ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-6654-0540-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICASSP43922.2022.9746446

DATA AUGMENTATION FOR LONG-TAILED AND IMBALANCED POLYPHONE DISAMBIGUATION IN MANDARIN

Yang Zhang , Haitong Zhang* , Yue Lin*

NetEase Games AI Lab, China
{zhangyang09,zhanghaitong01,gzlinyue}@corp.netease.com

ABSTRACT

Polyphone disambiguation is an important module in Mandarin Chinese text-to-speech (TTS). Recently, neural-network-based (NN-based) models have achieved a great improvement on polyphone disambiguation. However, a long-tailed and imbalanced distribution is usually observed in the training data of polyphone disambiguation, resulting in an unsatisfying performance on the low-frequent polyphone in the imbalanced pinyin set, and the least-frequent polyphonic characters and polyphones. In this paper, we proposed a simple data-augmentation method based on the pre-trained mask language model BERT to mitigate the long-tailed and imbalanced distribution problem. We incorporate a weighted sampling technique in the data augmentation method to balance the data distribution, and a useful filtering strategy to remove some noisy augmented data. Experimental results show that the proposed data-augmentation method can improve the prediction accuracy, especially for those low-frequent polyphone in the imbalanced pinyin set, and the least-frequent polyphonic characters and polyphones.

Index Terms— Polyphone disambiguation, long-tailed and imbalanced distribution, data-augmentation

1. INTRODUCTION

Grapheme-to-phoneme (G2P) conversion is an essential component in text-to-speech synthesis, which typically generates a sequence of phones given a series of characters or graphemes [1]. As in building Chinese G2P conversion system, the major challenge lies in how to disambiguate the pronunciation of polyphones, i.e., characters having more than one pronunciation.

Traditional approaches for polyphone disambiguation are rule-based algorithms [2, 3] and statistical machine learning methods [4, 5]. Recently, deep neutral network (DNN) is successfully applied in polyphone disambiguation, and has been proved to achieve a great performance improvement [6–8]. Especially, state-of-the-art results are made by employing large pre-trained language models such as BERT [9].

However, two problems still exist in practice. First, the performance of lightweight G2P system still needs to be improved, because in real-time text-to-speech BERT based models are too heavy and time costing. Furthermore, large and balanced training data for polyphone disambiguation are hard to obtain, since the natural distribution of polyphones in Chinese text follows long-tail effect. The accuracy of those low frequent characters or pronunciations is still unsatisfied, although the overall performance of the dataset seems high enough.

* Equal contribution

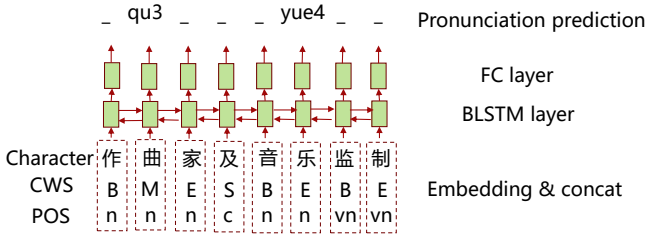

Fig. 1. The model structure with an example of input features.

1.1. Main Contribution

In this paper, we propose to improve polyphone disambiguation under long-tailed dataset. We use a lightweight structure instead of BERT for model construction to match practical use, while carefully designed data augmentation methods are used to improve the model performance on small amount of labelled data. The contribution of this work can be summarized in three aspects:

- We proposed a data augmentation method based on BERT language model for polyphone disambiguation in Mandarin.
- In addition, we proposed a weighted sampling strategy in the proposed data augmentation method to solve the long-tailed and imbalanced data distribution problem, and a novel filter strategy to filter the noisy augmented data.
- We conducted extensive and statistically significant experiments to verify the effectiveness of our method.

2. THE LIGHTWEIGHT MODEL

In real-time text-to-speech (TTS) application, such as human-robot conversation, there is a strict requirement of response latency (usually less than 200ms) to satisfy the user experience. As a text preprocess module of TTS, G2P module therefore needs to run very fast and be as lightweight as possible for on-device deployment.

In this work, we test the proposed augmentation method using a lightweight model structure from a practical view.

2.1. Input Features

According to the previous research [6], we used features including Chinese character (CC), Chinese word segmentation (CWS) , and Part-of-speech tagging (POS). Besides, there is a boolean token input , which indicates whether the character is polyphonic or not. It

Algorithm 1 Data Augmentation approach

Input:
Training dataset D_{train}
Augmentation constant C
Pre-trained BERT model G ;
Output:
Augmented dataset D_{Aug};

Function `AugWithBERT(`x , K, G`)`**:**
$x_{Aug} \leftarrow \{\}$;
for each token j in all non-polyphonic characters:
Mask the token j in x with [MASK] to format the input to masked language model G;
Use G to predict top K candidate replacement token.
for each R in K candidates:
$x_{new} \leftarrow$ Replace token j of x with R;
$x_{Aug} \leftarrow x_{Aug} \cup x_{new}$;
return x_{Aug}
End Function

$D_{Aug} \leftarrow \{\}$;
Get the number of occurrences $t_{c,p}$ of every tuple (polyphone c, pronunciation p) in D_{train}.
for each sentence $x \in D_{train}$ **do**
Get the augmentation scale S^* for x according to Eq. 3
Get the minimum $MaxRep$ for x according to Eq. 2
$x^0_{Aug} \leftarrow \{x\}$
for i in range(0, MaxRep-1, 1) **do**
$x^{i+1}_{Aug} \leftarrow \{\}$;
for each sentence x_i in x^i_{aug} **do**
$x^{i+1}_{aug} \leftarrow x^{i+1}_{aug} \cup$ AugWithBERT(x_i, K, G)
end for
end for
$x_{aug} \leftarrow \cup_{i=1}^{MaxRep} x^i_{aug}$
Sample S^* sentences from x_{aug} as x^*_{aug}
$D_{Aug} \leftarrow D_{Aug} \cup x^*_{aug}$
end for

also eliminates the contribution of non-polyphonic characters to the optimization loss.

In the example sentence "作曲家及音乐监制" (composer and music producer), the polyphonic characters are "曲" and "乐". In polyphone disambiguation, only polyphonic characters are labeled as their pronunciation, while non-polyphonic characters are labeled as "NULL". CWS and POS are got from external tools, CWS is labelled as {B,M,E,S} as commonly used in CWS sequence labelling, and POS tag is labelled with tags in Chinese POS set.

2.2. Model structure

The model structure is illustrated as Fig. 1. Four input features are first embedded and concatenated as inputs. All features are embedded using randomly initialized embedding matrix except Chinese characters. Pre-trained and fixed character embedding are used, which contain semantic information but do not lead to more computation such as BERT embedding. The whole model consists of one Bidirectional Long Short-Term Memory (BLSTM) layer and one fully-connected (FC) layer, which is simple but efficient.

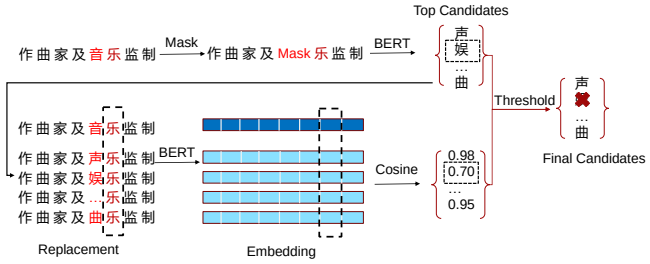

Fig. 2. An example of filtering strategy for words with polyphonic character in the proposed BERT-based data augmentation.

3. DATA AUGMENTATION

Data augmentation is proved to be promising in improving model performance on low resource data, and has been successfully applied in deep learning applications, such as automatic speech recognition [10, 11] and natural language processing [12, 13]. One fundamental idea is to create the plausible transformation of original data while preserving the label information. Inspired by this idea, we proposed an augmentation method for polyphone disambiguation, which is based on pre-trained language model BERT.

3.1. BERT Based Data Augmentation

BERT [14], is a transformer based language model pretrained with masked language modelling. With huge unlabelled data pretraining, BERT can capture rich contextual information, that BERT can give a reasonable prediction of masked words for any sentence. We make use of this ability of BERT, and replace non-polyphonic characters in original training sentences to form augmented data, while the label of polyphone is remained. Note that using BERT for augmentation won't increase any time in inference.

3.2. Weighted sampling

In general, the frequency of characters in the Chinese text is in an uneven distribution, so is the frequency distribution of pronunciation. This phenomenon leads to a very unbalanced training data of polyphone disambiguation, which is also observed in [8, 15]. The imbalanced distribution of our dataset will be discussed in the experimental part.

With this unbalanced data, G2P model may have a low accuracy on those low frequent characters or pronunciations, which is the main problem of mandarin G2P system. To address the problem, we added weighted sampling in our augmentation process, that is to control the augmentation scale of each training sentence.

For a sentence, we have three parameters relate to the augmentation scale: number of multiple replaced positions (N), number of allowed positions that can be replaced (M), number of top candidates for replacing one position (K). Then one sentence will be augmented S times following Equation 1.

$$S \sim \frac{M!}{(M-N)!} K^N \tag{1}$$

In which M is no more than the number of non-polyphonic characters in that sentence, K is set to a small number to avoid unreasonable substitution candidates. Multiple replacement N is restricted as small as possible, since masking more positions will lose more context, which result in lower generation quality. Iterative replacement

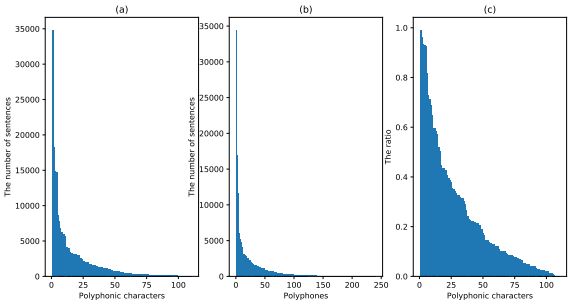

Fig. 3. Long-tailed and imbalanced distribution in the training dataset. (a) The frequency distribution of polyphonic characters; (b) the frequency distribution of polyphones; (c) the ratio between the least frequent polyphone and the most frequent polyphone of each polyphonic character.

in our algorithm may partly ease that problem. Considering different N, the total augmented scale S is

$$S \sim \sum_{N=1}^{MaxRep} \left(\frac{M!}{(M-N)!}K^N\right) \tag{2}$$

The aim of weighted sampling is to make a large and balanced dataset for each polyphone. For simplicity, the amount of every pronunciation in each polyphone need to be close to a given constant C. Let $t_{c,p}$ be the frequency of tuple (character c, pronunciation p), the augmentation scale needed for sentence x is determined by

$$S^* = \max_{(c,p)\in x} (C/t_{c,p}) \tag{3}$$

Then weighted sampling is done by choosing the minimum $MaxRep$ that makes $S \geq S^*$, and sample S^* generated sentences as the final augmentation data of x. The pseudo-code of that is show in Algorithm 1.

3.3. Filtering Strategy

Although BERT can generate meaningful sentences, it does not guarantee that pronunciation being unchanged. In fact, it is observed that replacing the character(s) close to the polyphonic character may change the pronunciation label, especially when these character(s) and the polyphonic character are in the same word. Besides, in most situations, the pronunciation of a polyphonic character is related to its meaning in the context, which can also be represented by the embedding of BERT. Thus, we can compare the BERT embedding for the target polyphone character before and after replacing its neighbor non-polyphonic characters, and filter those candidates with cosine similarity lower than a threshold. Fig. 2 illustrates this filtering process.

4. EXPERIMENT

4.1. Dataset

To verify the proposed data augmentation methods, we conducted experiments on an internal dataset. The dataset contains 110 frequently used polyphonic characters, and includes 100,000 sentences as training set and 20,000 sentences as testing set. Each sentence contains at least one target polyphonic character.

Table 1. Results of the testing data.

Model	Accuracy(%)
BM	94.92
BM-US	93.26
BM-OS	94.79
SDA [16]	94.37
BDA	95.24
BDA-W	95.20
BDA-WF (Proposed)	**96.71**

Table 2. The accuracy (%) of polyphonic characters, where (Train/Test) denotes the frequency in the training/testing data.

Char	Polyphone (Train/Test)	BM	BM-OS	BDA-WF
数	shu4 (6115/338)	99.11	**99.41**	**99.41**
	shu3 (133/40)	65.00	55.00	**67.50**
中	zhong1 (34510/2874)	**99.79**	**99.79**	**99.79**
	zhong4 (378/75)	58.67	58.67	**73.33**
得	de2 (5275/2874)	94.15	93.75	**95.16**
	dei3 (59/80)	58.75	32.50	**73.75**

As discussed above, the Chinese text usually has an imbalanced distribution in characters and pronunciations. Fig. 3 provides a visualization of the frequency distribution of polyphonic characters, the frequency distribution of polyphones, and the ratio between the most frequent polyphone and the least frequent polyphone of each polyphonic character. As shown in the figure, our dataset also severely suffer from the long-tailed and imbalanced distribution problem. The most frequent character occurs in more than 30 thousand sentences while the least frequent one occurs less than 10 times (see Fig. 3 (a)). Similarly, the frequency of the most common polyphone is over 30000, while that of the least common ones is less than 10 (see Fig. 3 (b)). According to Fig. 3 (c), about half of polyphonic characters has a ratio below 0.2. Namely, the frequency of the most-frequent polyphone is more than five times of that of the least-frequent polyphone in 50% of the polyphonic characters.

4.2. Experimental setup

To verify our proposed method, we trained the following model variants:

- BM: Baseline model trained using the original data;
- BM-US: Baseline model trained using the original data with undersampling;
- BM-OS: Baseline model trained using the original data with oversampling;
- SDA (Similar word data augmentation): Model trained with the original data and the augmented data as [16]
- BDA (BERT data augmentation): Model trained with the original data and the BERT-based augmented data;
- BDA-W: BDA with weighted sampling in data augmentation;
- BDA-F: BDA with filtering strategy in data augmentation;
- BDA-WF: BDA with weighted sampling and filtering strategy in data augmentation.

By undersampling, we set the upper limit on the frequency of each (character, pronunciation) tuple into 500. By oversampling, we set the frequency of each tuple into 10000. We also implemented

Table 3. The accuracy (%) of the least-frequent-char testing data.

Char	BM	BDA	BDA-F	BDA-W	BDA-WF
朴	87.75	87.18	88.01	89.14	**89.84**
荫	89.77	88.47	89.77	88.06	**90.57**
喝	92.52	92.63	92.52	91.75	**96.92**
肖	95.54	93.77	93.35	93.35	**97.73**
		...			
Total	92.95	92.42	92.40	92.28	**94.10**

Table 4. The accuracy (%) of the least-frequent-polyphone testing data, where PP means polyphone.

Char(PP)	BM	BDA	BDA-F	BDA-W	BDA-WF
系(ji4)	27.27	23.23	27.23	38.38	**53.54**
扎(za1)	53.33	35.56	52.33	44.44	**66.67**
泡(pao1)	37.25	60.78	60.78	53.92	**76.47**
禅(shan4)	38.84	57.90	61.40	61.40	**82.46**
		...			
Total	56.53	53.59	56.53	57.07	**69.34**

the data augmentation method proposed in [16] for comparisons, in which all words in the sentence have an equal probability to be swapped by a similar word using cosine similarity of their word embedding. We used Tencent AI Lab Word Embedding [17] in this method.

We set the number of the BLSTM layer to 1 and the hidden size to 256, and the hidden unit of the FC layer to 256. We used a 128-dimension word2vec trained on a Chinese Wikipedia corpus for the character embedding. For Chinese word segmentation and part-of-speech tagging, we used an open-source tool called Jieba [1]. The embedding size of CWS and POS feature are 16 and 64, respectively. In the experiments, we set the hyper-parameter C, K in Algorithm 1 to 10,000, 10. The cosine similarity threshold to filter augmented data is set to 0.9.

The training loss is cross-entropy over all possible pronunciation labels. We trained the model using Adam optimizer with an initial learning rate of 0.01 and a batch size of 128. We trained each model until convergence.

4.3. Results on all testing data

As shown in Table 1, with undersampling, the model performance gets worse while oversampling provides a comparable performance to the baseline model. Model SDA under-performs over the baseline model slightly. We speculate that simply replacing with similar words does not ensure the semantic soundness of the augmented sentences, which may bright some noise to the model.

It is clearly shown that models trained with the proposed augmented data outperform the baseline model. Specifically, model BDA-WF achieve an accuracy of over 96%, with nearly 1.8% absolute improvement over the baseline model. In addition, we find that our proposed filtering strategy is useful because using the BERT-based data augmentation without filtering some noisy data points only provides comparable performances to the baseline model (See BDA and BDA-W).

[1]https://github.com/fxsjy/jieba

4.4. Impacts on the imbalanced data

To look into the impact on the imbalanced cases in the testing data, we collected the accuracy of several polyphones suffered from imbalanced distribution mentioned in section 4.1 and Fig. 3 (c). The accuracy from the system BM, BM-OS and BDA-WF are listed as Table 2.

The experimental results revealed that the proposed method is highly conductive to mitigate the negative impact of imbalanced distribution within Pinyin set. The accuracy of "shu3" in BDA-WF is 2.5% higher than that of BM and 12.5% higher than that of BM-OS. As the case of 'zhong4', BDA-WF is 14% higher than that of system BM and BM-OS. Besides, system BDA-WF got a slightly improvement compared to the other systems on massive examples such as "de2". It indicates that the proposed method can improve competency of the model in classifying rare of the hard examples without harming that of the massive examples.

4.5. Impacts on the long-tailed data

We further investigated the impact of the proposed data-augmentation methods on the long-tailed data (see Fig.3 (a) and (b)). Since the number of the least frequent polyphonic characters and polyphones in the original test dataset is small, the results would become less statistically significant. Thus, we then constructed two new test datasets specifically designed for these cases. The first dataset (the least-frequent-char testing data) consists of 20k sentences and each sentence contains at least one of top-20 least-frequent polyphonic characters (with no more than 200 cases for each character in the original training data). Each top-20 least-frequent polyphonic character has at least 500 testing cases. The second dataset (the least-frequent-polyphone testing data) includes 1.5k sentences in which there is at least one of top-20 least-frequent polyphones (with no more than 10 cases for each polyphones in the original training data). Each top-20 least-frequent polyphone has at least 40 testing cases.

The results of the least-frequent-char testing data and the least-frequent-polyphone testing data are provided in Table 3 and Table 4, respectively. Regarding the least frequent characters, the accuracy of the proposed model is 1.1% higher than that of the baseline model, while the least-frequent-polyphone testing data is concerned, 12% accuracy improvement is achieved. We also do the ablation study for the proposed method, as shown in Table 3 and Table 4, the accuracy of BDA, BDA-F, BDA-W are all comparable with the baseline, which indicates that the proposed BDA-WF, i.e., the combination of weighted sampling and filtering strategy, is the best solution to improve BDA in long-tailed and imbalanced settings.

5. CONCLUSION

In this paper, we proposed a data augmentation method for improving the performance of long-tailed and imbalanced polyphone disambiguation in Mandarin. With only 100000 training sentences, the proposed method achieves an accuracy rate at 96.71%, a 1.8% improvement compared with that of the baseline model. The experimental results demonstrate that the proposed data augmentation method can ease the distribution imbalance of pinyin set. For the long-tailed distribution, the proposed method achieves an accuracy rate at 94.10% and 69.34%, with 1.1% and 12.8% improvement over the baseline model. In addition, the proposed method also significantly outperforms the oversampling strategy, which is a common practice to mitigate the long-tailed distribution problem.

6. REFERENCES

[1] Lifu Yi, Jian Li, Jie Hao, and Ziyu Xiong, "Improved grapheme-to-phoneme conversion for mandarin tts," *Tsinghua Science & Technology*, vol. 14, no. 5, pp. 606–611, 2009.

[2] Zi-Rong Zhang, Min Chu, and Eric Chang, "An efficient way to learn rules for grapheme-to-phoneme conversion in chinese," in *International Symposium on Chinese Spoken Language Processing*, 2002.

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

[4] Hong Zhang, JiangSheng Yu, WeiDong Zhan, and Shiwen Yu, "Disambiguation of chinese polyphonic characters," in *The First International Workshop on MultiMedia Annotation (MMA2001)*, 2001, vol. 1, pp. 30–1.

[5] Jinke Liu, Weiguang Qu, Xuri Tang, Yizhe Zhang, and Yuxia Sun, "Polyphonic word disambiguation with machine learning approaches," in *2010 Fourth International Conference on Genetic and Evolutionary Computing*. IEEE, 2010, pp. 244–247.

[6] Changhao Shan, Lei Xie, and Kaisheng 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.

[7] Zexin Cai, Yaogen Yang, C. Zhang, Xiaoyi Qin, and Ming Li, "Polyphone disambiguation for mandarin chinese using conditional neural network with multi-level embedding features," in *INTERSPEECH*, 2019.

[8] Haiteng Zhang, Huashan Pan, and Xiulin Li, "A mask-based model for mandarin chinese polyphone disambiguation," *Proc. Interspeech 2020*, pp. 1728–1732, 2020.

[9] Junjie Li, Zhiyu Zhang, Minchuan Chen, Jun Ma, Shaojun Wang, and Jing Xiao, "Improving polyphone disambiguation for mandarin chinese by combining mix-pooling strategy and window-based attention," *Proc. Interspeech 2021*, pp. 4104–4108, 2021.

[10] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin Dogus Cubuk, and Quoc V. Le, "Specaugment: A simple augmentation method for automatic speech recognition," in *INTERSPEECH*, 2019.

[11] Ilyes Rebai, Yessine BenAyed, Walid Mahdi, and Jean-Pierre Lorré, "Improving speech recognition using data augmentation and acoustic model fusion," *Procedia Computer Science*, vol. 112, pp. 316–322, 2017.

[12] Rico Sennrich, Barry Haddow, and Alexandra Birch, "Improving neural machine translation models with monolingual data," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany, Aug. 2016, pp. 86–96, Association for Computational Linguistics.

[13] Jason Wei and Kai Zou, "EDA: Easy data augmentation techniques for boosting performance on text classification tasks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China, Nov. 2019, pp. 6382–6388, Association for Computational Linguistics.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota, June 2019, pp. 4171–4186, Association for Computational Linguistics.

[15] Kyubyong Park and Seanie 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.

[16] Yi Shi, Congyi Wang, Yu Chen, and Bin Wang, "Polyphone disambiguition in mandarin chinese with semi-supervised learning," *arXiv preprint arXiv:2102.00621*, 2021.

[17] Yan Song, Shuming Shi, Jing Li, and Haisong Zhang, "Directional skip-gram: Explicitly distinguishing left and right context for word embeddings," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 2018, pp. 175–180.