgansing: A multi-voice singing voice synthesizer based on the wasserstein-gan. *CoRR*, abs/1903.10729, 2019.

[12] F. Chen, R. Huang, C. Cui, Y. Ren, J. Liu, and Z. Zhao. Singgan: Generative adversarial network for high-fidelity singing voice generation, 2021.

[13] J. Chen and T. Huang. A robust feature extraction algorithm for audio fingerprinting. In *Pacific-Rim Conference on Multimedia*, pages 887–890. Springer, 2008.

[14] J. Chen, X. Tan, J. Luan, T. Qin, and T.-Y. Liu. Hifisinger: Towards high-fidelity neural singing voice synthesis. *arXiv preprint arXiv:2009.01776*, 2020.

[15] Y.-R. Chien, H.-M. Wang, and S.-K. Jeng. Alignment of lyrics with accompanied singing audio based on acoustic-phonetic vowel likelihood modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(11):1998–2008, 2016.

[16] Y.-P. Cho, F.-R. Yang, Y.-C. Chang, C.-T. Cheng, X.-H. Wang, and Y.-W. Liu. A survey on recent deep learning-driven singing voice synthesis systems. In *2021 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, pages 319–323, 2021.

[17] S. Colton and G. A. Wiggins. others. 2012. Computational creativity: The final frontier. In *Proceedings of the 20th European Conference on Artificial Intelligence*, pages 21–26.

[18] G. R. Dabike and J. Barker. Automatic Lyric Transcription from karaoke vocal tracks: Resources and a baseline System. In *Proc. Interspeech 2019*, pages 579–583, 2019.

[19] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson. FMA: A dataset for music analysis. In *18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017.

[20] E. Demirel, S. Ahlbäck, and S. Dixon. Automatic lyrics transcription using dilated convolutional neural networks with self-attention. In *International Joint Conference on Neural Networks*. IEEE, 2020.

[21] E. Demirel, S. Ahlbäck, and S. Dixon. Mstre-net: Multistreaming acoustic modeling for automatic lyrics transcription. *CoRR*, abs/2108.02625, 2021.

[22] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever. Jukebox: A generative model for music, 2020.

[23] F. Faraji, Y. Attabi, B. Champagne, and W.-P. Zhu. On the use of audio fingerprinting features for speech enhancement with generative adversarial network. In *2020 IEEE Workshop on Signal Processing Systems (SiPS)*, pages 1–6. IEEE, 2020.

[24] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 369–376, New York, NY, USA, 2006. Association for Computing Machinery.

[25] M. Guzdial and M. O. Riedl. An interaction framework for studying co-creative AI. *CoRR*, abs/1903.09709, 2019.

[26] M. R. Hasan, M. M. Hasan, and M. Z. Hossain. How many mel-frequency cepstral coefficients to be utilized in speech recognition? a study with the Bengali language. *The Journal of Engineering*, 2021(12):817–827, 2021.

[27] R. Hennequin, A. Khlif, F. Voituret, and M. Moussallam. Spleeter: a fast and efficient music source separation tool with pre-trained models. *Journal of Open Source Software*, 5(50):2154, 2020. Deezer Research.

[28] D. Herremans, C.-H. Chuan, and E. Chew. A functional taxonomy of music generation systems. *ACM Comput. Surv.*, 50(5), sep 2017.

[29] J. H. Holland. Genetic algorithms. *Scientific american*, 267(1):66–73, 1992.

[30] R. Huang, F. Chen, Y. Ren, J. Liu, C. Cui, and Z. Zhao. Multi-singer: Fast multi-singer singing voice vocoder with a large-scale corpus. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3945–3954, 2021.

[31] D. Iclănzan. The creativity potential within evolutionary algorithms. In *European Conference on Artificial Life*, pages 845–854. Springer, 2007.

[32] J. Kim, H. Choi, J. Park, M. Hahn, S. Kim, and J.-J. Kim. Korean singing voice synthesis system based on an lstm recurrent neural network. In *Proc. Interspeech*, pages 1551–1555, 2018.

[33] A. Koestler. The act of creation. In *Brain Function, Volume IV: Brain Function and Learning*, pages 327–346. University of California Press, 2020.

[34] J. Lee, H. Choi, C. Jeon, J. Koo, and K. Lee. Adversarially trained end-to-end korean singing voice synthesis system. *CoRR*, abs/1908.01919, 2019.

[35] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu. Neural speech synthesis with transformer network. AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019.

[36] C.-C. Liu and P.-F. Chang. An efficient audio fingerprint design for mp3 music. In *Proceedings of the 9th International Conference on Advances in Mobile Computing and Multimedia*, pages 190–193, 2011.

[37] J. Liu, C. Li, Y. Ren, F. Chen, P. Liu, and Z. Zhao. Diffsinger: Diffusion acoustic model for singing voice synthesis. 2021.

[38] J. Lu, K. Zhou, B. Sisman, and H. Li. Vaw-gan for singing voice conversion with non-parallel training data, 2020.

[39] B. Manaris, D. Hughes, and Y. Vassilandonakis. Monterey mirror: combining markov models, genetic algorithms, and power laws. In *Proceedings of 1st Workshop in Evolutionary Music, 2011 IEEE Congress on Evolutionary Computation (CEC 2011)*, pages 33–40. IEEE New York, NY, 2011.

[40] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger. Montreal forced aligner: Trainable text-speech alignment using kaldi. In *Interspeech*, volume 2017, pages 498–502, 2017.

[41] G. Meseguer-Brocal, A. Cohen-Hadria, Gomez, and P. Geoffroy. DALI: a large dataset of synchronized audio, lyrics and notes, automatically created using teacher-student machine learning paradigm. In ISMIR, editor, *19th International Society for Music Information Retrieval Conference*, September 2018.

[42] F. Mo, X. Wang, S. Li, and H. Qian. A music generation model for robotic composers. In *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1483–1488, 2018.

[43] L. Muda, M. Begam, and I. Elamvazuthi. Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques. *CoRR*, abs/1003.4083, 2010.

[44] M. Müller. *Fundamentals of Music Processing – Audio, Analysis, Algorithms, Applications*. 01 2015.

[45] M. Nishimura, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda. Singing voice synthesis based on deep neural networks. In *INTERSPEECH*, 2016.

[46] T. Okamoto, T. Toda, Y. Shiga, and H. Kawai. Real-time neural text-to-speech with sequence-to-sequence acoustic model and waveglow or single gaussian wavernn vocoders. In *INTERSPEECH*, pages 1308–1312, 2019.

[47] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[48] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210, 2015.

[49] J. Park, Kyubyong & Kim. g2pe. https://github.com/Kyubyong/g2p, 2019.

[50] K. Pearson and F. Galton. Vii. note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58(347-352):240–242, 1895.

[51] V. Peddinti, D. Povey, and S. Khudanpur. A time delay neural network architecture for efficient modeling of long temporal contexts. In *Sixteenth annual conference of the international speech communication association*, 2015.

[52] S. Phon-Amnuaisuk, A. Tuson, and G. Wiggins. Evolving musical harmonisation. In *Artificial Neural Nets and Genetic Algorithms*, pages 229–234. Springer, 1999.

[53] I. M. Pires, R. Santos, N. Pombo, N. M. Garcia, F. Flórez-Revuelta, S. Spinsante, R. Goleva, and E. Zdravevski. Recognition of activities of daily living based on environmental analyses using audio fingerprinting techniques: A systematic review. *Sensors*, 18(1):160, 2018.

[54] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, et al. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, number CONF. IEEE Signal Processing Society, 2011.

[55] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner. The MUSDB18 corpus for music separation, Dec. 2017.

[56] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu. Fastspeech: Fast, robust and controllable text to speech. *Advances in Neural Information Processing Systems*, 32, 2019.

[57] Y. Ren, X. Tan, T. Qin, J. Luan, Z. Zhao, and T.-Y. Liu. Deepsinger: Singing voice synthesis with data mined from the web. New York, NY, USA, 2020. Association for Computing Machinery.

[58] G. Ritchie. Assessing creativity. In *Proc. of AISB'01 Symposium*. Citeseer, 2001.

[59] E. Schubert, S. Canazza, G. De Poli, and A. Rodà. Algorithms can mimic human piano performance: the deep blues of music. *Journal of New Music Research*, 46(2):175–186, 2017.

[60] K. Schulze-Forster, C. S. J. Doire, G. Richard, and R. Badeau. Phoneme level lyrics alignment and text-informed singing voice separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:2382–2395, 2021.

[61] X. Tan, T. Qin, F. Soong, and T.-Y. Liu. A survey on neural speech synthesis, 2021.

[62] M. Umbert, J. Bonada, M. Goto, T. Nakano, and J. Sundberg. Expression control in singing voice synthesis: Features, approaches, evaluation, and challenges. *IEEE Signal Processing Magazine*, 32(6):55–73, 2015.

[63] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[64] M. Xu, L.-Y. Duan, J. Cai, L.-T. Chia, C. Xu, and Q. Tian. Hmm-based audio keyword generation. In K. Aizawa, Y. Nakamura, and S. Satoh, editors, *Advances in Multimedia Information Processing - PCM 2004*, pages 566–574, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[65] D. Yu and L. Deng. *Automatic speech recognition*, volume 1. Springer, 2016.

[66] L. Yujian and L. Bo. A normalized levenshtein distance metric. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1091–1095, 2007.

[67] C. Zhang, J. Yu, L. Chang, X. Tan, J. Chen, T. Qin, and K. Zhang. Pdaugment: Data augmentation by pitch and duration adjustments for automatic lyrics transcription, 2021.

[68] H. Zhu, S. Wang, and Z. Wang. Emotional music generation using interactive genetic algorithm. In *2008 International Conference on Computer Science and Software Engineering*, volume 1, pages 345–348, 2008.

[69] J. Zhu, C. Zhang, and D. Jurgens. Phone-to-audio alignment without text: A semi-supervised approach. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.