Interspeech 2022
18-22 September 2022, Incheon, Korea

Japanese ASR-Robust Pre-trained Language Model with Pseudo-Error Sentences Generated by Grapheme-Phoneme Conversion

Yasuhito Ohsugi, Itsumi Saito, Kyosuke Nishida, Sen Yoshida

NTT Human Informatics Laboratories, NTT Corporation

yasuhito.ohsugi.va@hco.ntt.co.jp

Abstract

Spoken language understanding systems typically consist of a pipeline of automatic speech recognition (ASR) and natural language processing (NLP) modules. Although pre-trained language models (PLMs) have been successful in NLP by training on large corpora of written texts; spoken language with serious ASR errors that change its meaning is difficult to understand. We propose a method for pre-training Japanese LMs robust against ASR errors without using ASR. With the proposed method using written texts, sentences containing pseudo-ASR errors are generated using a pseudo-error dictionary constructed using grapheme-to-phoneme and phoneme-to-grapheme models based on neural networks. Experiments on spoken dialogue summarization showed that the ASR-robust LM pre-trained with the proposed method outperformed the LM pre-trained with standard masked language modeling by 3.17 points on ROUGE-L when fine-tuning with dialogues including ASR errors.

Index Terms: pre-trained language model, ASR-robust, spoken dialogue summarization

1. Introduction

With the increase in demand for web conferencing and smart devices, spoken language understanding (SLU) systems such as spoken dialogue summarization [1] have been gaining attention. SLU systems can be roughly classified into pipeline approaches of automatic speech recognition (ASR) and natural language processing (NLP) modules, as well as end-to-end approaches. End-to-end approaches can be robust against ASR errors by training on large corpora of the speech and transcript dataset of the target task [2], but have been difficult to apply to low-resource languages such as Japanese. The pipeline approaches of ASR and NLP modules are more common because of the availability of excellent ASR modules and pre-trained language models (PLMs). These approaches fine-tune a PLM to adapt to the target task and achieve high performance even when the amount of data in the target task is relatively small. However, because PLMs are typically learned from written texts, it is difficult to understand texts that contain serious ASR errors that change the meaning of the spoken language.

To improve the performance of PLMs in pipelined SLU systems, ASR-error correction (EC) methods have been proposed [3, 4]. However, it is difficult to obtain a large number of speech and transcript pairs, so approaches that can improve the ASR-robustness of PLMs from written texts are more promising. Two types of methods for constructing a pre-training text corpus including errors from written texts have been proposed. One type is based on word-by-word replacement, for example, using homophone dictionaries [3]. However, these methods cannot take into account crucial errors such as phonetic connections and contextualized errors in multiple words. In particular, Japanese has a large number of Kanji, Hiragana, and Katakana characters, resulting in a larger number of graphemes than phonemes and a variety of contextualized errors. The other type is based on sentence-by-sentence conversion by recognizing speeches generated from text-to-speech (TTS) modules [5, 6, 7, 8]. Although these methods can generate various ASR errors, it takes a large amount of time to generate enough data for pre-training language models.

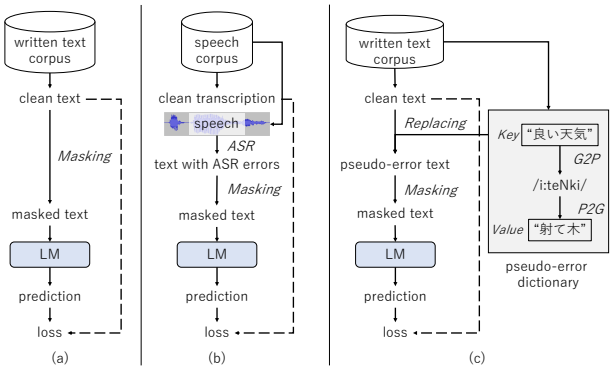

Figure 1: *Pre-training tasks for language models (LMs). (a)* ***MLM*** *(Masked Language Modeling) with written-text corpus. (b)* ***MLM+EC*** *(Error Correction of ASR) with speech corpus. (c) our* ***MLM+PEC*** *method with written corpus that can learn MLM and pseudo-ASR-error correction generated by grapheme-to-phoneme (G2P) and phoneme-to-grapheme (P2G) conversions.*

We propose a method to pre-train Japanese language models robust against errors. Our contributions are as follows.

- We propose a method for generating pseudo ASR errors by using grapheme-to-phoneme (G2P) and phoneme-to-grapheme (P2G) based on neural networks, without ASR and TTS. These G2P and P2G models take a short sequence of tokens as input and learn phonetic connections over multiple graphemes.
- We introduce a method for building a pseudo-error dictionary using the G2P and P2G models with a beam search and generating sentences containing phrase-level errors with the dictionary during pre-training.
- Our experiments on a Japanese spoken dialogue summarization task in a call center show that our method was more robust against ASR errors than using standard masked language modeling (MLM) and error correction of ASR.

10.21437/Interspeech.2022-327

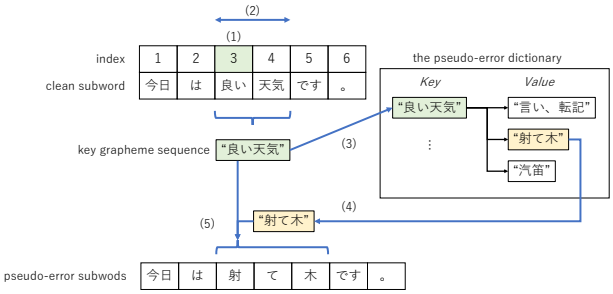

Figure 2: *Example of making pseudo-error sentence. Phonemes of this input grapheme sequence are /kyo:wa**i:teNki**desu/, and those of generated sequence are /kyo:wa**iteki**desu/.*

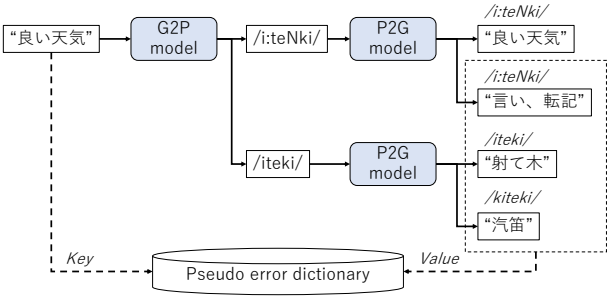

Figure 3: *Example of our pseudo-error dictionary consisting of key-values pairs. A key grapheme sequence is converted to two phoneme sequences by using a G2P model. Then, each phoneme sequence is converted to two grapheme sequences by using a P2G model. Three of four obtained grapheme sequences are incorrect in terms of homophones, punctuation insertion, phoneme deletion, and phoneme replacement. These incorrect sequences are used as values associated with the key grapheme sequence.*

2. Proposed Method

The proposed method pre-trains an LM by restoring masked sentences with ASR-like errors generated with a pseudo-error dictionary to clean text. In the dictionary, one key grapheme sequence sampled from the written-text corpus is associated with incorrect grapheme sequences that have similar phonemes. These incorrect grapheme sequences are generated using two sequence-to-sequence (seq2seq) models, i.e., grapheme-to-phoneme (G2P) and phoneme-to-grapheme (P2G) models. By sequence conversion, our method can generate ASR-like errors more flexibly than using homophone dictionaries.

2.1. Pseudo-error Generation

Let $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ be a clean text (grapheme sequence) and a text with noise, respectively. The training objective of a PLM with parameters θ is to reconstruct $\boldsymbol{x}$ from $\hat{\boldsymbol{x}}$:

$$\max_{\theta} \sum_{t=1}^{T} \log p_\theta(x_t|\hat{\boldsymbol{x}}). \quad (1)$$

First, in standard MLM pre-training with a written-text corpus, $\hat{\boldsymbol{x}}$ can be generated by token masking, as shown in Figure 1(a),

$$\hat{\boldsymbol{x}} = \mathrm{Masking}(\boldsymbol{x}) \quad (2)$$

In MLM pre-training with a speech corpus, however, $\hat{\boldsymbol{x}}$ is obtained by ASR of speech $\boldsymbol{s}$ in a spoken corpus, as shown in Figure 1(b),

$$\hat{\boldsymbol{x}} = \mathrm{Masking}(\mathrm{ASR}(\boldsymbol{s})), \quad (3)$$

where $\boldsymbol{x}$ in Eq. (1) is a transcript of speech $\boldsymbol{s}$. Note that this pre-training contains ASR-error correction in addition to MLM.

Our method combines both approaches but uses a pseudo-error dictionary instead of ASR to generate texts containing errors,

$$\hat{\boldsymbol{x}} = \mathrm{Masking}(\mathrm{PseudoError}(\boldsymbol{x})), \quad (4)$$

where the input can be a written text with our method.

As shown in Figure 2, the procedure of $\mathrm{PseudoError}(\cdot)$ for generating a pseudo-error sentence with the pseudo-error dictionary consisting of pairs of an original grapheme sequence (key) and multiple incorrect grapheme sequences (values), involves the following five steps. (1) One index of the original grapheme sequence (subwords) is selected randomly as the start index. (2) The length of subwords for G2P conversion m is selected randomly from one to M, where M is a hyperparameter. (3) Since the key subwords are defined, incorrect subwords are looked up in the pseudo-error dictionary, and (4) one incorrect grapheme sequence is sampled from them. (5) Finally, a pseudo-error sentence is generated by replacing the original subwords with sampled subwords.

2.2. Construction of Pseudo-error Dictionary

To construct a pseudo-error dictionary, G2P and P2G models are pre-trained using a written-text corpus with phoneme labels. As shown in Figure 3, N phoneme sequences are obtained as hypotheses by converting one key grapheme sequence with the G2P model. By using N-best hypotheses, obtained sequences are similar to correct phonemes but have phoneme insertions, deletions, and substitutions. Then, one hypothesis is converted to N-best grapheme sequences with the P2G model. If a correct sequence is obtained, we remove it and the other sequences are used as the values of the dictionary. Therefore, one key grapheme sequence is associated with a maximum $N \times N$ incorrect grapheme sequences. We note that various errors can be generated without conversions in a full-sentence level by using the sub-sequence level pseudo-error dictionary constructed by G2P and P2G conversions.

3. Experiments

3.1. Datasets

We used two datasets for pre-training and a dataset for the downstream task.

- **10GB written-text corpus.** For pre-training with a written-text corpus shown in Figure 1(a), we used 10GB of written texts collected from web sites including Wikipedia.
- **CSJ corpus.** For pre-training with a speech corpus shown in Figure 1(b), we used the training set of Corpus of Spontaneous Japanese (CSJ) [9] containing 233 hours of raw speech data and 35MB of transcribed texts. We used two ASR modules to obtain texts from the speech data. The first one is a publicly available ASR model based on Conformer [10] trained on the CSJ corpus [1], and the other is the inhouse model based on DNN-HMM

[1] https://zenodo.org/record/4065140

Table 1: *Statistics of our dataset for spoken dialogue summarization. These values are averages over dialogues.*

	input dialogue			summary
	# of utterances	# of tokens (transcript)	# of tokens (ASR)	# of tokens (transcript)
train	58.9	435.3	418.1	78.7
dev	59.7	435.3	421.8	73.1
test	58.8	434.6	414.8	77.0

[11]. The character error rates of the two models on the CSJ corpus were 1.6 and 15.8%, respectively.

- **Spoken dialogue summarization.** As the downstream task, we used our in-house dataset consisting of 684 simulated call-center dialogues between two speakers, the utterances of which were manually transcribed, and summaries were extracted manually from the utterances. We split the dataset into 499, 48, and 137 dialogues for the training, development, and test sets, respectively.

  To investigate the effect of the presence or absence of ASR errors on the accuracy of spoken dialogue summarization, we conducted experiments with two settings: the **main** setting used pairs of ASR utterances and transcribed summary, and the **sub** setting used pairs of transcribed utterances and transcribed summary. The ASR texts were extracted with the Conformer-based ASR module. The character error rate on this dataset was 16.4%. Table 1 shows the statistics of this dataset.

3.2. Baselines

We used two encoder-decoder models based on BERT2BERT [12] as the baseline models. These models were initialized with our Japanese BERT-base model pre-trained on 30GB of written texts. The tokenizer was based on SentencePiece [13], and the vocab size was 32,000.

- **MLM**. This baseline model, shown in Figure 1(a), was trained with standard MLM [14] on the 10GB written-text corpus. Following BERT, token masking with a mask probability of 12% and keeping probability of 1.5% were carried out for input texts. This model is different from the model pre-trained with the proposed method in that it does not contain pseudo-errors in the sentences of the corpus.
- **MLM→MLM+EC**. We also additionally trained the first baseline model with MLM+EC shown in Figure 1(b), which is a seq2seq pre-training using masked ASR text and transcript pairs on a speech corpus.

3.3. Implementation

3.3.1. Construction of pseudo-error dictionary

The G2P and P2G models had the same architecture. The encoder had two 256-dimensional bidirectional LSTM layers and the decoder had two 512-dimensional unidirectional LSTM layers. We randomly sampled 2,170,443 and 10,000 subword sequences from the written-text corpus for the training and development sets, respectively. Phoneme labels were created on the basis of our in-house phoneme-tagging model. These models were trained with a batch size of 128 of 20,000 steps using Adam [15] with a learning rate of 2e-4 and 10,000 warm-up steps, respectively. Other parameters were the default values of OpenNMT [16]. The final accuracy scores on the validation set were 98.2 and 76.2% for G2P and P2G conversions, respectively. To construct the pseudo-error dictionary, 207,894 subword-based N-grams that occurred at least 11 times in the written-text corpus were used, where $N = 1$ to 5. In the conversion between graphemes and phonemes, five hypotheses were used, and the same candidates as the original subwords were removed. The training time with two P100 GPUs was 9 hours. The construction time with five P100 GPUs was 5 hours.

3.3.2. Pre-training using pseudo-error dictionary

We used two BERT2BERT models pre-trained using the proposed method, corresponding to the baselines. These models were also initialized with our Japanese BERT-base model.

- **MLM+PEC**. This model was pre-trained using our proposed method which learns MLM and pseudo-error correction, on the 10GB written-text corpus. During pre-training, some spans from the input sequence are randomly sampled and converted into erroneous grapheme sequences using the pseudo-error dictionary. At this time, the total number of tokens in all spans is set to 1.5% the input sequence, which is the same as the replacement probability of BERT's token masking.

  We trained this model with a batch size of 256 for 29,348 steps, which was one epoch over the written-text corpus, using Adam with a learning rate of 3e-5 and 3,000 warmup steps. Other parameters were the default values of Huggingface transformers [17]. The hyperparameters of the MLM baseline were the same as this model.
- **MLM+PEC→MLM+EC**. As in the MLM→MLM+EC baseline, this LM was further pre-trained using the CSJ corpus. When training this model (and the MLM→MLM+EC baseline), we changed the training and warmup steps to 10,000 and 1,000, respectively.

3.3.3. Fine-tuning on spoken dialogue summarization

We fine-tuned the models with a batch size of 4 for 6,000 steps using Adam with a learning rate of 1e-6 and 1,000 warmup steps. On the test set, we generated texts with a beam size of 5 and blocking 5-gram repetitions. The minimum and maximum lengths of produced summary were 10 and 256, respectively. We used segment embeddings in the encoder that was used to distinguish between two speakers in a dialogue.

3.4. Experimental Results

We investigated whether our proposed method can improve the performance on the spoken dialogue summarization task. As described in §3.1, we conducted experiments in the two settings to investigate the effect of the presence (the main setting) or absence (the sub-setting) of ASR errors in the input utterances on the performance of the spoken dialogue summarization task. We report ROUGE scores using Ginza as the tokenizer [2].

3.4.1. Can proposed method with pseudo-error correction improve ASR-robustness without speech corpus?

The top rows of Table 2 in the main setting (fine-tuned on ASR) show that the MLM+PEC method, which uses pseudo-ASR-

[2] https://megagonlabs.github.io/ginza/

Table 2: *ROUGE scores of dialogue summarization on test set. x indicates written texts. s_{asr_1} and s_{asr_2} indicate ASR texts of speech s using Conformer-based and DNN-HMM-based ASR modules, respectively. Bold (underlined) numbers indicate best score between baselines and our models (among all models). Results were averaged over three runs.*

	pre-training		fine-tuned on ASR			fine-tuned on transcripts		
	1st	2nd	R-1	R-2	R-L	R-1	R-2	R-L
baseline (MLM)	MLM(x)	–	42.30	28.09	42.56	46.88	35.35	47.80
ours (MLM+PEC)	MLM+PEC(x)	–	**46.10**	**30.80**	**45.73**	**50.53**	**37.88**	**50.33**
baseline (MLM→MLM+EC)	MLM(x)	MLM+EC(s_{asr1})	44.75	30.40	45.16	51.24	**38.59**	**51.34**
ours (MLM+PEC→MLM+EC)	MLM+PEC(x)	MLM+EC(s_{asr1})	**47.07**	**32.00**	**46.45**	**51.90**	38.36	**51.34**
baseline (MLM→MLM+EC)	MLM(x)	MLM+EC(s_{asr2})	47.03	32.00	46.89	**52.39**	39.58	51.66
ours (MLM+PEC→MLM+EC)	MLM+PEC(x)	MLM+EC(s_{asr2})	**48.70**	**33.90**	**48.21**	52.17	**39.62**	**52.67**

error correction generated from written texts using the pseudo-error dictionary, clearly outperformed the MLM baseline.

3.4.2. Can proposed method with pseudo-error correction improve performance when speech corpus is available?

In the main setting, the MLM+PEC method (when not conducting the second additional pre-training with speech corpus) was comparable to the MLM→MLM+EC baselines (error correction of ASR) pre-training with the speech corpus containing transcribed texts, shown in Table 2. These results indicate the effectiveness of the proposed method since it is difficult to collect a large enough speech corpus. We confirmed that the MLM+PEC→MLM+EC method performed the best in terms of the ROUGE-L metric.

Moreover, the second MLM+EC pre-trainings were less effective when using the Conformer-based ASR, the character error rate of which was low (1.6%) on the speech corpus, than using DNN-HMM-based ASR, which had a higher error rate (15.8%). In the former case, the model was not sufficiently pre-trained for error correction because there were few errors occurred by ASR. That is, the ASR module with poor performance in the pre-training corpus was useful for generating diverse ASR errors.

3.4.3. Can proposed method with pseudo error correction improve summarization performance for clean text?

Our models outperformed the baselines in the sub-setting (fine-tuned on transcribed texts). This is because MLM+PEC, which is a more difficult pre-training task than standard MLM due to the replacement of multiple graphemes, was also effective in improving natural language understanding.

4. Related work

4.1. Text-level data augmentation

ASR hypotheses can be used for data augmentation [18, 19, 20, 21]. For example, to build an ASR-robust speech translation system, Ruiz et al. generated pseudo-error sentences with a phrase-based machine translation that converts phoneme sequences obtained by a pronunciation dictionary of TTS system to ASR texts [18]. Although this is similar to our proposed method in terms of conversion through phonemes, our method does not need ASR texts to train conversion models. Although a pseudo-error sentence can be generated with a homophone dictionary without using ASR hypotheses [3], the phonetic relationship among multiple words cannot be taken into account.

4.2. Audio-level data augmentation

Several methods using TTS and ASR have been proposed to get texts containing ASR errors [5, 6, 7, 8]. This approach adopts a TTS module to convert text corpus into audio files. It then adds noise into the audio and passes the audio through an ASR module. However, the use of TTS is time-consuming, and it is difficult to use this approach for acquiring a large-scale text corpus containing ASR errors.

4.3. ASR-robust PLMs and word embeddings

Several pre-training methods of LMs, including Masked Language Modeling (MLM) [22], were proposed [14, 23, 24]. As mentioned in [5], ASR errors can deteriorate the performance of downstream tasks, but pre-training methods of LMs robust against ASR errors have not yet been fully investigated. ARoBERT [25], which is a concurrent work to ours, proposed to integrate ASR ambiguities into the pre-training stages of LMs by injecting heuristic-based phonetic similarities or ASR-based substitution scores into the loss function of MLM. Our proposed method is different from ARoBERT that our method changes the input text of MLM with phonetic information, not the loss function. Also, some studies proposed to learn word embeddings considering phonetic information for SLU tasks [26, 27, 28, 29].

5. Conclusions

We proposed a method of constructing language models robust against ASR errors through pre-training using pseudo-error sentences. During pre-training, pseudo-error sentences are generated using a pseudo-error dictionary created by grapheme-to-phoneme and phoneme-to-grapheme conversions. In the speech dialogue summarization task, the models pre-trained with our method outperformed the standard MLM baseline based only on token masking when the written-text corpus is used for the pre-training. The ROUGE-L score increased by 3.17 points when fine-tuned on dialogues involving ASR errors. By additional MLM with ASR-error correction using the CSJ corpus to our models, further accuracy improvement was observed.

A limitation of this study is that we did not fully evaluate the coverage of the pseudo-error dictionary over the errors that occurred by actual ASR modules. In the future, we will investigate the performance when we use a manually maintained error dictionary and will tackle improving the quality of the pseudo-error dictionary. We will also verify the effectiveness of the proposed method in tasks other than speech dialogue summarization.

6. References

[1] R. Sharma, S. Palaskar, A. W. Black, and F. Metze, "Speech summarization using restricted self-attention," *arXiv preprint arXiv:2110.06263*, 2021.

[2] E. Palogiannidi, I. Gkinis, G. Mastrapas, P. Mizera, and T. Stafylakis, "End-to-end architectures for asr-free spoken language understanding," in *ICASSP*, 2020, pp. 7974–7978.

[3] Y. Leng, X. Tan, R. Wang, L. Zhu, J. Xu, W. Liu, L. Liu, X.-Y. Li, T. Qin, E. Lin, and T.-Y. Liu, "FastCorrect 2: Fast error correction on multiple candidates for automatic speech recognition," in *EMNLP*, 2021, pp. 4328–4337.

[4] S. Dutta, S. Jain, A. Maheshwari, G. Ramakrishnan, and P. Jyothi, "Error correction in ASR using sequence-to-sequence models," *arXiv preprint arXiv:2202.01157*, 2022.

[5] L. Feng, J. Yu, D. Cai, S. Liu, H. Zheng, and Y. Wang, "ASR-GLUE: A new multi-task benchmark for asr-robust natural language understanding," *arXiv preprint arXiv:2108.13048*, 2021.

[6] W. Li, H. Di, L. Wang, K. Ouchi, and J. Lu, "Boost transformer with bert and copying mechanism for asr error correction," in *IJCNN*, 2021, pp. 1–6.

[7] M. Fazel-Zarandi, L. Wang, A. Tiwari, and S. Matsoukas, "Investigation of error simulation techniques for learning dialog policies for conversational error recovery," *The 3rd NeurIPS workshop on Conversational AI*, 2019.

[8] C. Park, J. Seo, S. Lee, C. Lee, H. Moon, S. Eo, and H. Lim, "BTS: Back TranScription for speech-to-text post-processor using text-to-speech-to-text," in *The 8th Workshop on Asian Translation*, 2021, pp. 106–116.

[9] K. Maekawa, "Corpus of spontaneous japanese: its design and evaluation," in *SSPR*, 2003, pp. 7–12.

[10] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Interspeech*, 2020, pp. 5036–5040.

[11] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, 2012.

[12] S. Rothe, S. Narayan, and A. Severyn, "Leveraging pre-trained checkpoints for sequence generation tasks," *Trans. Assoc. Comput. Linguistics*, vol. 8, pp. 264–280, 2020.

[13] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *EMNLP*, 2018, pp. 66–71.

[14] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *ACL*, 2020, pp. 7871–7880.

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, Y. Bengio and Y. LeCun, Eds., 2015.

[16] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. Rush, "OpenNMT: Open-source toolkit for neural machine translation," in *ACL*, 2017, pp. 67–72.

[17] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. Rush, "Transformers: State-of-the-art natural language processing," in *EMNLP*, 2020, pp. 38–45.

[18] N. Ruiz, Q. Gao, W. Lewis, and M. Federico, "Adapting machine translation models toward misrecognized speech with text-to-speech pronunciation rules and acoustic confusability," in *Interspeech*, 2015, pp. 2247–2251.

[19] L. Wang, M. Fazel-Zarandi, A. Tiwari, S. Matsoukas, and L. Polymenakos, "Data augmentation for training dialog models robust to speech recognition errors," in *The 2nd Workshop on NLP for Conversational AI*, 2020, pp. 63–70.

[20] T. Cui, J. Xiao, L. Li, X. Jiang, and Q. Liu, "An approach to improve robustness of NLP systems against ASR errors," *arXiv preprint arXiv:2103.13610*, 2021.

[21] K. Beneš and L. Burget, "Text Augmentation for Language Models in High Error Recognition Scenario," in *Interspeech*, 2021, pp. 1872–1876.

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT*, 2019, pp. 4171–4186.

[23] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 140, pp. 1–67, 2020.

[24] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *NeurIPS*, 2020.

[25] C. Wang, S. Dai, Y. Wang, F. Yang, M. Qiu, K. Chen, W. Zhou, and J. Huang, "Arobert: An asr robust pre-trained language model for spoken language understanding," *IEEE ACM Trans. Audio Speech Lang. Process.*, pp. 1–1, 2022.

[26] X. Cheng, W. Xu, K. Chen, S. Jiang, F. Wang, T. Wang, W. Chu, and Y. Qi, "SpellGCN: Incorporating phonological and visual similarities into language models for Chinese spelling check," in *ACL*, 2020, pp. 871–881.

[27] M. N. Sundararaman, A. Kumar, and J. Vepa, "PhonemeBERT: Joint Language Modelling of Phoneme Sequence and ASR Transcript," in *Interspeech*, 2021, pp. 3236–3240.

[28] P. Gurunath Shivakumar and P. Georgiou, "Confusion2vec: towards enriching vector space word representations with representational ambiguities," *PeerJ Computer Science*, p. e195, 2019.

[29] C.-W. Huang and Y.-N. Chen, "Learning asr-robust contextualized embeddings for spoken language understanding," in *ICASSP*, 2020, pp. 8009–8013.