*INTERSPEECH 2021*
30 August – 3 September, 2021, Brno, Czechia

# Improving Polyphone Disambiguation for Mandarin Chinese by Combining Mix-pooling Strategy and Window-based Attention

*Junjie Li*[1,*], *Zhiyu Zhang*[2,*], *Minchuan Chen*[1], *Jun Ma*[1], *Shaojun Wang*[1], *Jing Xiao*[1]

[1]Ping An Technology, China
[2]National Tsing Hua University, Taiwan

junjielee815@gmail.com, zzy1024458849@gmail.com

## Abstract

In this paper, we propose a novel system based on word-level features and window-based attention for polyphone disambiguation, which is a fundamental task for Grapheme-to-phoneme (G2P) conversion of Mandarin Chinese. The framework aims to combine a pre-trained language model with explicit word-level information in order to get meaningful context extraction. Particularly, we employ a pre-trained bidirectional encoder from Transformers (BERT) model to extract character-level features, and an external Chinese word segmentation (CWS) tool is used to obtain the word units. We adopt a mixed pooling mechanism to convert character-level features into word-level features based on the segmentation results. A window-based attention module is utilized to incorporate contextual word-level features for the polyphonic characters. Experimental results show that our method achieves an accuracy of 99.06% on an open benchmark dataset for Mandarin Chinese polyphone disambiguation, which outperforms the baseline systems.

**Index Terms**: polyphone disambiguation, pre-trained BERT, attention mechanism, text-to-speech

## 1. Introduction

Converting graphemes to a sequence of phonemes [1] (grapheme-to-phoneme conversion, G2P) is essential for text-to-speech (TTS) systems. Mandarin Chinese G2P systems convert Chinese characters to their corresponding pronunciations, called pinyin. There are at least 13000 commonly used Chinese characters, about 10% of which are polyphonic ones [2], meaning that the same character has different pronunciations in different contexts. For the Mandarin Chinese TTS system, it is essential to develop an effective polyphone disambiguation system to predict the correct pinyin transcription from several candidate pronunciations of the polyphonic character according to the context.

Traditional approaches for polyphone disambiguation are rule-based algorithms [3, 4] and statistical machine learning methods [5, 6]. Recently, inspired by tremendous successes in English G2P [1, 7] and multi-lingual G2P [8, 9] conversion, many researchers use the deep neural network (DNN) to improve the model performance of polyphone disambiguation. Shan et al. [10] treated polyphone disambiguation as a sequential labeling task and proposed a Bi-LSTM-based [11] approach jointed with part-of-speech (POS) tagging information. Cai et al. [2] proposed a data-driven approach using Bi-LSTM with word and sentence level embeddings. However, the capability of feature extraction with LSTM is limited, and the above methods cannot solve the out of vocabulary (OOV) problem due to the restricted word dictionary. Zhang et al. [12] proposed a mask-based architecture composed of Bi-LSTM and convolutional neural network (CNN) and Zhang et al. [13] constructed a framework based on CNN in a distantly supervised way. Concerning pre-trained model, Dai et al. [14] proposed a method combining the pre-trained BERT [15] with a neural-network based classifier and Sun et al. [16] distilled the knowledge from the standard BERT model into a smaller BERT model for polyphone disambiguation. Thus, BERT has shown its splendid ability to learn semantic representations from raw character sequences.

The pronunciation of a polyphonic character is related to the meanings of the word, which carries more semantic information than a single character. For example, the pronunciation of "倒" in word "倒塌" (collapse) is different from another word "倒立" (handstand). The pronunciations are marked as "dao3" and "dao4" respectively, where the different numbers represent different tones of pinyin. Therefore, we assume that combining the word segmentation information with the pre-trained BERT can improve the results of polyphone disambiguation. However, the standard Chinese pre-trained BERT model takes characters as the basic units, which ignores the semantic representations of Chinese words. Additionally, some existing CWS tools using for word segmentation are prone to cause cascading errors, leading to a mismatch with the golden-standard result. To tackle these issues, Li et al. [17] developed a model which integrates word segmentation information into a self-attention mechanism to enhance the pre-trained Chinese character representation. Tian et al. [18] proposed a neural approach with a two-way attention mechanism to incorporate both contextual feature and corresponding syntactic knowledge for the joint CWS and Part of Speech (PoS) tagging task.

To summarize, the main contributions of this paper are listed as follows: (1) we propose a novel neural architecture based on the pre-trained BERT with mixed pooling mechanism and window-based attention to improve polyphone disambiguation; (2) the proposed method can effectively avoid the OOV problem and alleviate cascading errors caused by the external CWS tools; (3) the proposed model achieves 99.06% accuracy which outperforms the baselines on a public benchmark dataset of Chinese polyphonic characters [19].

## 2. Method

The proposed architecture, named BERT with Mixed context Features Attention (BERT-MFA), is illustrated in Fig.1. The backbone of the model for polyphone disambiguation is the pre-trained BERT encoder with a dense layer, where the input text is a sequence of $n$ characters $\mathrm{X} = [x_1, x_2, ..., x_i, ..., x_n]$ ($x_i$ denotes the polyphonic character needs to be disambiguated), the output is the correct pinyin transcription of the correspond-

*Indicates equal contribution. Listing order is random.



http://dx.doi.org/10.21437/Interspeech.2021-1232

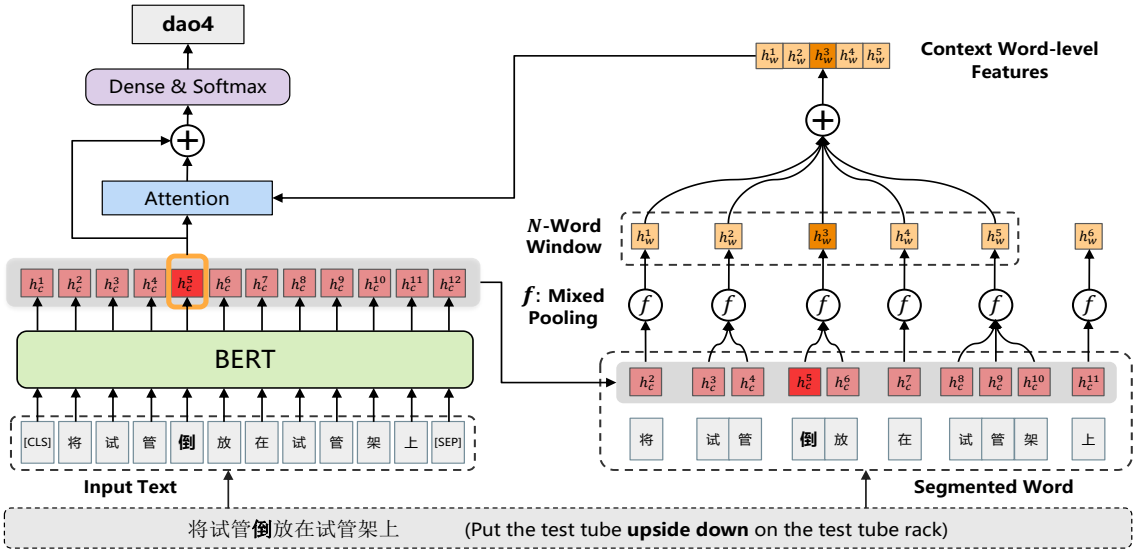


Figure 1: *Overview of the proposed architecture.*

ing character. In this work, we explore the mixed pooling mechanism and window-based attention module to exploit explicit word information. We first introduce the proposed method of the word-level feature extraction, then discuss the window-based attention and ways to integrate context word-level features with the polyphonic character. Lastly, we describe the polyphone disambiguation network and the loss function.

### 2.1. Word-level Feature Abstraction

Firstly, a character sequence $\mathrm{X} = [x_1, x_2, ..., x_i, ..., x_n]$ is fed into the pre-trained BERT encoder to get the character-level hidden features:

$$\mathbf{H}_\mathrm{c} = [\mathbf{h}_\mathrm{c}^1, \mathbf{h}_\mathrm{c}^2, ..., \mathbf{h}_\mathrm{c}^i, ..., \mathbf{h}_\mathrm{c}^n] \tag{1}$$

where $\mathbf{h}_\mathrm{c}^i \in \mathbb{R}^d$ is the hidden vector of polyphone character and $d$ is the hidden dimension of the encoder. Secondly, we use an external CWS tool $\tau$ to segment X into a sequence of words, where $m$ is the length of the sequence:

$$\tau(\mathrm{X}) = [\omega_1, \omega_2, ..., \omega_j, ..., \omega_m], (m \leq n) \tag{2}$$

where $w_j = \{x_s, x_{s+1}, ..., x_i, ..., x_{s+l-1}\}$ is the segmented words with the length of $l$, which contains the polyphone character $x_i$ , and $s$ is the index of $w_j$'s first character. We segment $\mathbf{H}_\mathrm{c}$ according to the result of $\tau(\mathrm{X})$. For example, if $\tau(\mathrm{X}) = [\{x_1, x_2, x_3\}, .., \{x_i, x_{i+1}\}, .., \{x_{n-1}, x_n\}]$, then we get:

$$\tau(\mathbf{H}_\mathrm{c}) = [\{\mathbf{h}_\mathrm{c}^1, \mathbf{h}_\mathrm{c}^2, \mathbf{h}_\mathrm{c}^3\}, .., \{\mathbf{h}_\mathrm{c}^i, \mathbf{h}_\mathrm{c}^{i+1}\}, .., \{\mathbf{h}_\mathrm{c}^{n-1}, \mathbf{h}_\mathrm{c}^n\}] \tag{3}$$

As a result, $\mathbf{H}_\mathrm{c}$ is divided into several sublists and each sublist includes character-level hidden features of the segmented words. Motivated by Li et al. [17], we transform the sublist into a word-level hidden feature $\mathbf{h}_\mathrm{w}^j$ with a mixed pooling strategy defined as follows:

$$\begin{aligned}\mathbf{h}_\mathrm{w}^j = {} & \lambda\,\mathrm{Maxpooling}(\{\mathbf{h}_\mathrm{c}^s, ..., \mathbf{h}_\mathrm{c}^{s+l-1}\}) \\ & + (1-\lambda)\,\mathrm{Meanpooling}(\{\mathbf{h}_\mathrm{c}^s, ..., \mathbf{h}_\mathrm{c}^{s+l-1}\})\end{aligned} \tag{4}$$

where $\lambda$ is a hyper-parameter used to balance the mean pooling and max pooling. For each sublist, we can obtain the word-level hidden feature list $\mathbf{H}_\mathrm{w} = [\mathbf{h}_\mathrm{w}^1, \mathbf{h}_\mathrm{w}^2, ..., \mathbf{h}_\mathrm{w}^j, ..., \mathbf{h}_\mathrm{w}^m]$, where $\mathbf{h}_\mathrm{w}^j \in \mathbb{R}^d$ denotes the hidden vector of the word which contains the polyphone character.

### 2.2. Window-based Attention

The work in [10] shows that information between the neighboring words of a polyphonic character is essential for polyphone disambiguation. Therefore, we use $N$-word window to improve neiboring information extraction in our model, where $\mathbf{h}_\mathrm{w}^j$ is the center of the sequence and its left and right neighboring word-level features with a size of $N$ are considered as the context. Then, we extract window-based context features S from $\mathbf{H}_\mathrm{w}$ as follows:

$$\mathrm{S} = [\mathbf{h}_\mathrm{w}^{j-N}, ..., \mathbf{h}_\mathrm{w}^j, ..., \mathbf{h}_\mathrm{w}^{j+N}] = [\mathbf{s}_1, ..., \mathbf{s}_k, ..., \mathbf{s}_{2N+1}] \tag{5}$$

where $\mathbf{s}_k$ denotes the $k$th word-level hidden feature in S, and padding is used if the window exceeds the range of $\mathbf{H}_\mathrm{w}$.

Inspired by [18], we adopt the window-based attention mechanism to incorporate external knowledge instead of directly concatenating the embeddings extracted from context features in previous studies [2, 10]. The attention module can effectively learn the weights of the corresponding features for polyphone and it benefits model performance. In our work, we use a feed-forward attention module [20] to associate the polyphone's character-level feature $\mathbf{h}_\mathrm{c}^i$ derived from BERT with the context feature S. For each word-level feature $\mathbf{s}_k$ in S, the attention weight $a_k \in \mathbb{R}^1$ is computed as:

$$a_k = \frac{exp[(\mathbf{h}_\mathrm{c}^i)^\mathsf{T} \cdot \mathbf{s}_k]}{\sum_{k=1}^{2N+1} exp[(\mathbf{h}_\mathrm{c}^i)^\mathsf{T} \cdot \mathbf{s}_k]} \tag{6}$$

Then the weighted embedding $\mathbf{a} \in \mathbb{R}^d$ can be calculated as:

$$\mathbf{a} = \sum_{k=1}^{2N+1} a_k \mathbf{s}_k \tag{7}$$

### 2.3. Polyphone Disambiguation Network

After obtaining the output of attention module $\mathbf{a}$, we concatenate it with the polyphone's character-level feature $\mathbf{h}_\mathrm{c}^i$. Next we feed the result into a fully-connected layer followed by a soft-

max layer. Finally, the pinyin probability distribution $\hat{\mathbf{y}}$ can be expressed as follows:

$$\hat{\mathbf{y}} = [\hat{y}_1, \hat{y}_2, ..., \hat{y}_t] = \text{softmax}[\,\mathbf{W} \cdot (\mathbf{h}_c^i \oplus \mathbf{a}) + \mathbf{b}\,] \quad (8)$$

where $\mathbf{W}$ and $\mathbf{b}$ are trainable parameters and $t$ is the number of all possible classes of pinyin for polyphonic characters in the dataset. In our experiment, we use cross-entropy as the loss function to train the model.

# 3. Experiments

## 3.1. Dataset

We use an open benchmark of the Chinese polyphonic character dataset named CPP Dataset[1] in our experiments. It consists of 99,264 sentences, each of which includes a polyphone with the correct pinyin transcription with an average length of approximately 31 characters. The dataset contains 623 different polyphonic characters, each of which has at least two pronunciations. More details about the dataset can be found in [19].

Table 1: *Comparison of different systems.*

| System | Accuracy (%) |
|---|---|
| pypinyin | 86.13 |
| g2pM | 97.31 |
| Seq2Seq (CNN+LM) | 97.51 |
| BERT | 97.85 |
| TwASP (BERT+POS) | 98.80 |
| TwASP (BERT+Syn.) | 98.71 |
| TwASP (BERT+Dep.) | 98.70 |
| BERT-MFA (Ours) | **99.06** |

## 3.2. Experimental Setup

We test the proposed BERT-MFA model with the publicly available Chinese pre-trained BERT [21] model as the basic encoder, which consists of 12 Transformer [22] layers, with 768 hidden dimensions. We keep the default hyper-parameter of BERT and use Adam [23] as the optimizer with a learning rate of 5e-5 to fine-tune the parameters. The batch size and the maximum sequence length are both set to 64. The accuracy of polyphone disambiguation is used as the evaluation metric in all experiments.

For comparison, we adopt the following four systems as baselines: the first one is an open-source Chinese G2P library called pypinyin[2], which is a rule-based system; the second one named g2pM[19] is based on the Bi-LSTM approach; the third one named Seq2Seq (CNN+LM)[13] is a convolutional seq2seq model with augmented data by audio-generated texts containing polyphonic characters; the last one is the vanilla Chinese BERT model. For the Chinese BERT model, we add a fully connected layer to the pre-trained BERT [21] and fine-tune it with the CPP dataset. Besides, we compare BERT-MFA with a new proposed model named TwASP [18], which achieves state-of-the-art performance for joint CWS and POS tagging tasks. This model utilizes a two-way attention mechanism to incorporate both word information and their corresponding syntactic knowledge. We implement it for polyphone disambiguation task by using Chinese BERT as the encoder and combining it with POS labels (BERT+POS), syntactic constituents (BERT+Syn.), and dependency relations (BERT+Dep.) as syntactic knowledge, respectively. The Stanford CoreNLP Toolkit (SCT)[3] [24] is employed as CWS and POS tagging tool in our experiments.

[1]dataset: https://github.com/kakaobrain/g2pM/tree/master/data

[2]pypinyin: https://github.com/mozillazg/python-pinyin

## 3.3. Performance Evaluation

Table 1 shows the results of all the mentioned approaches above. Overall, the proposed BERT-MFA model outperforms the baseline systems and achieves state-of-the-art performance with an accuracy of 99.06%. Since BERT-MFA is based on the pre-trained language model and word information, it has significantly improved compared with the g2pM and Seq2Seq (CNN+LM) model. We can also observe that the TwASP model improves the performance remarkably comparing with the vanilla BERT. However, TwASP randomly initializes the embeddings for words and features of syntactic knowledge can easily lead to the OOV problem during inference. By contrast, BERT-MFA utilizes a mixed pooling strategy to convert character-level features derived from BERT into word-level features which can effectively avoid this issue. Additionally, although BERT-MFA only exploits word segmentation information, it achieves better performance than the TwASP model.

Table 2: *The accuracy (%) on polyphonic characters.*

| Statistics (Polyphone frequency) | g2pM | BERT | BERT-MFA |
|---|---|---|---|
| 更<br>geng4: 117 geng1: 44 | 76.19 | 85.71 | **90.48** |
| 丧<br>sang4: 106 sang1: 45 | 84.21 | 94.74 | **100.00** |
| 薄<br>bo2: 81 bao2: 42 bo4: 5 | 76.47 | 88.24 | **94.12** |
| 调<br>diao4: 109 tiao2: 41 | 84.21 | 89.47 | **94.74** |
| 哄<br>hong1:16 hong3:16 hong4:6 | 60.00 | 80.00 | **100.00** |
| 捋<br>lu3: 7 luo1: 5 | 50.00 | 50.00 | **100.00** |
| 劲<br>jing4: 92 jin4: 68 | 80.95 | 90.48 | **95.24** |
| 率<br>lu4: 97 shuai4: 63 | 85.71 | 90.48 | **100.00** |

As shown in table 2, we compare our model with g2pM and vanilla Chinese BERT for some typical polyphonic characters suffering from imbalanced distribution or data scarcity in the training set. For those characters with serious imbalance problems, e.g., "更" and "薄", BERT-MFA gets higher predicting accuracy compared with other systems. Especially, our proposed model can properly handle the characters with insufficient data size, e.g., "哄" and "捋". These may attribute

[3]Stanford CoreNLP Toolkit: https://stanfordnlp.github.io/CoreNLP/

to the window-based feed-forward attention which allows the model to effectively refer to in-window and critical word information neighboring polyphonic characters, without the need for distant and irrelevant word information. This enhances BERT-MFA performance noticeably comparing with g2pM and BERT even on the unbalanced or inadequate training set.

Table 3: *Ablation study of BERT-MFA.*

| **Model** | **Accuracy (%)** |
|---|---|
| BERT | 97.85 |
| BERT-RC | 98.33 (+0.48) |
| BERT-RA | 98.65 (+0.80) |
| BERT-MC | 98.72 (+0.87) |
| BERT-MFA (Ours) | **99.06 (+1.21)** |

Table 4: *Test accuracy (%) of different CWS tools.*

| **Segmenter** | **BERT-RC** | **BERT-MFA** |
|---|---|---|
| Thulac | 98.08 | 98.95 |
| Hanlp | 98.25 | 98.98 |
| Ictclas | 98.18 | 99.02 |
| SCT | **98.33** | **99.06** |

### 3.4. Ablation Study

To demonstrate the effectiveness of the proposed model, we conduct some ablation experiments. The first model is vanilla Chinese BERT introduced in 3.2. The second one is BERT-RC, where the word information is directly added to the first model by randomly initializing word-level embeddings and concatenating with polyphone character-level features derived from BERT. The third model is BERT-RA where we replace concatenating operation in BERT-RC with the proposed window-based attention module. The last one is BERT-MC, in which we adopt the mixed mechanism instead of randomly initializing the embeddings of BERT-RC. We set the window size $N = 2$, hyper-parameter $\lambda = 0.5$ and employ SCT as the CWS tool $\tau$ for all ablation experiments. As shown in table 3, all models that integrate word segmentation information outperform the vanilla BERT. Comparing with BERT-RC, BERT-MC with the mixed mechanism obtains preferable word-level features and avoids the OOV problem. Comparing BERT-RC and BERT-RA, we find that the proposed attention module can learn and benefit more from word-level features. In conclusion, all components of our proposed model are necessary for achieving the highest accuracy.

### 3.5. Factor Analysis

#### *3.5.1. Chinese word segmentation*

Since the result of word segmentation has a great impact on polyphone disambiguation, we employ four popular CWS tools as $\tau$ in our model to segment the input sequences: Thulac [25], Hanlp [26], Ictclas [27] and SCT. As shown in table 4, our model achieves better performance regardless of the CWS tools comparing with BERT-RC, which confirms the effectiveness of the mixed pooling mechanism and attention module in our proposed model. It can be also observed that the results of BERT-RC have visible differences among different CWS tools, while our model achieves a closer performance for these tools. This indicates that the cascading errors caused by external tools can be alleviated by our model.

#### *3.5.2. Attention window size*

We also analyze the influence of the window size by increasing $N$ from 0 to 3. Table 5 shows that without the left and right context features ($N$=0), the accuracy is lower than other conditions. The best performance is achieved at $N$=2 (consider the context of two words), but the result degrades when expanding N to 3. This means larger window size may not benefit context extraction.

#### *3.5.3. Ratio of max and mean pooling*

We explore different ratios of the mixed mechanism by changing the interval from 0 to 1. Fig.2 presents the results with different values of hyper-parameter $\lambda$ in Equation 4. When we set $\lambda$ to the extreme situations ($\lambda$=0 or 1), the results are worse than other conditions. The highest accuracy is obtained when $\lambda$ is set to 0.5, which gets a good compromise between the max and mean pooling.

Table 5: *The results of using different window sizes.*

| **Window size** $N$ | **0** | **1** | **2** | **3** |
|---|---|---|---|---|
| **Accuracy (%)** | 98.89 | 98.97 | **99.06** | 98.90 |

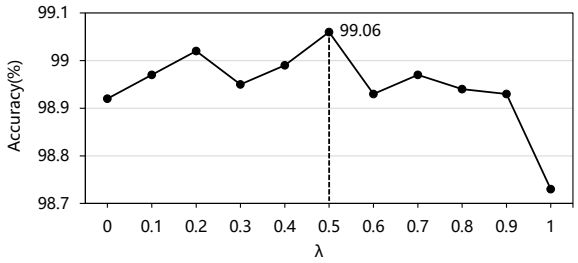


Figure 2: *The impact of different values of* $\lambda$.

## 4. Conclusions

In this paper, we propose a novel BERT-MFA model based on mixed pooling mechanism and window-based attention to improve polyphone disambiguation for Chinese G2P conversion. From the experimental results on the benchmark polyphonic character dataset, BERT-MFA significantly improves the accuracy by over 1.2% and achieves better performance for polyphonic characters with distribution imbalance or data scarcity comparing with the vanilla Chinese BERT. Ablation experiments prove that all the major components of BERT-MFA are essential for achieving the best performance. Moreover, BERT-MFA can avoid the OOV issue and alleviate cascading errors caused by CWS tools effectively. Future work includes employing our method to Chinese dialects featured by a relatively small training set of polyphonic characters sensitive to word-based semantic information and incorporating multiple types of knowledge like POS labels for our proposed model.

# 5. References


[1] K. Rao, F. Peng, H. Sak, and F. Beaufays, "Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks," in *ICASSP*, 2015, pp. 4225–4229.

[2] Z. Cai, Y. Yang, C. Zhang, X. Qin, and M. Li, "Polyphone disambiguation for mandarin chinese using conditional neural network with multi-level embedding features," *Proc. Interspeech 2019*, pp. 2110–2114, 2019.

[3] Z.-R. Zhang, M. Chu, and E. Chang, "An efficient way to learn rules for grapheme-to-phoneme conversion in chinese," in *International Symposium on Chinese Spoken Language Processing*, 2002.

[4] F.-L. Huang, "Disambiguating effectively chinese polyphonic ambiguity based on unify approach," in *2008 International Conference on Machine Learning and Cybernetics*, vol. 6. IEEE, 2008, pp. 3242–3246.

[5] X. Mao, Y. Dong, J. Han, D. Huang, and H. Wang, "Inequality maximum entropy classifier with character features for polyphone disambiguation in mandarin tts systems," in *ICASSP*, vol. 4, 2007, pp. IV–705.

[6] F. M. H. G. W. Renhua, "Multi-level polyphone disambiguation for mandarin grapheme-phoneme conversion," *Computer Engineering and Applications*, no. 2, p. 49, 2006.

[7] P. Jyothi and M. Hasegawa-Johnson, "Low-resource grapheme-to-phoneme conversion using recurrent neural networks," in *ICASSP*, 2017, pp. 5030–5034.

[8] M. Yu, H. D. Nguyen, A. Sokolov, J. Lepird, K. M. Sathyendra, S. Choudhary, A. Mouchtaris, and S. Kunzmann, "Multilingual grapheme-to-phoneme conversion with byte representation," in *ICASSP*, 2020, pp. 8234–8238.

[9] A. Sokolov, T. Rohlin, and A. Rastrow, "Neural machine translation for multilingual grapheme-to-phoneme conversion," *arXiv preprint arXiv:2006.14194*, 2020.

[10] C. Shan, L. Xie, and K. Yao, "A bi-directional lstm approach for polyphone disambiguation in mandarin chinese," in *2016 10th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. IEEE, 2016, pp. 1–5.

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[12] H. Zhang, H. Pan, and X. Li, "A mask-based model for mandarin chinese polyphone disambiguation," *Proc. Interspeech 2020*, pp. 1728–1732, 2020.

[13] J. Zhang, Y. Zhao, J. Zhu, and J. Xiao, "Distant supervision for polyphone disambiguation in mandarin chinese," *Proc. Interspeech 2020*, pp. 1753–1757, 2020.

[14] D. Dai, Z. Wu, S. Kang, X. Wu, J. Jia, D. Su, D. Yu, and H. Meng, "Disambiguation of chinese polyphones in an end-to-end framework with semantic features extracted by pre-trained bert." in *INTERSPEECH*, 2019, pp. 2090–2094.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[16] H. Sun, X. Tan, J.-W. Gan, S. Zhao, D. Han, H. Liu, T. Qin, and T.-Y. Liu, "Knowledge distillation from bert in pre-training and fine-tuning for polyphone disambiguation," in *ASRU*. IEEE, 2019, pp. 168–175.

[17] Y. Li, B. Yu, M. Xue, and T. Liu, "Enhancing pre-trained chinese character representation with word-aligned attention," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 3442–3448.

[18] Y. Tian, Y. Song, X. Ao, F. Xia, X. Quan, T. Zhang, and Y. Wang, "Joint chinese word segmentation and part-of-speech tagging via two-way attentions of auto-analyzed knowledge," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 8286–8296.

[19] K. Park and S. Lee, "g2pm: A neural grapheme-to-phoneme conversion package for mandarinchinese based on a new open benchmark dataset," *Proc. Interspeech 2020*, pp. 1723–1727, 2020.

[20] C. Raffel and D. P. Ellis, "Feed-forward networks with attention can solve some long-term memory problems," *arXiv preprint arXiv:1512.08756*, 2015.

[21] Y. Cui, W. Che, T. Liu, B. Qin, Z. Yang, S. Wang, and G. Hu, "Pre-training with whole word masking for chinese bert," *arXiv preprint arXiv:1906.08101*, 2019.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

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

[24] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit," in *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, 2014, pp. 55–60.

[25] M. Sun, X. Chen, K. Zhang, Z. Guo, and Z. Liu, "Thulac: An efficient lexical analyzer for chinese," 2016.

[26] H. He, "HanLP: Han Language Processing," 2020. [Online]. Available: https://github.com/hankcs/HanLP

[27] H.-P. Zhang, H.-K. Yu, D. Xiong, and Q. Liu, "Hhmm-based chinese lexical analyzer ictclas," in *Proceedings of the second SIGHAN workshop on Chinese language processing*, 2003, pp. 184–187.