The 33rd Conference on Computational Linguistics and Speech Processing (ROCLING 2021)
Taoyuan, Taiwan, October 15-16, 2021. The Association for Computational Linguistics and Chinese Language Processing

Integrated Semantic and Phonetic Post-correction for Chinese Speech Recognition

Yi-Chang Chen
E.SUN Financial Holding Co., Ltd.
ycc.tw.email@gmail.com

Chun-Yen Cheng
E.SUN Financial Holding Co., Ltd.
quadratic999@gmail.com

Chien-An Chen
E.SUN Financial Holding Co., Ltd.
lukechen419@gmail.com

Ming-Chieh Sung
E.SUN Financial Holding Co., Ltd.
mingchieh-17908@email.esunbank.com.tw

Yi-Ren Yeh
Department of Mathematics
National Kaohsiung Normal University
yryeh@nknu.edu.tw

Abstract

Due to the recent advances of natural language processing, several works have applied the pre-trained masked language model (MLM) of BERT to the post-correction of speech recognition. However, existing pre-trained models only consider the semantic correction while the phonetic features of words is neglected. The semantic-only post-correction will consequently decrease the performance since homophonic errors are fairly common in Chinese ASR. In this paper, we proposed a novel approach to collectively exploit the contextualized representation and the phonetic information between the error and its replacing candidates to alleviate the error rate of Chinese ASR. Our experiment results on real world speech recognition datasets showed that our proposed method has evidently lower CER than the baseline model, which utilized a pre-trained BERT MLM as the corrector.

Keywords: language error correction, masked language modeling, phonetic distance

1 Introduction

A variety of real-world applications have been benefited from the recent advances of Automatic speech recognition (ASR), such as voice-activated banking, meeting minutes transcription, and voice content inspection. In ASR, hidden Markov model (HMM) based models (Rabiner and Juang, 1986; Rabiner, 1989; Povey et al., 2011) and end-to-end models (Chan et al., 2016; Bahdanau et al., 2016; Graves, 2012; Jaitly et al., 2016) are two popular types of modeling methods. For end-to-end models, it typically requires a huge amount of data for the model training due to the complicated architectures of neural networks. However, it is not easy to collect sufficient voice data in many real-world scenarios.

In contrast to end-to-end models, conventional HMM-based models, such as Kaldi (Povey et al., 2011), require less data and are quite popular in practice. HMM-based models are comprised of the acoustic model and language model. The acoustic model is used to produce phonetic units from the speech signals. Language models are responsible for obtaining the probabilities of next words by given past words. Typically the N-gram model is used as the language model in HMM-based models. One drawback of the N-gram model is the lack of long-term contextual clues by comparing with RNN-based or transformer-based language models.

For Chinese speech recognition, we found that many homo-phonic errors are produced in HMM-based models with the N-gram model. It shows that the näive N-gram model might sacrifice the performance of HMM-based models even a well-trained acoustic model is given. However, it is not easy to replace the N-gram model due to the structure of interaction between the acoustic model and language model within HMM-based models. To overcome this problem, many methods have been proposed for the post-correction of speech recognition (Kumar et al., 2017; Xie et al., 2016; Guo et al.,

2019; Liu et al., 2013; Zhang et al., 2020).

Recently, many successful methods have been proposed in natural language processing, such as BERT (Devlin et al., 2019). For those pretraining tasks in BERT, masked language modeling (MLM) is a task of interest for our post-correction. The goal of MLM is to predict those masked tokens within a sentence in which certain input tokens are randomly masked. The prediction of masked tokens can be regarded as a kind of error correction. As shown in (Devlin et al., 2019), MLM also could be applied as a post-correction for speech recognition. To be more precise, we apply the fine-tuned BERT to detect the errors within a recognized sentence from ASR. Followed by the detection, MLM is applied to correct these words.

The post-correction by MLM could reduce the deficiency of long-term contextual information in the N-gram model. However, the conventional MLM did not take the phoneme into account. To address this issue, we aim to propose a phonetic MLM as the post-correction for speech recognition by leveraging the phoneme information from the predicted words.

2 Related Work

Many methods have been proposed for correcting the outputs of ASR systems (Errattahi et al., 2018). These existing approaches of language correction typically can be divided into three categories. The first group of them uses external language models to rescore k-best candidates in ASR system. For example, (Kumar et al., 2017) picks k-best candidates of each word from the original ASR system. Once these k-best candidates are determined, RNN-LM is applied to re-score the k-best candidates of each word. From (Kumar et al., 2017), it also shows that the improved performance can be achieved since RNN-LM is a more effective model for the representation of natural languages.

The second category of language correction methods adopts the sequence to sequence learning framework (Sutskever et al., 2014). Based on this architecture, (Xie et al., 2016) adopts a character-based attention mechanism to generate a corrected sentence. On the other hand, (Guo et al., 2019) also proposes a RNN with attention to correct the output from Listen, Attend, and Spell (LAS) model.

The third group of language correction methods adopts a two-step correction. For example, (Liu et al., 2013) uses the language model and statistical machine translation model to detect error words in a sentence. After the error detection, SVM is used to replace the predicted error words with the most likely word. In (Zhang et al., 2020), the authors proposed a bi-GRU model as the error detection network. Given a sequence of embeddings from BERT, the detection networks generate the probability of being an incorrect word. Followed by the detection network, the input of the correction model is the convex combination of mask token embedding and token embedding with the probability of incorrectness. Once the integrated embedding is calculated, a sequential multi-class labeling model based on BERT is applied to generate the corrected sentence.

3 Methodology

In our proposed method, we integrate semantic and phonetic information for the post-correction of ASR. More specifically, the mask language model (MLM) based on BERT is used for semantic error correction. Besides, we also apply a phonetic distance to re-rank the candidates of being corrected from MLM. The details will be addressed in Section 3.1 and Section 3.2 respectively.

3.1 Semantic Post-correction by MLM

In our work, we first apply a token classifier to detect the errors within a recognized sentence from ASR. To learn the binary classifier, we regard the incorrect words within a sentence as the positive examples and fine-tune the model with Chinese pre-trained BERT. Followed by the detection, MLM is applied to correct these words. MLM is one of the pre-training tasks of BERT and originally aims to predict those masked tokens within a sentence in which certain input tokens are randomly masked. In the original design for the pre-training BERT, MLM predicts all masked tokens (i.e., the error words in our task) in a sentence simultaneously as shown in Figure 1(a). That is, the

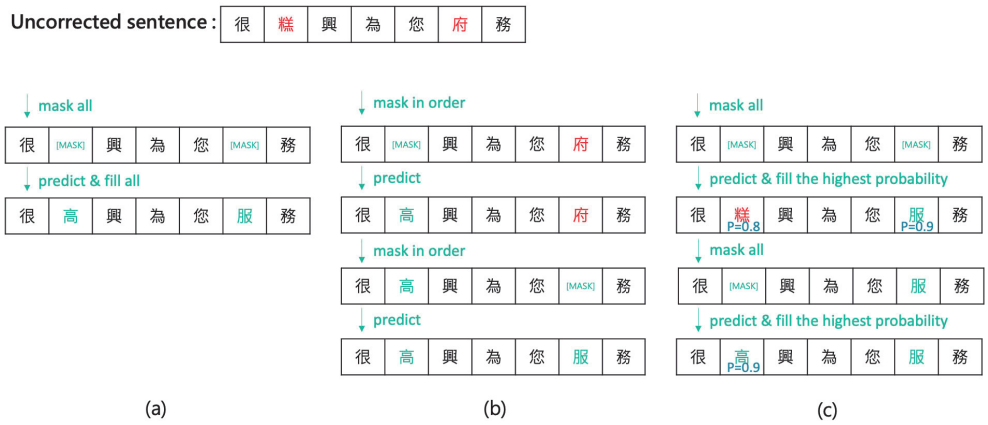

Figure 1: Different masking and replacement strategies of MLM for post-correction: (a) mask-all-and-replace-all, (b) mask-one-and-replace-one, and (c) mask-all-and-replace-one.

mask-all-and-replace-all strategy applied the error token classifier to detect all candidates of incorrect words. Once the detected error words are determined by the token classifier, we replace all of them by the "[MASK]" token and predict the correct words by MLM at the same time.

In addition to the mask-all-and-replace-all strategy, we also propose two other strategies to investigate the influence of the sequential masking and replacement of the detected error words. Different to mask-all-and-replace-all, our first strategy, mask-one-and-replace-one as shown in Figure 1(b), applies MLM to predict the correct words for each error token sequentially from left to right after the positions of error tokens are determined.

Similar to mask-all-and-replace-all, our second strategy, mask-all-and-replace-one, also masks all the candidates at the beginning. Rather than replace all the candidates at once, only one candidate associated with the highest probability will be replaced at one time as shown in Figure 1(c).

Based on the strategies mentioned above, the edited sentence will go through the same process all over again until all detected error words has been corrected. In our experiments, we also evaluate the performance of using these different strategies. The detailed results will be discussed in Section 4.1.

3.2 Phonetic MLM for Post-correction

Using conventional MLM as post-correction of speech recognition only takes the semantic context into account. As the example recognized sentences shown in Figure 2, we found that many homo-phonic errors of correction are made in HMM-based models with the N-gram language model. To overcome this problem, we proposed a phonetic MLM by leveraging the phonetic distance to integrate semantic and phonetic information for the post-correction.

In our proposed framework as shown in Figure 2, we first apply the fine-tuned BERT of token classification to detect the positions of errors. Once the errors are determined, we simply mask them and apply MLM to get the probabilities of candidates denoted by $P_{candidate}$. As the example in Figure 2, we first detect the error "糕" in the recognized sentence, and then "糕" is replaced by "[MASK]". After masking "糕", our MLM will predict candidates of replacement, such as "有", "高", and "羔", with the corresponding probabilities 0.4, 0.2, and 0.1 respectively.

In addition to the semantic correction by the conventional MLM, we also take the phonetic information into account. To obtain the phonetic information, we apply DIMSIM (Li et al., 2018) to obtain the Chinese phonetic distance. In DIMSIM, each pronunciation of Chinese characters is encoded in a high dimensional space. The phonetic distance S between Chinese characters c and c' is defined as follows:

$$S(c, c') = S_p(p_c^I, p_{c'}^I) + S_p(p_c^F, p_{c'}^F) + S_T(p_c^T, p_{c'}^T), \quad (1)$$

where p_c^I, p_c^F and p_c^T represent the initial, final, and tone components of c in Pinyin, re-

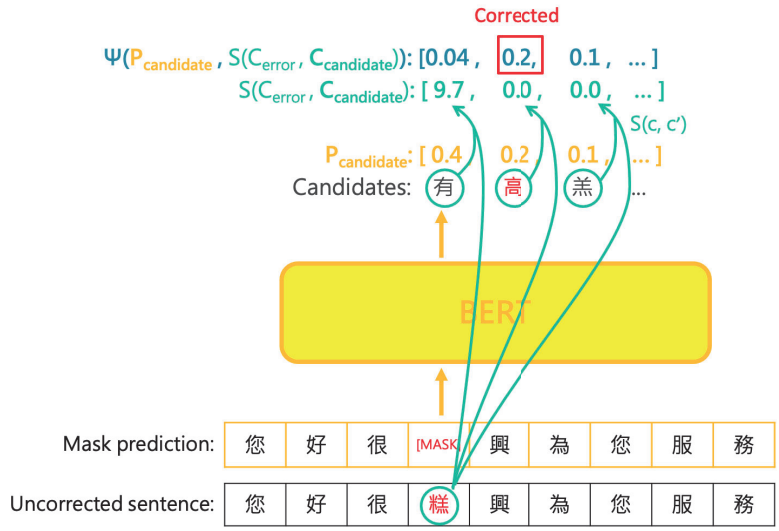

Figure 2: An example of our proposed semantic and phonetic post-correction. $P_{candidate}$ is the probabilities of candidates from MLM. $S(c_{error}, c_{candidate})$ is the the phonetic distances between the detected error character of interest (c_{error}) and the candidates ($c_{candidate}$) based on (1). $\Psi(\cdot,\cdot)$ controls the trade-off between semantic and phonetic metrics as defined in (2).

spectively. S_p and S_T are denoted as the Euclidean distance and phonetic tone distance between c and c', respectively. We note that the phonetic distance S between two homophonic characters is 0, and the phonetic distance $S(c, c') \geq 0$. In (1), by given two Chinese characters, the phonetic distance will be larger while the phonic difference is more significant.

Based on (1), we could calculate the phonetic distances between the detected error character of interest (c_{error}) and the candidates ($c_{candidate}$) of replacing c_{error} by $S(c_{error}, c_{candidate})$. For example, we will calculate S(“糕”,“有”), S(“糕”,“高”), and S(“糕”,“羔”) as their phonetic distances in Figure 2. To consider the semantic correction and phonetic distance for the selection of candidates simultaneously, we first estimate $P_{candidate}$ of all candidates by MLM. Once $P_{candidate}$ and $S(c_{error}, c_{candidate})$ are obtained, we balance these two metrics by the function Ψ as follows:

$$\Psi(P_{candidate}, S(c_{error}, c_{candidate})) = P_{candidate} \times exp(-\alpha \times S(c_{error}, c_{candidate})), \quad (2)$$

where α is a positive number that controls the trade-off between semantic and phonetic information. In our experiments, this hyperparameter is determined by grid search with a validation set. As the example in Figure 2, given the error of interest (i,e., “糕”), S(“糕”,“有”), S(“糕”,“高”), and S(“糕”,“羔”) are calculated as 9.7, 0.0, and 0.0 by (1), respectively. For the correction, we use (2) to obtain the final scores 0.04, 0.2, and 0.1 for “有”, “高”, and “羔”, respectively. Based on the scores from (2), we chose the character with the highest score as the replacement (i.e., “高” in Figure 2).

4 Experiments

Different to the conventional typo correction, we aim to correct the error after ASR in this work. To obtain the results of ASR, we use Kaldi (Povey et al., 2011) as the speech recognizer in our experiments. Once the ASR results are generated, the correction methods are applied to refine the sentences. To evaluate our proposed methods, we conduct two experiments in this section. For the first one, we evaluate the performance of the semantic-only post-correction with MLM in Section 3.1. In the second experiment, our proposed semantic and phonetic post-correction in Section 3.2 is also evaluated. The details will be addressed in the following sections.

	Datasets	
	AISHELL-3	Wiki
mask-all-and-replace-all	11.69 %	75.14 %
mask-one-and-replace-one	9.89 %	73.84 %
mask-all-and-replace-one	11.75 %	75.62 %

Table 1: The correction accuracies for different masking and replacement strategies.

	Correction			
	Pre.	Rec.	F_1	CER
MLM	0.099	0.061	0.075	10%
Ours ($\alpha = 500$)	**0.404**	**0.179**	**0.248**	**8.3%**

Table 2: The evaluation results of our proposed method and the baseline model on AISHELL-3 dataset. Pre., Rec., F_1 represent the correction precision, recall and F_1-score denoted in (Tseng et al., 2015), respectively.

4.1 Evaluation on Semantic-only Post-correction

In this experiment, we aim to evaluate the effectiveness on the semantic-only post-correction with MLM by considering different masking and replacement strategies as described in Section 3.1. For the error detection, we assume that our detection network could detect all the incorrect words perfectly. Based on the setting, we calculate the accuracy of correction by given the detected incorrect characters. In our evaluation, we use two benchmark datasets in this experiment. The first one is a Chinese open speech dataset: AISHELL-3 (Shi et al., 2020). AISHELL-3 contains 63,262 and 24,773 sentences as the training set and test set respectively. It is worth noting that we directly use the pre-trained MLM of BERT with different masking strategies. Thus, we did not use the training set and only sampling 20,000 sentences from the testing set for the evaluation. The second one is Wiki dataset. The dataset contains 286,975 sentences, and all of them are used for the evaluation.

From the evaluation on Wiki dataset, as the results are shown in Table 1, the mask-one-and-replace-one strategy produces the lowest accuracy. This indicates that if we only mask one incorrect character, the other unmasked incorrect characters will sacrifice the performance of MLM. On the other hand, if the incorrect characters are all masked, such as mask-all-and-replace-all and mask-all-and-replace-one strategies, the incorrect semantic information will not propagate to the task of token replacement. For AISHELL-3 dataset, we also can obtain similar results from the evaluation even if there are a lot of proper nouns in the sentences. Besides, the results from Table 1 also show that mask-all-and-replace-all and mask-all-and-replace-one strategies produce similar results for the token correction. For the sake of simplicity, we applied the mask-all-and-replace-all strategy in our experiment as the origin MLM of BERT did.

4.2 Evaluation on Our Semantic and Phonetic Post-correction

In the second experiment, we evaluate our proposed phonetic MLM post-correction mentioned in Section 3.2 with only AISHELL-3 dataset since the phonetic information is not available in Wiki dataset. Different to the setting in Section 4.1, we randomly split 6,000 sentences from the training set as the validation set to find the proper hyper-parameters in our proposed method, and all the testing data are used for the evaluation. To evaluate the performance of the post-correction for ASR, we adopt correction F_1-score and CER (character error rate) as the metrics. Correction F_1-score is calculated by examining whether each error is corrected or not. Most Chinese error correction tasks adopt this metric as the evaluation (Tseng et al., 2015). On the other hand, CER is calculated by the average error rate in every sentence. It is often used to evaluate the results of speech recognition. To evaluate the

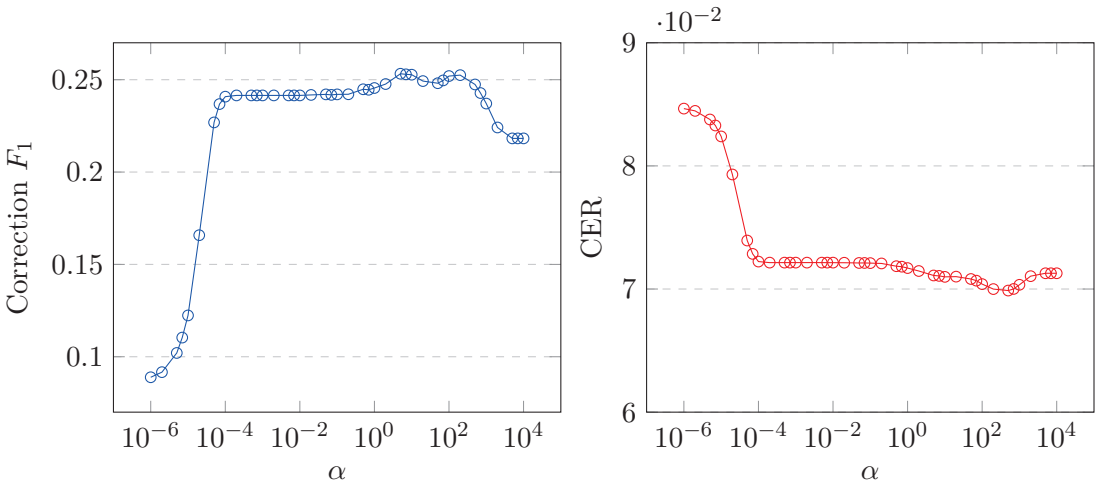

Figure 3: Comparisons of correction F_1 and CER using different α in (2) for ALSHELL-3 dataset.

performance in practices, we also report CER of the correction results in our experiments.

Followed by experimental results in Section 4.1, we use the pre-trained MLM model from the official bert-base-chinese package[1] for the semantic correction. This semantic-only approach is also the baseline in this experiment. As shown in Table 2, our proposed method could achieve 0.248 correction F_1-score while the baseline model only has 0.075 correction F_1-score. It shows that our proposed improve the performance of post-correction by leveraging the phonetic distance defined in (2).

In addition to the correction F_1-score, we also evaluate the performance of these two models with CER due to the practical usage. Similar to the results with correction F_1-score, our proposed method also achieves better CER by comparing with the baseline model. Based on the results from Table 2, we confirmed that the usage of phonetic information of characters is beneficial to post-correction of ASR.

4.3 Sensitivity of Phonetic Distance

As discussed in Section 3.2, we need to determine the hyper-parameter α in (2). This hyper-parameter controls the trade-off between semantic and phonetic information. In our experiments, we use the validation set to determine the value of α by the grid search. According to the range of phonetic distances from DIMSIM, we set 10^{-6} to 10^{4} as the search range, and calculate correction F_1-score and CER with the validation data. Typically the larger α value we have, the more influence of the phonetic distance it will increase. As shown in Figure 3, we plot the correction F_1-score and CER according to different values of α. It can be observed that slightly increasing the value of α will improve the performance dramatically. This also indicates that many homo-phonic errors can be corrected by our proposed method. On the other hand, a too large value of α will also cause the opposite effect due to the over-emphasizing of phonetic information. Besides, it also shows that the results are quite robust within a wide range of α. Thus, the proper value of α in (2) could be easily searched.

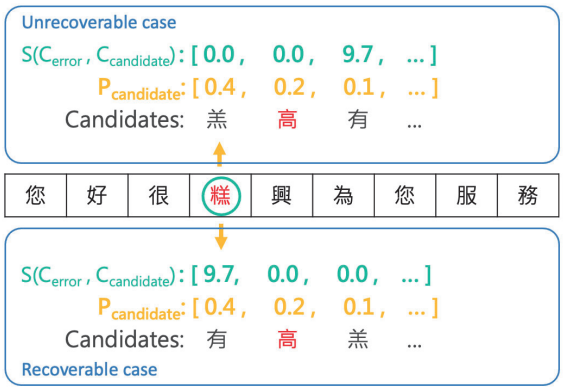

Figure 4: Examples of recoverable and unrecoverable cases in our scenario.

4.4 Recoverable Ability of Phonetic Distance

In our proposed method, it is obvious that not all the incorrect characters can be corrected by

[1] https://github.com/huggingface/transformers

adding the phonetic information. To be more precise, an error word of interest is unrecoverable if there exists a candidate that satisfies the following two conditions:

$$P_{error\ candidate} \geq P_{correct\ candidate} \quad (3)$$

and

$$\begin{aligned} &S(C_{error}, C_{error\ candidate}) \\ \leq &S(C_{error}, C_{correct\ candidate}), \end{aligned} \quad (4)$$

where C_{error} is the error word of interest, $C_{correct\ candidate}$ is the ground truth, and $C_{error\ candidate}$ is the incorrect word of the candidates. For example, as the unrecoverable case shown in Figure 4, it is not possible to recover the correct character "高" since "羔" satisfies (3) and (4). On the other hand, one can recover the correct character "高" as shown in the recoverable case of Figure 4 since no candidate satisfies (3) and (4).

In our experiments, we have 21,865 Chinese characters that are not able to be corrected properly by the baseline model. Among these error corrections, we have 6,483 recoverable characters (∼29.7%). By given these recoverable characters, our proposed method can refine 4,671 characters (∼72.1%) correctly by using the phonetic distance. This indicates that our proposed phonetic feature could fix most recoverable characters.

5 Conclusion

In this paper, we proposed a novel approach for the post-correction of speech recognition. By exploring the phonetic distance derived from DIMSIM, we integrated semantic and phonetic information based on the pre-trained MLM of BERT. By taking the phonetic distance into account, many homophonic errors can be corrected by our proposed method. Experimental results on a real-world speech recognition dataset confirmed the use of our proposed method for improved post-correction of ASR.

Acknowledgments

This work was supported in part by the Ministry of Science and Technology of Taiwan under Grants MOST 108-2221-E-017-008-MY3.

References

Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philémon Brakel, and Yoshua Bengio. 2016. End-to-end attention-based large vocabulary speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*.

Rahhal Errattahi, Asmaa El Hannani, and Hassan Ouahmane. 2018. Automatic speech recognition errors detection and correction: A review. In *Procedia Computer Science, 2018*.

Alex Graves. 2012. Sequence transduction with recurrent neural networks. In *International Conference of Machine Learning (ICML) 2012 Workshop on Representation Learning*.

Jinxi Guo, Tara N. Sainath, and Ron J. Weiss. 2019. A spelling correction model for end-to-end speech recognition. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Navdeep Jaitly, David Sussillo, Quoc V. Le, Oriol Vinyals, Ilya Sutskever, and Samy Bengio. 2016. An online sequence-to-sequence model using partial conditioning. In *NIPS*.

Shankar Kumar, Michael Nirschl, Daniel Holtmann-Rice, Hank Liao, Ananda Theertha Suresh, and Felix Yu. 2017. Lattice rescoring strategies for long short term memory language models in speech recognition. In *Proceedings of ASRU*.

Min Li, Marina Danilevsky, Sara Noeman, and Yunyao Li. 2018. Dimsim: An accurate chinese phonetic similarity algorithm based on learned high dimensional encoding. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*.

Xiaodong Liu, Fei Cheng, Yanyan Luo, Kevin Duh, and Yuji Matsumoto. 2013. A hybrid chinese spelling correction using language model and statistical machine translation with reranking. In *Proceedings of the 7th SIGHAN Workshop on Chinese Language Processing (SIGHAN-13)*.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. 2011. The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*.

L. R. Rabiner. 1989. A tutorial on hmm and selected applications in speech recognition. *IEEE Proceedings*, pages 257–286.

L. R. Rabiner and B. H. Juang. 1986. An introduction to hidden markov models. *IEEEASSP Mag. (June)*, pages 4–16.

Yao Shi, Hui Bu, Xin Xu, Shaoji Zhang, and Ming Li. 2020. Aishell-3: A multi-speaker mandarin tts corpus and the baselines. *arXiv preprint arXiv:2010.11567*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *NIPS, 2014*, pages 3104–3112.

Yuen-Hsien Tseng, Lung-Hao Lee, Li-Ping Chang, and Hsin-Hsi Chen. 2015. Introduction to sighan 2015 bake-off for chinese spelling check. In *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing*, pages 32–37.

Ziang Xie, Anand Avati, Naveen Arivazhagan, Dan Jurafsky, and Andrew Y. Ng. 2016. Neural language correction with character-based attention. *arXiv:1603.09727*.

Shaohua Zhang, Haoran Huang, Jicong Liu, and Hang Li. 2020. Spelling error correction with soft-masked bert. In *Association for Computational Linguistics*, pages 882–890.