Interspeech 2022
18-22 September 2022, Incheon, Korea

ASR-Robust Spoken Language Understanding on ASR-GLUE dataset

Lingyun Feng[1,*], *Jianwei Yu*[2,*], *Deng Cai*[2], *Songxiang Liu*[2], *Hai-Tao Zheng*[1,3,†], *Yan Wang*[2,†]

[1]Shenzhen International Graduate School, Tsinghua University, China
[2]Tencent AI lab, ShenZhen; [3]Peng Cheng Laboratory

fly19@mails.tsinghua.edu.cn, zheng.haitao@sz.tsinghua.edu.cn, yanwang.branden@gmail.com, tomasyu@tencent.com

Abstract

In recent years, with the increasing demand for voice interface applications, more and more attention has been paid to language understanding in speech systems. These speech-based intelligent systems usually comprise an automatic speech recognition (ASR) component and a natural language understanding (NLU) component which takes the output of the ASR component as input. Despite the rapid development of speech recognition over the past few decades, recognition errors are still inevitable, especially in noisy environments. However, the robustness of natural language understanding (NLU) systems to errors introduced by ASR is under-examined. In this paper, we propose three empirical approaches to improve the robustness of the NLU models. The first one is ASR correction which attempts to make error corrections for the mistranscriptions. The later two methods focus on simulating a noisy training scenario to train more robust NLU models. Extensive experimental results and analyses show that the proposed methods can effectively improve the robustness of NLU models.

Index Terms: Spoken language Understanding, Noise Robust

1. Introduction

Language understanding in speech-based systems has attracted much attention in recent years with the growing demand for voice interface applications and devices such as Alexa [1], Siri [2], and Cortana [3]. These speech-based intelligent systems usually comprise an automatic speech recognition (ASR) component which converts audio signals to readable natural language text, and a natural language understanding (NLU) component which takes the output of the ASR component as input and fulfills downstream tasks such as sentiment analysis, natural language inference, and response selection. The upstream ASR error can propagate to the downstream NLU component, degrading the overall performance [4]. In real-world scenarios, ASR error can be ubiquitous due to poor articulation and acoustic variability caused by environmental noise and reverberation [5]. The persistence of ASR error indicates a need for ASR-robust natural language understanding.

Previous work in this area is limited to task-oriented language understanding such as hotel reservation and meeting scheduling through human-machine interactions [6, 7, 8]. However, ASR error also affects those general NLU tasks, such as sentiment analysis and natural language inference. In [9], it has been found that ASR error can significantly degrade the performance of recent state-of-the-art (SOTA) NLU models, especially under challenging environment with strong acoustic noise and reverberation.

To alleviate the effect of ASR error, we need not only an accurate speech recognition system, but also a powerful and robust NLU model. In this work, we comprehensively studied three approaches to improve the robustness of NLU models to ASR error. The most straightforward approach is to introduce an additional ASR post-processing step to make error corrected. Due to the diversity of ASR errors, the correction model may still far from prefect, so the latter two methods, audio-level augmentation and text-level augmentation, try to simulate a noisy training scenario so that the NLU model can be more robust to ASR error. Specifically, both methods attempt to injects the ASR error to the NLU training corpus, so that NLU model can be trained on noisy training datasets. The former adopts a Text-to-Speech system (TTS) - ASR pipeline to first transforms the training corpus to audio and then convert them back to text (with errors). On the other hand, instead of the high-cost TTS-ASR pipeline, the later one uses text generation models [10] or a confusion-matrix-based method [11, 12] to directly generate erroneous text according to the input clean text. Experiments conducted on ASR-GLUE dataset demonstrate that the proposed methods can effectively improve the model robustness against ASR error on various downstream NLU tasks.

Our contributions are as follows: 1) In this work, we emphasize the importance of a robust NLU model and provide a comprehensive study of the sensitivity of state-of-the-art NLU models to ASR error. 2) Three approaches based on ASR correction and data augmentation are investigated in this paper. Experimental results demonstrate their effectiveness, and they can be considered as baseline methods on the ASR-GLUE dataset in future studies.

2. Related Work

Large-scale pretrained language models have achieved striking performance on NLU in recent years [13, 14]. Recently, many works test their robustness by human-crafted adversarial examples [15] or generated examples by adversarial attacks [13, 16, 17, 18]. [19] projects the input data to a latent space by generative adversarial networks (GANs), and then retrieves adversaries close to the original instance in the latent space. [20] proposes controlled paraphrase networks to generate syntactically adversarial examples that both fool pre-trained models and improve the robustness of these models to syntactic variation when used to augment their training data. However, the robustness of pre-trained model to speech recognition error in real conditions has not been fully explored.

*Equal contribution. †Corresponding authors. Work was done when Lingyun Feng is an intern at Tencent AI Lab. This research is partially supported by National Natural Science Foundation of China (Grant No. 6201101015) , Beijing Academy of Artificial Intelligence(BAAI) Natural Science Foundation of Guangdong Province Grant No. 2021A1515012640, the Basic Research Fund of Shenzhen City (Grant No. JCYJ20210324120012033 and JCYJ20190813165003837), and Overseas Cooperation Research Fund of Tsinghua Shenzhen International Graduate School (Grant No. HW2021008).

10.21437/Interspeech.2022-10097

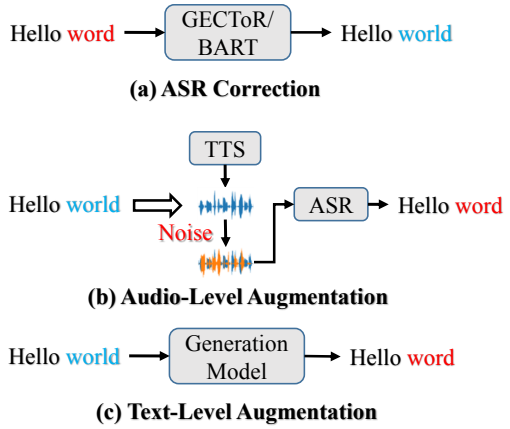

Figure 1: *An illustration of the proposed methods. (a) In ASR Correction, models recover the erroneous input "hello word" to correct text "hello world". (b) and (c): In these two augmentation methods, the source input is clean text data "hello world" and the output is noisy data with ASR error "hello word". They only differ in the way they simulate noise.*

3. ASR-Robust NLU

To mitigate the effects of ASR error on NLU models, we propose three empirical methods: ASR correction, Audio-Level Augmentation, and Text-Level Augmentation. The relationships and differences between them are shown in Figure 1. The first one (Section 3.1) tries to eliminate errors in the ASR hypothesis, so it trains models to recover clean text from erroneous text. The latter two methods follow the same intuition that NLU models trained in noisy scenarios would be more robust. To this end, we simulate two scenarios for training: 1) Audio-Level Augmentation (Section 3.2), using the TTS-ASR pipeline, converts the training corpus to audio files and then recognize them to text with ASR error; 2) Text-Level Augmentation (Section 3.3), using generation models [10, 21] and a confusing-matrix based method [11, 12] as a substitution of the TTS-ASR pipeline to direct generate erroneous text.

3.1. ASR Correction

Obviously, the most straightforward motivation to alleviate the effect of ASR error, is to eliminate them from the ASR hypothesis. To this end, we propose to add an extra ASR post-processing process to the ASR-NLU pipeline: After recognition, a correction model is adopted to recover the clean text from the ASR hypothesis that may contain some ASR errors. Pre-trained models adopted in ASR Correction are listed below:
1) **GETToR [22]** : A typical sequencing labeling model, and it is also the State-of-the-art (SOTA) grammatical error correction model. Since the task of grammatical error correction is really similar to our task, we directly take it as our starting point and fine-tune it on ASR data.
2) **Bart [21] (referred to as BART-C for disambiguation)** : a popular auto-regressive model with encoder-decoder architecture. It is fine-tuned to generate clean text when take the ASR hypothesis as input.

3.2. Audio-Level Augmentation

Although the ASR correction model can partly eliminate the errors in ASR hypothesis, due to the diversity of ASR error types, it is still far from perfect. We further attempt to simulate a noisy training scenario so that the NLU model can be more robust to ASR error. One straightforward way is to hire human speakers to record all NLU training corpora to audio files and then recognize them to text. However, it is unpractical due to the high labor cost, so we adopt a TTS system as the substitute for human speakers. As shown in Figure 1 (b), the detailed pipeline is as follows:
1) **Audio Recording**: a TTS system[1] is adopted to convert the text-form NLU training corpus in into audio files.
Noise simulation: different levels of environmental noise and reverberation are introduced into the audio files so that more ASR errors will be made in next step.
2) **Speech Recognition**: An ASR system is trained to recognize all audio files into text. In this way we got noisy NLU training corpora that contain ASR errors. The implementation details of Noise Simulation and Speech Recognition will be introduced in section 4.3.

In NLU model training, we use these augmented corpora as additional training data, along with the original clean training datasets to train the NLU model. Note that the original training dataset cannot be discarded in training, otherwise the model cannot maintain its original performance on clean text.

3.3. Text-Level Augmentation

In audio-level augmentation, the audio from the TTS system is still quite different from the human voice, which may cause a different error distribution. Besides, the TTS-ASR pipeline is time-consuming and expensive. So we propose text-level augmentation which uses text generation models as a substitution for the TTS-ASR pipeline and directly generates the "pseudo" ASR hypothesis from NLU training corpora.
Concretely, we adopt two pretrained language models and a Confusion-matrix-based method (abbreviated as **CM**) [12]) to generate the pseudo ASR hypothesis:
1) **Pretrained Models**: Two auto-regressive model, **GPT-2** [10] and BART [21] (referred to as **BART-S** for disambiguation), are used to generate ASR hypotheses. As sequence-to-sequence models, they take the ASR transcript as the input sequence and learn to output its corresponding hypothesis.
2) **CM**: In CM, each ASR transcript and its corresponding hypothesis are aligned at the word level by minimizing the Levenshtein distance between them. Then we conduct the confusion matrix based on the aligned n-grams and add ASR error according to the frequencies of confusions.

Similar to Text-level Augmentation, in training we merge these "pesudo" training corpora to the training datasets to train the NLU model. Further comparisons of the aforementioned methods are presented in experiments.

4. Experiment Setup

In this section, we first provide a brief introduction of the ASR-GLUE dataset used in our experiment. Then we present our implementation details.

4.1. ASR-GLUE dataset

The experiments in this paper are conducted on the ASR-GLUE dataset, which is constructed on the basis of GLUE [23]. Specifically, five different NLU tasks are contained in ASR-GLUE, including Sentiment classification (SST-2 [24]), Semantic Tex-

[1] https://cloud.tencent.com/product/tts

tual Similarity (STS-B [25]), paraphrase (QQP [2]), Question-answering NLI (QNLI [26]), Recognizing Textual Entailment (RTE [27]) and incorporate with a Science NLI task (SciTail [28]). The audio part of ASR-GLUE is produced by hiring six native speakers to convert the test data into audio recordings with 3 different levels of environment noise, which results in 32k utterance and 91.4h audio recording. ASR-GLUE is adopted as the dev and test set in following experiment. We follow the the original data partition of ASR-GLUE to split the dev and test set in our experiments. Details of ASR-GLUE can be found in [9].

4.2. Generation of training data

Noise Simulation: In both Text Correction and Audio-level Augmentation, we need to introduce some environmental noise into the speech so that hypotheses with more ASR errors can be made. Specifically, the background noise caused by such as phone ring, alarm clock and incoming vehicles are randomly sampled and added into the original LibriSpeech [29] utterance with the signal-to-noise-ratio (SNR) from 0dB to 15dB. In addition, the room reverberation is also introduced by involving the recorded audio signals with the Room Impulse Responses (RIRs) [3] generated by the image-source method [30]. The simulation process totally covers 843 kinds of different background noise and 417 types of different RIRs.

ASR: A 6000h trained LF-MMI TDNN ASR system [4] [31] is used to convert audio recordings into ASR hypothesis. This ASR system used 40 dimesnion high resolution mfcc feature as input and achieves 5.1% word error rate on the widely used WSJ[32] benchmark.

Speech Data In both Text Correction and Text-level Augmentation, we require paired transcriptions and hypotheses as the training data. We generate these data from the public 1000h LibriSpeech [29] dataset. After Noise introduction and Speech Recognition, we obtain the hypotheses corresponding to the transcriptions in LibriSpeech, which can be taken as the training data for correction or augmentation models.

4.3. Implementation details

For ASR correction methods, the input is hypothesis and the training target is its corresponding clean transcript. For augmentation-based methods, the input is the clean transcript while the output is its corresponding hypothesis which contains ASR error. In NLU training, the proportion between the augmented data and original data is 1:1 and we combine them together as the training set.

The NLU model we used in our experiment is BERT_{base} [5]. We use Adam [33] with an initial learning rate of 5e-5. For GPT-2 [6] and BART [7] used in text-level augmentation (BART-S) and ASR correction (BART-C), sentences are tokenized with byte-pair encoding (BPE) [34]. Both of them use beam search [35] as their decoding strategy. For GETToR, the sequence tagging model is an encoder made up of RoBERTa [36] stacked with two linear layers with softmax layers on the top. BPE is used for tokenization. Early stopping was used, stopping criteria was 3 epochs of 10K updates each without improvement.

[2] https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs

[3] The noise and RIR files can be found at http://www.openslr.org/resources/28/rirs_noises.zip

[4] https://github.com/kaldi-asr/kaldi

[5] https://huggingface.co/bert-base-uncased

[6] https://huggingface.co/gpt2

[7] https://huggingface.co/facebook/bart-large/tree/main

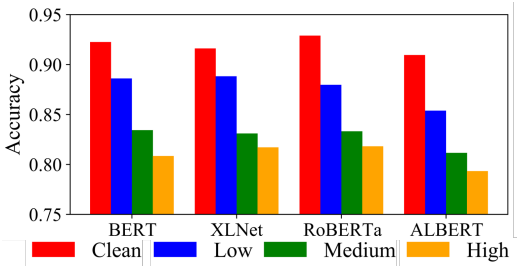

Figure 2: *Accuracy of different NLU models on SST-2 task. Here "Clean" stands for test on clean text data. "Low/Medium/High" stands for test in low/medium/high-level noise respectively.*

5. Experimental Results

In this section, we present our experimental results on ASR-GLUE test set. We first investigate the effect of ASR error on different NLU models. Then Bert is selected as a baseline to examine the effectiveness of proposed methods. The main results based on the proposed Audio-level and Text-level augmentation methods are present in Section 5.2. Finally the generalization of the proposed method are further evaluated on an industrial system, google ASR.

5.1. Performance of Existing NLU Models

To investigate how ASR error affects NLU models, we train NLU models with different model architectures. Specifically, we adopt base version of BERT [37], RoBERTa [36], ALBERT [38], XLNet [14] as the base model and train them on clean training data in ASR-GLUE. Then we test their performance under different noise levels. As shown in Fig. 2 we can observe that all these pretrained language models are sensitive to ASR error and the performance degrades with the increase of the noise level. Since these models performed similarly in the noisy environment, we selected Bert as their representative for follow-up experiments.

5.2. Results of proposed augmentation based methods

The main result of proposed methods are shown in Table 1, we can observe that the NLU models are sensitive to ASR error and the performance of NLU degrade severely across various task. The proposed methods can effectively improve the robustness of the model to a certain extent in most scenarios, but are still far from human performance. For example, the accuracy on SciTail task under high environment noise decline by 32.35% and is restored from 62.39% to 78.62% by audio-level augmentation method. In contrast, human are almost unaffected by the ASR error and still maintain high accuracy of 90.13%. Moreover, the human performance is more stable than the NLU model across all noise-levels for various tasks. We can also observe that although audio-level augmentation method achieves promising results on most tasks, it is even worse than the original BERT on RTE task under high environment noise (52.73% vs 52.86%).

We also find that the performance of different method varies across tasks, and it is difficult to find a universal method for all scenarios. Although the audio-level data augmentation method always gains the highest accuracy on the data with ASR errors, it can not maintain the original performance on clean data for certain tasks, e.g., worse than original BERT on clean text data

Table 1: *Baseline performance on the **ASR-GLUE** test sets. Here "Clean" stands for test on clean text data. "Low/Medium/High" stand for test on low-level/medium-level/high-level noise version respectively. Spearman correlations are reported for STS-B, and accuracy scores are reported for the other tasks. All values are scaled by 100.*

		BERT	Audio-level Augmentation	Text-level Augmentation			Correction		BEST(Δ)	Human
				CM	GPT-2	BART-S	GECToR	BART-C		
SST-2	Clean	92.25	92.90	91.61	90.96	92.90	92.25	92.25	92.90(+0.65)	-
	Low	88.60	89.35	86.99	87.31	89.78	87.74	87.85	89.78(+1.18)	90.40
	Medium	83.42	85.68	84.39	82.45	84.71	83.42	82.35	84.71(+1.29)	87.90
	High	80.84	81.92	81.59	81.16	81.16	81.70	78.69	81.70(+0.86)	86.18
STS-B	Clean	91.13	91.54	90.90	90.76	92.22	91.13	91.13	92.22(+1.09)	-
	Low	85.39	87.21	86.95	86.17	86.58	84.68	83.39	87.21(+1.82)	88.00
	Medium	69.93	75.49	74.16	73.14	71.38	68.38	69.73	75.49(+5.56)	86.59
	High	63.89	70.66	68.70	68.03	65.89	62.83	65.42	70.66(+6.77)	85.44
QQP	Clean	87.16	87.75	88.03	88.40	87.75	87.16	87.16	88.40(+1.24)	-
	Low	76.75	81.98	80.58	81.33	79.18	76.66	73.67	81.98(+5.23)	83.22
	Medium	69.47	76.94	73.76	75.07	70.40	70.59	67.51	76.94(+7.47)	81.02
	High	67.77	76.18	72.85	72.21	68.79	69.62	66.67	76.18(+8.41)	77.60
SciTail	Clean	94.74	90.79	94.74	94.74	91.45	94.74	94.74	94.74(+0.00)	-
	Low	75.33	85.53	85.31	85.31	77.63	79.61	80.26	85.53(+10.20)	90.82
	Medium	64.91	81.69	76.54	77.41	71.05	70.07	71.05	81.69(+16.78)	91.08
	High	62.39	78.62	74.34	73.90	69.52	67.98	69.08	78.62(+16.23)	90.13
QNLI	Clean	90.73	87.42	90.07	89.40	88.74	90.73	90.73	90.73(+0.00)	-
	Low	83.33	84.11	86.09	86.64	85.43	83.77	84.66	86.64(+3.31)	87.23
	Medium	78.48	82.23	84.11	84.44	83.33	79.80	79.91	84.44(+5.96)	86.29
	High	77.15	82.12	82.23	83.44	81.57	76.93	76.71	83.44(+6.29)	83.68
RTE	Clean	68.93	63.76	66.37	64.69	66.27	68.93	68.93	68.93(+0.00)	-
	Low	60.06	62.36	58.91	61.78	59.91	60.78	59.05	62.36(+2.30)	65.43
	Medium	53.87	57.33	54.89	60.78	55.60	56.35	53.59	60.78(+6.91)	64.21
	High	52.86	52.73	50.29	56.90	49.86	55.02	51.44	56.90(+4.04)	63.14

Table 2: *Effect of different ASR systems on STS-B.*

		Clean	Low	Medium	High
WER (Google ASR)		0%	7.1%	11.4%	12.5%
BERT(Google ASR)		91.13	86.97	83.65	82.20
Audio-level Augmentation		91.54	89.66	86.75	85.95
Text-level Augmentation	CM	90.90	89.00	85.82	84.63
	GPT-2	90.76	89.35	85.78	84.44
	BART-S	92.22	88.56	84.62	83.38
Correction	GECToR	91.13	86.29	83.13	82.04
	BART-C	91.13	86.48	83.80	82.06

(90.79% vs 94.74 on SciTail, 87.42% vs 90.73% on QNLI, and 63.76% vs 68.93% on RTE).

Another interesting phenomena is that the ASR error correction is less effective in many cases. On half of the tasks, such as SST-2, STS-B, QNLI, BERT with correction performs similar or even worse than the original one. One possible reason for its poor performance is that the ASR system already integrates a strong n-gram language model to guarantee the quality of system output. So an additional language model is redundant and cannot make further improvement.

By comparison on various tasks, we can observe that the text-level augmentation achieves more promising results than other methods. It works well in most situations, without degrading the NLU performance on clean text in general. We further make comparisons between the three text-level augmentation methods and find that none have absolute advantages. GPT-2 based augmentation performs better in most situations, but on RTE task with clean text, the accuracy is much lower than original BERT (64.69% vs 68.93%).

Overall, the proposed methods can effectively improve the model robustness to some extent, but a large gap still remains between these approaches and human, which indicates much potential for work on robust NLU models.

5.3. Generalization to industry ASR system

To verify this assumption that the ASR error may no longer be a serious problem in NLU for a good enough ASR system, we conduct a further experiment that replace our kaldi-based ASR system with a SOTA public ASR system: Google ASR. We replace the ASR system with Google ASR on dev and test sets (not on training set due to its high price) and test the performance of BERT on these new sets. As shown in Table 2, on STS-B task, the performance of most methods are consistent with the results in Table 1. We can observe the most data augmentation-based methods still perform well while the correction-based methods are less effective. Although in training these methods are based on another ASR system, they can generalize to Google ASR systems as well, which proves the generalization ability of augmentation-based methods.

6. Conclusion

On the basis of ASR-GLUE, we propose three empirical ways to improve robustness of the NLU model. Obviously these methods performs well, and effectively improve the robustness of NLU models to some extent. However, there is still a gap between the NLU capability of the model and humans, so this task is still far away from being solved. Given the difficulty of ASR-GLUE, we hope the proposed methods in this work can be regarded as baseline methods in this area, and expect more interesting studies in the future.

7. References

[1] L. Wang, M. Fazel-Zarandi, A. Tiwari, S. Matsoukas, and L. Polymenakos, "Data augmentation for training dialog models robust to speech recognition errors," arXiv preprint arXiv:2006.05635, 2020.

[2] J. D. Williams and S. Young, "Partially observable markov decision processes for spoken dialog systems," Computer Speech & Language, vol. 21, no. 2, pp. 393–422, 2007.

[3] S. Wang, T. Gunter, and D. VanDyke, "On modelling uncertainty in neural language generation for policy optimisation in voice-triggered dialog assistants," in 2nd Workshop on Conversational AI: Today's Practice and Tomorrow's Potential, NeurIPS, 2018.

[4] D. Serdyuk, Y. Wang, C. Fuegen, A. Kumar, B. Liu, and Y. Bengio, "Towards end-to-end spoken language understanding," in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018, pp. 5754–5758.

[5] R. Errattahi, A. El Hannani, and H. Ouahmane, "Automatic speech recognition errors detection and correction: A review," Procedia Computer Science, vol. 128, pp. 32–37, 2018.

[6] R. Schumann and P. Angkititrakul, "Incorporating asr errors with attention-based, jointly trained rnn for intent detection and slot filling," in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018, pp. 6059–6063.

[7] M. Rao, A. Raju, P. Dheram, B. Bui, and A. Rastrow, "Speech to semantics: Improve asr and nlu jointly via all-neural interfaces," arXiv preprint arXiv:2008.06173, 2020.

[8] C.-W. Huang and Y.-N. Chen, "Learning asr-robust contextualized embeddings for spoken language understanding," in ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020, pp. 8009–8013.

[9] 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.

[10] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," OpenAI blog, vol. 1, no. 8, p. 9, 2019.

[11] M. Fazel-Zarandi, L. Wang, A. Tiwari, and S. Matsoukas, "Investigation of error simulation techniques for learning dialog policies for conversational error recovery," arXiv preprint arXiv:1911.03378, 2019.

[12] J. Schatzmann, B. Thomson, and S. Young, "Error simulation for training statistical dialogue systems," in 2007 IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU). IEEE, 2007, pp. 526–531.

[13] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits, "Is bert really robust? a strong baseline for natural language attack on text classification and entailment," in Proceedings of the AAAI conference on artificial intelligence, vol. 34, no. 05, 2020, pp. 8018–8025.

[14] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," arXiv preprint arXiv:1906.08237, 2019.

[15] Y. Nie, A. Williams, E. Dinan, M. Bansal, J. Weston, and D. Kiela, "Adversarial nli: A new benchmark for natural language understanding," arXiv preprint arXiv:1910.14599, 2019.

[16] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," arXiv preprint arXiv:1706.06083, 2017.

[17] C. Zhu, Y. Cheng, Z. Gan, S. Sun, T. Goldstein, and J. Liu, "Freelb: Enhanced adversarial training for natural language understanding," arXiv preprint arXiv:1909.11764, 2019.

[18] X. Dong, A. T. Luu, R. Ji, and H. Liu, "Towards robustness against natural language word substitutions," in 9th International Conference on Learning Representations (ICLR), 2021.

[19] Z. Zhao, D. Dua, and S. Singh, "Generating natural adversarial examples," in 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net, 2018.

[20] M. Iyyer, J. Wieting, K. Gimpel, and L. Zettlemoyer, "Adversarial example generation with syntactically controlled paraphrase networks," arXiv preprint arXiv:1804.06059, 2018.

[21] 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," arXiv preprint arXiv:1910.13461, 2019.

[22] K. Omelianchuk, V. Atrasevych, A. Chernodub, and O. Skurzhanskyi, "Gector–grammatical error correction: Tag, not rewrite," arXiv preprint arXiv:2005.12592, 2020.

[23] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Glue: A multi-task benchmark and analysis platform for natural language understanding," arXiv preprint arXiv:1804.07461, 2018.

[24] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in Proceedings of EMNLP 2013, 2013, pp. 1631–1642.

[25] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia, "Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation," arXiv preprint arXiv:1708.00055, 2017.

[26] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," arXiv preprint arXiv:1606.05250, 2016.

[27] L. Bentivogli, P. Clark, I. Dagan, and D. Giampiccolo, "The fifth pascal recognizing textual entailment challenge." in TAC, 2009.

[28] T. Khot, A. Sabharwal, and P. Clark, "Scitail: A textual entailment dataset from science question answering," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1, 2018.

[29] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2015, pp. 5206–5210.

[30] E. A. Habets, "Room impulse response generator," Technische Universiteit Eindhoven, Tech. Rep, vol. 2, no. 2.4, p. 1, 2006.

[31] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz et al., "The kaldi speech recognition toolkit," in IEEE 2011 workshop on automatic speech recognition and understanding, no. CONF. IEEE Signal Processing Society, 2011.

[32] D. B. Paul and J. Baker, "The design for the wall street journal-based csr corpus," in Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992, 1992.

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

[34] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," arXiv preprint arXiv:1508.07909, 2015.

[35] P. Koehn, "Pharaoh: a beam search decoder for phrase-based statistical machine translation models," in Conference of the Association for Machine Translation in the Americas. Springer, 2004, pp. 115–124.

[36] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," arXiv preprint arXiv:1907.11692, 2019.

[37] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.

[38] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," arXiv preprint arXiv:1909.11942, 2019.