Automatic Singing Transcription

Victor Shan

December 1, 2023

Abstract

1 Introduction

Songs play an important part of society because they are conduits of cultural expression, foster emotional connection and is a great source of entertainment. Automatic Singing Transcription (AST) is a branch of Automatic Speech Recognition (ASR) can facilitate a deeper way to interact with it. In ASR, the goal is to transcribe normal speech to text, whether that's a conversation between two people, a lecture, or a podcast. Examples of ASR include Youtube's auto-generated captions, assistants like Siri and Alexa, and even live captioning of meetings through applications like Zoom. AST is a similar task, but instead of transcribing normal speech, AST transcribes singing. Applications of AST range from karaoke applications to Music Information Retrieval (MIR) systems. These MIR systems can be used to search for songs based on lyrics, categorize songs based on their lyrics, or even aid in the generation of new songs.

While AST is similar to ASR, there are a few key differences that make it a more challenging task. First, singing is a more complex signal than normal speech. Singing has a more dynamic signal, with more variation in pitch, duration and vibrato [1]. For example, vowels sounds are often held for longer in singing than in normal speech like dragging the last part of "bye" in word "goodbye". Songs also often include music in the background, which can make it more difficult to distinguish between the singer's voice and the background music. Finally, there is a lack of large, publicly available datasets for AST [2]. This is in contrast to ASR, where there are many large datasets available, such as LibriSpeech [3]. This lack of data makes it especially difficult to train AST models.

There are many possible outputs for AST but one of the most detailed and useful outputs is the time-aligned phoneme sequence. This is a sequence of phonemes, or the smallest unit of sound in a language, that is aligned with the audio. Time-aligned phoneme sequences are useful because it can contains enough information to reconstruct the not only the lyrics, but also how they match with the music. This output can be used to generate a karaoke application that highlights the lyrics as they are sung, or even train a model to sing covers of songs. Word-level alignement is an easier alternative but it misses the crucial information of the duration of phonemes within words. The ideal output is one that includes even more information such as the notes to sing at so that singers would be able to read and sing the song without any additional information. In this paper, we will mainly focus on the phoneme-level alignment but use word-level alignment to compare the performance of our model to other models.

2 Datasets

2.1 Requirements

The requirements of a good dataset are audio files, lyrics and timestamps. The audio files can be in any format such as mp3 or wav. The lyrics can be in any format as well, but the timestamps must be in a format that can be used to align the lyrics with the audio. The timestamps can be in the form of a phoneme sequence or word sequence but should also include the onset and offeset times (beginning and end). Sentence level timestamps are too coarse and would need to be broken down into word or phoneme level timestamps. Most lyrics used by popular music services such as Spotify only use sentence/line level timestamps. Ideally, the dataset would also be large enough to train deep neural networks and publicly available so that other researchers can use it and compare their models to existing models.

2.2 Challenges

With these requirements in mind, it is easy to see why there are so few datasets available. First, it is difficult to create a dataset with lyrics and timestamps. The lyrics must be manually transcribed and the timestamps must be manually aligned with the audio. This is a time consuming process that requires a lot of effort. Second, it is difficult to obtain the rights to use the audio files. Most songs are owned by record labels and it is difficult to obtain the rights to use these files. These two challenges make it difficult to create new datasets and is the reason why there are so few datasets available and why most of the existing datasets are small.

2.3 Dataset Augmentation

2.3.1 SpecAugment

Due to the lack of datasets, it is more important to make the most of the existing datasets. One way to do this is to augment the existing datasets. SpecAugment is a series of techniques that augment the audio spectrogram to improve the performance of ASR models. A frequent intermediate representation are Mel-Frequency Cepstral Coefficients (MFCC). These are image representations of energy at certain frequencies on a scale that more closely matches human hearing [4]. These techniques include time warping, frequency masking and time masking [5]. Time warping is a technique that stretches or compresses the audio spectrogram in the time dimension. Frequency masking is a technique that masks a random number of frequency channels in the audio spectrogram. Time masking is a technique that masks a random number of time steps in the audio spectrogram. These techniques are used to augment the audio spectrogram before it is fed into the ASR model. [5]

2.3.2 Transforming Existing Datasets Into Pseudo-Singing Datasets

Another way to augment the existing datasets is to transform the existing datasets. This can be done by shifting the pitch, duration or vibrato [6]. The advantage of this technique is that is can also be applied to speech datasets and transform them into pseudo-singing datasets. The disadvantage is that the results will contain artifacts from the transformation based on the techiniques applied. Neural network models showed an almost 15% improvement on the transformed TIMIT dataset than ones trained on the original TIMIT dataset [6].

2.3.3 Transforming Utterance level datasets into Phoneme level datasets

A technique from ASR that can also be applied to AST is transforming utterance/sentence level datasets into phoneme level datasets. There are many ASR datasets that contain single utterances such as LibriSpeech and MUSDB18 [3], [7]. These datasets can be effectively transformed into phoneme level datasets by using a phoneme dictionary such as the CMU Pronouncing Dictionary [8]. This dictionary contains a mapping from words to phonemes. Using this dictionary, the utterances can be transformed into phoneme sequences. These sequences can then be used with a Connectionist Temporal Classification (CTC) loss function to train AST models [9]. CTC allows for the model to output a sequence of phonemes per time step and the repetitions are collapsed into a single phoneme as shown in Figure 1. This technique was used to allow a model to incorperate the utterance level LibriSpeech dataset into the training for a phoneme ASR model [10]. Timestamps can be retrieved from the pre-CTC output that had time-aligned phoneme classifications. The same process can be applied to AST models to allow them to use the utterance level song datasets.

2.3.4 Teacher-Student Approach

The teacher-student approach is inspired by the technique of the same name that was intended to reduce the size of large deep neural networks. The idea was first train a large deep neural network, the teacher, and then train a smaller deep neural network, the student, to mimic the teacher [11]. However, this technique can also be applied in cases of

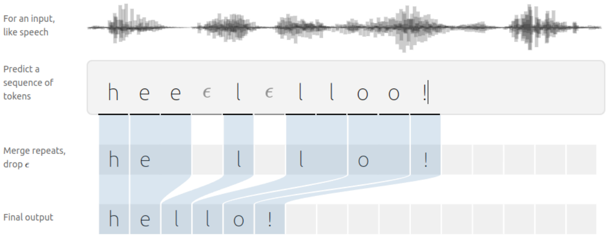

Figure 1: CTC Function collapsing repetitions [9]

low labeled data availablility but high unlabeled data availablility. To start off, a model would be trained on a small dataset of labeled data. Then, the model would be used to label a large dataset of unlabeled data. Finally, a new model would be trained on the newly labeled dataset [2]. This has proven to be effective for transcribing drums in music with the student model outperforming the teacher model [12].

2.4 Existing Datasets

2.4.1 TIMIT (ASR)

TIMIT is a dataset of speech recordings of 630 speakers of eight major dialects of American English with time-aligned phoneme sequences [13]. It is a popular dataset for ASR and has been used to train many ASR models. However, since this is not a singing dataset, it does not contain any of the characteristics of singing such as pitch, duration or vibrato. This is an excellent dataset to apply transformation into a pseudo-singing dataset metioned in 2.3.2. The popularity of this dataset makes it a good candidate for applying transformations to create a pseudo-singing dataset and also for a general benchmark to compare against other models.

2.4.2 LibriSpeech (ASR)

LibriSpeech is a 1000 hour dataset of audiobook recordings where each recording has a matching sentence [3]. This dataset is also a popular dataset for ASR and has been used to train many ASR models including recent breakthroughs like wav2vec 2.0 [10]. This dataset is also a good candidate for training AST models because it is large and publicly available. This dataset is special because of the clarity of the audio and previous success by other models such as the wav2vec 2.0 model in detecting phonemes in this dataset when fine tuned with the TIMIT dataset [10]. This is done through the CTC technique mentioned in 2.3.3. Since speech and singing both use the same phonemes, because they are in the same language, this large dataset can be used to train base models before fine-tuning them AST models.

2.4.3 Jamendo Dataset

This dataset one of the most popular datasets for AST and has been used by many state-of-the-art AST models. It contains 20 English songs and 60 songs in other languages with word-aligned timestamped sequences [14]. This dataset is a good candidate for training AST models because it is publicly available and it's popularity makes it an excellent benchmark to compare against other models. However, it is still a relatively small dataset and does not contain any phoneme sequences. The authors of this dataset were able to achieve a 77.8% Word Error Rate (WER) which still leaves a lot of room for improvement.

2.4.4 Children's Songs Dataset

This is a dataset of 50 English and 50 Korean children's songs with grapheme level timestamps and pitches. This dataset was sung by a single professional Korean singer in two separate keys for each song [**ChildrensSongs**]. This dataset is a good candidate for training AST models because it includes the onset and offset information, pitch, and grapheme level timestamps that could be preprocessed to phoneme or word level timestamps. It is also a very clean dataset because of the professional performance and setting. The downside of this dataset is that graphemes are not as popular as phonemes, word or utterance labels in either ASR or AST.

2.4.5 MUSDB18

MUSEDB18 is a dataset of 150 songs with isolated vocals and accompaniment tracks [7]. This dataset is a good candidate for training AST models because it is publicly available and it has clean isolated vocals. The downside of this dataset is that it doesn't contain any word level timestamps. However, with the CTC technique mentioned in 2.3.3, this dataset can still be used to train AST models.

2.4.6 NUS Dataset

This dataset is one of the few datasets that contains phoneme level timestamps [15]. There are 169 minutes of 20 unique English songs by 12 different people. The CMUDict was used for the phoneme vocabulary and timestamps were manually annotated. This dataset is the ideal dataset type for training AST models due to this level of detail. It also includes a mix of slow to fast melodies annd a balanced gender distribution [15].

2.4.7 Free Music Archive

The Free Music Archive (FMA) dataset is a dataset of 106,574 tracks with 161 genres [16]. This dataset is not a good candidate for direclty training AST models because it does not contain any lyrics and some music may not even contain any singing at all. However, it is a good source of unlabeled songs that could be labeled through the teacher-student technique in 2.3.4. It can also provide a good source of general singing audio that can be used in the training of wav2vec2.0 models that will be described in section 3.2.

2.4.8 VocalSet

VocalSet is a 10 hour dataset of a capella singing from 20 professional singers demonstrating a variety of singing techniques [17]. This dataset is a good candidate for training Voice Activity Detection (VAD) models because it contains onset and offset timestamps for each vocal segment. This is also a good a good dataset to help train AST models to know what singing sounds like.

2.4.9 Other Datasets

Many datasets were considered but left out due to the lack of availablility. Some of the most popular datasets such as Mauch's Dataset [18] and Hansen's Dataset [19] are not publicly available anymore. Newer datasets such as DALI [2] and DAMP! [20] are are hidden behind institutional logins and require manually requesting access.

While both the Mauch's Dataset and Hansen's Datasets are quite small (Mauch has 20 songs, Hansen has 9 [18]), the DALI and DAMP! datasets are much larger. The DALI dataset in particular used a version of the Teacher Student technique mentioned in 2.3.4 to label 105 songs with timestamps for the word and phoneme level [2]. The DAMP! dataset is even larger with 300x30x2 song dataset. Both of these datasets would be excellent candidates for training AST models from their size alone.

3 Related Works

3.1 HMM Based Acoustic Models

The traditional approach to AST is can be separated into a pipeline of distinct steps. The first step is to extract the features from the audio. This usually includes some form of spectral analysis such as MFCCs. The second step is to use an acoustic model to classify the features into phonemes and generate a sequence of phonemes using a Hidden Markov Model (HMM). This is the traditional approach to ASR and is also used in AST [19].

Other models focus on the transcription of notes instead of lyrics. These models use a similar to the ones that are transcribing lyrics TODO: Add more detail about HMMs TODO: Write about Dynamic Time Warping TODO: Write about some results of HMMs

3.2 wav2vec 2.0 and Transfer Learning

3.2.1 wav2vec 2.0

One of the most recent breakthroughs in ASR is wav2vec 2.0 [10]. This model uses a self-supervised learning approach was used during the training of the model. Unlabeled audio was fed into the model to learn discrete speech units. These discrete speech units required the Gumbel-Softmax [21] to allow for backpropagation. The model was then fine-tuned with a linear layer and CTC loss on labeled data to perform ASR [10].

This approach was able to achieve state-of-the-art results on the LibriSpeech dataset [3] and the TIMIT dataset [13]. However, the thing that makes this approach the most promising is the fact that after pre-training on a large amount of unlabeled data, the model can be fine-tuned on a 10 minute subset of labeled data to achieve 5.2 WER on the LibriSpeech clean dataset [10]. This is very important for AST because there are so few labeled datasets available.

3.2.2 wav2vec 2.0 Transfer Learning

Using transfer learning for AST using wav2vec 2.0 was attempted in 2022 and was used to achieve state-of-the-art results on the Jamendo dataset as well as on the DALI, Hansen, Mauch, and DAMP! datasets [22]. The approach they performed did not change the first part of the model that generated the discrete speech units. Instead, they changed the last part of the model to have another branch that outputs the probability of the current word given previous words and context. This approach achieved a 33.13 WER on the Jamendo dataset [22].

Apart from their results, there were also a few other interesting things about their approach. Since the initial wav2vec 2.0 model was trained on the LibriSpeech dataset, they wanted to make the input audio of their model as similar to the LibriSpeech dataset as possible. To do this, they used Demucs v3 [23] to separate the vocals from the accompaniment. They did not futher remove any noise or singing specific features from the audio. For their labeled datasets, they used utterance level labels excluding the instrumental parts. For the output, they used character level labels.

3.3 Whisper and WhisperX Word-Level Alignment

3.3.1 Whisper

Whisper a ASR model trained on a extremely large dataset of 680,000 hours of audio [24]. Instead of using self-supervised learning like in wav2vec 2.0, Whisper uses a weakly supervised approach. This approach uses a large dataset of audio and text that is not high quality as many of the smaller datasets. Instead the dataset was generated by scraping the internet for audio and their corresponding transcriptions. Although Whisper did not improve the state-of-the-art on the LibriSpeech dataset when compared against wav2vec 2.0, did achieve improvements and therefore state-of-the-art results on the a variety of other speech datasets [24]. The model performs well in zero-shot scenarios so it can achieve good results without fine-tuning on the target dataset [24]. This is important for AST because there are so many genres and types of music and it would be difficult to fine-tune on all of them.

3.3.2 WhisperX

The original Whisper model was also able to provide some timestamps but they were not the focus of the paper. It was more of a proof of concept that the same model can be used for multiple tasks [24]. WhisperX is a model that was trained specifically for word-level alignment and the speedup of the Whisper model [25].

On the TED-LIUM dataset, a dataset of TED talks [26], WhisperX was able to acheive a 11.8x speedup over Whisper. This is comparable to wav2vec 2.0 which was a 10.3x speedup over Whisper. However, WhisperX was also able to achieve a WER of 9.7 compared to the 10.5 WER from Whisper and 19.8 WER from wav2vec 2.0 [25]. This is a significant improvement over of speed with a slight increase in performance. This is important for AST because it would allow for faster training and inference times.

For the word alignment task, WhisperX uses an external phoneme alignment approach to generate word level timestamps using phoneme level timestamps [25]. The external phoneme alignment model was actually a wav2vec 2.0 model that used Dynamic Time Warping (DTW) [27] to align the phonemes [25]. The first and last phonemes of each word were used to generate the word level timestamps [25].

The most important improvement of WhisperX over Whisper is VAD Cut & Merge technique. This technique uses a Voice Activity Detection (VAD) model to cut the audio into segments at low voice activity sections. This decreases the hallucination of Whisper if a word was partly spoken in a audio segment which lead to the slight increase in WER [25]. The speedup of WhisperX was also due to this technique it allowed parallel processing of the audio segments where Whisper had to process sequenctially [25].

4 Method

In order to produce a model that can perform AST, we train a model to first take the audio and align it with the lyrics like in the last step of the WhisperX pipeline. This helps us create a model that can poperly annotate phonemes from the audio. Then, a secondary model can be trained to take the audio and the phoneme annotations to generate the lyrics. With both a lyric and phoneme generator, any song will be able to be transcribed into lyrics and time-aligned phonemes for an end-to-end AST.

Each step in this process allows for the use of teacher-student techniques mentioned in section 2.3.4 to generate more labeled data from unlabeled data. This is important because there are so few labeled datasets available for AST.

4.1 Preprocessing Datasets

To preprocess datasets, the audio files were preprocessed in the same way WhisperX preprocessed data with the VAD Cut & Merge technique [25] except with a slight modification before trying to detect voices. Since the audio files were songs, most of them will have both singing and background music. This means that the VAD model will have a harder time detecting when the singing stops. To help the VAD model, the audio files were first passed through Demucs v4 [23] to separate the vocals from the background music. Then, the VAD model was used to detect the voices. Finally, the audio files were cut into segments at the detected voice sections. To reduce memory requirements and because lyrics utterances are often shorter than 30 seconds, the segments will be cut into 8-15 second segments. Segments that are shorter than 8 seconds can either be padded or merged with other segments into segments of 8-15 seconds.

The first subsystem of our AST model will be to take the audio and generate the phoneme annotations. This subsystem will be trained and tested on the TIMIT, Children's Song and NUS datasets. These are all datasets that contain phoneme level annotations. TIMIT, although it is a speech dataset, can be transformed into a pseudo-singing dataset as mentioned in 2.3.2 or used directly as a different variety of vocalizing phonemes.

4.2 Augmenting Datasets

There are many ways to augment datasets. The three ways mentioned in SpecAugment includes time warping, frequency masking and time masking [5]. These techniques can be applied to the audio spectrogram before it is fed into the model. Time warping is a technique that stretches or compresses the audio spectrogram in the time dimension so it must be handled carefully to update the labels accodingly. Another way to augment the datasets is to transform the audio pitch. This can be done by shifting the pitch up or down.

5 Training

6 Results

6.1 VAD Performance

6.2 Phoneme Alignment Performance

6.3 Lyric Generation Performance

7 Discussion

8 Future Work

8.1 Adversarial Training

8.2 Transform Singing to Speech Data

remove pitch from singing

8.3 Phoneme to Grapheme

[28] or [29]

9 Conclusion

[30]

References

[1] C. Gupta, R. Tong, H. Li, and Y. Wang, *Semi-supervised lyrics and solo-singing alignment*, Jul. 2018.

[2] G. Meseguer-Brocal, A. Cohen-Hadria, and G. Peeters, "Dali: A large dataset of synchronized audio, lyrics and notes, automatically created using teacher-student machine learning paradigm.," in *19th International Society for Music Information Retrieval Conference*, ISMIR, Ed., Sep. 2018.

[3] 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)*, 2015, pp. 5206–5210. DOI: 10.1109/ICASSP.2015.7178964.

[4] M. Raissi. "Mel-spectrogram and mfccs." (2021), [Online]. Available: https://www.youtube.com/watch?v=hF72sY70_IQ.

[5] D. S. Park, W. Chan, Y. Zhang, *et al.*, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Interspeech 2019*, ISCA, 2019. DOI: 10.21437/interspeech.2019-2680. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2680.

[6] A. Kruspe, "Training phoneme models for singing with "songified" speech data," Jan. 2015.

[7] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, *The MUSDB18 corpus for music separation*, 2017. DOI: 10.5281/zenodo.1117372. [Online]. Available: https://doi.org/10.5281/zenodo.1117372.

[8] CMU, *Cmu pronouncing dictionary*, 2023. [Online]. Available: http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

[9] A. Hannun, "Sequence modeling with ctc," *Distill*, 2017, https://distill.pub/2017/ctc. DOI: 10.23915/distill.00008.

[10] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, *Wav2vec 2.0: A framework for self-supervised learning of speech representations*, 2020. arXiv: 2006.11477 [cs.CL].

[11] S. Abbasi, M. Hajabdollahi, N. Karimi, and S. Samavi, *Modeling teacher-student techniques in deep neural networks for knowledge distillation*, 2019. arXiv: 1912.13179 [cs.CV].

[12] C.-W. Wu and A. Lerch, "Automatic drum transcription using the student-teacher learning paradigm with unlabeled music data," Oct. 2017.

[13] J. S. Garofolo, L. F. Lamel, W. M. Fisher, *et al.*, *Semi-supervised lyrics and solo-singing alignment*, 1993.

[14] D. Stoller, S. Durand, and S. Ewert, *End-to-end lyrics alignment for polyphonic music using an audio-to-character recognition model*, 2019. arXiv: 1902.06797 [cs.SD].

[15] Z. Duan, H. Fang, B. Li, K. C. Sim, and Y. Wang, "The nus sung and spoken lyrics corpus: A quantitative comparison of singing and speech," in *2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 2013, pp. 1–9. DOI: 10.1109/APSIPA.2013.6694316.

[16] 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. arXiv: 1612.01840. [Online]. Available: https://arxiv.org/abs/1612.01840.

[17] J. Wilkins, P. Seetharaman, A. Wahl, and B. Pardo, "Vocalset: A singing voice dataset," in *International Society for Music Information Retrieval Conference*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:53875542.

[18] MIREX, *2021:automatic lyrics transcription*, 2021. [Online]. Available: https://music-ir.org/mirex/wiki/2021:Automatic_Lyrics_Transcription.

[19] J. K. Hansen, "Recognition of phonemes in a-cappella recordings using temporal patterns and mel frequency cepstral coefficients," 2012. [Online]. Available: https://api.semanticscholar.org/CorpusID:37419482.

[20] I. Smule, *Damp-mvp: Digital archive of mobile performances - smule multilingual vocal performance 300x30x2*, 2018. [Online]. Available: https://zenodo.org/records/2747436.

[21] E. Jang, S. Gu, and B. Poole, *Categorical reparameterization with gumbel-softmax*, 2017. arXiv: 1611.01144 [stat.ML].

[22] L. Ou, X. Gu, and Y. Wang, *Transfer learning of wav2vec 2.0 for automatic lyric transcription*, 2022. arXiv: 2207.09747 [eess.AS].

[23] S. Rouard, F. Massa, and A. Défossez, "Hybrid transformers for music source separation," in *ICASSP 23*, 2023.

[24] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, *Robust speech recognition via large-scale weak supervision*, 2022. arXiv: 2212.04356 [eess.AS].

[25] M. Bain, J. Huh, T. Han, and A. Zisserman, "Whisperx: Time-accurate speech transcription of long-form audio," *INTERSPEECH 2023*, 2023.

[26] F. Hernandez, V. Nguyen, S. Ghannay, N. Tomashenko, and Y. Estève, "Ted-lium 3: Twice as much data and corpus repartition for experiments on

speaker adaptation," in *Speech and Computer*, Springer International Publishing, 2018, pp. 198–208.

[27] K. Bringmann, N. Fischer, I. van der Hoog, E. Kipouridis, T. Kociumaka, and E. Rotenberg, *Dynamic dynamic time warping*, 2023. arXiv: `2310.18128 [cs.CG]`.

[28] E. Engelhart, M. Elyasi, and G. Bharaj, *Grapheme-to-phoneme transformer model for transfer learning dialects*, 2021. arXiv: `2104.04091 [cs.CL]`.

[29] R. Masumura, N. Makishima, M. Ihori, A. Takashima, T. Tanaka, and S. Orihashi, "Phoneme-to-Grapheme Conversion Based Large-Scale Pre-Training for End-to-End Automatic Speech Recognition," in *Proc. Interspeech 2020*, 2020, pp. 2822–2826. DOI: `10.21437/Interspeech.2020-1930`. [Online]. Available: `https://www.isca-speech.org/archive/interspeech_2020/masumura20_interspeech.html`.

[30] H. Bredin, "pyannote.audio 2.1 speaker diarization pipeline: principle, benchmark, and recipe," in *Proc. INTERSPEECH 2023*, 2023.