MPop600: A Mandarin Popular Song Database with Aligned Audio, Lyrics, and Musical Scores for Singing Voice Synthesis

Chan-Chuan Chu, Fu-Rong Yang, Yi-Jhe Lee, Yi-Wen Liu, and Shan-Hung Wu
National Tsing Hua University, Hsinchu, Taiwan
E-mail: chu850503@gmail.com, fjbcrs34@gmail.com, ywliu@ee.nthu.edu.tw
Tel/Fax: +886-3-5162205

***Abstract*—The purpose of singing voice synthesis (SVS) is to generate human-like singing voice from lyrics and the corresponding musical score. Nowadays, mainstream SVS approaches rely on neural networks (NNs) which can map linguistic and musical contextual factors to acoustic features for producing audio outputs. For SVS in Mandarin or other Chinese languages in particular, a sufficiently large and adequately labeled database has not been publicly available. To proceed with Mandarin SVS research, we built a singing voice database from scratch, with 600 pop songs sung by 2 male and 2 female vocalists. Each audio contains single vocal only, without any background music. This paper describes the recording of the dataset and necessary steps of data preprocessing for training NNs to perform SVS. Several simple neural network architectures were adopted so preliminary SVS performance can be compared. Both subjective and objective evaluations show that these networks could learn from the MPop600 database to generate singing voice with unseen musical scores. MPop600 is available in both the MIDI and the MusicXML formats. In the future, we believe that more advanced and recently developed networks can be applied to model the singing behaviors in this database and help advance research in Mandarin SVS.**

I. Introduction

Natural singing voice synthesis (SVS) is an emerging research topic, and previous attempts have aimed to synthesize singing in different languages, including Japanese [1], Spanish [2], Korean [3] and so on. Even though Mandarin is the second most spoken language in the world — only after English — however, Mandarin SVS remains relatively under-explored [4], [5], [6]. One possible reason might be the lack of publicly available datasets that are sufficiently large and meticulously labeled to enable supervised learning. Therefore, we aimed to create and share a Mandarin singing database so as to facilitate future research in Mandarin SVS. Our database is named MPop600 to indicate explicitly that it contains 600 popular songs in Mandarin, sung by native speakers of the language.

SVS differs from speech synthesis in that the synthesized singing voice needs to follow the musical scores; performance of pitch and rhythm synthesis would directly influence the perceived quality. According to the techniques adopted by different systems, existing SVS methods can be categorized into concatenation-based SVS (CSVS) [7], hidden Markov Model (HMM)-based SVS [1], or neural network(NN)-based SVS. Within the realm of CSVS, commercially available tools such as Vocaloid [8] and Synthesizer V[1] have successfully gathered loyal groups of users. In principle, human-like singing voice is synthesized by concatenating sample units that are found in a corpus. Hence, the performance of CSVS directly depends on that the corpus covers all possible phonemes and syllables of the language of interest. A shortcoming of CSVS is perhaps the lack of flexibility to change the voice characteristics. Auxiliary systems such as the Vocalistener [9] have been built to grant the users with certain degrees of freedom so the synthesized voice characteristics can be adjusted in artistic ways.

In contrast, HMM-based SVS [4], [10] can model the spectral envelopes, excitation, and the singing voice duration separately. Then, speech parameter generation algorithms [11] are used to produce singing voice parameter trajectories. As HMM predates the advances in deep learning, the naturalness of HMM-based SVS is outperformed by what could now be achieved by neural networks. Over the past few years, several types of neural networks have been adopted for SVS, such as a generic deep neural network (DNN) [12], a recurrent neural network with long-short term memory (LSTM-RNN) [3], and generative adversarial networks (GAN) [13], [14] most recently.

In this paper, besides presenting the MPop600 database, we also built SVS systems based on DNN, LSTM, and bi-directional LSTM (BiLSTM) so as to compare performance and evaluate whether it takes certain additional care to synthesize Mandarin singing voice due to the tonal nature of the language. Our systems separately consider the fundamental frequency (F0) and "pronunciation", loosely referring to all aspects of word enunciation other than controlling the pitch. To train the neural nets, we first annotated the lyrics and musical scores for all the songs. Then, contextual features were carefully designed and extracted from the lyrics and scores so they can be fed as the input to the NNs. Meanwhile, the audio signals were aligned to the lyrics and the notes, and acoustic features were computed from the audio via the WORLD vocoder [15]. Thus, the contexual factors were paired with the corresponding acoustic features with frame-level accuracy,

[1] https://synthesizerv.com/en/

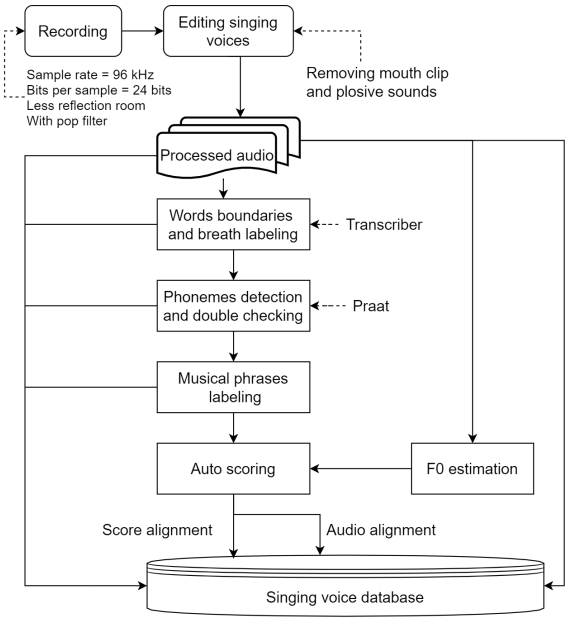

Fig. 1. The structure of Mandarin singing voice database creation.

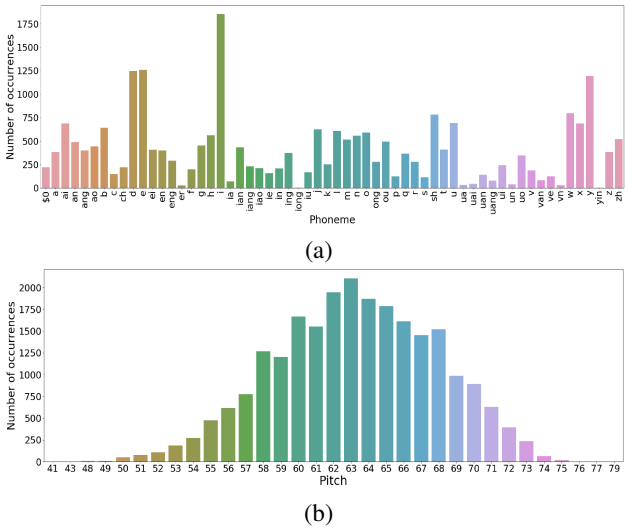

Fig. 2. (a) The statistical distribution of phonemes in the MPop600 database. The phoneme symbols are initial/final of Pinyin. (b)The statistical distribution of musical pitches in the MPop600 database. Here, 60 equals C4 in musical notation.

so that the mapping from the contextual domain to the acoustic domain can be learnt. To compare the performance of different NNs, subjective and objective evaluations were conducted in terms of the mean opinion score (MOS) and the Mel-cepstral distortion (MCD), respectively.

The rest of this paper is organized as follows. Section II describes the details about the establishment of Mandarin singing voice database and auto transcription, which is the main part in this research. Section III introduces the design of the voice synthesis system as well as input and output features. Section IV describes the experimental condition and the result evaluations. Conclusions are given in Section V.

II. Database Creation

The flowchart for creating the MPop600 database is shown in Fig. 1. First, each of the four participating singers, two male and two female, was invited to freely choose 150 songs they would like to record. So, in total, 600 Mandarin pop songs were recorded. Next, we labeled the linguistic information manually and musically transcribed the singing voices semi-automatically. Thus, the database consists of 600 sets of aligned lyrics, musical scores, and the corresponding vocal audios, and statistical distributions of the phonemes and note pitches are shown in Fig. 2.

A. Considerations for recordings and sound engineering

Two male and two female singers were asked to sing 150 songs each. To ensure the steadiness of tempo and accuracy of pitch, the singers were required to sing synchronously with the accompany audio and tempo click, which were played into the headphone, so each recorded audio only contains pure vocal, without any background music. What's more, to increase the diversity of phonetics and musical expressiveness within limited time, only the first verse and the chorus were recorded for each song. The recording data length was about 2.5 hours per singer and about 10 hours in total.

In addition, every song preserves two beats in front of the measure before the singing starts, so that the score could easily align with the singing voice. Besides, to reduce the noise caused by the singers, we applied iZotope RX7[2], a commonly used software in the music production industry, to remove the mouth clip and plosive sounds.

The singing voices were recorded by a condenser microphone and RME UFX, a high resolution audio interface, in a room with sufficiently little reverb. The recording setup met the professional studio standards. A filter was also used to obstruct fast airflow and prevent popping sounds from being recorded during singing. Finally, the data were recorded in 96kHz sampling rate with 24 bits per sample in `.wav` (Waveform Audio File Format) format.

B. Labeling

The word boundaries of lyrics were labeled manually on Transcriber[3], a tool for annotation of speech signals. To ensure the precision of word boundaries, we double checked the labels by Praat [19], which provides the visualization of spectrum, pitch, intensity, and formants of the audio, and this helped us to discover some erroneous boundaries. After word transcription, the Mandarin characters were translated into Pinyin. As for phoneme duration labeling, an open source phoneme-level alignment tool called Speech-Aligner[4] was applied. In this

[2]https://www.izotope.com/en/products/rx/features.html
[3]http://trans.sourceforge.net/en/presentation.php
[4]https://github.com/open-speech/speech-aligner

research, the initial/final of Pinyin was utilized to represent Mandarin word pronunciation. By marking the boundaries between initial/finals of Pinyin, pronunciation of the lyrics has been unambiguously encoded and aligned to the audio temporally.

C. Automatic singing voice transcription

The musical score of a song includes the note pitch and duration, the tempo, the key signature, and the time signature, and so on. Instead of creating the MIDI score directly by operating on a electronic piano keyboard, we came up with a semi-automatic singing voice transcription. The method consists of three steps:

- *Pitch determination*: Due to some personal skills such as vibrato or glissando, the mean of F0 of each whole word may not correspond to the musically correct note intended by the singer. Fig. 3 (a) shows a typical result in which the note pitch is calculated by the mean of F0 within a whole word. In order to obtain the note pitch accurately, each word is cut into 100-ms segments, and then the mean F0 of the segment which has the smallest variance was defined as the pitch of the note. Fig 3 (b) shows a typical result in which the note pitch is determined by the above method. During the creation of MPop600 database, we assumed that each word corresponds to only one note; exceptions remain to be handled in this research.
- *Duration quantization*: We assumed that any note duration in a pop song is longer than 1/16 note. Therefore, the length of a word (or note) is quantized to the closest integer multiples of a half of note, a 1/4 note, a 1/8 note or a 1/16 note.
- *Adjustment of rhythm*: Continuing from above, a dynamic quantization method was used to modify the rhythm so the transcription would be more musically reasonable while allowing syncopation. Depending upon the note duration, the onset of each note was moved to an integer multiple different lengths; a note equal to a 1/4 note or longer would be moved to align with the 1/8 note grid line. Similarly, a note duration originally equal to or shorter than 1/16 note would be moved to align with the 1/32 note grid line. While the note duration is shorter than a 1/4 note but longer than a 1/16 note, it would be moved to align with the 1/16 note grid line. Fig. 3 (c) shows a typical result of the proposed automatic singing voice transcription approach.

After transcribing the singing voice automatically, musical scores were saved in two formats: Musical Instrument Digital Interface (MIDI) and Music eXtensible Markup Language (MusicXML), so that a score could be manually labeled or corrected if necessary. We envision that, if necessary, the pitches, durations, and the relative positions of the notes in a bar can be easily modified in the MIDI format on a digital audio workstation, and the MusicXML format can be used to label high-level information, such as the onset and offset of a musical phrase, or to add meta-data, such as the key signature and the time signature.

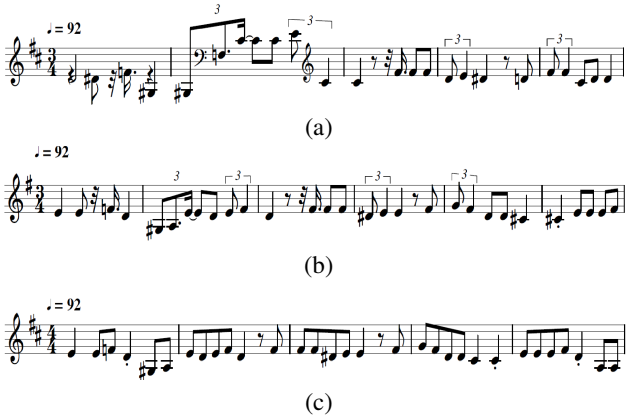

Fig. 3. The typical result of the note pitch estimation method. (a) is the one in which the note pitch is calculated by using the mean of F0 contour within whole word. (b) is the result of the proposed pitch determination method. (c) is the result of the proposed automatic singing voice transcription approach.

III. Synthesis System

A. System design

In Fig. 4 the system is divided into the training and the synthesis part. In the training part, the input and the output of the system are, respectively, the contextual features and the acoustic features extracted from the singing voice database. The contextual features were extracted from the lyrics and the musical scores, which had been automatically transcribed from singing voice as mentioned in Sec. II. The acoustic features consisted of the fundamental frequency (F0), the mel-generalized cepstrum (MGC) [17], the band-aperiodicity (BAP) [18] and the voice/unvoice decision (VUV) [15]; the MGC and BAP were encoded from spectral parameters provided by the WORLD vocoder. Hereafter, we abbreviate MGC, BAP and VUV as *MBV*. Subsequently, F0 and MBV were trained separately, because empirically we found that, if all of the output features are trained in the same network, the model might not be able to find the correct learning direction even when the output loss is weighted.

Furthermore, we found that different representations of F0 could influence the performance of prediction. In this work, instead of predicting the F0 directly, a ΔlogF0 term which denotes the difference between the singing voice logF0 and the musical note in logarithm was adopted to be the representation of F0. Thus, ΔlogF0 was supposed to be easier to model than logF0 itself, because the distribution of logF0 could be too sparse. However, there was a jumping problem of ΔlogF0 because, in Mandarin as well as other languages, an unvoiced period might occur at the beginning of a word instead of during a rest. This situation would make ΔlogF0 ill-defined. A pitch normalization method [12] was incorporated to mend the jumping problem.

Therefore, in the synthesis part, Fig. 4 shows that the note pitch of the input is used in combination with the predicted ΔlogF0 and threshold VUV to calculate the final F0 contour in Hertz. Happening in parallel, the predicted MGC and BAP

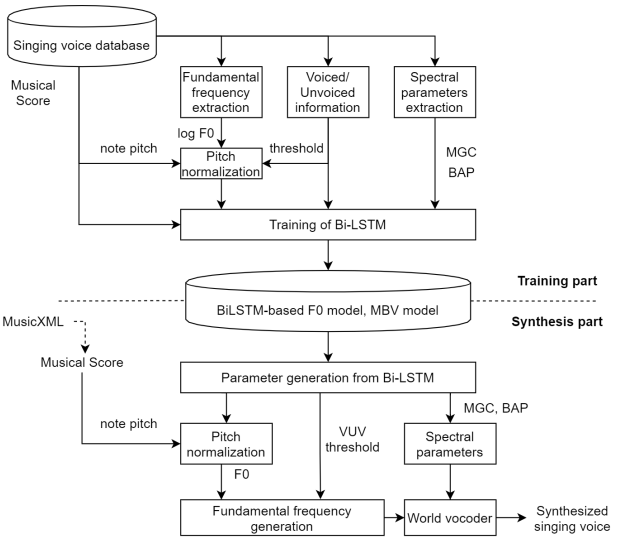

Fig. 4. Overview of the singing voice synthesis system based on Bi-LSTM.

are decoded back to the spectral parameters. After that, they are delivered to WORLD vocoder for synthesizing the audio.

B. Organization of the input and output features

Contextual factors regarding the musical scores with lyrics include phoneme identity, note pitch, note duration, rhythm, tempo, musical phrase, and so on. In most SVS systems, contextual factors are encoded to become the input feature vectors. According to our team's previous work [16], the F0 contour may slightly depend on the tonality of the lyrics when a singer sings in Mandarin. Thus, we designed a set of contextual factors that takes tonality into account. The contextual factors are organized into five layers, listed as follows with the smallest unit first:

- phoneme – phoneme identity, tone, position of the phoneme in a Mandarin character (initial/final)
- note – pitch, musical key, note duration, position of the note in a musical measure
- character – tone of the character (high and level, rising, falling and rising, or falling), number of notes
- phrase – number of phonemes, notes, and characters
- song – number of phonemes, notes, measures, and phrases

After analysis, the contextual features are represented in the phoneme level. Let $\mathbf{x}$ be the input sequence, which comes from the contextual factor analysis. Thus, we have

$$\mathbf{x} = \{x_1, x_2, ..., x_{N_p}\}, \tag{1}$$

where N_p denotes the number of the phonemes, and x_i denote the feature vector corresponding to the i-th phoneme in the sequence, $i = 1, 2, ..., N_p$. On the output side, let $\mathbf{y}$ be the acoustic feature sequence comprised of ΔlogF0, MGC, BAP and VUV; that is,

$$\mathbf{y} = \{y_1, y_2, ..., y_{N_f}\}, \tag{2}$$

TABLE I
THE ARCHITECTURES AND HYPERPARAMETERS OF DNN LSTM MODELS

DNN		LSTM	
Item	**Details**	**Item**	**Details**
N_{layer}	4	N_{neuron} in latent layers	2048
N_{neuron}, layer 1	1024	N_{layer}	2
N_{neuron}, layer 2	512	N_{neuron} in LSTM kernel	1024
N_{neuron}, layer 3	256	N_{neuron} in dense layer	128
N_{neuron}, layer 4	128	Batch size	16
Batch size	16	Step size	20

where N_f is the number of frames, and y_t denotes the output feature vector of the t-th frame, $t = 1, 2, ...N_f$. With the input $\mathbf{x}$, the model predicts the acoustic features at each time step from the input. To synchronize the input and the output sequences, the time stamp of contextual feature vectors have to be converted from the phoneme level to the frame level.

C. Feature Extraction and Waveform Synthesis

Based on the WORLD vocoder, singing voice signals are represented by F0 plus two spectral parameters, namely the spectral envelopes (SP) and the aperiodicity (AP). They were estimated by the state-of-the-art methods HARVEST [20], CheapTrick [21], and D4C [22], respectively. In this research, the frame shift was set at 5 ms. The FFT length was set to 4096 at a sampling rate of 48000 Hz. Instead of taking the vocoder features directly, SP and AP were encoded into 60-dimensional MGC [17] and 5-dimensional BAP [18] respectively, since reducing the dimension of the output feature space can make it easier for the model to learn the mapping. In addition, VUV was also extracted to switch on or off the fundamental frequency generation module (see Fig. 4). After the prediction, they were converted back to the SP, AP, F0 correspondingly. Finally, the frame-level SP, AP and F0 were passed to the WORLD vocoder for synthesizing the singing voice.

IV. EXPERIMENTS AND DISCUSSION

Here we report results of selecting 40 songs for training the neural networks. All the songs were sung by the same female singer. Additionally, we selected one song for validation, and two songs for evaluation. The total audio length of the training dataset is about 1 hour. As mentioned previously, the training was separately conducted by the F0 model and the MBV model.

In this research, we constructed and compared a deep neural network (DNN), an LSTM RNN, and a bidirectional LSTM (Bi-LSTM) network. A 4-layer DNN was trained and regarded as the baseline system, while both LSTM and Bi-LSTM had two hidden layers. The details of the neural network architectures and the setting of the hyper-parameters are tabulated in Table I. Additionally, in order to assess whether tonaliy is important for Mandarin SVS, we also compared two LSTM models; one of them considered word tonality while the other did not.

All models were trained with a mini-batch stochastic gradient descent (SGD)-based back-propagation algorithm, and with Adam [23] as the optimizer. Also, the learning rate was

TABLE II
RESULTS OF OBJECTIVE EVALUATION

Model	model 1	model 2	model 3	model 4
Method	DNN	LSTM	LSTM+*	BiLSTM+*
	Slow song		**Fast song**	
Model	**F0-RMSE**	**MCD**	**logF0-RMSE**	**MCD**
model 1	0.057	11.16	0.061	11.41
model 2	0.055	8.77	0.054	8.85
model 3	0.046	8.59	0.049	8.85
model 4	**0.042**	**8.45**	**0.047**	**8.66**

exponential decayed; it was initially set to 0.001, and decayed with a base of 0.95 per 10 epochs. The loss function was the measurement of mean squared error between target values and predicted values. In addition, the activation functions of all models were Sigmoid [24]. Regarding feature normalization, we followed Saino et al. [10] so that the input features were normalized to the range from 0 to 1 and output features were scaled to the range from 0.01 to 0.99.

A. Objective Evaluations

Four frameworks based on different neural networks and features were trained to observe if the proposed dataset is trainable for SVS. The differences between the four frameworks are described in Table II, including: (1) DNN without tonality consideration (baseline), (2) LSTM without tonality consideration, (3) LSTM with tonality consideration (+*), and (4) Bi-LSTM with tonality consideration (+*).

To evaluate the performance of the proposed systems objectively, we calculated logF0 root mean squared error (logF0-RMSE) and the MCD. The base of logF0-RMSE was 10 (instead of the more commonly used base-2 logarithm in music theory),

$$\text{logF0-RMSE} = \sqrt{\frac{1}{N}\sum_{i}(\log_{10} F_i - \log_{10} \hat{F}_i)^2}, \quad (3)$$

where F_i denotes the target F0 at the ith frame, $\hat{F}_i$ denotes the predicted F0 for the ith frame, and N is the total number of frames for a song.

MCD is commonly used for synthesized speech quality assessment. It quantifies the distance between two sequences of mel cepstra. In this research, a small MCD between the synthesized singing and the ground truth is preferred; the following equation defines the MCD for this research,

$$\text{MCD (dB)} = \frac{10\sqrt{2}}{ln10}\frac{1}{T}\sum_{t=1}^{T}\sqrt{\sum_{i}(C_{ti} - \hat{C}_{ti})^2}, \quad (4)$$

where C_{ti} denotes the i-th Mel cepstral component at the t-th target frame and $\hat{C}_{ti}$ denotes the same component for the synthesized frame.

The objective evaluation was conducted over 2 songs that were previously unseen during the training phase. The evaluation results in Table II shows that three LSTM-based frameworks outperform the baseline method in terms of both logF0-RMSE and MCD. This is not surprising, because temporal relationship among input frames was not explicitly considered by the DNN. In addition, adding tonality as input features helps to predict both the F0 contour and, somewhat suprisingly, the acoustic output. To explain this, note that in Mandarin, word meanings are partially conveyed by tones. Consequently, when a Mandarin song is sung, word meanings might potentially be misunderstood because the direction of the tone — rising, falling, or flat — might conflict with the melodic direction.

Furthermore, Bi-LSTM achieves the lowest logF0-RMSE and MCD. This is reasonable because, like an experienced singer, a neural network should consider both the past and the future notes to "sing" expressively instead of just looking to the past. In this sense, Bi-LSTM might have implicitly characterized the position of a note *within a musical phrase* and thus become the most successful singing network here.

The logF0-RMSE and MCD achieved by the three LSTM-based networks are in a reasonable range; for comparison purposes, logF0-RMSE in [4] was in the range of 0.045 to 0.049, and MCD in [3] was in the range of 5.43 to 8.61.

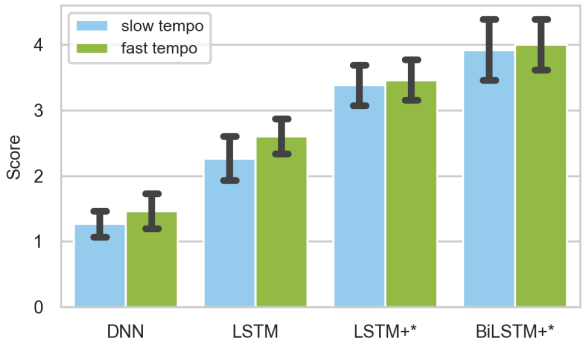

Fig. 5. Results of MOS test for each methods.

B. Subjective Evaluations

A subjective listening test was carried out to evaluate the synthetized singing voices. 15 subjects participated in the evaluation to rate the pitch and pronunciation accuracy of the generated singing voice. The mean opinion score (MOS) with a scale from 1 (poor) to 5 (good) was adopted. All subjects are native Mandarin speakers. The listening material consisted of the same two songs that were chosen for objective evaluation.

Figure 5 shows the results of subjective evaluation. The height of the bar shows the mean score across 15 subjects, and the error bar shows the 95% confidence intervals determined as follows,

$$\text{CI} = \left[\hat{\mu} - 1.96\frac{\hat{\sigma}}{\sqrt{N}}, \hat{\mu} + 1.96\frac{\hat{\sigma}}{\sqrt{N}}\right], \quad (5)$$

where μ denotes the mean score responded by the subjects, $\hat{\sigma}$ is the corresponding standard deviation, and $N = 15$ denotes the number of the subjects [25].

The present results suggest that all the LSTM-based systems achieve higher MOS than the baseline. Also, songs with slow and fast tempo have similar MOS range. Note that in the present experiment, only 40 songs were included in the training set. While we continue to label the data, we believe that better performance could be obtained in the future when the size of the training dataset increases.

V. Conclusions

In this research, we intend to build a system that can sing a Mandarin song by "sight-reading" — including the lyrics and the sheet music. In order to implement it on neural networks, we created a dataset that contains 600 Mandarin pop songs from scratch. The dataset was prepared with care during the recording and the labeling phase. In particular, singing voice transcription had to be achieved automatically because the participating singers freely chose the songs to sing and thus the musical score of the songs were mostly unavailable. The data was preprocessed to obtain the contextual and acoustic features. Four neural-network models were adopted to find the mapping from the contextual feature space to the acoustic feature space. We verified that, by using the proposed dataset, a Mandarin SVS system is trainable; results from both the objective and the subjective evaluation suggest that (i) the sequential nature of music matters, as LSTM-based model consistently outperformed the DNN, and (ii) explicitly telling the system to consider word tonality helps to improve the synthesized sound quality as far as Mandarin SVS is concerned. In the future, we believe that techniques developed more recently can be applied to model Mandarin pop song singing using this dataset. Therefore, we will gladly release the dataset to academia and maintain it after the conference.

Acknowledgement

This research is supported by the Ministry of Science and Technology of Taiwan under Grant No. 108-2634-F-007-003 awarded to SHW. The authors thank Dr. Yi-Hsuan Yang for research consulting and offering critiques during the writing process.

References

[1] K. Nakamura, K. Oura, Y. Nankaku, and K. Tokuda, "HMM-based singing voice synthesis and its application to Japanese and English," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, pp. 265-269, May 2014.

[2] M. Blaauw and J. Bonada, "A neural parametric singing synthesizer modeling timbre and expression from natural songs," *Applied Sciences,* vol. 7, no. 12, p. 1313, Dec. 2017.

[3] J. Kim, H. Choi, J. Park, M. Hahn, S. Kim, and J. Kim, "Korean singing voice synthesis based on an LSTM recurrent neural network," in *Proc. INTERSPEECH*, pp. 1551-1555, Sep. 2018.

[4] Li, Xian, and Zengfu Wang, "A HMM-based mandarin Chinese singing voice synthesis system," *IEEE/CAA Journal of Automatica Sinica*, 3.2, pp. 192-202, 2016.

[5] Gu, Y., Yin, X., Rao, Y., Wan, Y., Tang, B., Zhang, Y., ... and Ma, Z. (2020). "ByteSing: A Chinese singing voice synthesis system using duration allocated encoder-decoder acoustic models and WaveRNN vocoders," *arXiv* preprint arXiv:2004.11012.

[6] Lu, P., Wu, J., Luan, J., Tan, X., and Zhou, L. (2020). "XiaoiceSing: A high-quality and integrated singing voice synthesis system," *arXiv* preprint arXiv:2006.06261.

[7] Bonada J, Serra X, "Synthesis of the singing voice by performance sampling and spectral models," *IEEE Signal Processing Magazine,* 24.2, pp. 69-79, 2007.

[8] Kenmochi H, Ohshita H. "VOCALOID-commercial singing synthesizer based on sample concatenation," in *Proc. 8th Annual Conf. Int. Speech Communication Assoc.*, Antwerp, Belgium, pp. 4009-4010, 2007.

[9] Nakano, T., and Goto, M. (2009). "VocaListener: A singing-to-singing synthesis system based on iterative parameter estimation," *Proc. SMC,* pp. 343-348.

[10] K. Saino, H. Zen, Y. Nankaku, A. Lee, and K. Tokuda, "HMM-based singing voice synthesis system," in *Proc.Ĩnt. Conf. Spoken Language Processing (ICSLP)*, pp. 2274-2277, 2006.

[11] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)* vol.3, IEEE, pp.1315-1318, 2000.

[12] M. Nishimura, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Singing voice synthesis based on deep neural networks," in *INTERSPEECH 2016,* pp. 2478-2482, Sep 2016.

[13] P. Chandna, M. Blaauw, J. Bonada, and E. Gómez, "WGANSing: A multivoice singing voice synthesizer based on the Wasserstein-GAN," in *2019 27th European Signal Processing Conference (EUSIPCO),* IEEE, pp. 1-5, 2019.

[14] Y. Hono, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Singing voice synthesis based on generative adversarial networks," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* IEEE, pp. 6955-6959, May 2019.

[15] Morise, Masanori, Fumiya Yokomori, and Kenji Ozawa, "WORLD: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE transaction on information and systems*, 99.7, pp. 1877-1884, 2016.

[16] Y.-J. Lee, B.-Y. Chen, Y.-T. Lai, H.-W. Liao, T.-C. Liao, S.-L. Kao, K.-Y. Kang, C.-T. Hsu, and Y.-W. Liu, "Examining the influence of word tonality on pitch contours when singing in mandarin," in *Proc. Oriental COCOSDA - International Conference on Speech Database and Assessments,* pp. 89-94, May 2018.

[17] K. Tokuda, T. Kobayashi, T. Masuko, and S. Imai, "Mel-generalized cepstral analysis," in *Proc. ICSLP*, pp. 1043-1046, 1994.

[18] H. Zen, T. Toda, M. Nakamura, and K. Tokuda, "Details of the nitech HMM-based speech synthesis system for the blizzard challenge 2005," *IEICE transaction on information and systems*, vol. 90-D, pp. 325-333, 2007.

[19] P. Boersma and D. Weenink, "Praat, a system for doing phonetics by computer," *Glot international,* vol. 5, pp. 341-345, Jan. 2001.

[20] M. Morise, "Harvest: A high-performance fundamental frequency estimator from speech signals," in *Proc. INTERSPEECH*, pp. 2321-2325, Aug. 2017.

[21] M. Morise, "CheapTrick, a spectral envelope estimator for high-quality speech synthesis," *Speech Communication*, vol. 67, pp. 1-7, 2015.

[22] M. Morise, "D4C, a band-aperiodicity estimator for high-quality speech synthesis," *Speech Communication*, vol. 84, pp. 57-65, 2016.

[23] D. Kingma and J. Ba. (2014). "Adam: A method for stochastic optimization," *arXiv* preprint arXiv:1412.6980.

[24] J. Han and C. Moraga, "The influence of the sigmoid function parameters on the speed of backpropagation learning," in *From Natural to Artificial Neural Computation* (J. Mira and F. Sandoval, eds.), (Berlin, Heidelberg), pp. 195-201, Springer Berlin Heidelberg, 1995.

[25] K. Kumar, R. Kumar, T. de Boissiere, et al., "MelGAN: Generative adversarial networks for conditional waveform synthesis," in *Advances in Neural Information Processing Systems,* pp. 14910-14921, 2019.