ASR ERROR CORRECTION WITH DUAL-CHANNEL SELF-SUPERVISED LEARNING

Fan Zhang[1,2]*, Mei Tu*[2]*, Song Liu*[2]*, Jinyao Yan*[1]

[1]State Key Laboratory of Media Convergence and Communication, Communication University of China
[2]Samsung Research China – Beijing (SRC-B)

ABSTRACT

To improve the performance of Automatic Speech Recognition (ASR), it is common to deploy an error correction module at the post-processing stage to correct recognition errors. In this paper, we propose 1) an error correction model, which takes account of both contextual information and phonetic information by dual-channel; 2) a self-supervised learning method for the model. Firstly, an error region detection model is used to detect the error regions of ASR output. Then, we perform dual-channel feature extraction for the error regions, where one channel extracts their contextual information with a pre-trained language model, while the other channel builds their phonetic information. At the training stage, we construct error patterns at the phoneme level, which simplifies the data annotation procedure, thus allowing us to leverage a large scale of unlabeled data to train our model in a self-supervised learning manner. Experimental results on different test sets demonstrate the effectiveness and robustness of our model.

Index Terms— Error correction, dual-channel, self-supervised learning, pre-trained language model

1. INTRODUCTION

Automatic speech recognition has made great progress in recent years [1, 2]. However, it still suffers errors from words or phrases with similar pronunciation. To improve recognition performance, an Error Correction (EC) module is usually integrated as a post-processing technique to correct recognition errors in spoken language [3, 4, 5, 6]. Since high-level language understanding ability is required to solve the complex recognition errors, EC remains a challenging task.

Pre-trained language models have achieved state of the art in many language understanding tasks. [7] uses BERT [8] to re-predict the suspicious characters in the Chinese error correction task. It shows BERT's powerful ability for contextual information extraction. However, because the error patterns of Chinese are limited to only one-to-one word substitution, this work is hard to adapt to heterogeneous languages (such as English or German). To train an EC model, at the data preparation stage, previous works synthesize error-to-reference text pairs and learn error patterns at the grapheme level [4, 7, 9]. However, it is difficult to construct error-to-reference text pairs of good universality for supervised training because the error form varies irregularly along with complicated surrounding context and pronunciation. In addition, previous works [9, 10, 11] confirm that the phonetic information plays an important role in error correction.

Based on these observations above, we propose a dual-channel model for error correction, which is trained with self-supervised learning. Beforehand, the error regions of ASR output are detected by an error region detection model. Then, we propose a dual-channel way to extract features and re-predict words to replace the error regions, which takes both contextual information and phonetic information of the error regions into account.

Since the pronunciations of recognition errors and correct transcripts are quite similar (or even the same), the difference between these errors and references at the phonetic level has smaller granularity than at the grapheme level. Therefore, during training, instead of constructing complex error patterns at the grapheme level, we construct error patterns at the phoneme level by introducing noise to phonemes based on their pronunciation similarity. In this way, the data construction procedure is simplified. So we can leverage a large scale of unlabeled data and train the model with self-supervised learning, which makes our model more robust for different data sets. A more detailed description can be found in subsection 2.2. We test our model with three different kinds of test sets. Empirical results demonstrate the effectiveness and robustness of our model. The main contributions of our work are as follows: (1) we propose a dual-channel way to extract features and re-predict words replacing the error regions; (2) we guide the model to learn error patterns at phoneme level, which allows us to leverage a large scale of unlabeled textual data and train the model with self-supervised learning.

2. PROPOSED METHOD

2.1. Model structure

The proposed error correction model contains five parts: error region detection (ERD), contextual channel, phonetic chan-

This work was supported in part by the National Key R&D Program of China (No. 2021YFF0900701), by the National Natural Science Foundation of China (No. 61971382).

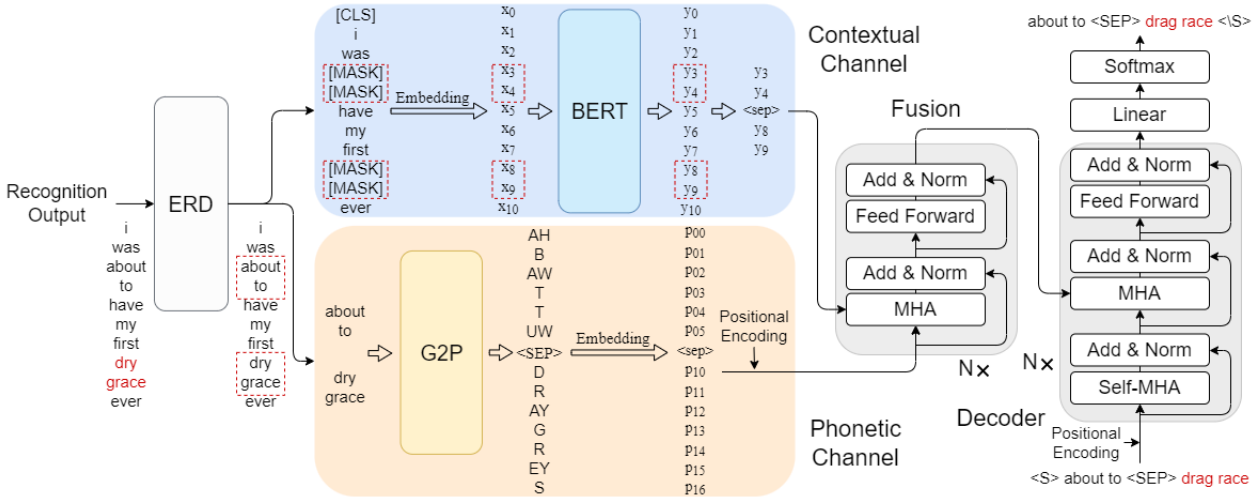

Fig. 1. Overall framework. Given the recognition output sequence, the detected error regions (positions of "*about to*" and "*dry grace*") of the sequence are first detected; with dual-channel, the contextual information and the phonetic information of the regions are obtained respectively; by decoding the fused dual-channel information, words of the error regions are re-predicted as "*about to*" and "*drag race*", the error "*dry grace*" is corrected.

nel, fusion, and decoder. Figure 1 illustrates the procedures with a specific case. Next, we introduce each part in detail.

Error region detection. Error detection has been widely studied in spoken language understanding [12, 13, 7]. As ERD is not our main focus of interest, for simplicity, it complies with a typical sequence labeling structure, which consists of a Transformer encoder for feature extraction and a conditional random field model for labeling. Positions of adjacent detected words belong to one error region.

Contextual channel. The contextual channel transforms the sentence into its high-level semantic representation. The detected regions are replaced by "[MASK]" symbol and then BERT is used for extracting their contextual information.

Phonetic channel. In the phonetic channel, with the CMU G2P tool [14], words in the error regions are converted into phonemes of ARPABET [15] to represent their phonetic information.

Fusion. The dual-channel information is fused with the Multi-Head Attention (MHA) mechanism in the fusion part. Same with Transformer [16], a feed-forward layer is added after the attention layer. Meanwhile, the residual network and normalization are applied after each layer.

Decoder. The structure of the decoder is implemented as described in Transformer [16]. Unlike [9] that produces a whole sequence, the decoder only produces words that fill these detected error regions, as we only focus on the error region correction.

2.2. Self-supervised Learning

In this work, we consider training ERD and all other components (namely the dual-channel part) as two independent parts. The ERD training process complies with [17], while the dual-channel part is in a manner of self-supervised learning. However, the error region information is necessary for the dual-channel part. Thus, to train this part with unlabeled textual data, two problems still remain to be solved: (i) how to label a fake error region in the training corpus without ERD and (ii) how to guide the model to learn error patterns at the phoneme level.

For problem (i), we refer to the masking strategy of BERT and mask words in each sentence at random. The positions of the adjacent words are regarded as an error region. Considering that a word with long pronunciation may be wrongly recognized into multiple words with short pronunciation (e.g. housetop - how step) and vice versa, in the contextual channel, we randomly drop or add a mask token in each error region with a probability of 50% to ensure the output length is not limited by the input.

For problem (ii), based on the prior knowledge of pronunciation, we perform error calibration at phoneme level, which has the effect of adding some pronunciation noise. In practice, a pronunciation similarity matrix $S \in R^{|V_p|\times|V_p|}$ is first built for all phonemes[1], where $|V_p|$ is the vocabulary size of phonemes, S_{ij} is the similarity between phonemes of the i-th and the j-th, $S_{ii} = 1$ and $S_{ij} = S_{ji}$. Then the transition probability matrix $T \in R^{|V_p|\times|V_p|}$ is computed by S:

$$T_{ij} = \frac{S_{ij}}{\sum_{j=0}^{|V_p|} S_{ij}} \tag{1}$$

For each masked region, we randomly change each phoneme according to the transition probability matrix. With this pro-

[1]https://github.com/lost-libra/phoneme_similarity

Table 1. Training set of different models

	Training set
A-Trans	TED-LIUM3 & DATA2
Ours	WIKIPEDIA
Ours-FT	+ TED-LIUM3 & DATA2

Table 2. Detection performance

Test set	$Prec.$	$Rec.$	F
WIKIPEDIA	0.69	0.89	0.78
TED-LIUM3	0.66	0.90	0.76
DATA2	0.70	0.94	0.81

posed training method, the training set can be created from ground truth text-only data in a simple way. Compared to previous works [4, 7], we do not need to use text-to-speech pipelined with speech recognition to collect training data, which simplifies the data collection process.

3. EXPERIMENTS

Our error correction task is based on the End-to-End (E2E) ASR model with the structure of [2]. The Augmented Transformer (A-Trans) [9] which also uses phonetic information in a different way is implemented for comparison. The correction performance of different models is tested on three test sets with different data distributions.

3.1. Corpus

Three different corpora are collected to validate our method.

WIKIPEDIA. The WIKIPEDIA corpus includes kinds of domains and can be regarded as an open domain corpus. It contains about 30 million sentences after cleaning. 3000 sentences are randomly selected as the ground truth of the test set and the rest as the training set. To get ASR transcripts, each ground truth sentence of the test set is first converted into speech by the TTS model, and then this speech is translated with the E2E ASR model. The error measurements between these texts and their ASR transcripts are regarded as the original baseline of the WIKIPEDIA test set. Furthermore, $100,000$ ASR transcripts are also generated to train the error region detection model.

TED-LIUM3. The TED-LIUM3 [18] corpus is selected from the TED conference videos, which has $225,018$ speech-to-text data pairs of the training set and 1155 of the test set.

DATA2. The DATA2 [19] corpus is built for the end-to-end name entity recognition tasks, which contains $70,763$ speech-to-text pairs. 2000 pairs are randomly selected as the test set and the rest as the training set.

Constructing error patterns from text-only data requires both TTS and ASR processing [4], which is quite time-consuming. To collect the error patterns with 30 million Wikipedia data roughly consumes about 20 to 30 days in our platform, so we only use the two speech corpora to build the training set of A-Trans. The beam size of the ASR model is set to 8 to generate the candidate transcripts. We make these candidates and the corresponding ground truth texts of error-to-reference data pairs as the training set, which contains about 3.6 million data pairs in the final. Considering the specific data distribution of the two speech corpora, we also fine-tune our model (Ours-FT) with the textual data of them and analyze the difference. Table 1 shows the training set information of each model.

3.2. Implementation

The experimental models are implemented with OpenNMT-tf [20]. In A-Trans, all hyper-parameters and training strategies are the same as the original paper. In our proposed method, for the ERD model, we use a Transformer encoder with the hidden size of 512, the feed-forward layer size of 1024, and the stack number of 4. In the dual-channel part, the hidden size d is set to 768 to be equal with that of BERT. The stack number of both the fusion part and the decoder is set to 4. The attention head number is 8. The size of the feed-forward layer is 1024. The parameters of BERT are fixed. Phonemes are generated using the CMU G2P tool [14] and the accent information of phonemes is deleted (e.g. "AA0", "AA1" and "AA2" are all represented as "AA").

3.3. Error region detection

In subsection 2.2, it is mentioned that ERD model is trained separately in a supervised learning manner. To build its training set, we first align the recognition results and references of each training set. We only focus on substitution errors and tag them with the BIO labeling strategy. An ERD model is trained for each of the three corpora.

The detected error region which totally covers an error is regarded as a correct case. The precision rate $Prec. = \frac{N_c}{N_m}$ and the recall rate $Rec. = \frac{N_c}{N_e}$ are used to measure the performance of ERD models, where N_c, N_m and N_e represent the number of the correct cases, the detected error regions and the true error regions respectively. Table 2 shows the detection performance of each ERD model. Next, we re-predict words in these error regions with our models for error correction.

3.4. Error correction

We test the error correction performance of each model on the three test sets, measured by the Word Error Rate (WER) and the Sentence Error Rate (SER). Table 3 shows the measurements over different models.

WER results show although our model (Ours) is trained with WIKIPEDIA, it also performs well on TED-LIUM3 and DATA2 test sets. In contrast, WER results of A-Trans show it

Table 3. Measurements of error correction performance

	WIKIPEDIA	**TED-LIUM3**	**DATA2**
	WER / SER	WER / SER	WER / SER
Original	8.96 / 47.10	10.10 / 57.83	7.56 / 52.85
A-Trans	12.57 / 60.70	10.08 / 63.11	6.12 / 50.70
Ours	**7.63 / 43.13**	8.27 / 54.98	6.43 / 49.95
Ours-FT	7.74 / 43.37	**8.10 / 54.03**	**5.75 / 42.65**

Table 4. Upper bound analysis

Metrics	**WIKIPEDIA**	**TED-LIUM3**	**DATA2**
P_{rr}	0.97	0.98	0.97
P_{er}	0.27	0.25	0.33
WER / SER	7.27 / 42.37	7.82 / 52.47	5.53 / 41.05

only performs well on DATA2, and there is little improvement on TED-LIUM3. For WIKIPEDIA which is not in its training set (see Table 1), WER scores are even worse than Original. It is reasonable that A-Trans is designed to solve entity retrieval recognition errors which is a domain-specific problem. So it performs well on DATA2, which has a limited number of error patterns. For TED-LIUM3 that has a relatively big domain area, the complex error patterns at the grapheme level make A-Trans perform poorly. When testing on an open domain test set (WIKIPEDIA), A-Trans even worsens the original recognition results. Since we leveraged a large scale of unlabeled data by introducing error patterns at the phoneme-level, our model shows better robustness. Because A-Trans re-predicts every word of recognition results, it increases the risk of over correction. Our model only re-predicts the error regions to avoid this kind of risk. The SER results in Table 3 prove the effectiveness of this predicting strategy.

We further fine-tune our model with TED-LIUM3 and DATA2. Results of Ours-FT show it further improves the recognition performance, especially on DATA2. After analyzing the DATA2 test set, we find the improvement is mainly reflected in the naming entities such as person names and location names. It is reasonable that DATA2 is built for the end-to-end name entity recognition tasks.

3.5. Performance bound

Since the detected error regions contain not only true error regions (containing errors in this region) but also false error regions (no error in this region), we count the proportion P_{rr} of the false error regions that are not over corrected (i.e. keeping the false error regions unchanged) as well as the proportion P_{er} of the true error regions that are corrected (i.e. errors in the true error regions are corrected) with the fine-tuned model and report the results in Table 4.

The statistical results of P_{rr} show that our model reaches high rates of keeping false error regions unchanged. However, it seems that the proportion P_{er} is relatively low. We check these errors and find some of them are partially corrected, but most of them cannot be corrected by only text. The possible reasons are analyzed with case study in subsection 3.6. We didn't deliberately optimize our model for testing.

As the detected error regions of the ERD are not so accurate, the model performance cannot achieve its upper bound. To test the upper bound performance of the fine-tuned model, instead of using the ERD model, the oracle error regions are detected by aligning recognition transcripts with references, which means both $Prec. = 1$ and $Rec. = 1$. Table 4 shows that with oracle ERD beforehand, the performance of our model has the potential to reach much lower WER/SER, which can be a promising direction in the future.

Table 5. Sample analysis

NO.1	**housetop** is a compound word **how step** is a compound word **housetop** is a compound word
NO.2	**well it 's** about thirty percent he said **that 's** about thirty percent he said **that 's** about thirty percent he said
NO.3	by this time **charley** was as enraged as the greek by this time **charlie** was as enraged as the greek by this time **charley** was as enraged as the greek

3.6. Sample analysis

To better support our analysis above, some typical samples from the three test sets with oracle ERD are listed in Table 5. In each sample, the reference, the ASR transcript, and the corrected transcript are listed in order. The NO.1 sample shows our model has the ability to handle many-to-many error cases, which benefits from the training strategy described in subsection 2.2. The NO.2 sample is a bad case, which results in low P_{er}. It is listed here to illustrate that many recognition errors in an open domain can be corrected only by original speech because they are syntactically and semantically correct. The NO.3 sample is a name entity case from DATA2. It shows in some error correction cases, the similar data distribution between the training set and the test set is important.

4. CONCLUSIONS

In this paper, we propose an error correction model with dual-channel processing to improve the performance of an ASR model. Instead of constructing error patterns at the grapheme level, we construct error patterns at the phoneme level by introducing phonetic noise, which simplifies the data annotation procedure, thus allowing us to leverage a large scale of unlabeled data to train the model in a self-supervised learning manner. Empirical results show the effectiveness and robustness of our model.

5. REFERENCES

[1] Alex Graves and Navdeep Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International conference on machine learning*. PMLR, 2014, pp. 1764–1772.

[2] Linhao Dong, Shuang Xu, and Bo Xu, "Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5884–5888.

[3] Tomohiro Tanaka, Ryo Masumura, Hirokazu Masataki, and Yushi Aono, "Neural error corrective language models for automatic speech recognition.," in *INTERSPEECH*, 2018, pp. 401–405.

[4] Jinxi Guo, Tara N Sainath, and Ron J Weiss, "A spelling correction model for end-to-end speech recognition," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5651–5655.

[5] Yue Weng, Sai Sumanth Miryala, Chandra Khatri, Runze Wang, Huaixiu Zheng, Piero Molino, Mahdi Namazifar, Alexandros Papangelis, Hugh Williams, Franziska Bell, et al., "Joint contextual modeling for asr correction and language understanding," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6349–6353.

[6] Shuai Zhang, Jiangyan Yi, Zhengkun Tian, Ye Bai, Jianhua Tao, Xuefei Liu, and Zhengqi Wen, "End-to-end spelling correction conditioned on acoustic feature for code-switching speech recognition," *Proc. Interspeech 2021*, pp. 266–270, 2021.

[7] Shaohua Zhang, Haoran Huang, Jicong Liu, and Hang Li, "Spelling error correction with soft-masked bert," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 882–890.

[8] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.

[9] Haoyu Wang, Shuyan Dong, Yue Liu, James Logan, Ashish Kumar Agrawal, and Yang Liu, "Asr error correction with augmented transformer for entity retrieval," *Proc. Interspeech 2020*, pp. 1550–1554, 2020.

[10] Arushi Raghuvanshi, Vijay Ramakrishnan, Varsha Embar, Lucien Carroll, and Karthik Raghunathan, "Entity resolution for noisy asr transcripts," 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): System Demonstrations*, 2019, pp. 61–66.

[11] Abhinav Garg, Ashutosh Gupta, Dhananjaya Gowda, Shatrughan Singh, and Chanwoo Kim, "Hierarchical multi-stage word-to-grapheme named entity corrector for automatic speech recognition," in *Proc. Interspeech*, 2020, vol. 2020, pp. 1793–1797.

[12] Mariano Felice and Ted Briscoe, "Towards a standard evaluation method for grammatical error detection and correction," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2015, pp. 578–587.

[13] Jinhua Xiong, Qiao Zhang, Shuiyuan Zhang, Jianpeng Hou, and Xueqi Cheng, "Hanspeller: a unified framework for chinese spelling correction," in *International Journal of Computational Linguistics & Chinese Language Processing, Volume 20, Number 1, June 2015-Special Issue on Chinese as a Foreign Language*, 2015.

[14] CMU Sphinx, "Tool of grapheme to phoneme," https://github.com/cmusphinx/g2p-seq2seq.

[15] Lloyd Rice, "Hardware and software for speech synthesis," *Dr. Dobb's Journal of Computer Calisthenics and Orthodontia*, vol. 4, no. 1, pp. 6–8, Apr. 1976.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[17] Zhiheng Huang, Wei Xu, and Kai Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.

[18] François Hernandez, Vincent Nguyen, Sahar Ghannay, Natalia Tomashenko, and Yannick Estève, "Ted-lium 3: twice as much data and corpus repartition for experiments on speaker adaptation," in *International Conference on Speech and Computer*. Springer, 2018, pp. 198–208.

[19] Hemant Yadav, Sreyan Ghosh, Yi Yu, and Rajiv Ratn Shah, "End-to-end named entity recognition from english speech," *Organization*, vol. 2299, no. 1473, pp. 3772, 2020.

[20] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush, "Opennmt: Open-source toolkit for neural machine translation," in *Proceedings of ACL 2017, System Demonstrations*, 2017, pp. 67–72.