Interspeech 2024
1-5 September 2024, Kos, Greece

G2PA: G2P with Aligned Audio for Mandarin Chinese

Xingxing Yang

Division of Emerging Interdisciplinary Areas,
Hong Kong University of Science and Technology,
Hong Kong, China
xyangbx@connect.ust.hk

Abstract

During the training of a Mandarin Chinese Text-To-Speech (TTS) system, it is necessary to preprocess the training speech data, which mainly consists of audio and text pairs. One crucial preprocessing step involves converting Chinese graphemes to phonemes (G2P) to obtain phoneme representations when using phonemes as input. However, relying solely on the text for G2P conversion may lead to inaccurate results due to the pronunciation ambiguity of polyphones - characters with multiple pronunciations. Although previous research has attempted to address this issue, most approaches solely rely on text-based methods, disregarding the valuable audio information that can only be captured from the raw audio. To overcome this limitation, we propose a G2P pipeline that leverages both audio and text inputs to resolve pronunciation ambiguity. The code and model weights for our approach are publicly available at the following GitHub repository: https://github.com/iooops/G2PA.

Index Terms: G2P, Grapheme-to-phoneme conversion, Chinese polyphone disambiguation, text-to-speech

1. Introduction

Many Mandarin Text-To-Speech (TTS) datasets include text and audio pairs, but they often lack complete pinyin/phoneme annotations. As a result, current TTS models commonly rely on text-based Grapheme-to-Phoneme (G2P) modules during data preprocessing when training TTS systems. However, these models face a challenge when determining the accurate alignment of pronunciation with the corresponding audio, especially for words that have multiple valid pronunciations within the same context. This issue is particularly prominent for specific words, even if they constitute a small portion of the dataset. Consequently, when such processed datasets are used for TTS training, the models often produce incorrect predicted phonemes for these specific words.

The purpose of Mandarin Grapheme-to-Phoneme (G2P) is to accurately convert Chinese characters into their corresponding Pinyin/phoneme pronunciations. This process is crucial in Chinese Text-To-Speech (TTS) systems because the same character can have different pinyin pronunciations depending on its context. For instance, in the given example, the first occurrence of the character "地" is pronounced as "de," while the second occurrence is pronounced as "dì."

Input: 他悄悄地来到地头。
Translation: He crept to the edge of the field.
Output: tā qiāo qiāo **de** lái dào **dì** tóu.

More specifically, there are also some occasions when a single character can have different pronunciations even in the same context depending on the speaker's style. In the example below, "这" can be both validly pronounced as "zhè" and "zhèi" within the context.

Input: 这是一只猫。
Translation: This is a cat.
Output: **zhè/zhèi** shì yì zhī māo.

Motivated by this observation, we propose a G2P pipeline that utilizes not only text but also aligned audio to resolve the issue of pronunciation ambiguity in Chinese polyphones. This method is useful, especially for TTS frontend.

Our main contributions can be summarized as follows:

1. We trained a classification model for pinyin classification with processed audio and pinyin pairs and proposed an inference pipeline for G2P.

2. We validated that G2P with aligned audio can better identify the actual pinyin phonemes in the datasets.

3. We released all the code and model weights trained from Biaobei dataset via our GitHub repository.

2. Related Work

Previous works in Chinese G2P are usually based on textual content only[1, 2, 3, 4]. State-of-the-art G2P utilizes deep-learning-based methods. For example, G2PM utilizes a Bi-directional LSTM for choosing the most possible Pinyin from polyphony candidates[1]. G2PW adapts learnable softmax-weights to condition the outputs of BERT with the polyphonic character of interest and its POS tagging[2]. Nonetheless, due to the absence of corresponding audio data, the utilization of G2P modules in TTS frontend data preprocessing results in the model encountering imprecise alignment issues with certain phonemes, despite their usefulness in TTS inference.

We regard a part of G2PA as an audio classification task, which can be followed by cropping a piece of audio into segments. We found that many research works utilize convolutional neural networks (CNNs) to classify audio pieces by extracting audio information first[5, 6]. Extensive evidence supports the effectiveness of utilizing CNNs trained through offline Knowledge Distillation (KD) from sophisticated and high-performing transformers for audio classification purposes[6]. Given the nature of the task, where audio is transformed into textual information, pre-trained speech understanding models, specifically those designed for Automatic Speech Recognition (ASR), can be applied for audio feature extraction. Notable examples of such models include Wav2vec [7] and Hubert [8], both of which are based on transformer architecture.

10.21437/Interspeech.2024-19

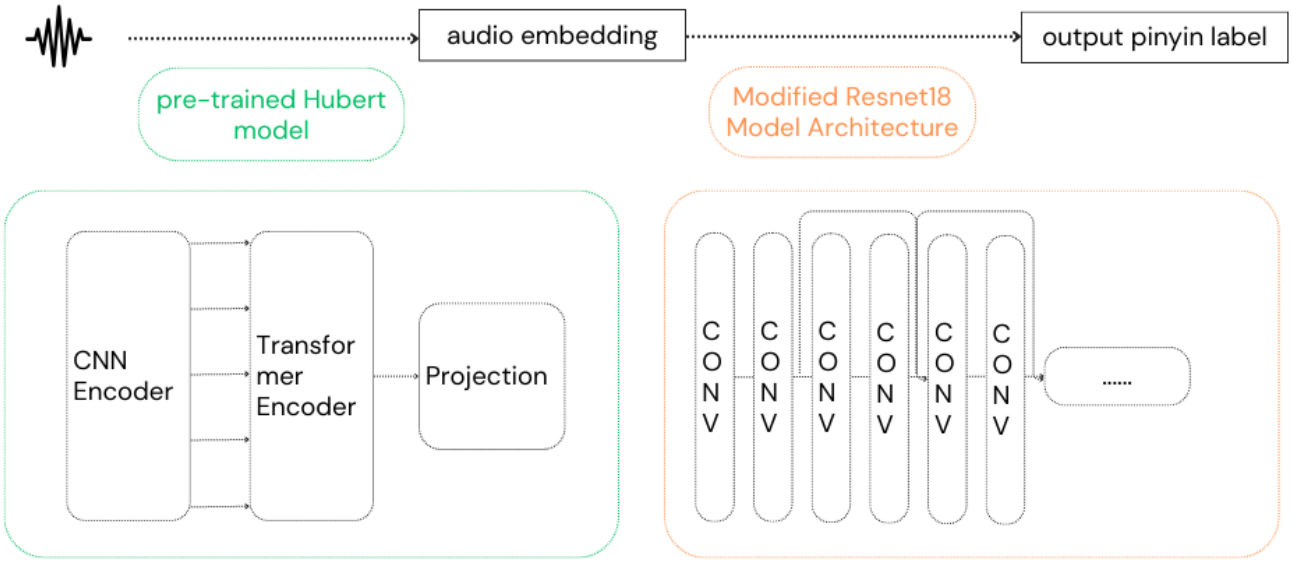

Figure 1: *Audio classifier for Pinyin*

3. Proposed Method

The problem is defined as follows: Given a piece of audio and its corresponding checked text, how to convert the text into its correct phonemes that follow the audio piece?

3.1. Inference Pipeline

The proposed inference pipeline is illustrated in Figure 2. Given a dataset comprising text and audio pairs, the key steps involved in the pipeline are as follows:

1. **Text-to-Pinyin Conversion**: Convert the textual information into Pinyin representation.
2. **Pinyin-Audio Alignment**: Align the Pinyin text with the corresponding audio clips.
3. **Classification of Aligned Audio**: Classify the cropped, Pinyin-audio aligned segments.
4. **Disambiguation and Comparison**: Compare the predicted results with polyphonic variations to resolve any ambiguities.

3.1.1. Text to Pinyin

The first stage of the procedure entails transforming the provided text into Pinyin representations, which serve as phonetic representations in Chinese indicating pronunciation. Each line of text is converted into a collection of the most likely Pinyin representations, with each word corresponding to a single Pinyin representation. Moreover, we extract all the accessible polyphonic variations for each word in the line.

3.1.2. Text-audio Aligner

Subsequently, the process proceeds to align the Pinyin/text with the corresponding audio. To accomplish this, an external aligner Montreal Forced Aligner with a a standard GMM/HMM architecture is employed, which enables us to obtain precise timing information for each word/Pinyin[9, 10]. Using this timing information, we segment the original audio into smaller clips. Consequently, we obtain a collection of audio-Pinyin pairs, where each pair comprises a small audio clip and its corresponding Pinyin label.

3.1.3. Audio Classifier

With the cropped audio segments, we subsequently employ a trained audio classifier to categorize the audio clips based on a predefined set of Pinyin labels.

3.1.4. Polyphone Disambiguation

Next, we compare the outcomes obtained from our trained audio classifier with the corresponding Pinyin labels generated by the Pinyin converter. In the event of a mismatch, we verify whether the predicted label exists within the polyphonic variations associated with that particular word. If a match is found, we substitute the initially chosen Pinyin label with the predicted one.

3.2. Audio Classifier for Pinyin

We consider Chinese polyphone disambiguation as a classification problem. Basically, with an audio clip of a specific word/pinyin and its corresponding polyphones, the goal is to get its actual pronunciation in pinyin.

During the training process, the primary focus is on training a classifier model. The basic model architecture can be observed in Figure 1.

3.2.1. Data Preprocessing

To facilitate the training of the audio classifier, the initial step involves obtaining aligned pairs of audio and corresponding text. The objective is to ensure that each audio clip is associated with a corresponding Pinyin label. It is achieved by training an audio aligner to obtain aligned pairs of Pinyin and text.

3.2.2. Getting Audio Embeddings

To obtain audio embeddings, we employ the pre-trained Hubert model as the underlying backbone. Hubert, short for Hidden-Unit BERT, incorporates an offline clustering stage to generate aligned target labels for a prediction loss with a CNN and transformer architecture[8, 11, 12]. The rationale behind utilizing a pre-trained Hubert model is that it has already acquired internal

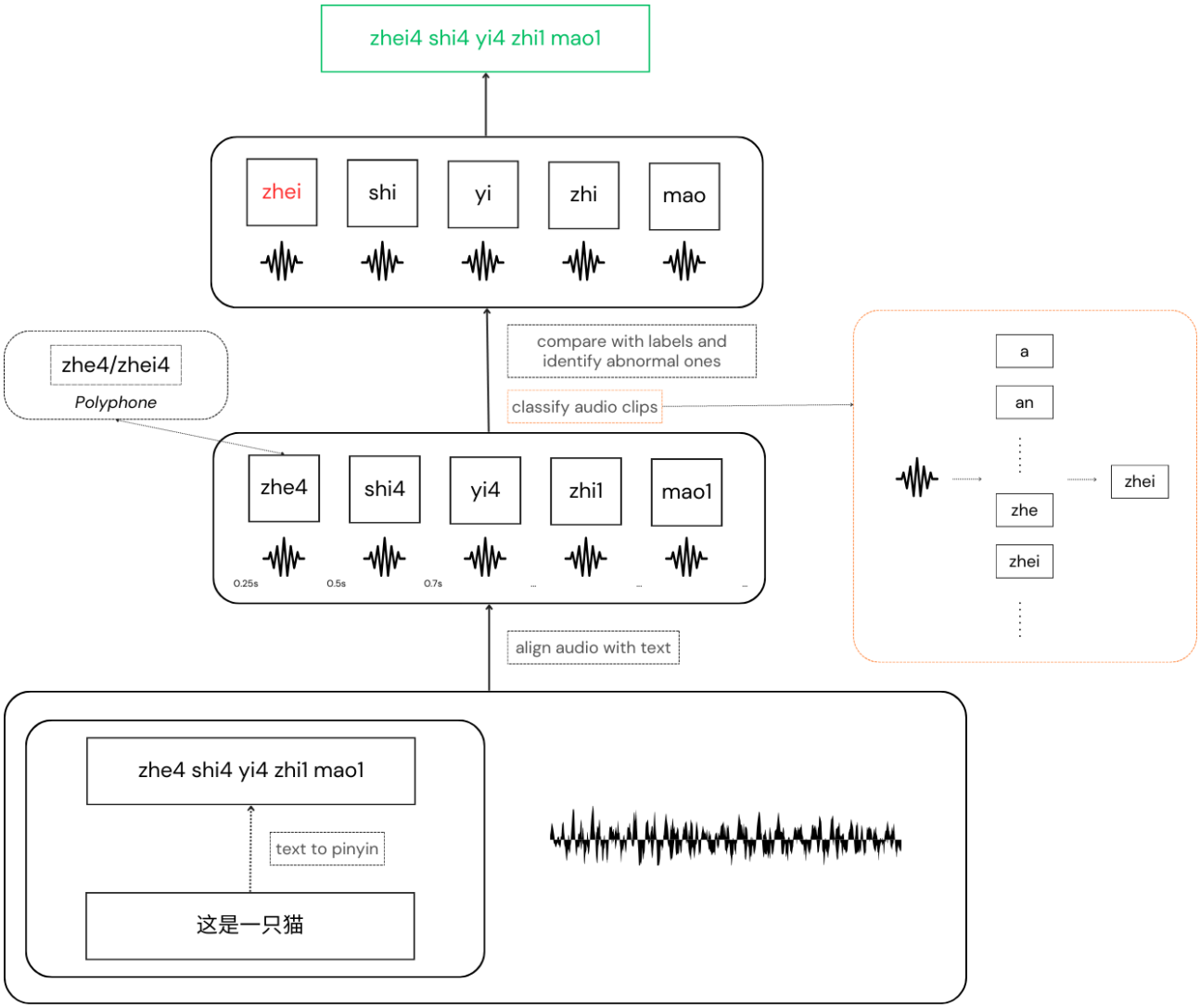

Figure 2: *Inference pipeline of our proposed method*

audio representations, thereby enhancing the classification task.

3.2.3. Classifier

In our classification setup, we compile a comprehensive list of all possible Pinyin labels as our classes. To simplify the task and account for the complex nature of spoken Chinese with its nuanced tones, we have intentionally excluded tone information from our Pinyin classes.

Our primary architecture is based on the ResNet18 model[13], which comprises 18 residual convolutional layers. However, we make modifications to the initial input layer and the final fully connected output layer to accommodate the dimensions of audio embeddings and the number of Pinyin labels. To train the model, we utilize the cross-entropy loss function, which enables us to obtain predicted output labels as the final outcome.

4. Experiments

In our experiments, we evaluate our G2PA with Biaobei dataset[1]. We remove one line with English characters. We use the first 8000 lines for audio classifier training and evaluation. The remaining 2000 lines are used for G2P evaluation.

4.1. Data Preprocessing

We used Montreal Forced Aligner[2] to train the alignment model and we used the trained alignment model to get aligned audio-pinyin pairs. For each piece of cropped audio, we got its embeddings through the Chinese Hubert model[3]. We dropped abnormal audio pieces that have a duration shorter than 0.03 seconds or longer than 0.6 seconds and did data augmentation for Pinyin snippets with fewer data. Finally, we got 84167 pairs of audio embeddings and pinyin. We split the dataset to train and test set

[1] https://www.data-baker.com/open_source.html

[2] https://montreal-forced-aligner.readthedocs.io/en/latest/

[3] https://huggingface.co/TencentGameMate/chinese-hubert-base

with the proportion of 9:1.

4.2. Training

Our primary focus during the training process was on training the classifier. For the architecture of our model, we primarily utilized Resnet18[4], which is a pre-existing model provided by the Torchvision library. However, we made modifications to the input layer to accommodate our specific input dimension, which was (1, 49, 768). Additionally, we adapted the final fully connected layer to have the same number of output features as the number of Pinyin labels, which amounted to 472 in our case.

To optimize the model, we employed the Adam optimizer with a learning rate of 1e-3. The loss function utilized was cross-entropy. The training procedure was carried out using 9 NVIDIA RTX 2080 GPUs, and the training duration lasted for a few minutes.

5. Results

5.1. Classifier

To assess the performance of our trained classifier, we employed accuracy as the evaluation metric. Moreover, we incorporated top-k sampling during the prediction phase and conducted a comparative analysis of the obtained results.

Top-k	Test Accuracy
1	96.727%
2	99.310%
3	99.663%
5	**99.854%**

Table 1: *Test set accuracy of the classifier*

The presented table demonstrates that the utilization of top-1 sampling leads to reasonably satisfactory accuracy. Furthermore, as the value of k increases, the accuracy further improves. These outcomes serve as validation for the effectiveness of our proposed method.

5.2. G2P

We conducted a comparative analysis of our model with 2 existing open-source Chinese Grapheme-to-Phoneme (G2P) libraries, namely Pypinyin[5] and G2PM[6]. Pypinyin utilizes rule-based approaches, while G2PM employs neural network-based methodologies. The libraries are readily accessible through the PyPi package repository. We ignored tone information and erhua(儿化音) when doing the comparison.

As shown in Table 2, our model slightly outperforms Pypinyin. By leveraging a combination of textual and auditory information, our methodology demonstrates the ability to attain enhanced accuracy in G2P tasks.

System	Top-k	Test Accuracy
Pypinyin		99.5427%
G2PM		99.2743%
Ours(w/Pypinyin)	1	99.8263%
Ours(w/Pypinyin)	2	99.8472%
Ours(w/Pypinyin)	3	99.8502%
Ours(w/Pypinyin)	5	**99.8532**%

Table 2: *Test set accuracy of Chinese G2P systems*

6. Limitations

In our study, the audio-pinyin alignment process was performed using the Montreal Forced Aligner. To ensure data quality, we excluded abnormal audio segments that had durations shorter than 0.03 seconds or longer than 0.6 seconds. And we did not conduct a specific assessment of the alignment accuracy. During the inference stage, it is also important to acknowledge that the processed Pinyin data has not undergone validation. It is possible that certain audio-pinyin pairs were not accurately recognized, such as cases where the cropped audio segments did not encompass the complete auditory information for a single word.

It should also be noted that certain words in Pinyin may encompass multiple tones, but we did not incorporate tone information in our study. It is partly because there are too many nuances for neural tone pronunciation in oral Mandarin.

Another observation is that our method requires much more processing time when compared to a plain-text-based approach. However, considering that it is primarily designed for TTS frontend purposes, such increased processing time can be deemed acceptable. This can be attributed to the inclusion of additional steps, such as audio alignment and audio embedding extraction.

7. Future Work

In the preceding section, it was mentioned that certain constraints exist. Employing the alignment of Chinese characters with audio pieces tends to be more suitable for this task as it eradicates potential alignment problems stemming from incorrect Pinyins. Nevertheless, the direct alignment of Chinese characters with audio remains an underexplored domain, necessitating further investigation. Additionally, as mentioned above, the act of cropping audio segments may lead to incomplete segmentation of a single word. Consequently, it is intriguing to explore whether an improved approach exists, enabling a more comprehensive end-to-end alignment of grapheme-to-phoneme (G2P) with audio, wherein text is directly converted into phonemes using audio information.

In addition, despite the abundance of datasets containing both audio and text, there is a scarcity of datasets that provide labeled correct Pinyins, text, and audio. This limited availability of datasets presents a new challenge for G2P in the speech training process, as the size of the datasets becomes a limiting factor. To overcome this challenge, it is helpful to construct a larger dataset with more comprehensive contexts, as it will contribute to the development of a more effective solution for this task.

8. Conclusion

We proposed a new method for Chinese polyphony disambiguation, which is especially useful in data preprocessing for TTS frontend during the training period. We trained G2PA models using Biaobei dataset and released the code and weights. We hope our method will be helpful for researchers and practitioners.

[4] https://pytorch.org/vision/main/models/generated/torchvision.models.resnet18.html

[5] https://pypi.org/project/pypinyin/

[6] https://github.com/kakaobrain/g2pm

9. References

[1] K. Park and S. Lee, "g2pm: A neural grapheme-to-phoneme conversion package for mandarin chinese based on a new open benchmark dataset," *arXiv preprint arXiv:2004.03136*, 2020.

[2] Y.-C. Chen, Y.-C. Chang, Y.-C. Chang, and Y.-R. Yeh, "g2pw: A conditional weighted softmax bert for polyphone disambiguation in mandarin," *arXiv preprint arXiv:2203.10430*, 2022.

[3] J. Kim, C. Han, G. Nam, and G. Chae, "Good neighbors are all you need for chinese grapheme-to-phoneme conversion," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[4] S. Zhang, K. Zheng, X. Zhu, and B. Li, "A polyphone bert for polyphone disambiguation in mandarin chinese," *arXiv preprint arXiv:2207.12089*, 2022.

[5] S. Verbitskiy, V. Berikov, and V. Vyshegorodtsev, "Eranns: Efficient residual audio neural networks for audio pattern recognition," *Pattern Recognition Letters*, vol. 161, pp. 38–44, 2022.

[6] F. Schmid, K. Koutini, and G. Widmer, "Efficient large-scale audio tagging via transformer-to-cnn knowledge distillation," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[7] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.

[8] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[9] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal forced aligner: Trainable text-speech alignment using kaldi." in *Interspeech*, vol. 2017, 2017, pp. 498–502.

[10] L. Rabiner and B. Juang, "An introduction to hidden markov models," *ieee assp magazine*, vol. 3, no. 1, pp. 4–16, 1986.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.