SEMI-SUPERVISED LYRICS AND SOLO-SINGING ALIGNMENT

Chitralekha Gupta[1,2] **Rong Tong**[4] **Haizhou Li**[3] **Ye Wang**[1,2]

[1] NUS Graduate School for Integrative Sciences and Engineering, [2] School of Computing,
[3] Electrical and Computer Engineering Dept., National University of Singapore, Singapore
[4] Alibaba Inc. Singapore R&D Center, Singapore

chitralekha@u.nus.edu, rong.tong@alibaba-inc.com, haizhou.li@nus.edu.sg,
wangye@comp.nus.edu.sg

ABSTRACT

We propose a semi-supervised algorithm to align lyrics to the corresponding singing vocals. The proposed method transcribes and aligns lyrics to solo-singing vocals using the imperfect transcripts from an automatic speech recognition (ASR) system and the published lyrics. The ASR provides time alignment between vocals and hypothesized lyrical content, while the non-aligned published lyrics correct the hypothesized lyrical content. The effectiveness of the proposed method is validated through three experiments. First, a human listening test shows that 73.32% of our automatically aligned sentence-level transcriptions are correct. Second, the automatically aligned sung segments are used for singing acoustic model adaptation, which reduces the word error rate (WER) of automatic transcription of sung lyrics from 72.08% to 37.15% in an open test. Third, another iteration of decoding and model adaptation increases the amount of reliably decoded segments from 44.40% to 91.96% and further reduces the WER to 36.32%. The proposed framework offers an automatic way to generate reliable alignments between lyrics and solo-singing. A large-scale solo-singing and lyrics aligned corpus can be derived with the proposed method, which will be beneficial for music and singing voice related research.

1. INTRODUCTION

Lyrics serve as an important component of music, that often defines the mood of the song [2, 4], affects the opinion of a listener about the song [3], and even improves the vocabulary and pronunciation of a foreign language learner [14, 30]. Research in Music Information Retrieval (MIR) in the past has explored tasks involving lyrics such as automatic lyrics recognition [15, 19, 26, 28] and automatic lyrics alignment [5, 11, 27] for various applications such as karaoke singing, song subtitling, query-by-singing as well as acoustic modeling for singing voice. In spite of huge advances in speech technology, automatic lyrics transcription and alignment in singing face challenges due to the differences between sung and spoken voices [11, 26], and a lack of transcribed singing data to train phonetic models for singing [11, 15, 26–28].

As singing and speech differ in many ways such as pitch dynamics, duration of phonemes, and vibrato [11, 26], the direct use of ASR systems for lyrics alignment or transcription of singing voice will result in erroneous output. Therefore, speech acoustic models need to be adapted to singing voice [27]. For training singing-adapted acoustic models, lyrics-aligned singing dataset is necessary. Lack of annotated singing datasets has been a bottleneck for research in this field. Duan et al. [8] published a small singing dataset (1.92 hours) with phone-level annotations, which were done manually that requires a lot of time and effort, and is not scalable. One way of getting data for training is to force-align the lyrics with singing using speech models, and use this aligned singing data for model training and adaptation. But due to the differences in speech and singing acoustic characteristics, alignment of lyrics with speech acoustic models will be prone to errors, that will result in badly adapted singing acoustic models.

With the increase in popularity of mobile phone karaoke applications, singing data collected from such apps are being made available for research. Smule's *Sing!* karaoke dataset, called Digital Archive of Mobile Performances (DAMP) [33], is one such dataset that contains more than 34K a capella (solo) singing recordings of 301 songs. But it does not have time-aligned lyrics, although the textual lyrics are available on Smule's website. The data also contains inconsistencies in recording conditions, out-of-vocabulary words, and incorrectly pronounced words because of unfamiliar lyrics or non-native language speakers. Although the presence of such datasets is a huge boon to MIR research, we need tools to further clean up such data to make them more usable. There is a need for aligned lyrics transcriptions for singing vocals while also eliminating inconsistent or noisy recordings. To address this need, we propose a simple yet effective solution to produce clean audio segments with aligned transcriptions.

In this work, we study a strategy to obtain time-aligned sung-lyrics dataset with the help of the state-of-the-art ASR as well as an external resource, i.e. published lyrics.

We use the speech acoustic models to transcribe solo-singing audio segments, and then align this imperfect transcription with the published lyrics of the song to obtain a better transcription of the sung segments. We hypothesize that this strategy will help in correcting the imperfect transcriptions from the ASR module and in cleaning up bad audio recordings. We validate our hypothesis by a human listening experiment. Moreover we show that a semi-supervised adaptation of speech acoustic models with this cleaned-up annotated dataset results in further improvement in alignment as well as transcription, iteratively. Hence, such an algorithm will potentially automate the labor-intensive process of time aligning lyrics such as in karaoke or MTV. Furthermore, it will enable large-scale singing transcription generation, thus increasing the scope of research in music information retrieval. We have applied our algorithm on a subset of the DAMP dataset, and have published the resulting dataset and code [1] .

2. RELATED WORK

One of the traditional methods of aligning lyrics to music is with the help of the timing information from the musical structure such as chords [17, 24, 25, 35], and chorus [21], but such methods are more suitable for singing in the presence of background accompaniments. Another study uses musical score to align lyrics [13], but such methods would be applicable for professional singing where the notes are correctly sung. In karaoke applications, as addressed in this work, correctness of notes is less likely.

One of the pioneering studies of applying speech recognition for lyric alignment was by Mesaros and Virtanen [27], who used 49 fragments of songs, 20-30 seconds long, along with their manually acquired transcriptions to adapt Gaussian Mixture Model-Hidden Markov Model (GMM-HMM) speech models for singing in the same way as speaker adaptation is done. They then used these singing-adapted speech models to align vocal sections of songs with their manually paired lyrics lines using the Viterbi algorithm. In [28], the authors used the same alignment method to automatically obtain the singing-to-lyrics aligned lines, and then explored multiple model adaptation techniques, to report the best phoneme error rate (PER) of 80%. This work has provided a direction for solving the problem of lyrics alignment and recognition in singing, but it suffers from manual post-processing and the models are based on a small number of annotated singing samples.

Recently, with the availability of more singing data, a subset of the DAMP solo-singing dataset was used for the task of sung phoneme recognition by Kruspe [19, 20]. In this work, the author builds new phonetic models trained only on singing data (DAMP data subset) and compares it with a pitch-shifted, time-stretched, and vibrato-applied version of a speech dataset called *songified* speech data TimitM [18]. Their best reported PER was 80%, and weighted PER (that gives 0.5 weights to deletions and insertions) was 56%, using the DAMP data subset, which outperformed the songified dataset. This work shows an effective use of the available (unannotated) singing data to build improved singing phonetic models. But there is still room for improvement.

[1] Dataset: https://drive.google.com/open?id=1hGuE0Drv3tbN-YNRDzJJMHfzKH6e4O2A;
Code: https://github.com/chitralekha18/AutomaticSungLyricsAnnotation_ISMIR2018.git

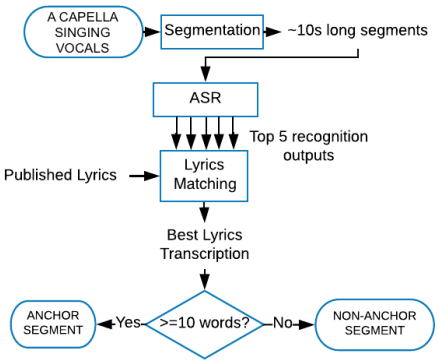

Figure 1: The diagram of lyrics to singing vocal alignment algorithm.

The first step in Kruspe's work was to obtain aligned lyrics annotations of every song, for which the whole lyrics of a song was force-aligned with the audio using speech-trained models. These force-aligned sung phonemes were then used to build the new acoustic phonetic models for singing. This approach of forced-alignment of singing using speech acoustic models has also been applied in the earlier attempts of automatic lyrics alignment in a capella singing as well as in singing with background music [11, 16, 17, 35]. But, as noted by Kruspe [19], forced-alignment of singing with speech models causes unavoidable errors, because of the mismatch between speech and singing acoustic characteristics [10,23], as well as the mismatch between the actual lyrics and what the singer sings. Thus, the lack of appropriate lyrics-aligned song dataset and the eventual use of forced-alignment with speech models to obtain this annotation is a source of errors.

3. SEMI-SUPERVISED LYRICS AND SINGING VOCALS ALIGNMENT ALGORITHM

We propose an algorithm to align lyrics to singing vocals, that consists of two main steps: dividing the singing vocals into shorter segments (Segmentation), and obtaining the aligned lyrics for each segment (Lyrics Matching). Figure 1 shows the overview of our algorithm.

3.1 Segmentation

One way to automatically align the published lyrics with a solo-singing audio is to force-align the lyrics with the full rendition audio (2 to 4 minutes long) using speech trained acoustic models, as discussed in [19]. However, the Viterbi alignment algorithm used in forced-alignment, fails to scale well for long audio segments leading to accumulated alignment errors [29]. In our singing-lyrics transcription and alignment algorithm, we propose to first divide the audio into shorter segments such that the ASR is less prone to the alignment errors. We find silent regions in the rendition by imposing constraints on the magnitude of the short time energy and the silence duration (Algorithm 1). The center of these silent regions are marked

as boundaries of non-silent sub-segments. Such non-silent sub-segments are of varying lengths. So we stitch consecutive sub-segments together to make segments of ∼10 seconds duration. We also add silence samples before and after every such segment so that the ASR has some time to adapt to the utterance and start recognition in the beginning of the utterance, and to avoid abrupt termination at the end of the utterance.

Algorithm 1 Segmentation algorithm

Calculate short time energy E for 32 ms window with 16 ms hop
if $E > 0.1 \times mean(E)$ is true **then**
    non-silent region
else
    silent region
end if
if silent region duration>=200 ms **then**
    valid silence region
    center of this region marks the boundary
else
    invalid silent region
end if
sub-segment = boundary-to-boundary region
segment = stitch together such sub-segments for ∼10s duration
add 2s silence before and after every segment, to improve ASR performance

3.2 Lyrics Matching

We would like to obtain the best possible lyrics transcription for these short singing segments. Moreover, to obtain a clean transcribed dataset of singing vocals, we would also like to reject the noisy audio segments that contain out-of-vocabulary, incorrectly pronounced words, and background noise. We use ASR to decode these segments because such ASR transcription ideally suggests words that are actually sung and different from the published lyrics. The ASR transcription also help detect erroneous pronunciations, reject noise segments. We understand that the the state-of-the-art ASR is not perfect, and for singing it is even more unreliable, as the ASR is trained on speech while singing is acoustically different from speech. So we designed an algorithm to overcome these imperfections of the ASR. This algorithm produces time-aligned transcriptions of clean audio segments with the help of the published lyrics.

Algorithm 2 Lyrics Matching algorithm

$X_{N\times 5}$ s.t. $x_{i,j} = e$
where, X = error matrix,
N = number of words in published lyrics,
e = ratio of number of errors obtained from Levenshtein distance between ASR output and published lyrics window, to the total number of words in the lyrics window
$i_{min}, j_{min} = \text{argmin}\, X$
where i_{min} = minimum distance transcription start index in lyrics,
where j_{min} = minimum distance transcription slack window size
transcription = lyrics[$i_{min} : i_{min} + M + j_{min}$]
where, M is the number of words in ASR transcription

3.2.1 ASR Transcription of Lyrics

To obtain the transcription of each of the audio segments, we use the Google speech-to-text API package in python [36] that transcribes a given audio segment into a string of words, and gives a set of best possible transcriptions. We compare the top five of these transcriptions with the published lyrics of the song, and select the one that matches the most, as described in Algorithm 2. The idea is that the ASR provides a hypothesis of the aligned lyrics although imperfect, and the published lyrics helps in checking these hypothesized lyrics, and retrieving the correct lyrics. Also, we use the Google ASR to bootstrap, with a plan to improve our own ASR (as discussed further in Section 4.2). Different ASR systems have different error patterns, therefore we expect that the Google ASR would boost the performance of our ASR. We use the Google ASR only for bootstrapping, the rest of the experiments use our own ASR. Below is the description of the lyrics-matching algorithm.

For an ASR output of length M words, we took a lyrics window of size M, and also empirically decided to provide a slack of 0 to 4 words, i.e. the lyrics window size could be of length M to M+4. This slack provides room for accommodating insertions and deletions in the ASR output, thus allowing improvement in the alignment. So, starting from the first word of the published lyrics, we calculate the Levenshtein distance [22] between the ASR output and the lyrics window of different slack sizes, iterated through the entire lyrics by one word shifts. This distance represents the number of errors (substitutions, deletions, insertions) occurred in ASR output with respect to the actual lyrics.

For the lyrics of a song containing a total of N words, we obtain an error matrix X of dimensions Nx5, where 5 is the number of slack lyric window sizes ranging from M to M+4. Each element e of the matrix is the ratio of the number of errors obtained from Levenshtein distance between the ASR output and the lyrics window, to the total number of words in that lyrics window. If (i_{min}, j_{min}) is the coordinate of the minimum error element of this matrix, then i_{min} is the starting index of the minimum distance lyrics transcription, j_{min} is the slack lyric window size. Amongst the top five ASR outputs, we choose the one that gives minimum error e, and select the corresponding lyrics window from the error matrix to obtain the best lyrics transcription for that audio segment. We illustrate this with the help of the following example.

Let's assume that the ASR transcription of an audio segment is *"the snow glows on the mountain"*, therefore M=6. The slack window size will range from 6 to 10 words. The lyrics of this song contains a total of N words, where a word sub-sequence is *"the snow glows white on the mountain tonight not a footprint to be seen..."*. The corresponding error matrix X is shown in Figure 2. The error element $e_{1,2}$ is the distance between the ASR transcription and the slack lyric window "the snow glows white on the mountain" which is 1. The error element $e_{2,1}$ is the distance between the ASR transcription and the slack lyric window "snow glows white on the mountain" which is 2, and so on.

$e_{i_{min},j_{min}}$	1 (M=6)	2 (M=7)	3 (M=8)	4 (M=9)	5 (M=10)
Word 1	2	1	2	3	4
Word 2	2	3	4	5	6
.....	....	....	....	....	....
Word N	....	....	....	....	....

Figure 2: Example of an error matrix X where the ASR transcript is *"the snow glows on the mountain"*, and the published lyrics of this song has N words where a word sub-sequence is *"the snow glows white on the mountain tonight not a footprint to be seen..."*.

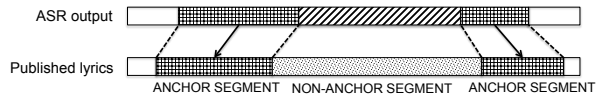

Figure 3: Anchor and non-anchor segments of a song based on sung-lyrics alignment algorithm. Anchor segments: ASR output and lyrics reliably match; Non-Anchor segments: ASR output and lyrics do not match.

So in this example, (i_{min}, j_{min}) is (1,2), i.e. the best lyrics transcription is "the snow glows white on the mountain".

3.2.2 Anchor and Non-Anchor Segments

From our preliminary study, we found that many of the ASR transcriptions had missing words because either the audio contained background noise or there were incorrectly pronounced words or deviation of singing acoustics from speech. For example, a 10 seconds long non-silent segment from a popular English song would rarely ever have as few as four or five words. In order to retrieve more reliable transcripts, we added a constraint on the number of words, as described below.

To check the reliability of the lyrics transcriptions, we marked the best lyrics transcriptions of a small subset of 360 singing segments as correct or incorrect, depending on whether the transcription matched with the audio. We found that all those segments for which the best lyrics transcription had less than 10 words were more likely to be incorrect matches, as shown in Figure 4. The segment transcriptions were 94.0% times incorrect (235 incorrect out of 250 total number of segments) when they contained less than 10 words, while they were 57.3% times incorrect (63 out of 110) when they contained more than or equal to 10 words. So we empirically set 10 words as the threshold for selecting reliable audio segments and transcriptions. By applying this constraint, we reject those audio segments that are noisy, or have wrongly pronounced words, or cause errors in transcription because of model mismatch, thus deriving a clean transcribed singing dataset.

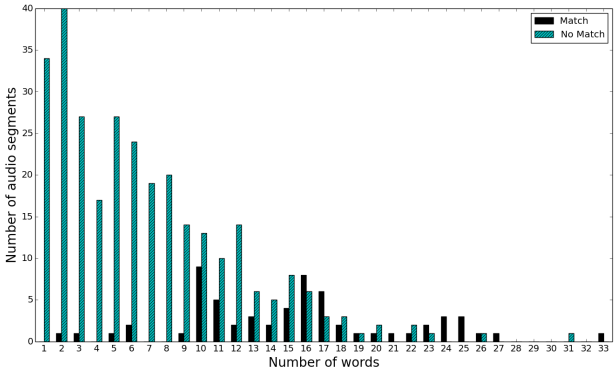

Figure 4: The number of audio segments with correct transcription (blue) or incorrect transcription (cyan) according to human judgment on y-axis versus the number of words in the transcription of an audio segment on x-axis. We set 10 words as the minimum threshold for a transcription to be valid for an approximately 10-seconds long segment.

The audio segments with reliable transcription are labeled as *Anchor segments*, and the audio segment(s) between two anchor segments that have unreliable transcription, are strung together and labeled as *Non-Anchor segments*, as illustrated in Figure 3.

One may argue that we could have used the error score e to evaluate the reliability of a segment. However, if the ASR output itself is wrong, then this lyrics-matching error score will be misleading. For example, if only 4 words get detected by the ASR, out of 12 words in the audio segment, and all the 4 words are correct according to the published lyrics, then e will be zero for this transcription, which is incorrect, and also undetectable. Thus we set a threshold on the number of detected words (i.e. 10 words) as a way to measure the reliability of the segment and its transcription.

4. EXPERIMENTS AND RESULTS

In order to validate our hypothesis that our algorithm can retrieve good quality aligned transcriptions, we conducted three experiments: A) Human verification of the quality of the aligned transcriptions through a listening test, B) Semi-supervised adaptation of speech models to singing using our aligned sung-lyrics transcriptions for assessing the performance of automatic lyrics recognition, and C) Second iteration of alignment, and re-training of acoustic models, to check for further improvement in lyrics recognition.

Our experiments are conducted on 6,000 audio recordings from the DAMP dataset that was used by Kruspe [19]. The list of recordings used by Kruspe is here [1], however the training and test subsets are not clearly marked. Therefore we have defined our training and test datasets, and they are subsets of Kruspe's dataset, as discussed in the following subsections. This data set contains recordings of amateur singing of English language pop songs with no background music but different recording conditions, which were obtained from the Smule Sing! karaoke app. Each performance is labeled with metadata such as the gender of the singer, the region of origin, the song title, etc. We obtained the textual lyrics of the songs from Smule Sing! website [34]. Since the songs in DAMP dataset were sung on Smule Sing! Karaoke app that uses these lyrics, it is safe to assume that these were the intended lyrics.

4.1 Experiment 1: Human Verification of the Quality of the Aligned-Transcriptions

In this experiment, we evaluate the quality of our aligned transcriptions (*segment transcriptions*), by asking participants to listen to the audio segments and verify if the given

transcription for the segment is correct or not. As opposed to word intelligibility evaluation tasks [6] where participants are asked to transcribe after listening to the stimuli once, in this task the participants were provided with the transcription and were free to listen to the audio as many times as they needed. Also the songs were popular English songs, that are less prone to perception errors [7].

4.1.1 Dataset

By applying our lyrics transcription and alignment algorithm (see Section 3), we obtained 19,873 of anchor segments ($\sim$58 hours) each $\sim$10 seconds long, out of which we asked humans to validate 5,400 (15 hours) anchor segment transcriptions through a listening test. The only criterion to qualify for the listening test was to be proficient in English language. 15 university graduate students were the human listeners. Every listener was given one hour of audio segments containing 360 anchor segments along with the obtained lyrics transcription. The task was to listen to each of the audio segments and compare the given transcription with the audio. If at least 90% of the words in the transcription match with that in the audio, then the audio segment was marked as correctly transcribed. If not, then it was marked as incorrectly transcribed.

Similarly, we also tested the quality of the non-anchor segments. Non-anchor segments could be of varying durations, greater than or equal to 10 seconds. We conducted the same human validation task for 2 hours (1,262 segments) of the non-anchor segments of different durations.

4.1.2 Results and Discussion

There were two types of successful segment transcriptions, one was verified by humans as correct and also matched perfectly with the ASR output, and was labeled as *correct transcriptions fully matching ASR*. Another was verified as correct by humans but did not match with the ASR output due to ASR errors, but our algorithm could successfully retrieve the correct transcriptions, that we call *correct transcriptions partially matching ASR*. And the ones that were verified as wrong by humans are labeled as *error transcriptions due to imperfect ASR or incorrect singing*.

Anchor Segments: Table 1 shows the validation results for the anchor segments. We found that a total of 73.32% of the segments were transcribed correctly, where 57.80% of the segments were *partially matching ASR*. This means that our algorithm could successfully retrieve many incorrect ASR transcriptions, which validates our hypothesis that the extra information provided by the published lyrics coupled with ASR decoding produces good aligned transcriptions. We also found that incorrect singing of lyrics and imperfect ASR output resulted in 26.68% erroneous transcriptions. A common error reported by the listeners was many missing words at the trailing end of the incorrectly aligned transcriptions, although the correct words were clearly audible, which is possibly a result of model mismatch between singing and speech.

Non-Anchor Segments: From the human validation of the non-anchor segments, we find that 62.07% of the total of 1,262 non-anchor segments transcriptions are correct. This suggests that these segments are relatively less reliable. Moreover, these audio segments could be long in duration (even more than a minute) that would cause errors in the Viterbi alignments. Thus in the subsequent experiments, we only use the anchor segments.

Segment Transcriptions	#	%	Total %
Correct transcriptions fully matching ASR	838	15.52	**73.32**
Correct transcriptions partially matching ASR	3,121	57.80	
Error transcriptions due to imperfect ASR or incorrect singing	1,441	26.68	**26.68**

Table 1: A summary of correct and error transcriptions by the proposed algorithm. Google ASR is used for singing transcription. Total # anchor segments = 5,400 (15 hours).

4.2 Experiment 2: Lyrics Transcription with Singing-Adapted Acoustic Models

In this experiment, we use our automatically generated aligned-transcriptions of sung audio segments in a semi-supervised adaptation of the speech models for singing. We use these singing-adapted models in an open test to validate our hypothesis that better aligned transcriptions for training singing acoustic models will result in improvement in automatic lyrics recognition compared to the best known baseline from the literature.

Adaptation of speech models for singing was previously attempted by Mesaros et al. [27, 28] who applied the speaker adaptation techniques to transform speech recognizer to singing voice. To reduce the mismatch between singing and speech, they used constrained maximum likelihood linear regression (CMLLR) to compute a set of transformations to shift the GMM means and variances of the speech models so that the resulting models are more likely to generate the adaptation singing data. In our work, we use CMLLR (also known as feature-space maximum likelihood linear regression (fMLLR)) [32] and our lyrics-aligned anchor segments to compute transformations for a semi-supervised adaptation of the speech models to singing. Adaptation can be done with the test dataset only, or the adaptation transformations can be applied at the time of training, called speaker adaptive training (SAT). Literature shows that the use of SAT with fMLLR transform requires minimum alteration to the standard code for training [12], and thus is a popular tool for speaker adaptation that we have used for singing adaptation here.

4.2.1 Dataset

The singing train set consists of 18,176 singing anchor segments from 2,395 singers while the singing test set consists of 1,697 singing anchor segments of 331 singers. The training set consists of both human verified and non-verified anchor segments, while the test set consists of only those anchor segment transcriptions that are verified as correct by humans. All of these anchor segments (training and test) are of $\sim$10 seconds duration. There is no speaker overlap between the acoustic model training and test sets. A language model is obtained by interpolating a speech language model trained from Librispeech [31] text and a

Models Adapted by Singing Data	%WER	%PER
(1) Baseline (speech acoustic models)	72.08	57.52
(2) Adapted with test data	47.42	39.34
(3) Adapted (SAT) with training data	40.25	33.18
(4) Adapted (SAT+DNN) with training data	37.15	31.20
(5) Repeat (3) and (4) for 2nd round	**36.32**	**28.49**

Table 2: The sung word and phone error rate (WER and PER) in the lyrics recognition experiments with the speech acoustic models (baseline) and the singing-adapted acoustic models, on 1,697 correctly transcribed test singing anchor segments.

lyric language model trained from lyrics of the 301 songs of the DAMP dataset. The same language model is used in all the recognition experiments.

4.2.2 Results and Discussion

Table 2 reports the automatic lyrics recognition results on the singing test set using different acoustic models to observe the effect of adapting speech models for singing using our sung segments with aligned transcriptions.

The baseline speech acoustic model is a tri-phone HMM model trained on Librispeech corpus using MFCC features. Due to the mismatch between speech and singing acoustic characteristics, the WER and PER are high (Table 2 (1)). Adapting the baseline model with the singing test data results in a significant improvement in the error rates (Table 2 (2)). Speaker adaptive training (SAT) further improves the recognition accuracy (Table 2 (3)). A DNN model [9] is trained on top of the SAT model with the same set of training data. During DNN training, temporal splicing is applied on each frame with left and right context window of 4. The SAT+DNN model has 3 hidden layers and 2,976 output targets. With DNN training, the WER is reduced by about 7.7% relative to the SAT model (Table 2 (4)) and PER is 31.20%.

Mesaros et al. [27] reported the best PER to be 80% with speech models adapted to singing, while Kruspe [19] reported the best PER to be 80% and weighted PER to be 56% with pure singing phonetic models trained on a subset of the DAMP dataset. Compared to [19] and [27], our results show a significant improvement, which is attributed to three factors. One is that leveraging on ASR along with the published lyrics as an external resource to validate and clean-up the transcriptions has led to better aligned transcriptions for training. Two, our automatic method for generating aligned transcriptions for singing provides us with a much larger training dataset. And three, the segment-wise alignment is more accurate than the whole-song forced-aligned with the speech acoustic models.

4.3 Experiment 3: Alignment with Singing-Adapted Acoustic Models and Re-training

We would like to test if the singing-adapted acoustic models can provide more number of reliably aligned transcriptions in a second round of alignment. Moreover whether re-training the models with this second round of transcriptions lead to better lyrics recognition.

Model	# anchor	total # segments	% anchor
Google ASR	5,400	12,162	44.40
Adapted (DNN) with training data	11,184	12,162	**91.96**

Table 3: Comparing the number of anchor segments obtained from the proposed transcription and alignment algorithm using Google ASR and the singing-adapted models.

4.3.1 Dataset

We used the singing-adapted models obtained in Experiment 2 to decode 12,162 segments, and then applied our lyrics-alignment algorithm to obtain new anchor and non-anchor segments. For comparison, we obtained the same from the Google ASR on the same dataset.

4.3.2 Results and Discussion

Table 3 shows that the number of anchor segments with the new models have increased from 44.40% with Google ASR to 91.96% with the singing-adapted models, which means that the number of reliable segment transcriptions have increased significantly. With the new anchor segments, we re-train our singing-adapted acoustic models. Table 2 (5) shows the free-decoding results after this second round of training. The WER and PER have dropped further to 36.32% and 28.49% respectively .

The results of this experiment are promising as they show iterative improvement in the quality of our alignment and transcription. This means that we can apply the following strategy: use only the reliably aligned segments from the Google ASR to adapt acoustic models for singing, and use these models to improve the quality of alignment and transcription, and then again use the reliable segments from the improved alignments for further adaptations.

5. CONCLUSIONS

We propose an algorithm to automatically obtain time-aligned transcriptions for singing by using the imperfect transcriptions from the state-of-the-art ASR along with the non-aligned published lyrics. Through a human listening test, we showed that the extra information provided by the published lyrics helps to correct many incorrect ASR transcriptions. Furthermore, using the time-aligned lyrics transcriptions for iterative semi-supervised adaptation of speech acoustic models for singing shows significant improvement in automatic lyrics transcription performance. Thus our strategy to obtain time-aligned transcriptions for large-scale singing dataset is useful to train improved acoustic models for singing.

Our contribution provides an automatic way to obtain reliable lyrics transcription for singing, that results in an annotated singing dataset. Lack of such datasets has been a bottleneck in the field of singing voice research in MIR. This will not only generate lyrics transcription and alignment for karaoke and subtitling applications, but also provide reliable data to improve acoustic models for singing, thus widening the scope of research in MIR.

6. REFERENCES

[1] MIREX 2017. 2017 Automatic Lyrics-to-Audio Alignment. http://www.music-ir.org/mirex/wiki/2017:Automatic_Lyrics-to-Audio_Alignment. [Online; accessed 15-March-2018].

[2] S Omar Ali and Zehra F Peynircioğlu. Songs and emotions: are lyrics and melodies equal partners? *Psychology of Music*, 34(4):511–534, 2006.

[3] Bruce Anderson, David G Berger, R Serge Denisoff, K Peter Etzkorn, and Peter Hesbacher. Love negative lyrics: Some shifts in stature and alterations in song. *Communications*, 7(1):3–20, 1981.

[4] Elvira Brattico, Vinoo Alluri, Brigitte Bogert, Thomas Jacobsen, Nuutti Vartiainen, Sirke Nieminen, and Mari Tervaniemi. A functional mri study of happy and sad emotions in music with and without lyrics. *Frontiers in psychology*, 2, 2011.

[5] Yu-Ren Chien, Hsin-Min Wang, and Shyh-Kang 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.

[6] Lauren B Collister and David Huron. Comparison of word intelligibility in spoken and sung phrases. 2008.

[7] Nathaniel Condit-Schultz and David Huron. Catching the lyrics: intelligibility in twelve song genres. *Music Perception: An Interdisciplinary Journal*, 32(5):470–483, 2015.

[8] Zhiyan Duan, Haotian Fang, Bo Li, Khe Chai Sim, and Ye Wang. The nus sung and spoken lyrics corpus: A quantitative comparison of singing and speech. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013 Asia-Pacific*, pages 1–9. IEEE, 2013.

[9] G. Hinton et al. Deep neural networks for acoustic modeling in speech recognition. In *IEEE Signal Processing Magazine*, volume 29, pages 82–97, 2012.

[10] Hiromasa Fujihara and Masataka Goto. Lyrics-to-audio alignment and its application. In *Dagstuhl Follow-Ups*, volume 3. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2012.

[11] Hiromasa Fujihara, Masataka Goto, Jun Ogata, and Hiroshi G Okuno. Lyricsynchronizer: Automatic synchronization system between musical audio signals and lyrics. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1252–1261, 2011.

[12] Mark JF Gales. Maximum likelihood linear transformations for hmm-based speech recognition. *Computer speech & language*, 12(2):75–98, 1998.

[13] Rong Gong, Philippe Cuvillier, Nicolas Obin, and Arshia Cont. Real-time audio-to-score alignment of singing voice based on melody and lyric information. In *Interspeech*, 2015.

[14] Arla J Good, Frank A Russo, and Jennifer Sullivan. The efficacy of singing in foreign-language learning. *Psychology of Music*, 43(5):627–640, 2015.

[15] Jens Kofod Hansen and IDMT Fraunhofer. Recognition of phonemes in a-cappella recordings using temporal patterns and mel frequency cepstral coefficients. In *9th Sound and Music Computing Conference (SMC)*, pages 494–499, 2012.

[16] Denny Iskandar, Ye Wang, Min-Yen Kan, and Haizhou Li. Syllabic level automatic synchronization of music signals and text lyrics. In *Proceedings of the 14th ACM international conference on Multimedia*, pages 659–662. ACM, 2006.

[17] Min-Yen Kan, Ye Wang, Denny Iskandar, Tin Lay Nwe, and Arun Shenoy. Lyrically: Automatic synchronization of textual lyrics to acoustic music signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):338–349, 2008.

[18] Anna M Kruspe. Training phoneme models for singing with "songified" speech data. In *ISMIR*, pages 336–342, 2015.

[19] Anna M Kruspe. Bootstrapping a system for phoneme recognition and keyword spotting in unaccompanied singing. In *ISMIR*, pages 358–364, 2016.

[20] Anna M Kruspe. Retrieval of textual song lyrics from sung inputs. In *INTERSPEECH*, pages 2140–2144, 2016.

[21] Kyogu Lee and Markus Cremer. Segmentation-based lyrics-audio alignment using dynamic programming. In *ISMIR*, pages 395–400, 2008.

[22] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966.

[23] Alex Loscos, Pedro Cano, and Jordi Bonada. Low-delay singing voice alignment to text. In *ICMC*, 1999.

[24] Matthias Mauch, Hiromasa Fujihara, and Masataka Goto. Lyrics-to-audio alignment and phrase-level segmentation using incomplete internet-style chord annotations. In *Proceedings of the 7th Sound and Music Computing Conference (SMC 2010)*, pages 9–16, 2010.

[25] Matthias Mauch, Hiromasa Fujihara, and Masataka Goto. Integrating additional chord information into hmm-based lyrics-to-audio alignment. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):200–210, 2012.

[26] Matt McVicar, Daniel PW Ellis, and Masataka Goto. Leveraging repetition for improved automatic lyric transcription in popular music. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 3117–3121. IEEE, 2014.

[27] Annamaria Mesaros and Tuomas Virtanen. Automatic alignment of music audio and lyrics. In *Proceedings of the 11th Int. Conference on Digital Audio Effects (DAFx-08)*, 2008.

[28] Annamaria Mesaros and Tuomas Virtanen. Automatic recognition of lyrics in singing. *EURASIP Journal on Audio, Speech, and Music Processing*, 2010(1):546047, 2010.

[29] Pedro J Moreno, Christopher F Joerg, Jean-Manuel Van Thong, and Oren Glickman. A recursive algorithm for the forced alignment of very long audio segments. In *ICSLP*, volume 98, pages 2711–2714, 1998.

[30] Hitomi Nakata and Linda Shockey. The effect of singing on improving syllabic pronunciation–vowel epenthesis in japanese. In *International Conference of Phonetic Sciences*, 2011.

[31] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *ICASSP*, 2015.

[32] Daniel Povey and George Saon. Feature and model space speaker adaptation with full covariance gaussians. In *Ninth International Conference on Spoken Language Processing*, 2006.

[33] Smule. Digital Archive Mobile Performances (DAMP). `https://ccrma.stanford.edu/damp/`. [Online; accessed 15-March-2018].

[34] Smule. Digital Archive Mobile Performances (DAMP). `https://www.smule.com/songs`. [Online; accessed 15-March-2018].

[35] Ye Wang, Min-Yen Kan, Tin Lay Nwe, Arun Shenoy, and Jun Yin. Lyrically: automatic synchronization of acoustic musical signals and textual lyrics. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 212–219. ACM, 2004.

[36] A Zhang. Speech Recognition (Version 3.7) [Software]. `https://github.com/Uberi/speech_recognition#readme`, 2017. [Online; accessed 14-Oct-2017].