PSSAT: A Perturbed Semantic Structure Awareness Transferring Method for Perturbation-Robust Slot Filling

Guanting Dong[1]*, Daichi Guo[1]*, Liwen Wang[1]*, Xuefeng Li[1]*, Zechen Wang[1], Chen Zeng[1], Keqing He[2], Jinzheng Zhao[3], Hao Lei[1], Xinyue Cui[1], Yi Huang[4], Junlan Feng[4], Weiran Xu[1]*

[1]Beijing University of Posts and Telecommunications, Beijing, China
[2]Meituan Group, Beijing, China
[3]University of Surrey, UK
[4]China Mobile Research Institute

Abstract

Most existing slot filling models tend to memorize inherent patterns of entities and corresponding contexts from training data. However, these models can lead to system failure or undesirable outputs when being exposed to spoken language perturbation or variation in practice. We propose a perturbed semantic structure awareness transferring method for training perturbation-robust slot filling models. Specifically, we introduce two MLM-based training strategies to respectively learn contextual semantic structure and word distribution from unsupervised language perturbation corpus. Then, we transfer semantic knowledge learned from upstream training procedure into the original samples and filter generated data by consistency processing. These procedures aim to enhance the robustness of slot filling models. Experimental results show that our method consistently outperforms the previous basic methods and gains strong generalization while preventing the model from memorizing inherent patterns of entities and contexts.

1 Introduction

The slot filling (SF) task in the goal-oriented dialog system aims to identify task-related slot types in certain domains for understanding user utterances. Traditional supervised slot filling models and sequence labeling methods (Liu and Lane, 2015, 2016; Goo et al., 2018; Niu et al., 2019; He et al., 2020a,b; Wang et al., 2022a) have shown remarkable performance. However, these models tend to memorize inherent patterns of entities and contexts (Wang et al., 2022b; Lin et al., 2021). Faced with uncertainty and diversity of human language expression, the perturbation of entities and contexts will lead to a decrease in the generalization ability of the SF model, which hinders its further application in practical dialog scenarios.

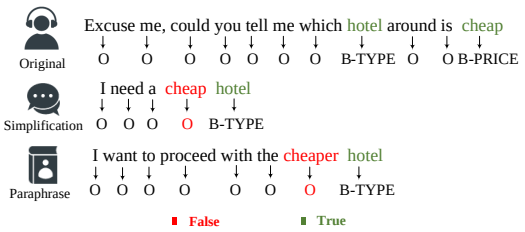

Figure 1: The impact of diverse spoken language perturbations on the slot filling system in real scenarios.

Due to the variety of expression habits, users may not interact with the dialogue system abiding by a rigid input mode in real dialog scenarios. Instead, the expression styles of users would be of high lexical and syntactic diversity while users express their intentions. An interesting finding is that, every expression retains the key semantic information of the sentence to ensure consistency of the intention, but it inevitably damages the semantic structure of the context. As shown in Figure 1, the original sentence comes from training data, while the other two sentences are real queries of users with different language habits. Firstly, paraphrase and simplification perturb the contextual semantic structure of the original sentence to various degrees. Secondly, some slot entities also suffer from word perturbations. However, they all retain price-related information to express the same intention. We refer to the above two perturbations collectively as Spoken Language Perturbation. The previous slot filling model, which tends to memorize entity patterns, has a significantly reduced generalization ability when faced with these situations. Therefore, it is necessary to train a robust slot filling model against perturbations in practical application.

Recently, improving the robustness of NLP systems against input perturbations has attracted increasing attention. Most existing studies (Wu et al., 2021; Moradi and Samwald, 2021; Gui et al., 2021) that explored the robustness problem are only about rule-based synthetic datasets, which have certain limitations. Further, Namysl et al. (2020) focused

*The first four authors contribute equally. Weiran Xu is the corresponding author. Email: dongguanting@bupt.edu.cn

Proceedings of the 29th International Conference on Computational Linguistics, pages 5327–5334
October 12–17, 2022.

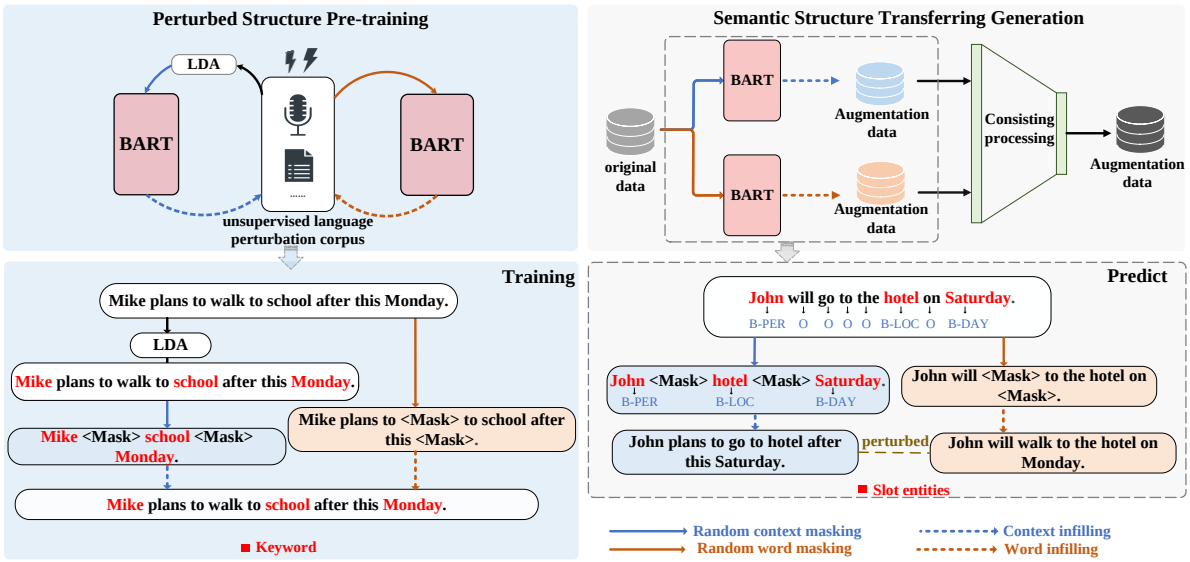

Figure 2: The overall architecture of the PSSAT framework. Two dotted boxes show the specific processes of the MLM-based strategies at pre-training and transferring generation stage, respectively.

on the robustness of the NER model against Optical Character Recognition (OCR) disturbance and misspellings. However, real-world dialogue systems face more diverse perturbations due to frequent interactions with users. Liu et al. (2020) proposed Language understanding augmentation, which contains four data augmentation methods, to simulate natural perturbations. Nevertheless, each method is designed for a specific perturbation, which cannot generalize for other unknown perturbations.

To solve the above issues, in this paper, we propose a **P**erturbed **S**emantic **S**tructure **A**wareness **T**ransferring method (**PSSAT**). It can generate augmented data based on human diversity expressions. In fact, it is not difficult to obtain unsupervised corpora containing spoken language perturbations in real-world scenarios (e.g. social media). Therefore, we extract the texts from two multi-modal datasets (Zhang et al., 2018; Lu et al., 2018) and construct an unsupervised language perturbation corpus, which helps the model learn the semantic structure of perturbed data. To be specific, we introduce a perturbed structure pre-training stage, which guides the model to directly learn contextual semantic structure and words distribution from unsupervised language perturbation corpus through two different MLM-based training strategies, respectively. To better eliminate the distribution gap between upstream and downstream data, we design a *Semantic Structure Transferring Generation* stage to transfer the upstream learned semantic structure knowledge to downstream original training samples. By doing so, the generated augmented samples are more in line with the spoken language perturbation. However, as there are mixed perturbations existed in upstream corpus, the model may generate some low-quality samples. To alleviate this problem, we introduce *Consistency Processing* to filter generated samples.

Our contributions are three-fold: (1) To the best of our knowledge, this is the first work to investigate spoken language perturbation of slot filling tasks and validate the vulnerability of existing rule-based methods in the condition of diverse language expressions. (2) We propose a perturbed semantic structure awareness transferring method, which transfers the learned contextual semantic structure and word distribution into the original samples through the MLM-based method. (3) Experiments demonstrate that our method outperforms all baseline methods and gains strong generalization while preventing the model from memorizing inherent patterns of entities and contexts.

2 Methodology

2.1 Problem Definition

Given a tokenized utterance $X = \{x_1, x_2, \ldots, x_N\}$ and its corresponding BIO format label $Y = \{y_1, y_2, \ldots, y_N\}$, we formulate the spoken language perturbation process in the real scenario as $X' = \mathcal{P}_x(X), Y' = \mathcal{P}_y(Y)$ such that $X' \neq X$ but Y' may be identical with Y or not. The perturbation-robust slot filling requires the model to be tested on the perturbed test dataset $\{(X', Y')\}$ but with no

access to the spoken language perturbation process $\mathcal{P}(\cdot)$ or perturbed data during the training phase.

2.2 Perturbed Structure Pre-training

The perturbed structure pre-training stage guides the model to learn the semantic structure from realistic perturbed data. We carefully collected several spoken language perturbation datasets to build an unsupervised language perturbation corpus[1]. Inspired by the key idea of masked language model (MLM) (Devlin et al., 2018), which randomly replaces a few tokens in a sentence with the special token *[MASK]* and recovers the original tokens by a neural network, we introduce two augmentation strategies, as shown in Figure 2.

Random Word Masking (RWM): words are randomly selected for masking and infilling to simulate the word perturbation, which guides the model to learn word distribution from real perturbed data.

Random Context Masking (RCM): we filter out the keywords of each sentence through Latent Dirichlet Allocation (LDA) (Blei et al., 2003) to keep the key information of the sentence. For non-keyword parts, we regard them as context spans of each sentence and conduct random masking and infilling. In this way, the model learns the contextual semantic structure from realistic perturbed data. Unlike word infilling, context infilling can generate multiple tokens for each *[MASK]* position.

2.3 Semantic Structure Transferring Generation

The Semantic Structure Transferring Generation stage aims to transfer learned contextual semantic structure and word distribution from upstream pre-trained model to downstream training samples. As shown in Figure 2, pre-trained models are separately loaded to conduct RWM and RCM. A slight difference from the pre-training stage is that slot entities are filtered out as keywords. It is worth noting that augmented data generated by two strategies explicitly contain diverse human expressions, which are learned from perturbed structure pre-training. Besides, we also generate coarse labels for two kinds of augmented data based on rules. Specifically, we label the infilling tokens as *O* while maintaining labels of other tokens. The case study (See Appendix D) shows that samples generated by semantic structure transferring generation can not only better fit spoken language perturbation, but also be more in line with human language diversity than those generated by rule-based methods.

[1]More details about the construction process of the perturbation corpus can be found at section 3.2

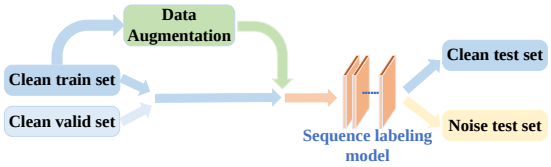

Figure 3: The process of downstream perturbation-robust slot filling task.

Consistency Processing Due to mixed perturbations in the upstream corpus, it is necessary to design a consistency processing to filter low-quality samples. Specifically, we train a tagging model with original training data and augmented samples. Then the model is used to predict labels for each augmented sentence. The labels which are consistent with the coarse labels and original labels are kept. The augmented samples filtered by consistency processing are mixed and input to the main task as the final augmented data.

Training and Inference As shown in figure 3, during the training stage, we first perform perturbed structure pre-training on the unsupervised language perturbation corpus to learn the contextual semantic structure and word distribution of perturbed data. We use the pre-trained model to obtain augmented data for the clean training dataset, and use all samples to train a perturbation-aware sequence labeling model. During the testing stage, we test the sequence labeling model on both clean and perturbed datasets.

3 Experiment

3.1 Dataset

RADDLE (Peng et al., 2020a) is a crowd-sourced diagnostic dataset to cover a broad range of real-world perturbations to study the robustness of end-to-end dialog system. We extract four kinds of realistic perturbed data from RADDLE and construct the slot filling dataset. In particular, the original dataset of the evaluation set in RADDLE is extracted from MultiWOZ (Lu et al., 2021). To introduce sufficient perturbed data for evaluating the model robustness against multiple perturbations, we extracted the clean user utterances and four kinds of perturbed utterances (Homophone, Simplify, Verbose and Paraphrase) from RADDLE. To be specific, **Homophone** perturbation comes from input text errors caused by recognition and synthesis errors. **Simplification** is generated by concise-

Methods	Clean	Homophone	Paraphrase	Verbose	Simplification	Overall
none	95.8	81.5 (-14.3)	87.5 (-8.3)	81.6 (-14.2)	85.3 (-10.5)	84 (-11.8)
Char-Random	96.0 (0.2)	84.1 (18.2%)	87.6 (1.2%)	83.2 (11.3%)	88.1 (26.7%)	85.8 (14.4%)
Word-Del	95.9 (0.1)	83.2 (11.9%)	89.3 (21.7%)	82.6 (7.0%)	87.5 (21.0%)	85.7 (15.4%)
Syn-Sub	96.1 (0.3)	83.5 (14.0%)	89.3 (21.7%)	82.2 (4.2%)	86.8 (14.3%)	85.5 (13.6%)
Word-Insert	95.8 (0.0)	81.2 (-2.1%)	88.2 (8.4%)	81.3 (-2.1%)	86.2 (8.6%)	84.2 (3.2%)
Hom-Sub	96.0 (0.2)	83.7 (15.4%)	89.3 (21.7%)	82.3 (4.9%)	87.7 (22.9%)	85.8 (16.3%)
NAT($\mathcal{L}_{aug}$)	96.0 (0.2)	84.3 (19.6%)	87.7 (2.4%)	82.8 (8.5%)	87.3 (19.0%)	85.5 (12.4%)
NAT($\mathcal{L}_{stabil}$)	96.0 (0.2)	83.9 (16.8%)	87.4 (-1.2%)	83.0 (9.9%)	87.3 (19.0%)	85.4 (11.1%)
PSSAT	96.2 (0.4)	**84.6 (21.7%)**	**90.1 (31.3%)**	**84.0 (16.9%)**	**89.3 (38.1%)**	**87.0 (27.0%)**
– RCM	96.2 (0.4)	83.8 (16.1%)	89.6 (25.3%)	83.5 (13.4%)	87.4 (20.0%)	86.1 (18.7%)
– RWM	96.3 (0.5)	83.3 (12.6%)	89.9 (28.9%)	83.8 (15.5%)	88.9 (34.3%)	86.5 (22.8%)
– CP	96.3 (0.5)	84.0 (17.5%)	90.0 (30.1%)	83.4 (12.7%)	88.3 (28.6%)	86.4 (22.2%)
– Pre-training	95.9 (0.1)	83.1 (11.2%)	89.4 (22.9%)	83.0 (9.9%)	86.9 (15.2%)	85.6 (14.8%)

Table 1: The performance (F1 score) of the PSSAT on RADDLE. For cells in *Baseline* row and *Clean test* column, the numbers in the parenthesis indicate the change of F1 score. For other cells, the numbers in the parenthesis indicate p_r. In Overall column, we calculate the average F1 and p_r of the four Spoken language perturbations, respectively. Both the best and the worst are marked, "−" denotes the model performance without a specific module. RWM, RCM, CP denotes Random Word Masking, Random Context Masking and Consistency Processing.

Method	Hom+App	Hom+Con	Con+App	Hom+Con+App
Baseline (LSTM)	47.9 (-46.0)	54.2 (-39.7)	73.4 (-20.5)	45.7 (-48.2)
best baseline	53.6 (12.4%)	61.1 (17.4%)	71.6 (-8.8%)	47.2 (3.1%)
PSSAT	59.6 (25.4%)	61.8 (19.1%)	78.3 (23.9%)	53.9 (17.0%)

Table 2: The performance of the best baseline and PSSAT on mixed perturbations.

word expression. On the contrary, **Verbose** refers to redundant expression. **Paraphrase** noise widely exists in our dataset, where users restate texts in different ways of expression according to their personal speaking habits. The training dataset consists of 61,117 clean data from four domains. We randomly select 5,000 data as the validation set. Our compared baselines and implementation details can be found in Section 3.4 and A.

3.2 Unsupervised Language Perturbation Corpus

In our perturbed structure pre-training stage, we employ two multi-modal datasets: Twitter-2015 (Zhang et al., 2018), Twitter-2017 (Lu et al., 2018). We only extract the corpus part and delete the useless details in sentences such as emoji and URL. We consider that the data on social media contains the real diversity of human expressions, and it is beneficial for the downstream generation to learn the knowledge of diverse human expressions in the pre-training stage.

3.3 Evaluation Metrics

We use F_1 score to measure the performance of the model. ${F_1}^c$, ${F_1}^p$ denote the performance on the clean and perturbed test set respectively. On this basis, we define (1) as Perturbation Recovery Rate (P_r) of a given perturbation-robust method m:

$$P_r = \frac{{F_1}^p_m - {F_1}^p_{\text{baseline}}}{{F_1}^c_{\text{baseline}} - {F_1}^p_{\text{baseline}}} \quad (1)$$

P_r indicates the improvement in performance of the model using the robust approach over the baseline model on the perturbed test set, as a percentage of the performance degradation of the baseline model due to the introduction of perturbation.

3.4 Implementation Details

For the upstream work, our model **PSSAT** is based on BART (Lewis et al., 2019), which is provided by the Huggingface Transformers[2]. The reason for choosing BART is that the pre-training tasks of BART include token masking and text filling, which is consistent with our PSSAT task. We set the batch size of BART to 8 and the pre-training takes an average of one hour for 10 epochs. The corresponding learning rates are set to 1e-5.

For the downstream work, we use two settings for perturbation-robust slot filling, Glove-Bi-LSTM and BERT-Bi-LSTM. Glove-6B-300d, char embedding and BERT-large-uncased are applied as the embedding layer. We take Bi-LSTM as the mainly analyzed model. The hidden size of Bi-LSTM is set to 128 and the dropout rate is set to 0.2. The transform probability p is set to 0.3. For all the experiments, we train and test our model on the 2080Ti GPU. It takes an average of 1.5 hours to run with 12 epochs on the training dataset.

[2]https://huggingface.co/docs/transformers

All experiments are repeated three times with different random seeds under the same settings. All the models are implemented with PyTorch (Paszke et al., 2019).

3.5 Main Results

Table 1 shows the main results of PSSAT compared to different baselines on the language perturbation dataset. The overall result of our PSSAT greatly outperforms the baseline by 27.0%. Especially, the P_r of paraphrase and simplification is about 40%, which is a remarkable enhancement. What's more, our method is not designed for any specific perturbation, but achieves the best results for various perturbations, which proves that our model not only improves the performance significantly, but generalizes better.

Ablation Studies. To better prove the effectiveness of the pre-training stage, we conduct ablation experiments. Table 1 illustrates the results that the model without RWM performs better than that without RCM, which shows that the change of context makes the semantic change more drastic. Meanwhile, all of RWM, RCM, CP and PSSAT without pre-training have a performance drop, which suggests that every part of design is necessary.

3.6 Mixed Perturbations Experiment

In real dialogue scenarios, mixed perturbations often appear in one input utterance at the same time. To verify the effectiveness of our method in more realistic scenarios, based on SNIPS (Coucke et al., 2018), we utilize TextFlint[3] (Gui et al., 2021) to introduce Homophone(Hom), Appendirr(App), ConcatSent(Con) and construct a mixed perturbations evaluation dataset [4]. As shown in Table 2 , the P_r of our PSSAT is over 20% against three different kinds of two-level perturbations, which far exceeds the best baseline (Hom-Sub). The model maintains an almost 17% P_r even with the joint disturbances from three-level perturbations, which shows the effectiveness and stability of our methods in real scenarios.

3.7 Error Analysis

We randomly selected 500 samples from all outputs and manually checking the error outputs for error analysis. Table 3 investigates 5 error types the model has made on the RADDLE. It can be seen that the number of PSSAT error outputs is less than the baseline in each category. Table 4 illustrates cases of each error type. Both the baseline model and PSSAT can correctly label clean text, but only PSSAT can correctly label texts with perturbation. After comprehensive analysis, the result shows that rote memorization of entity mention and contextual perturbation accounts for a large portion of the errors. Compared to the baseline, PSSAT can alleviate the problem of memorizing inherent patterns of entities and contexts.

Error Type	Baseline		PSSAT	
	Num	%	Num	%
Entity Location	12	20.0	9	18.8
Contextual Perturbation	16	26.7	11	22.8
Entity Mention	23	38.3	19	39.6
Others	9	15.0	9	18.8
Mixed Perturbation	11	–	9	–

Table 3: Error analysis on RADDLE.

Clean	are there any **museums** in the **centre** ?
Verbosity	could you please search for any **museums** in the town **centre** .
Baseline	O O O O O O B-type O O O O O
PSSAT	O O O O O O B-type O O O B-area O
Clean	i 'd like a **jamaican** restaurant please .
Simplification	find **jamaican** plz .
Baseline	O B-name I-name O
PSSAT	O B-food O O
Clean	i need to leave after **12:00** .
Homophone	i need to leave after **twelve** .
Baseline	O O O O O O O
PSSAT	O O O O O B-leave O
Clean	i need a booking for **4** people .
Paraphrase	i need seats for **4** .
Baseline	O O O O B-time O
PSSAT	O O O O B-people O
Clean	could you tell me which hotel around is **cheap** ?
Paraphrase	I want to proceed with the **cheaper** hotel .
Baseline	O O O O O O O B-type O
PSSAT	O O O O O O B-price B-type O

Table 4: The error cases. The bold texts are slot entities. Both wrong and correct labels are marked in red and green, respectively.

4 Conclusion

In this paper, we propose a perturbed semantic structure awareness transferring method for perturbation-robust slot filling task. Specifically, we design the perturbed structure pre-training and the semantic structure transferring generation to transfer the upstream learned semantic structure knowledge to downstream original training samples. Further, we filter low-quality samples through a consistency processing module. Sufficient experiments and error analysis demonstrate the effectiveness and generalization of our methods, and also prove that PSSAT alleviates the problem of memorizing inherent patterns of entities and contexts.

[3] http://textflint.io/

[4] We conducted single perturbation experiment on SNIPS. The results can be found in Appedix C

Acknowledgements

We thank all anonymous reviewers for their helpful comments and suggestions. This work was partially supported by MoE-CMCC "Artifical Intelligence" Project No. MCM20190701, National Key R&D Program of China No. 2019YFF0303300 and Subject II No. 2019YFF0303302, DOCOMO Beijing Communications Laboratories Co., Ltd.

References

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces.

Claude Coulombe. 2018. Text data augmentation made simple by leveraging nlp cloud apis.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757.

Tao Gui, Xiao Wang, Qi Zhang, Qin Liu, Yicheng Zou, Xin Zhou, Rui Zheng, Chong Zhang, Qinzhuo Wu, Jiacheng Ye, Zexiong Pang, Yongxin Zhang, Zhengyan Li, Ruotian Ma, Zichu Fei, Ruijian Cai, Jun Zhao, Xingwu Hu, Zhiheng Yan, Yiding Tan, Yuan Hu, Qiyuan Bian, Zhihua Liu, Bolin Zhu, Shan Qin, Xiaoyu Xing, Jinlan Fu, Yue Zhang, Minlong Peng, Xiaoqing Zheng, Yaqian Zhou, Zhongyu Wei, Xipeng Qiu, and Xuanjing Huang. 2021. Textflint: Unified multilingual robustness evaluation toolkit for natural language processing.

Keqing He, Weiran Xu, and Yuanmeng Yan. 2020a. Multi-level cross-lingual transfer learning with language shared and specific knowledge for spoken language understanding. *IEEE Access*, 8:29407–29416.

Keqing He, Yuanmeng Yan, and Weiran Xu. 2020b. Learning to tag oov tokens by integrating contextual representation and background knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 619–624.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Bill Yuchen Lin, Wenyang Gao, Jun Yan, Ryan Moreno, and Xiang Ren. 2021. Rockner: A simple method to create adversarial examples for evaluating the robustness of named entity recognition models. *arXiv preprint arXiv:2109.05620*.

Bing Liu and Ian Lane. 2015. Recurrent neural network structured output prediction for spoken language understanding. In *Proc. NIPS Workshop on Machine Learning for Spoken Language Understanding and Interactions*.

Bing Liu and Ian R. Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. In *INTERSPEECH*.

Jiexi Liu, Ryuichi Takanobu, Jiaxin Wen, Dazhen Wan, Hongguang Li, Weiran Nie, Cheng Li, Wei Peng, and Minlie Huang. 2020. Robustness testing of language understanding in task-oriented dialog. *arXiv preprint arXiv:2012.15262*.

Di Lu, Leonardo Neves, Vitor Carvalho, Ning Zhang, and Heng Ji. 2018. Visual attention model for name tagging in multimodal social media. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1990–1999.

Hengtong Lu, Zhuoxin Han, Caixia Yuan, Xiaojie Wang, Shuyu Lei, Huixing Jiang, and Wei Wu. 2021. Slot transferability for cross-domain slot filling. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4970–4979, Online. Association for Computational Linguistics.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Milad Moradi and Matthias Samwald. 2021. Evaluating the robustness of neural language models to input perturbations. *arXiv preprint arXiv:2108.12237*.

Marcin Namysl, Sven Behnke, and Joachim Köhler. 2020. Nat: noise-aware training for robust neural sequence labeling. *arXiv preprint arXiv:2005.07162*.

Peiqing Niu, Zhongfu Chen, Meina Song, et al. 2019. A novel bi-directional interrelated model for joint intent detection and slot filling. *arXiv preprint arXiv:1907.00390*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037.

Baolin Peng, Chunyuan Li, Zhu Zhang, Chenguang Zhu, Jinchao Li, and Jianfeng Gao. 2020a. Raddle: An evaluation benchmark and analysis platform for robust task-oriented dialog systems. *arXiv preprint arXiv:2012.14666*.

Baolin Peng, Chenguang Zhu, Michael Zeng, and Jianfeng Gao. 2020b. Data augmentation for spoken language understanding via pretrained models. *arXiv e-prints*, pages arXiv–2004.

Liwen Wang, Rumei Li, Yang Yan, Yuanmeng Yan, Sirui Wang, Wei Yu Wu, and Weiran Xu. 2022a. Instructionner: A multi-task instruction-based generative framework for few-shot ner. *ArXiv*, abs/2203.03903.

Xiao Wang, Shihan Dou, Limao Xiong, Yicheng Zou, Qi Zhang, Tao Gui, Liang Qiao, Zhanzhan Cheng, and Xuanjing Huang. 2022b. Miner: Improving out-of-vocabulary named entity recognition from an information theoretic perspective. *arXiv preprint arXiv:2204.04391*.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.

Di Wu, Yiren Chen, Liang Ding, and Dacheng Tao. 2021. Bridging the gap between clean data training and real-world inference for spoken language understanding. *arXiv preprint arXiv:2104.06393*.

Qi Zhang, Jinlan Fu, Xiaoyu Liu, and Xuanjing Huang. 2018. Adaptive co-attention network for named entity recognition in tweets. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

A Baselines

To simulate the input perturbation existing in realistic scenarios, we introduce five well-designed perturbation robust methods and a strong baseline:

Random Char Augmentation (Char-Random) is a character-level augmentation method that randomly adds, removes, and replaces characters in a token with a transformation probability p.

Random Word Deletion (Word-Del) aims to simulate the effect of simplification in input utterances in real-world scenarios (Wei and Zou, 2019). It randomly removes tokens with a probability p.

Random Word Insertion (Word-Insert) randomly insert words with probability p based on contextual embedding (Peng et al., 2020b). The method aims to model the effect of verbosity perturbation in input utterances.

Homophonic substitution (Hom-Sub) is designed for simulating word-level perturbation. We implement a homophone replacement dictionary, where words in the utterance are replaced by homophones with probability p.

Synonymous Substitution (Syn-Sub) is implemented based on WordNet's (Miller, 1995) synonymous thesaurus. We randomly select tokens in utterance with probability p for synonymous substitution (Coulombe, 2018). Note that our augmentations on training samples avoid slot words and only operate on contextual words.

Noise-Aware Training is proposed by (Namysl et al., 2020), which includes two Noise-Aware Training (NAT) objectives that improve robustness of sequence labeling performed on perturbed input. The data augmentation method trains a neural model using a mixture of clean and noisy samples, whereas the stability training algorithm encourages the model to create a noise-invariant latent representation.

B BERT Result on RADDLE

Table 5 shows the BERT-version results of PSSAT. Compared to several data augmentation methods, PSSAT method makes a great improvement in each field. The overall results are better than any type of data augmentation results. Furthermore, the whole PSSAT method outperforms the baseline by 18.8%. Similar to the results of LSTM, PSSAT also achieves the best results on each spoken language perturbation.

C SNIPS Single Perturbation Experiment

As shown in Table 6, we also explore the performance of various denoising methods on SNIPS dataset. Both entity mention and contextual semantics are corrupted in mixed multiple noise scenarios, resulting in a catastrophic degradation of model performance. The overall result combining the single-noise and multi-noise results achieves a 33% improvement.

D Case Study

Table 7 shows some samples generated by PSSAT in the way of RWM and RCM, respectively. It can be seen that the generated augmented samples are more in line with the Spoken language perturbation, while preserving the semantics of the original sentences.

Methods	Clean	Homophone	paraphrase	verbose	simplification	Overall
none	96.2	82.8(-13.4)	90.4(-5.8)	84.4(-11.8)	87.7(-8.5)	82.6(-13.6)
Char-Random	96.0 (-0.2)	85.0 (16.4%)	89.9 (-8.6%)	84.9 (4.2%)	88.1 (4.7%)	86.9 (4.2%)
Word-Del	95.9 (-0.3)	84.5 (12.7%)	90.0 (-6.9%)	84.5 (0.8%)	88.0 (3.5%)	86.8 (2.5%)
Word-Sub	96.3 (0.1)	84.1 (9.7%)	90.2 (-3.4%)	84.1 (-2.5%)	88.2 (5.9%)	86.7 (2.4%)
Word-Insert	96.3 (0.1)	84.3 (9.7%)	90.5 (1.7%)	83.9 (-4.2%)	88.5 (9.4%)	86.8 (4.2%)
Homophone	95.8 (-0.4)	**85.8 (22.4%)**	90.2 (-3.4%)	82.4 (-16.9%)	87.5 (-2.4%)	86.5 (-0.1%)
NAT($\mathcal{L}_{aug}$)	96.0 (0.2)	85.2 (17.7%)	90.5 (2.4%)	85.4 (8.3%)	88.0 (3.0%)	87.2 (7.9%)
NAT($\mathcal{L}_{stabil}$)	96.0 (0.2)	85.1 (16.8%)	90.3(-1.2%)	85.2 (6.6%)	88.0 (3.0%)	87.2 (6.3%)
PSSAT	96.4 (0.2)	85.6 (20.9%)	**91.5(19.0%)**	**85.8(11.9%)**	**89.7(23.5%)**	**88.1(18.8%)**
– RCM	96.6 (0.4)	84.7 (14.0%)	91.3 (15.5%)	85.1 (5.9%)	88.4 (8.2%)	87.4(10.9%)
– RWM	96.4 (0.2)	83.5(5.2%)	**91.5(19.0%)**	85.7(11.0%)	89.4(20.0%)	87.5(13.8%)
– Pre-training	95.9 (0.1)	83.1(2.2%)	90.7(4.9%)	84.9(4.4%)	88.2(5.6%)	86.7(4.3%)

Table 5: The performance (F1 score) of the PSSAT on RADDLE. For cells in *Baseline* row and *Clean test* column, the numbers in the parenthesis indicate the change of F1 score over the baseline (96.2), while for other cells, the numbers in the parenthesis indicate the perturbation recovery rate (p_r). In Overall column, we calculate the average F1 and p_r of the four Spoken language perturbations respectively.

Method	Clean	Hom	App	Concat	Overall
Baseline (LSTM)	93.9	62.2(-31.7)	71.2(-22.7)	85.0(-8.9)	72.8(-21.1)
Char-Random	93.7	**75.8(42.9%)**	74.0(12.3%)	85.2(2.2%)	78.3(19.1%)
Word-Del	93.8	61.6(-1.9%)	69.2(-8.8%)	85.3(3.4%)	72.0(-2.4%)
Word-Sub	93.8	65.7(11.0%)	73.3 (9.3%)	84.1 (-10.1%)	74.4(3.4%)
Word-Insert	92.8	63.9 (5.4%)	80.5 (41.0%)	82.1 (-32.6%)	75.5(4.6%)
Homephone	93.7	70.1 (24.9%)	72.8 (7.0%)	86.4 (15.7%)	76.4(15.9%)
NAT($\mathcal{L}_{aug}$)	93.6	69.1 (21.8%)	74.7 (15.3%)	85.5 (5.5%)	76.4 (14.2%)
NAT($\mathcal{L}_{stabil}$)	93.6	68.4 (19.6%)	74.3 (13.8%)	85.4 (4.7%)	76.0 (12.7%)
PSSAT	94.14	71.5(29.3%)	**82.7(50.7%)**	**86.7(19.1%)**	**80.3(33.0%)**

Table 6: The performance of the best baseline and PSSAT on mixed perturbations. Con, APP and Home stand for ConcatSent, Appendirr and Homophone, respectively.

	Ori.	Aug.
Text	can you please check for a **turkish restaurant** ? does it have **4 stars** ? **5** people for the train please . i 'll be leaving **kings lynn** after **13:15** .	so can you show me some **turkish restaurant** ? does that rated **4 stars** ? **5** tkts R needed please . i 'm gonna leave **kings lynn** @ **13:15** .
Word	ok , how about **scudamores punting company** then . how about a **museum** ? i am looking for a **hotel** please .	@HeyandIfhey , how about **scudamores punting company** then . how is a **museum** ? i am sorry for a **hotel** please .

Table 7: Some raw data and the corresponding enhanced data.