End-to-End Mongolian Text-to-Speech System

Jingdong Li, Hui Zhang, Rui Liu, Xueliang Zhang, Feilong Bao*

College of Computer Science, Inner Mongolia University
Inner Mongolia Key Laboratory of Mongolian Information Processing Technology
Hohhot 010021, China

jingdong.li@mail.imu.edu.cn, cszh@mail.imu.edu.cn

Abstract

Speech synthesis, or text-to-speech (TTS), generates a speech waveform of the given text. To build a satisfactory TTS system, a large natural speech corpus is requested. In the traditional approach, the corpus should be accompanied with precise annotations. However, the annotation is difficult and costly. Recently, end-to-end speech synthesis methods are proposed, which eliminated the requirement of annotation. The end-to-end methods make the development of TTS system less costly and easier. We used the state-of-the-art end-to-end Tacotron model in the Mongolian TTS task. With much more unannotated speech data (about 17 hours), the new system beats the old best Mongolian TTS system, which is trained on a small amount of annotated data (about 5 hours), with a big margin. The new mean opinion score (MOS) is 3.65 vs 2.08 which is the old one. The proposed system becomes the first Mongolian TTS system can be utilized in real applications.

Index Terms: Mongolian, speech synthesis, TTS, end-to-end

1. Introduction

Speech synthesis, or text-to-speech (TTS), generates a speech waveform of the given text [1]. TTS is one of the most important components of the voice user interface. With the widespread use of smart voice assistants (e.g. Siri from Apple), research on speech synthesis draws more and more attention [2]. Most traditional speech synthesis methods fall into two categories: unit-selection synthesis [3] and statistical parametric speech synthesis (SPSS) [4]. Unit-selection synthesis selects appropriate sub-word units from large corpora of natural speech, then concatenates the selected units to form the output. In contrast to selecting the actual speech instances, SPSS models the sub-words with parametric models and generates speech from these parametric models. Unit-selection owns higher naturalness, while SPSS over the unit-selection synthesis in flexibility and controllability.

Both of the unit-selection synthesis and the SPSS request a large natural speech corpus. It appears that the larger the corpus the better the quality of the synthesized speech. Unfortunately, recording large corpus is very difficult and costly [5]. To build such a speech corpus, firstly, we need carefully select the sentences to make them phonetically and prosodically balanced. Secondly, we need carefully select a suitable speaker who should give a pleasant voice, with good voice quality and professional recording experience. Thirdly, we need to be equipped with professional recording devices and environment which can obtain high-quality noise-free recordings. Lastly and the most costly, we need to give precise annotations on these recordings.

Data annotation for TTS is much difficult and costly than other applications. For example, speech recognition only needs to transcribe the recordings at word level. However, TTS system needs to model a series expressive factors of speech, which includes intonation, stress, rhythm, and so forth. To fit the requirement of the TTS, we need to transcribe the recordings not only in word level, but also in sub-word (e.g. phoneme) level. The non-speech events like breathing or clicking also need to be picked out. The intonation, stress, rhythm, syllable and prosody have to be annotated, too. All of these annotations should be aligned to the time-line of the recording. Annotation boundaries also need careful fine tuning. Because any misannotation often causes glitches in the synthesized speech, we need double check these annotations. All of these requirements make the data annotation difficult and costly. In our experience, the annotation of one hour's recording expends one man-month work, and involves at least two native speaking and professional annotators, where one for the first-phase annotation and another for revision. In average, one hour's TTS speech data cost over 1000 US dollars, in where the annotation cost the 95%.

The requirement of annotation much restricts the quality improvement of the TTS systems, especially for the languages whose resources are scarce. Mongolian is one of these languages. As far as our knowledge, the largest annotated Mongolian speech corpus, which can be used in TTS, only contains about 5 hours recordings. The state-of-the-art TTS system is built upon it only obtains 2.08 mean opinion score (MOS) [1], which means the perceived quality is "poor". In fact, this system cannot be used in any real applications. However, the requirement of a usable Mongolian TTS system is urgent. Mongolian is an influential language. There are about 6 million people who speak Mongolian language all over the word. Mongolian is one of the five major minority languages in China, and is one of the official languages in the Inner Mongolia Autonomous Region of China. To improve the quality of the synthesized Mongolian speech, and develop a TTS system can be used in real applications, we seek for some approaches which do not need the costly data annotation.

The end-to-end learning is a solution to our question. It takes all of multiple stages required by the conventional processing, and replaces them usually with just a single neural network. With the development of deep learning, end-to-end model have achieved significant improvement in many tasks in recent years [6–10]. Specifically, several end-to-end speech synthesis models has been successfully applied to English and other languages [11–14]. The end-to-end TTS system can be trained to predict audio from the text directly, which minimize the costly annotation work. In this work, we used the state-of-the-art end-to-end Tacotron model [14] in the Mongolian TTS task. Because the Tacotron model can be trained with

[1] See our experiments in section 4 for details.

<text, audio> pairs only, we can build a larger corpus with our limited budget. As a result, a Mongolian corpus is built, which contains about 17 hours recordings with word-level transcriptions. Although it smaller than the actually used English TTS corpus, it much larger than the existing largest Mongolian TTS corpus. A Mongolian TTS system is built with the Tacotron model trained with the new corpus. The new system obtains a MOS at 3.65, which beats the old best system with a big margin.

The rest of the paper is organized as follows: related work on end-to-end speech synthesis and Mongolian speech synthesis are described in the next section. Section 3 describes the proposed Mongolian speech synthesis system. Section 4 shows the experimental conditions and results. The conclusions are proposed in the last section.

2. Related Works

The Mongolian TTS research follows the SPSS methods. Hidden Markov model (HMM) [15] has dominated SPSS for a long time. Then, deep neural network (DNN) has been introduced into SPSS [16]. Zhao applied the HMM in the Mongolian speech synthesis in 2014 [17]. Then Liu introduced DNN to Mongolian TTS in 2017 [18]. But limited by available annotated data, both of these two methods cannot give a natural-sounding and even clear synthesized speech, which is not satisfied the requirements of application.

The costly annotation processing is the bottleneck to improve the performance of TTS systems. The end-to-end method, which only uses the <text, audio> pairs as its training data, is proposed to break the bottleneck. [11] is declared as the first step towards end-to-end parametric TTS synthesis. A sequence to sequence model with attention is used to convert the input phonemes to vocoder parameters. [12] proposed a Char2Wav model, which simplified the input. It converts the raw text (characters and not phonemes) to vocoder parameters, directly. [13] proposed a VoiceLoop model. It converts the input characters to vocoder parameters, and can be trained with multiple speakers. [14] proposed a Tacotron model. It converts the input characters to spectrogram. In this work, we adopted the Tacotron model, because it can generate high-quality speech and has open-source implementation. The original paper did not show the results on more languages than English. Our work provided that it can be used in Mongolian and can generate comparable results as in English.

Before we starting this work, a new version of Tacotron model [19] has been proposed, which improved the quality of generated speech. However, the new model involved a complex autoregressive WaveNet model [20] as vocoder. WaveNet generates output audio sample by sample, which results the new Tacotron model cannot generate audio in real-time. The processing speed is critical for our application, therefore we do not employ the new Tacotron model.

3. System Architecture

In this work, we apply the Tacotron model in the Mongolian TTS task. The Tacotron model is a neural text-to-speech model that learns to synthesize speech directly from <text, audio> pairs. In the Mongolian TTS, the training target is utterance which is recorded by a single professional speaker, and the input is the corresponding transcription in Mongolian.

Because of historical issues, nowadays, Mongolian language has two writing systems: 1) Traditional Mongolian scripts used in Inner Mongolia region and 2) Cyrillic scripts used in the Mongolia. Although the two writing systems use different alphabets, both of them are used to transcribe the same pronunciation. Therefore, there are some relaxed counterparts between their alphabets. Base on the alpha-level corresponding, the traditional Mongolian scripts can be converted to Cyrillic scripts, and vice versa [21, 22]. As the middle layer of the transformation, a Mongolian Latin transliteration scheme is used [23]. Both traditional and Cyrillic Mongolian scripts can be converted into this Latin form, and vice versa. To generate synthesized speech from both of the traditional and Cyrillic Mongolian scripts, in this work, we use the Latin scripts as the input of the Tacotron model.

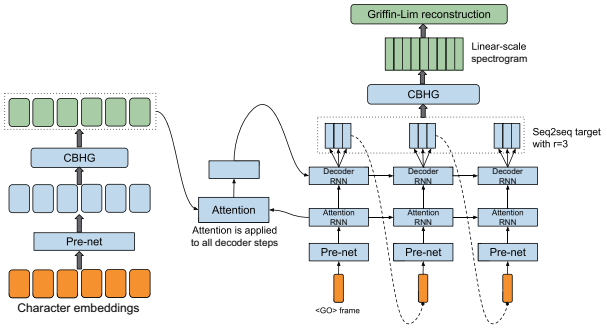

Figure 1: *Model architecture as originally presented in the Tacotron paper [14].*

The Tacotron model is a complicate neural network architecture, as shown in figure 1. It contains a multi-stage encoder-decoder based on the combination of convolutional neural network (CNN) and recurrent neural network (RNN). The raw Mongolian text in Latin script input is fed into an encoder which generates attention features. Then the generated features are fed in every step of the decoder before generating spectrograms. At last, the generated spectrograms are converted to waveform by the GriffinLim method [24].

Our implementation is forked from the TensorFlow implementation from Keith Ito [2], which is faithful to the original Tacotron paper. We first convert the Latin script Mongolian to character sequence. Then the character inputs are converted into one-hot vectors. In the length of the one-hot vectors is 40, corresponding to the number of characters in the Latin script Mongolian. The one-hot vectors then turn into character embeddings and are fed to a pre-net which is a multilayer perceptron. The output is then fed into the CBHG which stands for Convolutional Bank + Highway network + gated recurrent unit (GRU). The output from CBHG is the final encoder representation used by the attention module. Then the generated attention features are fed in a RNN-based decoder. In every step, the decoder generate few frames of spectrograms. The number of output frames is controlled by a hyperparameter reduction factor (r in Fig. 1). At last, the audio is reconstructed by the Griffin-Lim method. We set the mel-scale spectrogram and linear spectrogram as target. The above spectrograms is obtained from the original 22.05 kHz signal with a sliding window of 50 ms frame length, and 12.5 ms frame shift.

In the speech synthesis stage, we first convert the input traditional or Cyrillic Mongolian input text into Latin script, the inputs are split into sentences. We synthesis the output sentence

[2]https://github.com/keithito/tacotron

by sentence, which is limited by the modeling length of the Tacotron model. At last, we concatenate these synthesized waveforms together.

To ensure low latency of the Mongolian speech synthesis system, we compile the model into both the CPU and GPU instruct. The high parallel computing ability of GPU makes processing speed is much higher when a batch of requirements are occurring concurrently. For example, if the input text contains many sentences, we can synthesize each sentence in parallel. We deploy the trained model on a GPU platform as our production environment.

4. Experiments and Results

4.1. Dataset

Because of the annotation cost, as far as we know, there are no large scale speech synthesis corpus in Mongolian language. The largest Mongolian speech synthesis corpus is reported in [17] and [18]. With a private communication with their common corresponding author, we got known that the current largest speech synthesis corpus is contains about 5 hours Mongolian recording data, and has been used to train their DNN-based and HMM-based TTS system. A HMM-based TTS demo is accessible for public. We are really thank the author provide us with some synthesized outputs from the DNN-based method.

Since the proposed end-to-end framework required only few annotations, we can easily build up a larger corpus. A professional Mongolian native female speaker was invited to record audio. Two Mongolian native speakers double checked the corresponding between the text and the audio recording. Our large-scale corpus covers common daily usage scenarios of Mongolian. The sentences are selected to make phonetically and prosodically balanced. The texts in this dataset are normalized. For example, the number "1" will be normalized to Mongolian Latin representation "nige". This Mongolian speech synthesis dataset consists of 13645 utterances, about 17 hours in total. Speech signal is sampled at 22.05 kHz and quantized with 16-bit depth. In this experiment, we use the 90% of the whole corpus for training, 5% for developing, 5% for evaluating.

4.2. Experimental setup

We compared the proposed end-to-end Mongolian speech synthesis system with HMM-based [17] and DNN-based [18] Mongolian TTS system. The detailed model configuration is described as follows.

- **HMM**: In HMM-base Mongolian TTS system, the authors use the statistic Mongolian grapheme to phoneme (G2P) conventional method to generate phoneme sequence. Then syllable and prosody are added to the phoneme sequence as the linguistic input of the Model. HMM Model predicts acoustic vocoder parameters from input features. We did not implement this TTS system. We generated a batch of synthesized outputs by using the author provided web-based TTS system.
- **DNN:** In DNN-base method, DNN is used to alternative HMM model in SPSS framework. The target of DNN is acoustic vocoder parameters, predicting from same linguistic feature as the HMM-based one described above. In this system, the DNN has 2 hidden layers, each layer has 512 units. We also did not implement this TTS system by ourselves. We obtained a batch of synthesized outputs from the original author of [18].
- **End2End:** We implement the proposed end-to-end model as described in section 3. Hyperparameters character embedding size is set to 256, reduction factor (r) is set to 5, and batch size is set to 8. We use Adam optimizer with learning rate decay to train the model.
- **Nature:** We involved a real human speaker in our comparison. We use the test set of dataset for evaluation.

4.3. Subjective Evaluation

To evaluate the perceived quality of synthesized speech, we conduct a subjective listening test. 40 utterances from each method are taken into consideration. Subjects who participated in the test are asked to score heard utterances in the item of perceived quality. The score follows the MOS ranges from 1 to 5, where 1 is lowest, and 5 is the highest perceived quality. As a reference, 1 to 5 means "bad", "poor", "fair", "good", and "excellent".

We conducted a web-based listening test experiment. Subjects can participate in the experiment via a browser on a PC or mobile phone. We think this experimental setting can reflect real applications' scenario. In the experiment, each subject is showed with 4 utterances from different method, where these 4 utterances are randomly selected to avoid memory effects. Before submitting a score, the subject is required to answer a validation question which can determine whether the subject really know Mongolian. We received about 300*4 scores, where each utterance was scored about 8 times in average. Due to the experimental setting, we cannot determine the exact number of the subjects. As an alternative measures, we found about 60 unique IP addresses from the subjects. Since the subject selection, there are only few subjects failed to pass the validation question test, who come from 7 unique IP addresses, and submited near 50*4 scores. We think their scores also valuable, and give analysis later.

The subjective evaluation is shown in the table 1, where MOS is calculated by taking the arithmetic average of all scores assigned the subjects who passed the validation question test. From table 1, we can clearly see that the proposed method outperforms both of the HMM-based and DNN-based methods with a big margin: 1.57 to the HMM-based and 1.96 to the DNN-based method. Because of the lack of annotated data, the HMM-based and DNN-based methods give a poor perceived quality, whose voice naturalness is very bad. Both methods give annoying robotic artifacts. Compared to these methods, the proposed method gives more natural synthesized speech. Its perceived quality is fair, and is approaching to good.

Table 1: *Subjective listening evaluation: MOS scores (Mean $\pm$ Variance)*

System	MOS
HMM	2.08 ± 0.251
DNN	1.70 ± 0.235
End2End	$\mathbf{3.65 \pm 0.116}$
Nature	4.92 ± 0.079

An interesting phenomenon is found when analyzing the scores from the subjects who have not passed the validation question test. Clearly, they may have no knowledge on the

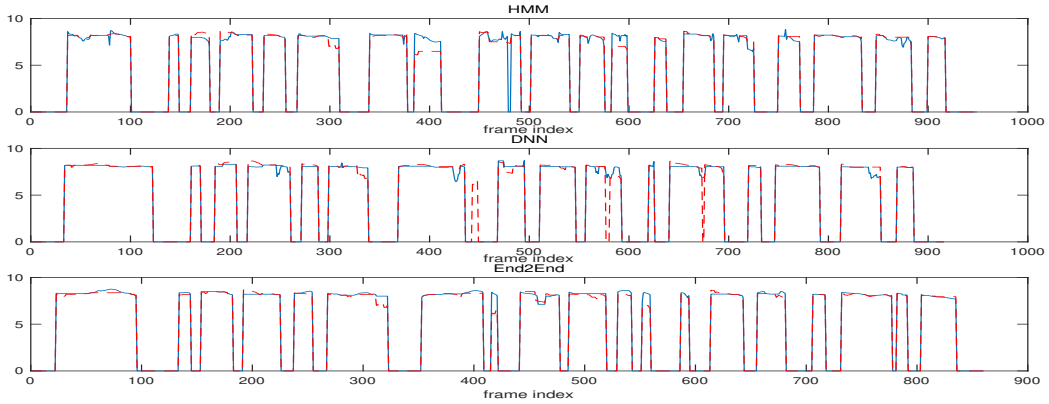

Figure 2: *Contour of $logF_0$ of natural speech and those predicted by HMM, DNN and End2End systems. The red dot line indicates $logF_0$ of natural speech. This sentence is "jarim ni ajil-aqa-ban bagvgsan-v seguler ober-un dvratai ajil-iyan hihu-du sedgil ondor haNhvlagvn amidvrajv bain_a", means "Some people can enjoy themselves after work".*

Mongolian. But we believe they have the knowledge of what is a better speech. Their scores are summarized in Table 2. In the world of subjects who have no idea on the Mongolian, the question is simple: there are just the good or bad utterances. They can clearly distinguish natural speech out of the speech from DNN-based and HMM-based models. They think the output of DNN-based and HMM-based methods only have small difference, both of them are bad. On the other side, they also think the output of the proposed method is difficult to distinguish from the human generated voice. It seems that the proposed method passes a low version of Turing test where the judges do not know the test language.

Table 2: *Subjective listening evaluation by subjects who have no idea with Mongolian: MOS scores (Mean $\pm$ Variance)*

System	MOS
HMM	1.62 ± 0.278
DNN	1.55 ± 0.306
End2End	$\mathbf{4.77 \pm 0.092}$
Nature	4.98 ± 0.038

4.4. Objective evaluation

The fundamental frequency, or F_0 is an essential acoustic feature that connected with the speech prosody. We compared the F_0 from the synthesized speech against that from human generated voice. An example is given in figure 2, in this figure a dynamic time warping (DTW) is used for alignment. Obviously, we can see the proposed method get a better F_0 contour which is closer to natural speech than other methods. The mean squared error (MSE) of $logF_0$ between the synthesized speech from three different methods to natural speech are 0.3104, 0.6454 and 0.0759 for HMM, DNN and End2End methods respectively. It indicates that the proposed method can generate more natural speech.

4.5. Deployment

We compile the model into both of the CPU and GPU instruction. We find that 30 times faster to use GPU than CPU in our test environment. In the GPU implementation, multiple NVIDIA Tesla K80 GPUs are used to synthesize speech waveform in parallel. It takes 1 second to synthesize a 13 seconds speech on average, which satisfying the real-time requirements.

5. Conclusion

In this paper, we proposed an end-to-end Mongolian speech synthesis system. Since the proposed system requires few annotations, we can afford the money and time to build a large-scale Mongolian corpus dataset for the model training. A Mongolian speech synthesis dataset consists of about 17 hours recording is built. As a result, the proposed end-to-end system becomes the new state-of-the-art in Mongolian speech synthesis task. We conduct the subjective and objective evaluation to compare the proposed system and the existing systems. The experimental results show that the proposed end-to-end model performs better than HMM-based and DNN-based models with a big margin.

We deploy the proposed system as a web-based service [3] and use multi GPU thread to generate speech. Our system achieves high availability and low latency, and becomes the first Mongolian TTS system can be used in real applications.

For mobile devices, like smartphones, the proposed model is too large to deploy. In the future, we will conduct research to explore a simpler network architecture, expected to achieve comparable performance.

6. Acknowledgements

We thank the reviewers for helpful comments and suggestions, and the volunteers who assisted us in annotating the large Mongolian corpus dataset in past year. This work is supported by the National Natural Science Foundation of China under Grant 61876214, 61866030, 61563040 and 61773224, the Natural Science Foundation of Inner Mongolia under Grant 2016ZD06, the Comprehensive Strength Enhancement Foundation of Inner Mougolia University.

[3] http://mtts.mglip.com/

7. References

[1] T. N. Lin and T. N. Lin, *Text-to-Speech Synthesis*. Cambridge University Press, 2009.

[2] M. Morise, F. Yokomori, and K. Ozawa, "World: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[3] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings*, 1996, pp. 373–376 vol. 1.

[4] A. W. Black, H. Zen, and K. Tokuda, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[5] J. Matousek, D. Tihelka, and J. Romportl, "Building of a speech corpus optimised for unit selection tts synthesis," in *International Conference on Language Resources and Evaluation, Lrec 2008, 26 May - 1 June 2008, Marrakech, Morocco*, 2008, pp. 1296–1299.

[6] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International Conference on Machine Learning*, 2014, pp. 1764–1772.

[7] Y. Miao, M. Gowayyed, and F. Metze, "Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding," pp. 167–174, 2015.

[8] T. H. Wen, D. Vandyke, N. Mrksic, M. Gasic, L. M. Rojas-Barahona, P. H. Su, S. Ultes, and S. Young, "A network-based end-to-end trainable task-oriented dialogue system," 2016.

[9] X. Ma and E. Hovy, "End-to-end sequence labeling via bi-directional lstm-cnns-crf," *arXiv preprint arXiv:1603.01354*, 2016.

[10] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, and J. Zhang, "End to end learning for self-driving cars," 2016.

[11] W. Wang, S. Xu, and B. Xu, "First step towards end-to-end parametric tts synthesis: Generating spectral parameters with neural attention," in *INTERSPEECH*, 2016, pp. 2243–2247.

[12] J. Sotelo, S. Mehri, K. S. Kumar, J. F. Santos, K. Kastner, A. Courville, and Y. Bengio, "Char2wav: End-to-end speech synthesis," 2017.

[13] Y. Taigman, L. Wolf, A. Polyak, and E. Nachmani, "Voiceloop: Voice fitting and synthesis via a phonological loop," 2017.

[14] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron: Towards end-to-end speech synthesis," *arXiv preprint arXiv:1703.10135*, 2017.

[15] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in hmm-based speech synthesis," in *European Conference on Speech Communication and Technology, Eurospeech 1999, Budapest, Hungary, September*, 1999, pp. 2099–2107.

[16] H. Ze, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 7962–7966.

[17] J. Zhao, G. Gao, and F. Bao, "Research on hmm-based mongolian speech synthesis," *Computer Science*, vol. 41, no. 1, pp. 80–104, 2014.

[18] R. Liu, F. Bao, G. Gao, and Y. Wang, "Mongolian text-to-speech system based on deep neural network," in *National Conference on Man-Machine Speech Communication*. Springer, 2017, pp. 99–108.

[19] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," *arXiv preprint arXiv:1712.05884*, 2017.

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

[21] H. Li and B. Sarina, "The study of comparison and conversion about traditional mongolian and cyrillic mongolian," *2011 4th International Conference on Intelligent Networks and Intelligent Systems*, pp. 199–202, 2011.

[22] F. Bao, G. Gao, X. Yan, and H. Wang, "Language model for cyrillic mongolian to traditional mongolian conversion," in *NLPCC*, 2013.

[23] M. Lu, F. Bao, and G. Gao, "Language model for mongolian polyphone proofreading," in *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*. Springer, 2017, pp. 461–471.

[24] D. W. Griffin and J. S. Lim, "Signal estimation from modified short-time fourier transform," in *ICASSP*, 1983.