Disambiguation of Chinese Polyphones in an End-to-End Framework with Semantic Features Extracted by Pre-trained BERT

Dongyang Dai[1,2], Zhiyong Wu[1,2,3,], Shiyin Kang[4], Xixin Wu[3], Jia Jia[1,2], Dan Su[4], Dong Yu[4], Helen Meng[1,3]*

Address - Line 1
Address - Line 2
Address - Line 3

`ddy17@mails.tsinghua.edu.cn`, `{zywu,wuxx,hmmeng}@se.cuhk.edu.hk`
`{shiyinkang,dansu,dyu}@tencent.com`, `jjia@tsinghua.edu.cn`

Abstract

Grapheme-to-phoneme (G2P) conversion serves as an essential component in Chinese Mandarin text-to-speech (TTS) system, where polyphone disambiguation is the core issue. In this paper, we propose an end-to-end framework to predict the pronunciation of polyphonic character, which accepts sentence containing polyphonic character as input in the form of Chinese character sequence without the necessity of any preprocessing. The proposed method consists of a pre-trained bidirectional encoder representations from Transformers (BERT) model and a neural network (NN) based classifier. The pre-trained BERT model extracts semantic features from raw Chinese character sequence and the NN based classifier predicts the polyphonic character's pronunciation according to BERT output. To explore the impact of contextual information on polyphone disambiguation, three different classifiers are investigated: a fully-connected network based classifier, a long short-term memory (LSTM) network based classifier and a Transformer block based classifier. Experimental results demonstrate the effectiveness of the proposed end-to-end framework for polyphone disambiguation and the semantic features extracted by BERT can greatly enhance the performance.

Index Terms: polyphone disambiguation, pre-trained BERT, end-to-end framework

1. Introduction

Text-to-speech (TTS) technology has been widely used in voice-assistants, car navigation, e-books and other products. For language based on graphic symbols like Chinese, it is necessary to convert the input character sequence into phoneme sequence before synthesizing speech. Therefore the grapheme-to-phoneme (G2P) conversion component is essential in Mandarin TTS system.

A Chinese character may have multiple corresponding pronunciations, which is called polyphonic character. Polyphone disambiguation which predicts the correct pronunciation of a polyphonic character is the core issue in Chinese G2P conversion. Fig.1 depicts the flow of Chinese G2P conversion. If the input character is not a polyphonic character, we can directly look up the dictionary to derive its pronunciation. Otherwise, we need a polyphone disambiguation model to predict its pronunciation based on its context information.

For Chinese polyphonic characters, their pronunciations are affected by the semantic context information [1] of neighboring
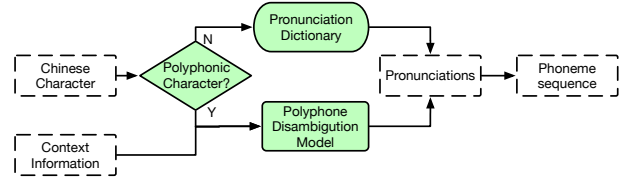
---

* Corresponding author

Figure 1: *Chinese G2P conversion flow*

characters that may occur before or after the polyphonic character with different spans. The earliest Chinese polyphone disambiguation system was based on manual rules [2, 3]. The laws for polyphone disambiguation were summarized by linguistic experts and written into computer-understandable forms. However, as the number of rules increases, a polyphonic character may be matched by multiple conflicting rules at the same time.

As the amount of data increases, more and more researchers tried to use statistical approaches for polyphone disambiguation. Decision trees were applied in [4] to classify the pronunciations of polyphonic characters. A study in [5] has shown that a Maxent model outperforms decision tree. However, these statistical approaches need handcrafted features extracted from sentence containing the polyphonic character as model's input. Feature engineering requires linguistic background knowledge and is expensive.

Deep neural network (DNN), which can learn high-level invariant features from raw data [6], provides an easier way to extract valid features and predict the pronunciation of polyphonic character. Shan et al. [7] addressed the polyphone disambiguation problem as a sequential labeling task, and proposed a bidirectional long short-term memory (BLSTM) approach to predict the pronunciation of polyphonic character which outperforms the max entropy model. This approach encodes the polyphonic character's surrounding observations including neighbor characters and part-of-speech (POS) of neighbor words. Only word tokenization and POS tagging are required in the preprocessing stage in Shan's approach, greatly reducing the work of feature engineering. However, since this model is trained on limited annotation data, it is difficult to learn enough semantic information for polyphone disambiguation. Besides, all the considered polyphonic characters share the same classifier with only one output layer listing the labels of all possible pronunciations, can neither avoid the circumstance of mis-predicting to the pronunciation of another polyphonic character, nor can it handle new polyphonic characters that have not yet appeared without changing the output layer and retraining of the shared
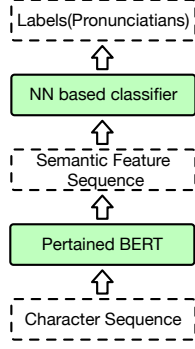
Figure 2: *Model architecture*

classifier.

With the development of neural network research, end-to-end TTS has become a new trend [8, 9, 10]. Towards end-to-end G2P conversion in Chinese Madarin, we propose an end-to-end framework for polyphone disambiguation consisting of a pertained BERT [11] and neural network (NN) based classifier. The advantage of proposed method is summarized as follows:

1. The proposed method predicts pronunciation in an end-to-end way, accepting raw Chinese character sequence containing polyphonic character as input, without the necessity of any preprocessing procedures.

2. A large amount unsupervised data can be adopted to pre-train the model for extracting semantic information, which will boost the performance of polyphone disambiguation.

3. The proposed method uses non-shared output layer among different polyphonic characters, eliminating the case of mis-predicting to pronunciation of other polyphonic characters. Furthermore, with this architecture, when new polyphonic characters are required to be processed, only output layers for these new characters are added and trained without affecting the existing classifiers.

2. The proposed approach

The proposed framework consists of a pre-trained BERT and NN based classifier. Depicted in Fig.2, the pertained BERT extracts semantic features from a raw Chinese character sequence containing polyphonic character, the following NN based classifier predicts polyphonic character's pronunciation according to BERT output. In our research, we explored the performance of classifiers based on fully-connected network, BLSTM and Transformer block respectively.

2.1. The pre-trained BERT

The pre-trained BERT accepts raw Chinese character sequence as input and outputs a sequence of semantic features. The BERT's architecture is shown in Fig.3, the character embedding layer and positional embedding layer process the input character sequence respectively before getting the embedding sequence. The following Transformer blocks convert embedding sequence to semantic feature sequence. Because the use of Transformer [12] and BERT have been ubiquitous, the structure of Transformer block will not be described in detail here.
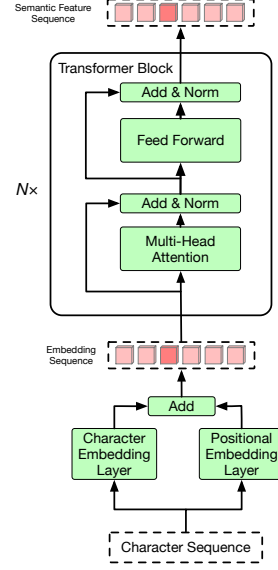

Figure 3: *BERT architecture*

The BERT model is pre-trained on a large amount of unlabeled data with two prediction tasks, predicting the masked input characters and predicting the next sentence. The pre-trained BERT model is expected to learn semantic representions from raw character sequence.

2.2. The NN based classifier

As the pre-trained BERT extracts semantic features from raw character sequence and the pronunciation of a polyphonic character is determined by its contextual semantics, we can directly predict a polyphonic character's pronunciation according to these semantic features. We assume that the polyphonic character is the ith element of the BERT input sequence, the NN based classifier predicts pronunciation based on BERT output and the value of the subscript i. We explored fully-connected network based classifier, LSTM based classifier and Transformer block based classifier in our research.

2.2.1. Fully-connected network based classifer

First of all, we use a two-layer fully-connected network to predict the pronunciation of polyphonic character according to the ith element of the BERT output sequence. The fully-connected network based classifier is depicted in Fig.4 (a). The first fully-connected layer is shared by all the polyphonic characters. As for the second fully-connected (output) layer, it is not shared. Each polyphonic character has a separate output layer whose units number is equal to the number of possible pronunciations. Softmax cross-entropy loss is adopted to train the classifier. The LSTM based classifier and Transformer block based classifier also use the same structure of output layer and loss function.

2.2.2. LSTM based classifer

Indicated by [7], contextual information such as the POS of polyphone's neighbor words can also affect the pronunciation of a polyphonic character. So instead of classifying according to only the ith element of BERT output sequence directly, we apply bidirectional LSTM (BLSTM) to model the contextual information before classifying. The LSTM based classifier is
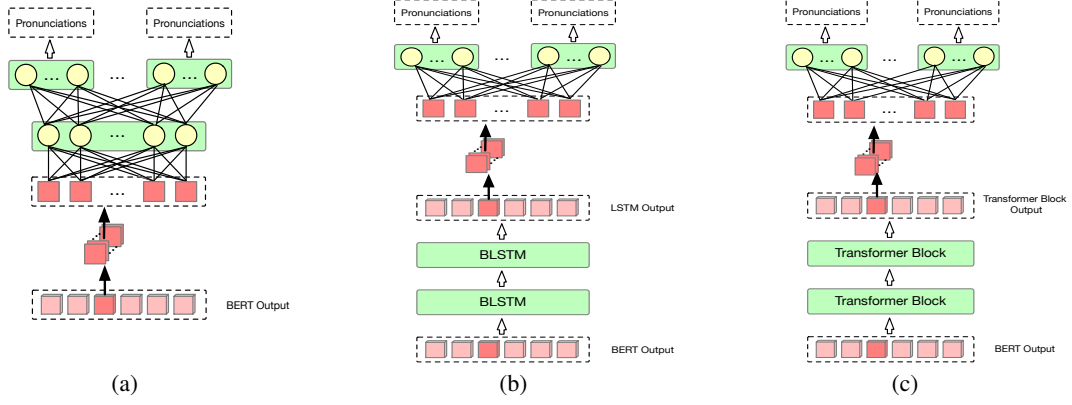
Figure 4: *The NN based classifiers, (a) fully-connected network based classifier, (b) LSTM based classifier, (c) Transformer block based classifier*

shown in Fig.4 (b). The BERT output sequence is processed by a two-layer BLSTM network first to model the contextual information, then a following unshared output layer predicts the pronunciation of corresponding polyphonic character according to the ith element of LSTM output sequence.

2.2.3. Transformer block based classifier

Due to the characteristics of the recurrent network, nearby locations have greater impacts than farther locations. In order to better analyze the impact of context information, we use Transformer block to model context information. From the perspective of model structure, information at any position is equally important in Transformer block. The Transformer block based classifier is depicted in Fig.4 (c). Two-layer Transformer blocks model the contextual information on the BERT output, the following unshared output layer accepts the ith element of Transformer block as input and predicting the pronunciation of corresponding polyphonic character.

3. Experiment and analysis

3.1. Dataset

The experiments were conducted on a dataset extracted from TTS corpus in Tencent AI Lab. There are 331,325 sentences containing polyphonic characters in the corpus. We selected polyphonic characters which appear in more than 2,000 sentences accounting for 83.7% of the total polyphonic samples. In our experiments, the dataset was divided into 10 subsets randomly keeping the distribution of polyphonic characters, 8 subsets were used for training, one subset was used as development set and the remaining subset as test set. We conducted 10-fold cross-validations to get the final average result.

3.2. Settings of baseline approach

We took Shan's LSTM approach for polyphone disambiguation in [7] as baseline (LSTM baseline). The LSTM baseline approach needs word tokenization and POS tagging first on the input character sequence. Then the LSTM based model accepts a character embedding sequence and a contextual POS embedding sequence as input. The character embedding is generated from characters composing the word that contains polyphonic character. The POS sequence considers the neighbor words besides the word containing polyphonic character.

Fig.5 depicts the LSTM baseline approach. The prediction of pronunciation is viewed as a sequence labeling task. The baseline model accepts embedding sequence concatenated by character embedding sequence and POS embedding sequence, and outputs a label sequence corresponding to the input characters. In our experiments, we set the hidden units of BLSTM to 512, the number of BLSTM layers to 2 and the contextual size to 1 when constructing POS embedding sequence, identical to the setting in [7].
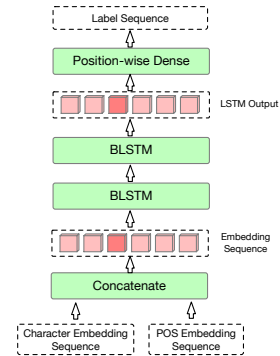

Figure 5: *LSTM baseline approach for polyphone disambiguation*

3.3. Settings of the proposed approach

In our experiments, we adopted the pre-trained BERT model provided by Google to extract semantic features from raw Chinese character sequence[1], and the detail of BERT is identical to the $\text{BERT}_{\text{BASE}}$ model described in [11] whose output size of Transformer block is 768. Due to limited data, fine-tuning the pre-trained BERT model did not achieve desired results. So during the training phase, we froze the parameters of the BERT model and only updated the parameters of the NN based classifier.

In the fully-connected network based classifier, the hidden units of the first fully-connected layer is 512 with a dropout rate of 0.5 during training phase. The hidden units of BLSTM is 512 in LSTM based classifier. In Transformer block based classifier,
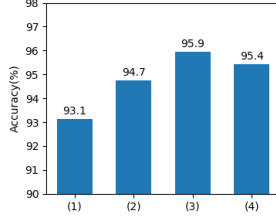
---

[1]https://github.com/google-research/bert

Figure 6: *Experimental results: accuracies of defferent methods, (1) LSTM baseline, (2) BERT + FC, (3) BERT + LSTM, (4) BERT + Transformer block*
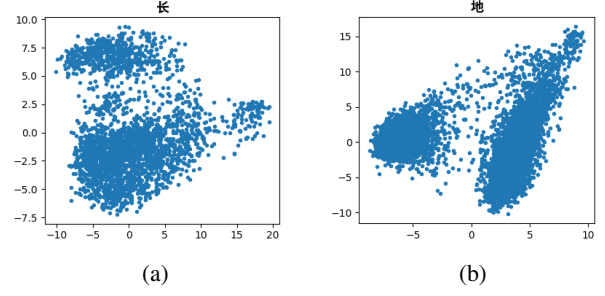

Figure 7: *PCA embedding of semantic features corresponding to typical Chinese character extracted by pre-trained BERT*

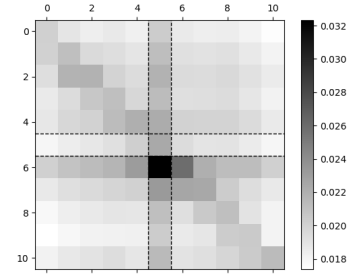Figure 8: *Attention average weights cropped around polyphonic character*

the dimension of Transformer block is 512, and the head number is 8 in multi-head attention which is identical to settings in [12]. During the training phase, we adopted the Adam[14] optimizer and set the learning rate to 5e-4, except the Transformer block based classifier. When training the classifier based on Transformer block, we use the training strategy described in [12].

3.4. Experimental results and analysis

The experimental results are depicted in Fig.6. All our proposed methods based on pretrained BERT outperform the LSTM baseline. Besides, *BERT + LSTM* method and *BERT + Transfomer* method, which model the context of BERT output, have achieved better results than the *BERT + FC* method. In particular, *BERT + LSTM* method achieves the highest prediction accuracy.

Fig.7 shows the PCA embedding of features corresponding to Chinese character "长" and "地" extracted by pre-trained BERT. "长" and "地" are typical Chinese characters with several distinct meanings. "长" mainly has two distinct meanings, which are "grow" (pronounced as "zhang3") and "long"(pronounced as "chang2"), and the PCA embedding in the Fig.7 (a) also mainly merges into two parts. Besides, there are two main usages of Chinese character "地", one stands for the "ground" (pronounced as "di4") and the other is to connect adverbials and predicates (pronounced as "de0"), corresponding to PCA embedding grouped in two parts in Fig.7 (b). Figure 7 illustrates that pre-trained BERT on a large amount of unlabeled data can extract valid semantic features. And supported by experimental results, these semantic features are useful to enhance the performance of pronunciation prediction.

To further illustrate the impact of contextual information for polyphone disambiguation, we draw the attention weights cropped around the polyphonic character. Fig.8 shows the average cropped attention weights of all the heads in the first Transformer block in classifier on test set. The attention weights is cropped around the polyphonic character with neighboring contextual size 5, which means the size of the cropped attention weight is (11, 11) and the location (5, 5) corresponding to the polyphonic character. It can be seen from Fig.8 that the closer to the position of the polyphonic character, the greater the weight of the attention, which means more adjacent contextual features have a greater effect in polyphone disambiguation task. This explains the reason why the LSTM based classifier performs better than the Transformer block based one, LSTM can better preserve adjacent contextal information due to the characteristics of the recurrent network.

4. Conclusions

In this paper, we proposed an end-to-end framework for Chinese polyphone disambiguation. The proposed framework accepts raw Chinese character sequence as input without the necessity of any preprocessing procedures such as word tokenization or POS tagging, which consists of a pre-trained BERT model for extracting semantic features from raw Chinese characters and a NN based classifier for predicting the polyphonic character's pronunciation from BERT output.

To explore the impact of contextual information on polyphone disambiguation, three different NN based classifiers are investigated and compared with the LSTM baseline approach [7]. Experimental results demonstrate the effectiveness of the proposed end-to-end framework and that the BERT model pretrained on a large amount of unsupervised data can effectively extract semantic features, which greatly enhances the performance of polyphone disambiguation. Meanwhile, the contextual information can also improve the result of polyphone disambiguation, especially the closer the context is to the polyphonic character, the greater its influence on polyphone disambiguation.

5. References

[1] L. Li, "The status quo of multi-syllable words and regular method of recognizing," *Economic and social development*, vol. 8, no. 7, pp. 125–128, 2010.

[2] D. Zhang, "Research and implementation of key techniques of chinese language switching system hj-tts," Ph.D. dissertation, Graduate School of the Chinese Academy of Sciences (Institute of Computing Technology), 2000.

[3] L. Cai, H. Wei, and X. Zhou, "Linguistic processing in chi-

nese text-to-speech conversion," *Chinese Journal of Information*, vol. 9, no. 1, pp. 31–36, 1995.

[4] W. Wang, S. Hwang, and S. Chen, "The broad study of homograph disambiguity for mandarin speech synthesis," in *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96*, vol. 3. IEEE, 1996, pp. 1389–1392.

[5] F. Liu, Q. Shi, and J. Tao, "Maximum entropy based homograph disambiguation," in *The 9th National Conference on Man-Machine Speech Communication*, 2007.

[6] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[7] 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.

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

[9] 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.

[10] N. Li, S. Liu, Y. Liu, S. Zhao, M. Liu, and M. Zhou, "Close to human quality tts with transformer," *arXiv preprint arXiv:1809.08895*, 2018.

[11] 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.

[12] 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.

[13] S. Li, Z. Zhao, R. Hu, W. Li, T. Liu, and X. Du, "Analogical reasoning on chinese morphological and semantic relations," *arXiv preprint arXiv:1805.06504*, 2018.

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