A Speech Recognition Model for Mandarin Chinese Pronunciation Evaluation

1st Jakub Kiliańczyk
Gdańsk University of Technology
Gdańsk, Poland

2nd Jakub Kwiatkowski
Gdańsk University of Technology
Gdańsk, Poland

3rd Anna Strzelecka
Gdańsk University of Technology
Gdańsk, Poland

4th Dawid Migowski
Gdańsk University of Technology
Gdańsk, Poland

5th Łukasz Smoliński
Gdańsk University of Technology
Gdańsk, Poland

***Abstract*—In recent years many Automatic Speech Recognition (ASR) solutions have been designed based on artificial intelligence algorithms. The Chinese language poses relatively new challenges when it comes to ASR or Mispronunciation Detection and Diagnosis (MDD) as the meaning of each word depends not only on pronunciation, but also on tone of speech. In order to correctly asses one's speech, both features have to be considered.**

***Index Terms*—Automatic Speech Recognition, Mispronunciation Detection and Diagnosis, Convolutional Neural Network, Transformer**

I. Introduction

In order to train new Speech Recognition models, we used AISHELL-3 [1] dataset and a self-made one consisting of recordings of students of Gdansk University of Technology. We developed and compared several approaches for the tasks of Speech and Scoring and Recognition and Tone Recognition.

II. Related Work

We conducted a literature review in search of existing research and solutions for the tasks of pronunciation and tone detection for Mandarin chinese. We specified search criteria to look for Neural Network models trained with limited data, published in years 2014-2024 in english. We based our research on the following keywords: "Speech Recognition", "Deep Learning", "Pronunciation Evaluation" and fetched a total of 878 research papers from several databases. After eliminating duplicates and assessing the titles and abstracts we chose 12 most suiting articles for further reading.

After the literature review, our first notion was the limited quantity of publicly available datasets, especially regarding the subject of Tone Classification (TC). [2] mentions difficulties in finding and accessing datasets for ASR in Chinese and suggests a new one for Keyword Detection. We picked two Deep Learning architectures to build on and train: [3] (a Convolutional Neural Network - CNN), and [4] (a Transformer).

III. Approach

We decided to first develop separate CNN models for the tasks of Pronunciation Evaluation (PE) and Tone Classification. Afterwards we attempted to solve both of these tasks using a Speech-Recognition architecture of a Transformer [4], [5]. It's possible to solve Pronunciation Scoring and Tone Classification tasks based on Speech-to-Text output in Mandarin chinese, as written chinese contains information about pronunciation and intonation. We trained all models with our dataset (in case of the Transformer we also used [1]) and compared the results to select the most accurate approach.

A. Data Processing

In order to ensure good training quality of our dataset we applied some preprocessing techniques. Firstly, we removed the incorrect recordings, for example only constituting of noise, or ones that didn't suffice the length requirements. We decided not to use recordings shorter than 0.5s as they were mostly empty or cut in the middle of a word. We have also excluded the top 5% of the longest recordings. We made sure not to lose data coverage of each class in the process. We've filtered the recordings with high noise levels using the Root Mean Square (RMS) as the energy measure. We assumed a threshold of 0.004 and only left recordings with RMS above that value. We verified that the ones below were in fact too noisy to use in training. A Mel Spectrogram of such recording is presented in "Fig. 1". For comparison, "Fig. 2" presents a spectrogram with acceptable noise levels.

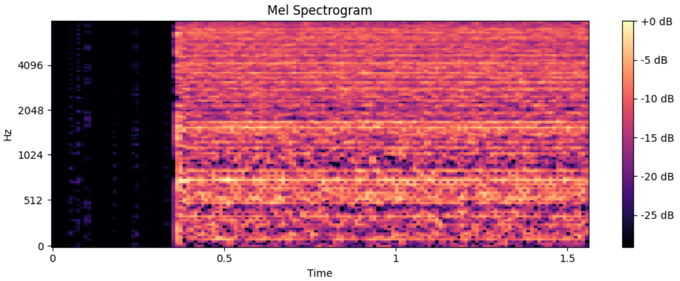

Fig. 1. Spectrogram with high noise levels

Later, we trimmed the silent parts at the edges of the recordings and applied noise-reduction filters. The effect of trimming silent parts is shown in "Fig. 3" and "Fig. 4".

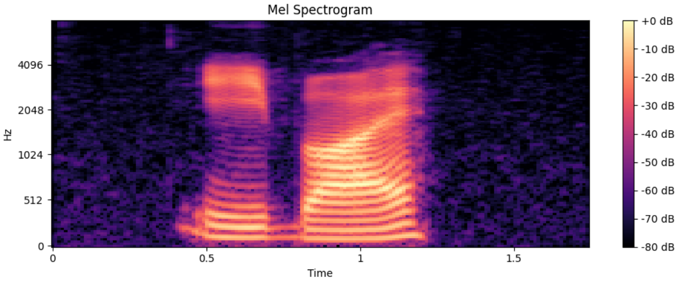

Fig. 2. Spectrogram with acceptable noise levels

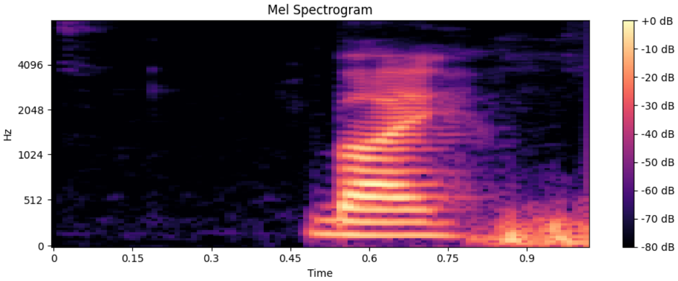

Fig. 3. Spectrogram before trimming silent parts

In order to reduce the noise of recordings, we applied techniques like noise profiling (based on noise from the beginning of recording), band-pass filters and spectral noise reduction. Results of noise reduction are presented in figures “Fig. 5” and “Fig. 6”.

We normalized the volume to achieve voice intensity consistency. After the normalization the amplitude of audio signal does not exceed the range from -1 to 1. For normalization we used the *librosa.util.normalize* function. We standardized the length of recordings by cutting and filling and balanced the representation of each class by equalizing the number of positive and negative samples. We split the dataset into training, validation and test sets (in proportion of 0.7, 0.15, 0.15). We conducted data augmentation on the training set, which included altering speed and voice level of recordings using functions from *librosa.effects* library. After obtaining the spectrograms required as an input for our models, we observed a significant improvement in the training.

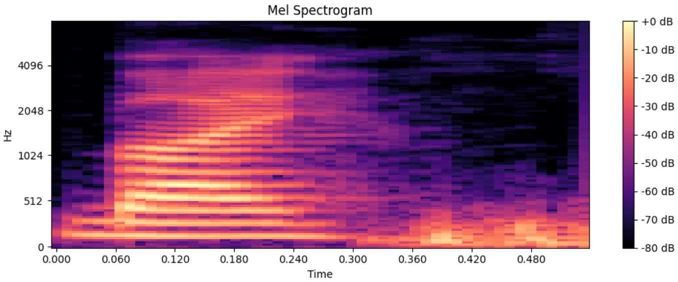

Fig. 4. Spectrogram after trimming silent parts

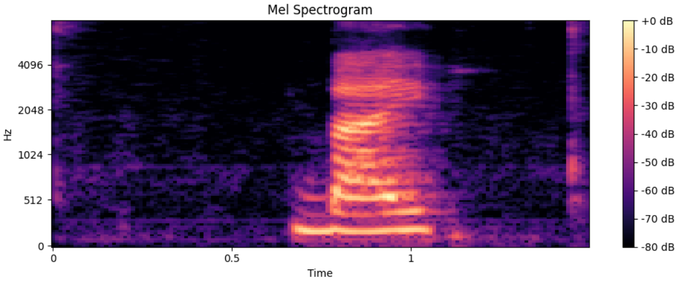

Fig. 5. Spectrogram before noise reduction

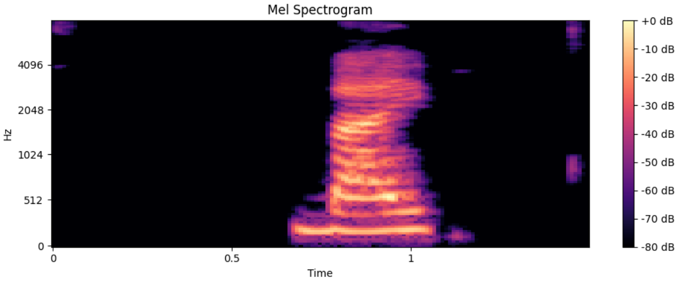

Fig. 6. Spectrogram after noise reduction

B. CNN for Tone Recognition

We used a multilayer CNN based on [3]. It classifies the tone of speech phonemes into one of 4 defined tones (1 to 4, not counting the neutral one, denoted by 0). The network consists of convolution layers, batch normalization, activation functions and pooling layers. The input layer takes two-dimensional data and passes the output to 4 convolution blocks with the number of filters ascending through values: 64, 128, 256, 512 and altering steps, which allow hierarchical feature extraction. Finally we apply Data Flattening combined with 2 Fully Connected Layers (FCLs) of 1024 units followed by a softmax layer to obtain probabilities. For this model we used the Stochastic gradient descent optimization algorithm [6] with a low learning rate and special weights applied for classes with inconsistent representation in training data. We used techniques of early stopping and saving checkpoints to avoid overfitting and ensure the best results are not lost.

C. CNN for Pronunciation Evaluation

For this task we used the numeric words from 0 to 10 and 100 from our Mandarin chinese audio dataset with labels provided by a native Chinese speaking expert in the field. For each word we trained a separate model, as the results for a single model trained on all the words haven’t been satisfactory. The network takes a 3-channel spectrogram as an input. Similarly to our Tone Recognition model, the architecture is based on convolution and pooling layers. The number of convolution filters grows to 128. Thanks to the MaxPooling layers, the model’s number of trainable parameters is reduced, alleviating the computational cost and risk of overfitting. After extracting the features with convolutional blocks, we use a

Flatten layer. Afterwards the tensors are passed to FCLs with 1024 and 128 units in this order and ReLU activation function. In order to prevent overfitting we use a Dropout layer that randomly disables half of the neurons, improving the model's generalization capability. In the end there is a 1-unit FCL with a sigmoid activation function, responsible for outputting a probability of given word, allowing binary classification. During the training we use the Adam optimizer [7] with Binary Cross-Entropy loss function. The hyperparameters we used for this model are summarized in “Tab. I”.

TABLE I
PRONUNCIATION EVALUATION - CNN HYPERPARAMETERS

Parameter	Value
batch size	32
learning rate	0.0001
dropout	0.2
smoothing	0.1

D. Speech-Transformer for complete ASR

With this last model, using an implementation of [4], [5], modified to enable training with [1] and finetuning with our dataset, we attempted solving both tasks of Pronunciation Evaluation and Tone Classification. Transformers have recently gained high popularity in Sequence-to-Sequence tasks such as Natural Language Processing. The architecture we used is based on Multi-Head Attention, which enables applying the attention mechanism in parallel, with separate learnable parameters. The Batch Normalization technique typical for CNNs, here has been replaced by Layer Normalization applied in each Encoder and Decoder layer. The architecture is presented in more detail in “Fig. 7”.

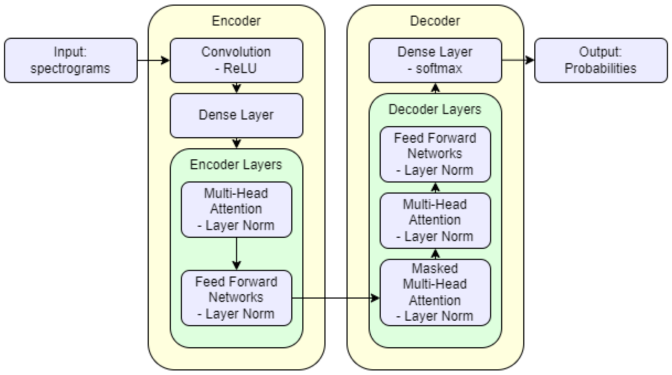
Encoder
Input: spectrograms
Convolution - ReLU
Dense Layer
Encoder Layers
Multi-Head Attention - Layer Norm
Feed Forward Networks - Layer Norm
Decoder
Dense Layer - softmax
Decoder Layers
Feed Forward Networks - Layer Norm
Multi-Head Attention - Layer Norm
Masked Multi-Head Attention - Layer Norm
Output: Probabilities

Fig. 7. Speech-Transformer Architecture

The model would originally generate transcription sequences based on processed audio input. In case of Mandarin chinese, the transcription can be represented in traditional alphabet or in pinyin notation, which consists of syllables written with latin letters and denotes the tone of each phone. The pinyin is popular for learning purposes for foreigners who don't know the Chinese alphabet. It also enables us to simply split the phoneme into pronunciation and tone information. The model has been configured with default hyperparameter values as in [4]. Due to lack of specific transcriptions in our dataset (we were limited to pronunciation scores and tone labels) the training split for the finetuning part had to be limited to correct recordings. For the testing, however, we used balanced amount of correct and incorrect recordings for pronunciation evaluation and all the recording for tone classification.

IV. RESULTS

A. CNN for Tone Classification

The accuracy initially achieved with our model was around 0.50. After data preprocessing we were able to raise that score to 0.57. The results of this CNN are also presented in a form of a confusion matrix in

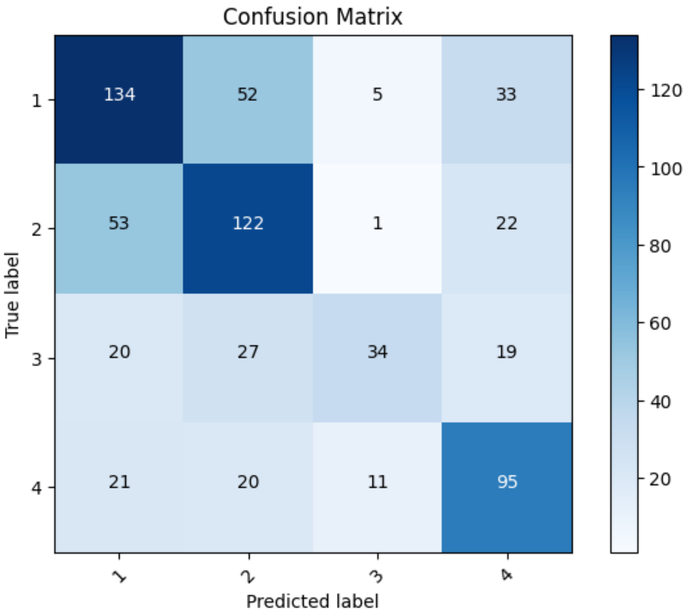

Fig. 8. CNN

B. CNN for Pronunciation Evaluation

Loss function and accuracy of the model we attempted to train for all the words are visualized in “Fig. 9” and “Fig. 10”. Similarly, we include loss function and accuracy plots for one of models trained for each word: “Fig. 11” and “Fig. 12”. The results of a model trained for each word are presented in “Tab. II”.

C. Speech-Transformer

Loss function for this model is plotted against time (training epochs) in “Fig. 13”. “Fig. 14” presents loss function during finetuning. We measured this model with an accuracy metric. The results are given in the table “Tab. III”. We have also plotted a confusion matrix “Fig. 15”.

Before the finetuning, this model would achieve at most 50% accuracy, meaning half of the times it would guess

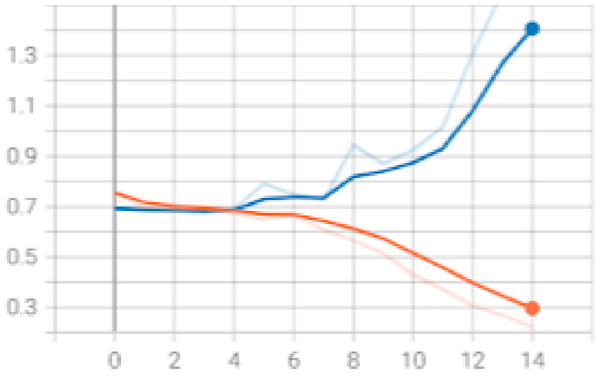

Fig. 9. CNN PE Loss function in time | blue - validation, orange - training

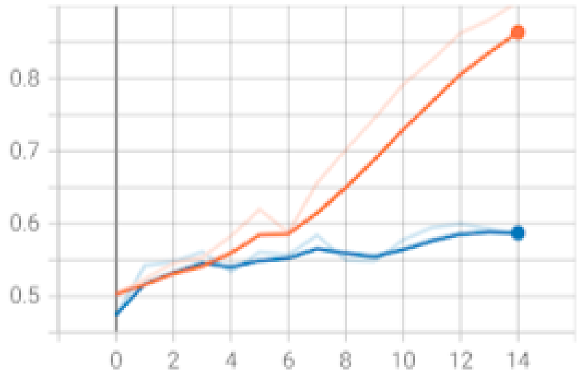

Fig. 10. CNN PE Accuracy in time | blue - validation, orange - training

one of the 4 available tones (excluding the neutral tone). Afterwards it actually dropped below that value, because it was not trained separately for this task and two tones out of four are underrepresented within the numeric words from 1 to 10, which can be observed on a confusion matrix “Fig. 16”.

V. Discussion

In the Pronunciation Scoring task, the Speech-Transformer has achieved a noticeably higher accuracy. Apart from signifi-

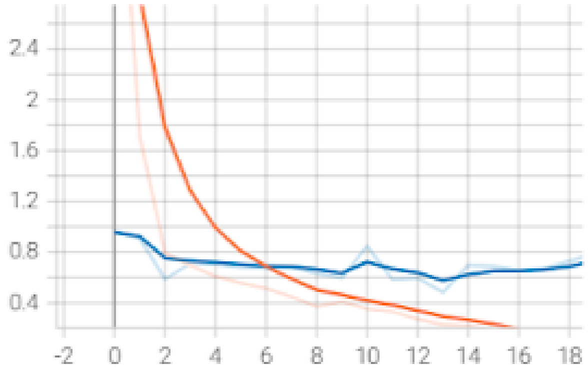

Fig. 11. CNN PE for number 3 Loss function in time | blue - validation, orange - training

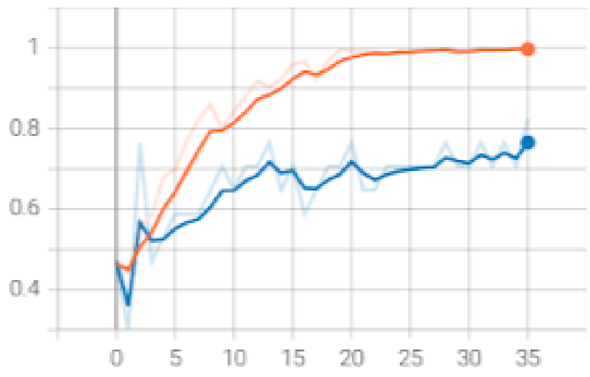

Fig. 12. CNN PE for number 3 Accuracy in time | blue - validation, orange - training

TABLE II
CNN - Pronunciation Evaluation Metrics

Word	Accuracy
a0	0.78
a1	0.77
a2	0.67
a3	0.61
a4	0.73
a5	0.72
a6	0.69
a7	0.68
a8	0.70
a9	0.70
a10	0.71
a100	0.73
Average	**0.7075**

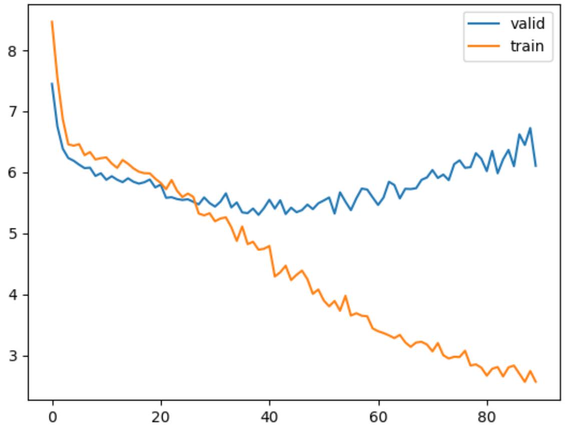

Fig. 13. Speech-Transformer - loss function in time

TABLE III
Speech-Transformer - Pronunciation Evaluation Metrics

Metric	Value
Tone Classification Accuracy	0.48
Articulation Scoring Accuracy	0.85

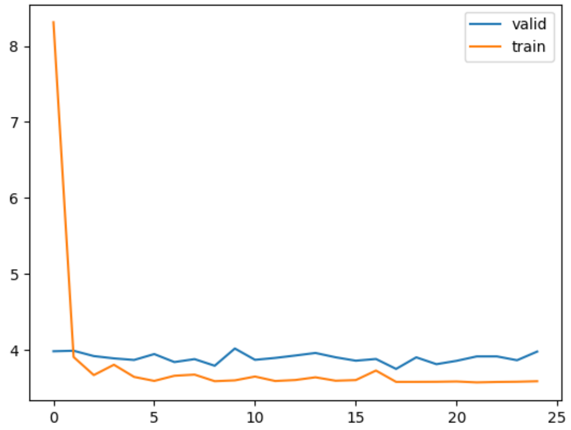

Fig. 14. Speech-Transformer finetuning - loss function in time

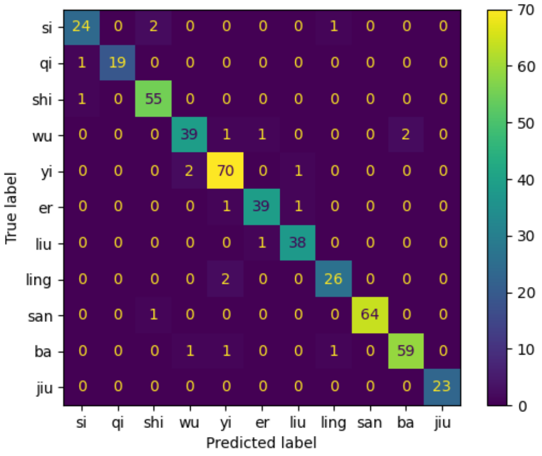

Fig. 15. Speech-Transformer - Confusion Matrix for PE

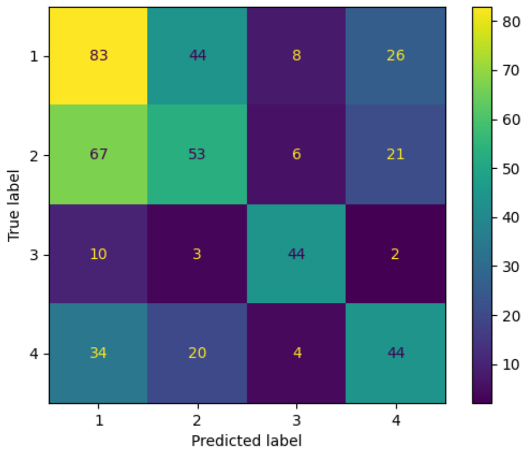

Fig. 16. Speech-Transformer - Confusion Matrix for TC

cant difference of these networks, the Speech-Transformer has been pre-trained with [1]. At this stage of research, accuracy of over 0.70 for a ten class problem is not a bad score. Ideally, the same result would be achieved with a single CNN for all the words. Another improvement would be to support multi-syllable words, which is currently possible only with the use of the Speech-Transformer as it's a sequence-processing-oriented approach. As per Tone Classification, the CNN has outperformed the Speech-Transformer. It's worth to note that in both cases the first two tones were likely to get confused. This can be observed in the upper-left corner of both "Fig. 8" and "Fig. 16". Unfortunately, even the accuracy of the CNN, which scored 0.57, can't be considered a success given there are only 4 tones to classify. To achieve a higher accuracy with our proposed models, it might be necessary to acquire more training data, enhance the data processing techniques, balance the tone samples in the dataset and remove the ones with any uncertainty (based on the confusion of both our models around the first two tones).

VI. Conclusion

The Pronunciation Scoring accuracy of our models shows that it's in fact possible to achieve results comparable to human experts in the field. Although the Tone Classification task hasn't been accomplished at the same level, it's certainly within the reach of our successors given some enhancements listed in the previous paragraph are there to be made. While the proposed solution shows promising initial results, it requires further refinement before it can be considered production-ready. Therefore, the main contribution of this work lies in providing a foundational implementation that can serve as a starting point for future research and improvements. To facilitate this, we have made our codebase publicly available on GitHub: https://github.com/lukmanis25/Speech-Recognition-Model-for-Mandarin-Chinese.

References

[1] Y. Shi, H. Bu, X. Xu, S. Zhang, and M. Li, "Aishell-3: A multi-speaker mandarin tts corpus and the baselines," 2021. [Online]. Available: https://arxiv.org/abs/2010.11567

[2] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," 2018. [Online]. Available: https://arxiv.org/abs/1804.03209

[3] Q. Gao, S. Sun, and Y. Yang, "Tonenet: A cnn model of tone classification of mandarin chinese," in *Interspeech*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:202727233

[4] L. Dong, S. Xu, and B. Xu, "Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5884–5888.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023. [Online]. Available: https://arxiv.org/abs/1706.03762

[6] W. Sung, I. Choi, J. Park, S. Choi, and S. Shin, "S-sgd: Symmetrical stochastic gradient descent with weight noise injection for reaching flat minima," 2020. [Online]. Available: https://arxiv.org/abs/2009.02479

[7] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:6628106