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

PDF: Polyphone Disambiguation in Chinese by Using FLAT

Haiteng Zhang

DataBaker (Beijing) Technology Co., Ltd, Beijing, China
zhanghaiteng@data-baker.com

Abstract

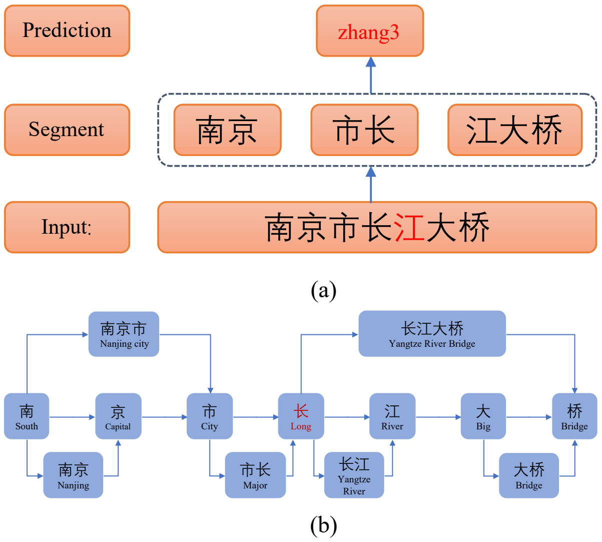
Polyphone disambiguation is an essential procedure in the front-end module of the Chinese text-to-speech (TTS) system. It serves to predict the pronunciation of the input polyphonic character. In the Chinese TTS system, a well-designed pronunciation dictionary plays a crucial role in supplying pinyin to words. However, the conventional system is unable to fully utilize the pronunciation dictionary while modelling because of the unavoidable Chinese segment errors and model structure. In this paper, we proposed a system named PDF: **P**olyphone **D**isambiguation by using **F**LAT. The proposed model encodes both the input character sequence and dictionary matched words of the sentence, enabling the model to both avoid segment errors and leverage the well-designed pronunciation dictionary in the model. Additionally, we also use the pre-trained language model (PLM) as an encoder to extract the contextual information of input sequence. The experimental results verified the effectiveness of the proposed PDF model. Our system obtains an improvement in accuracy by 0.98% compared to Bert on an open-source dataset. The experiential results demonstrate that leveraging pronunciation dictionary while modelling helps improve the performance of polyphone disambiguation system.

Index Terms: Polyphone Disambiguation, FLAT, PDF, pronunciation dictionary

1. Introduction

Polyphone disambiguation has a great impact on Mandarin text-to-speech system. As an important module of the front-end module, it serves to identify the correct pronunciation of the given polyphonic characters. A lot of academic efforts have been made to address the polyphone disambiguation problem. The previous methods can be divided into knowledge-based approaches and learning-based approaches [1]. In the Chinese TTS system, a well-designed pinyin dictionary supplies pinyin to a word, thus it is crucial to both types of system above. However, the previous systems didn't utilize the pinyin dictionary which is helpful to further improve the overall performance while modelling.

The knowledge-based system heavily relies on the well-designed pronunciation dictionary where abundant words that contain polyphonic characters and corresponding pronunciation are listed [2, 3]. During runtime, sentences are segmented into word pieces and the system starts to search the pinyin from the dictionary. Then the system will utilize the handcrafted rule to solve the remaining individual polyphonic characters. The knowledge-based system achieves good performance by applying the dictionary. However, as shown in Figure 1, due to the limitation of Chinese word segmentation, the unavoidable and incorrectly segmented result caused by the segment system leads to polyphone disambiguation errors.



(a)

(b)

Figure 1: (a) *The errors cause by Chinese segment system*; (b) *The lattice structure of input sentence.*

The learning-based can be concluded as the statistical method and the neural network method. Some statistical methods such as Decision trees (DT) or Maximum Entropy (ME) Model can reach reasonable performance [4, 5]. Recently, neural network model has been employed in the task of polyphone disambiguation. Shan and K.Park adopted bidirectional long-short-term memory (BLSTM) layer to predict the pronunciation of polyphonic character [6, 7] that is in accord with the context. Seq2seq model was also employed in the task in a distantly supervised way [8]. Taking advantage of the abundant corpus, researchers set out to apply pre-trained language model (PLM) in the task [9, 10]. With the adoption of representation of semantic information, PLM has been proved to boost the accuracy rate of the task. Nonetheless, modelling of the systems above was based on character-level and sentence-level, which did not utilize the lexicon-level feature. Zhang treated the segment result as lexicon-level feature [11], and Cai combined multi-granularity feature including lexicon-level embedding as input [12]. Both models demonstrated the effectiveness of the lexicon feature for the task.

However, the segment system is prior to the modelling preprocess, which means the model still cannot avoid the segment errors brought by Chinese word segment system. As a result, the lexicon feature with segment errors fed to the later model leads to misprediction. Moreover, the previous system still didn't utilize the pinyin dictionary while modelling.

Recently, lexicon features have been applied in the Named Entity Recognition (NER) task while modelling and successfully avoided the Chinese segment errors. Zhang et al. [13] first employed lattice structure composed of characters

http://dx.doi.org/10.21437/Interspeech.2021-1087

sequence and all potential words as shown in Figure 1(b) in Lattice-LSTM. Li et al proposed FLAT structure [14] which flatted the lattice structure and was applied in a Transformer module. By this means, FLAT structure could integrate the dictionary into the modelling process. With the power of the self-attention mechanism and relative position encoding, the FLAT outperformed other lexicon-based models both in performance and efficiency.

Inspired by FLAT, rather than sequence labelling, we employed a FLAT structure in the polyphone disambiguation and verified the FLAT structure's performance on classification. Besides, taking the advantage of transformer, we applied the pinyin from the pronunciation dictionary as an input feature to boost the performance. In this way, we proposed a novel polyphone disambiguation model and named it PDF: **P**olyphone **D**isambiguation by using **F**LAT. Owing to flatten lattice structure, PDF can not only model input sequence without segment errors but also utilize the pinyin feature from the pronunciation dictionary. To further explore the latent semantic feature, we employed the Chinese Bert as the PLM encoder. The contribution of this paper can be summarized as follows:

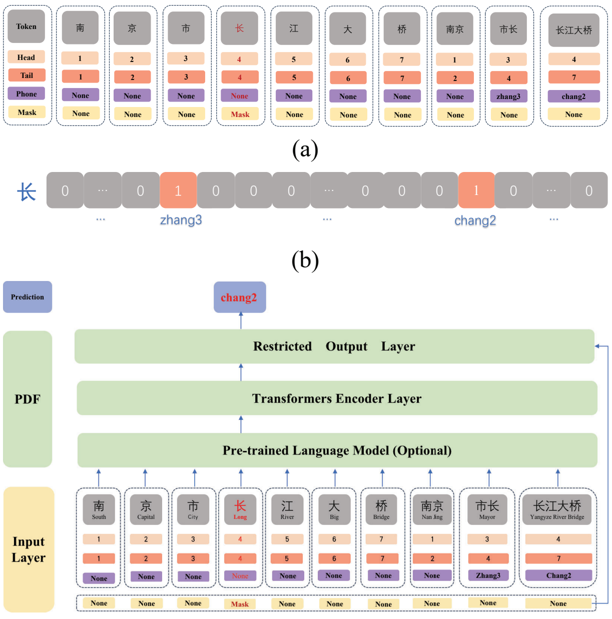
1. On the basis of FLAT structure, we proposed a novel model named PDF which avoided word segment errors.
2. We utilized the pronunciation dictionary while modelling, boosting the performance of the polyphone disambiguation system.
3. With and without the PLM encoder, the proposed model can surpass the Bert model in accuracy by 0.98% and 0.38% respectively on an open source dataset [7].

2. Method

The conventional polyphone disambiguation serves to convert the input polyphonic characters into their corresponding pinyin. We applied FLAT as the basic model architecture. The FLAT structure was first applied in the field of polyphone disambiguation, different from how the origin work [14] solved the sequence labelling task in a seq2seq way. We regarded the Polyphone disambiguation as a classification task and named the proposed model as PDF: **P**olyphone **D**isambiguation by using **F**LAT. Figure 2(c) depicts the overall structure of PDF model. The proposed model is mainly composed of the following four parts: Input span feature layer, PLM encoder layer (optional), Transformer encoder layer and finally Restricted output layer. In order to utilize the pronunciation dictionary in the process of modelling, we provided the model with the pronunciation of the potential words which contained the polyphonic character in the input span feature layer. The proposed PDF model also applied the weighted-softmax mechanism introduced by Zhang et al. [11] to prevent mis-prediction.

2.1. Input span feature layer

As shown in Figure 2(a), the input span features are composed of token, head index, tail index, mask vector and pinyin feature. In our work, the input sentence would first match the pronunciation dictionary to gain all the potential words into the lattice structure as Figure 1(b) before flattening into tokens. Besides, the head index and tail index would calculate according to the index of the token and potential words from the initial input sentence. Additionally, the mask vector is built upon the relation between the polyphonic character and its pronunciation. In order to make use of the pinyin dictionary while modelling, we supplied pinyin only to those potential words that carry polyphonic characters. The details of the input features are concluded as follows:



(a)

(b)

(c)

Figure 2: (a) *Input span feature*; (b) *Mask vector of "长"*; (c) *Structure of PDF model.*

- Token: the input character along with all potential words that matched the pronunciation dictionary inside of the input sentence.
- Head index: index denotes the first char of token inside the initial sentence;
- Tail index: index denotes the last char of token inside the initial sentence;
- Mask vector: A Boolean vector indicates the relation between polyphonic characters and their pronunciations.
- Pinyin feature: This feature provides the corresponding pinyin of the polyphonic character inside of the matched word.

As shown in Figure 2 (a), the input token index contains all the input character tokens and potential words. In this way, the PDF can model both characters and all the word pieces of the input sequences so as to avoid segment errors.

As described before, the head and tail index denote the position of the token. As for character token, the head and tail index denote the initial character position. As for words, the head and tail index denote the first character and last character's position of words in the sentence respectively. Based on the head and tail index, the relative distance between different input spans can be described as the followings:

$$d_{ij}^{(h,h)} = head[i] - head[j] \tag{1}$$

$$d_{ij}^{(h,t)} = head[i] - tail[j] \tag{2}$$

$$d_{ij}^{(t,h)} = tail[i] - head[j] \tag{3}$$

$$d_{ij}^{(t,t)} = tail[i] - tail[j] \tag{4}$$

While i, j denote different spans of the input token and $head[i]$, $tail[i]$ denote the head index and the tail index of the i^{th} span token. And the relative distance will be applied to calculate the relative position encoding which will be described in the following section.

Considering memory saving, different from [11], mask vector only sets up for the polyphonic character rather than all input characters of the sentence. The mask vector's dimension is equal to the amount of pinyin set. As shown in figure 2 (b), the value in the mask vector is set to 1 when the indexes of pinyin are "chang2" and "zhang3" which are the candidate prediction of polyphonic character "长". And the remaining value of mask vector is set to 0. Specifically, the mask vector is not fed into model but acts as the weighted factor in the weighted-softmax mechanism.

To employ the pronunciation dictionary while modelling, we provided the correct pronunciation of the potential word only if the matched word contains a polyphonic character. We would set the value to "<None>" for the remaining situation. The pinyin feature would be converted into a vector and fed for the model. In this way, we can apply the pronunciation from pinyin dictionary to the polyphone disambiguation system and reduce the complexity of predicting module.

2.2. Pre-trained Language Model Encoder

The pronunciation of the target polyphonic character is mainly affected by the context of a sentence. The dynamical representation computed by the self-attention mechanism from the contextual has a better representation of the token. The previous researches [10] have shown that PLM encoder such as BERT is effective to enhance the performance of a polyphone disambiguation system. Benefited from abundant, easy-accessed and unlabeled corpus, the PLM encoder is capable of extracting semantic and contextual representation from the raw Chinese input. In the PDF model, we applied the PLM model to gain better representation of the input character. As depicted in Figure 2(c), after getting the span features from the input layer, the PLM encoder layer accepts the raw token sequence and converts it into a sequence of contextual features.

2.3. Transformer Encoder Layer

Combined with the non-linear transformation, the relative position encoding of different span tokens can be described as follow:

$$R_{ij} = relu(W_r(Concat(p_{d_{ij}^{(h,h)}}, p_{d_{ij}^{(t,h)}}, p_{d_{ij}^{(h,t)}}, p_{d_{ij}^{(t,t)}}) + b)) \quad (5)$$

Where the parameter W_r and b are learnable and the p_d denotes the sinusoidal positional encoding mentioned in the previous work [15]. The relative position encoding calculated by relative distance can reflect the relation between different span features. In this way, we can leverage the relative position encoding rather than the absolute position encoding as the variant self-attention mechanism does [16]. Therefore, the polyphonic character can better interact with the dictionary matched words that contain the target polyphonic character through the self-attention mechanism and the relative position encoding.

After the Transformer Encoder Layer, we only gathered the hidden state of polyphonic characters in the original sentences and fed the character representation into the following Restricted Output Layer for prediction.

2.4. Restricted Output Layer

Subsequent to the transformer encoder layer, the Restricted Output Layer is handily established by a fully connected layer. In order to eliminate the case of mis-predicting pinyin from other polyphonic characters, similar to [11], we also set a restricted output layer and applied weighted-softmax mechanism here. The weighted-softmax can restrict to the candidate set of the input polyphonic character, and therefore eliminating influence of mis-prediction that comes from the non-candidate Pinyin set brought to the model. The weighted-softmax is described as follow:

$$p_i = \frac{w_i \times e^{v_i}}{\sum_{j=1}^{n} w_j \times e^{v_j}} \quad (6)$$

Where w denotes the mask vector we described in section 2.1. And e^{v_i} denotes the score of the pronunciation among the whole Pinyin set calculated by the model. As shown in the function, the boolean mask vector is employed here as the weight factor of each pronunciation in the Pinyin set. In this way, the model can assure that the probability of the non-candidate Pinyin would set to zero and it prevent predicting Pinyin outside of the candidate set.

3. Experiments

3.1. Dataset

To evaluate the proposed polyphones disambiguation system, we conducted several experiments on an open source dataset: CPP (Chinese Polyphone with Pinyin) [7]. There is a total of 99,264 sentences in the dataset and each sentence contains only one polyphonic character and its corresponding correct pinyin annotation. While the training set contains 79,117 sentences, the dev set contains 9,893 sentences and the test set contains 10,254 sentences. The dataset contains 623 polyphonic characters in total with a corresponding 876 pinyin. The length of the dataset ranges from 5 to 50 while the amount of different polyphones ranges from 10 to 250.

3.2. Experiment setting

To verify the proposed PDF model, we also implemented and compared the following systems, and reported the results of accuracy rate (ACC) on the CPP dataset in table 1. The details of the experimental systems are listed as follow:

1. **G2Pm (BLSTM):** A recent solution for polyphone disambiguation [7]. The model applied a BLSTM structure to encode the contextual information of the raw Chinese input. The system set cross-entropy as the loss function for training.
2. **Distant supervision:** A recently published polyphone disambiguation system [8] is trained in a distantly supervised way. It solves the polyphone disambiguation in a seq2seq way. The core module consists of a character-phoneme transformation module and a reranking module.
3. **MASK_BASED:** Strictly following the description in [11], we implemented the MASK_BASED model for comparison. We applied the Jieba tool for obtaining the word segment result and the POS feature in our implementation. The MASK_BASED model was constructed by both BLSTM and 1D-CNN structure. We applied the weighte-softmax mechanism in the experiments. Additionally, we chose the Modified Focal Loss as the loss function.

4. **G2Pm (BERT)**: Same as the structure described in [7], this system applied the BERT structure as an encoder to get the semantic representation. The gathered hidden state of the polyphonic character would later be fed to the fully connected layer for task prediction.
5. **BERT (with LSTM)**: According to [10], attached by PLM encoder, BLSTM achieved better performance by the better capability of modelling context. We conducted the experiments for comparison and the hypermeters setting was the same as G2Pm (BERT).
6. **PDF (without PLM):** To examine the efficiency of the PDF structure, we first implemented our model without PLM encoder layer. In the training process, we adopted Adamw as the optimizer and set the learning rate to 5.0e-4 with the batch size setting to 64. Besides, the layer of Transformer was set to 1. We further experimented with different PDF models by combining different features provided by the pronunciation dictionary and recorded the performance in table2. The pronunciation dictionary we employed comes from DataBaker.
7. **PDF (with PLM):** The experiment setting was similar to the system **PDF (without PLM)** but equipped with PLM encoder layer. Rather than freezing the PLM's weight, we trained the entire model including the PLM encoder layer. The learning rate of PLM layer was set to 5.0e-6 for further fine-tuning the training dataset. The learning rate of other modules was still set to 5.0e-4.

3.3. Result and analysis

3.3.1. Evaluation of different systems

In this session, we compared the experiential performance of the proposed model and the recent learned-based method. Table 1 lists the accuracy rates of different systems on the CPP dataset. Accuracy rates of system G2Pm (BLSTM), Distant Supervision, G2Pm (BERT) were recorded from the origin papers. The result of MASK-BASED model was implemented by us. The BERT (with LSTM) system was operated by us mainly on the basis of transformers packages.

Table 1: *The accuracy of different systems*

System	Test Accuracy
G2Pm (BLSTM) [7]	97.31
Distant supervision [8]	97.51
MASK-BASED [11]	97.68
G2Pm (BERT) [7]	97.85
BERT (with LSTM)	98.04
PDF (without PLM)	**98.23**

As shown in table 1, the proposed PDF (without PLM) model outperformed other systems on CPP. The PDF (without PLM) achieved an accuracy rate of 98.23% with 0.92% and 0.55% increase surpassing the system G2Pm (BLSTM) and MASK-BASED model respectively. Besides, the PDF (without PLM) was slightly superior to the system G2Pm (BERT) and BERT (with LSTM) even without the contextual feature offered by the PLM, demonstrating the effectiveness of the proposed PDF structure.

3.3.2. PDF based on pronunciation dictionary and PLM

Table 2: *The accuracy of a series of PDF model*

System	Test Accuracy
PDF (without dict.)	98.08
PDF (with dict.)	98.23
PDF (with PLM)	**98.83**

To further illustrate the impact of applying pronunciation dictionary, Table 2 describes the performance of a series of PDF model on CPP dataset. As shown in table, the PDF (with dict.) slightly surpassed the PDF (without dict.) with 0.15% higher rate than the latter system, which proves that the pinyin feature is helpful for the task. As a result, we believe that applying the pronunciation dictionary helps to improve the accuracy rate of the system. Additionally, the PDF (with PLM) saw a significant improvement of 0.60%, compared with the former system. It shows the PLM encoder provides extra semantic information for the polyphone disambiguation system.

3.3.3. Evaluation between PLM encoder

Table 3: *The performance among PLM encoder*

System	Test Accuracy
G2Pm (BERT)	97.85
BERT (with LSTM)	98.04
PDF (with PLM)	**98.83**

For a fair comparison, the PDF (with PLM) was implemented to compare the system G2Pm (BERT) and BERT (with LSTM). As represented in table3, the result shows our system surpassed the G2Pm (BERT) and BERT (with LSTM) model by 0.98% and 0.79% respectively.

Based on the performances in the above tables, the advantages of the proposed model, are clear: 1) The flat-lattice structure enables the system not merely to model all the potential words but also to avoid word segment errors. 2) Benefited from the self-attention mechanism, the polyphonic character can interact with its contextual word. 3) The auxiliary feature from the pronunciation dictionary can provide more information for the polyphonic character.

4. Conclusions

In this paper, we proposed a polyphone disambiguation model and named it PDF. Benefited from the flattened lattice structure, the proposed model can model both raw input sequence and all potential words. In this way, the proposed model can avoid segment errors and leverage the lexicon-level information. Besides, we first employed the pinyin in the modelling preprocess and the experiential result shows that the auxiliary feature helps boost the performance of polyphone disambiguation. The experimental results show that the proposed approach outperforms other models on the CPP dataset.

5. Acknowledgements

The authors would like to thank the team of FastNLP for providing open source framework to support the major work of this paper. Besides, the authors also would like to thank the organization of K.Park [7] for the dataset they released for academic promotion.

6. References

[1] C.Shan, L.xie, K.Yao“A Bi-directional LSTM Approach for Polyphone Disambiguation in Mandarin Chinese,” *10th International Symposium on Chinese Spoken Language Processing (ISCSLP), IEEE*, pp. 1-5, 2016.

[2] L. Yi, L. Jian, H. Jie and Z.Xiong, “Improved Grapheme-to-Phoneme Conversion for Mandarin TTS” *Tsinghua Science&Technology*, vol. 14, no. 5, pp. 606–611, 2009.

[3] H.Dong, J.Tao, and B.Xu, *“Grapheme-to-Phoneme Conversion in Chinese TTS System”. In 2004 International Symposium on Chinese Spoken Language Prtocessing (ISCSLP),* pp. 165-168, 2004.

[4] H.Zhang, J.Yu, W.Zhan and S.Yu, “Disambiguation of Chinese Polyphonic Characters,” in *The First International Workshop on MultiMedia Annotation(MMA2001), vol.1,* pp. 30–1, 2001.

[5] J.Liu, W. Qu, X. Tang, Y.Zhang and Y.Sun, “Polyphone Word Disambiguation with Machine Learning Approaches”, *2010 Fourth international Conference on Genetic and Evolutionary Computing(ICGEC),* pp. 244-247, 2010

[6] 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*, pp.1-5, 2016

[7] K.Park. Novak, N. Minematsu, and K. Hirose, “g2pm: A neural grapheme-to-phoneme conversion package for mandarin Chinese based on a new open benchmark dataset”, *in Interspeech 2020,* pp.1723-1727, 2020

[8] J. Zhang, Y. Zhao, J. Zhu and J.Xiao, “Distant supervision for polyphone disambiguation in mandarin chinese”, *in Interspeech 2020,* pp. 1753-1757, 2020

[9] 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, 2019.

[10] B. Yang, J.Zhong, and S.Liu,“Pre-trained text representations for improving front-end text processing in mandarin text-to-speech synthesis”, *in Interspeech 2019,* pp. 4480-4484, 2019.

[11] H. Zhang, P, Hua, X. Li, “A mask-based model for mandarin Chinese polyphone disambiguation”, *in Interspeech 2020*, pp.1728-1732, 2020.

[12] 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, ” *Interspeech 2019,* pp. 2110-2114, 2019

[13] Y.Zhang, J.Yang, “Chinese NER using lattice LSTM”, *ACL 2018,* pp. 1554-1564, 2018

[14] X.Li, H.Yan, X.Qiu, X.Huang, “FLAT: Chinese NER Using Flat-Lattice Transformer”, *ACL 2020,* pp.6836-6842, 2020

[15] A. Vaswani, N.Shazeer, N.Parmar, J.Uszkoreit, L.Jones, A.N. Gomez, and I.Polosukhin, “Attention is all you need”, *in Advances in Neural Information Processing Systems,* pp. 5998-6008, 2017.

[16] Z.Dai, Z.Yang, Y.Yang, J. Garbonell and Ruslan Salakhutdinov, “Transformer-xl: Attentive language models beyond a fixed-length context” in *CoRR,abs:1901.02860,* 2019