Robustness Testing of Language Understanding in Dialog Systems

Jiexi Liu[1*], Ryuichi Takanobu[1*], Jiaxin Wen[1], Dazhen Wan[1],
Weiran Nie[2], Hongyan Li[2], Cheng Li[2], Wei Peng[2], Minlie Huang[1†]
[1]DCST, BNRIST, Tsinghua University, Beijing, China
[2]Huawei Technologies, Shenzhen, China
{liujiexi19, gxly19}@mails.tsinghua.edu.cn, aihuang@tsinghua.edu.cn

Abstract

Most language understanding models in dialog systems are trained on a small amount of annotated training data, and evaluated in a small set from the same distribution. However, these models can lead to system failure or undesirable outputs when being exposed to natural perturbation in practice. In this paper, we conduct comprehensive evaluation and analysis with respect to the robustness of natural language understanding models, and introduce three important aspects related to language understanding in real-world dialog systems, namely, *language variety*, *speech characteristics*, and *noise perturbation*. We propose a model-agnostic toolkit LAUG to approximate natural perturbation for testing the robustness issues in dialog systems. Four data augmentation approaches covering the three aspects are assembled in LAUG, which reveals critical robustness issues in state-of-the-art models. The augmented dataset through LAUG can be used to facilitate future research on the robustness testing of language understanding in dialog systems.

1 Introduction

Recently task-oriented dialog systems have been attracting more and more research efforts (Gao et al., 2019; Zhang et al., 2020b), where understanding user utterances is a critical precursor to the success of such dialog systems. While modern neural networks have achieved state-of-the-art results on language understanding (LU) (Wang et al., 2018; Zhao and Feng, 2018; Goo et al., 2018; Liu et al., 2019; Shah et al., 2019), their robustness to changes in the input distribution is still one of the biggest challenges in practical use.

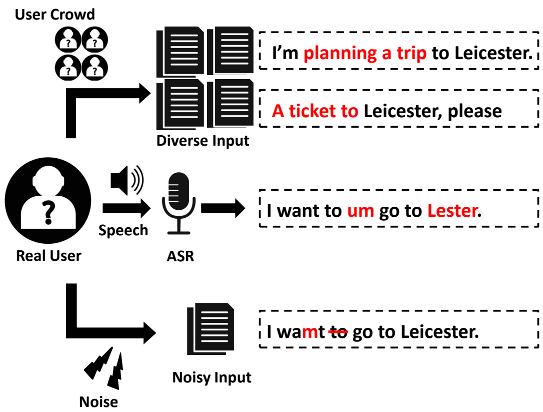
Real dialogs between human participants involve language phenomena that do not contribute so much to the intent of communication. As shown in Fig. 1, user expressions can be of high lexical and syntactic diversity when a system is deployed to users; typed texts may differ significantly from that recognized from voice speech; interaction environments may be full of chaos and even users themselves may introduce irrelevant noises such that the system can hardly get clean user inputs.

Unfortunately, neural LU models are vulnerable to these natural perturbations that are legitimate inputs but not observed in training data. For example, Bickmore et al. (2018) found that popular conversational assistants frequently failed to understand real health-related scenarios and were unable to deliver adequate responses on time. Although many studies have discussed the robustness of LU (Ray et al., 2018; Zhu et al., 2018; Iyyer et al., 2018; Yoo et al., 2019; Ren et al., 2019; Jin et al., 2020; He et al., 2020), there is a lack of systematic studies for real-life robustness issues and corresponding benchmarks for evaluating task-oriented dialog systems.

In order to study the real-world robustness issues, we define the LU robustness from three aspects: *language variety*, *speech characteristics* and *noise perturbation*. While collecting dialogs from deployed systems could obtain realistic data distribution, it is quite costly and not scalable since numerous conversational interactions with real users are required. Therefore, we propose an automatic method LAUG for **L**anguage understanding **AUG**mentation in this paper to approximate the natural perturbation and introduce it into existing data. LAUG is a black-box testing toolkit on LU robustness composed of four data augmentation methods, including word perturbation, text paraphrasing, speech recognition, and speech disfluency. We instantiate LAUG on a multi-domain dia-

*Equal contribution.
†Corresponding author.

(a) Dataset construction



(b) Real-world application

Figure 1: Difference between dialogs collected for research and those for real-world applications.

log corpus MultiWOZ (Budzianowski et al., 2018) to demonstrate the toolkit's effectiveness. Human evaluation indicates that the sentences augmented by LAUG are reasonable and appropriate with regards to each augmentation approach's target. A number of LU models with different categories and training paradigms are tested as base models with in-depth analysis. Experiments indicate a sharp performance decline in most baselines in terms of each robustness aspect. Since our toolkit is model-agnostic and does not require model parameters or gradients, the augmented data can be easily obtained for both training and testing towards building a robust system.

In summary, our contributions fall into the following:

- We test the robustness of language understanding (LU) models systematically from three aspects that occur in real-world dialog, including linguistic variety, speech characteristics and noise perturbation.
- We propose a general, model-agnostic toolkit *LAUG*, which is an integration of four data augmentation methods on LU that covers the three aspects. This toolkit is applicable for augmenting both training and test data.
- We conduct in-depth analysis of LU robustness on a multi-domain corpus with a variety of baselines and standardized evaluation measures. The augmented dataset can be used for further research in task-oriented dialog systems[1].

2 Robustness Type

We summarize several common interleaved challenges in language understanding from three aspects, as shown in Fig. 1b:

Language Variety A modern dialog system in a text form has to interact with a large variety of real users. The user utterances can be characterized by a series of linguistic phenomena with a long tail of variations in terms of spelling, vocabulary, lexical/syntactic/pragmatic choice (Ray et al., 2018; Jin et al., 2020; He et al., 2020; Zhao et al., 2019; Ganhotra et al., 2020).

Speech Characteristics The dialog system can take voice input or typed text, but these two differ in many ways. For example, written language tends to be more complex and intricate with longer sentences and many subordinate clauses, whereas spoken language can be full of repetitions, incomplete sentences, self-corrections and interruptions (Wang et al., 2020a; Park et al., 2019; Wang et al., 2020b; Honal and Schultz, 2003; Zhu et al., 2018).

Noise Perturbation Most dialog systems are trained only on noise-free interactions. However, there are various noises in the real world, including background noise, channel noise, misspelling, and grammar mistakes (Xu and Sarikaya, 2014; Li and Qiu, 2020; Yoo et al., 2019; Henderson et al., 2012; Ren et al., 2019).

3 Language Understanding Augmentation

This section introduces commonly observed out-of-distribution data in real-world dialog into an existing corpus. We approximate natural perturbation in an automatic way instead of collecting real data by asking users to converse with a dialog system.

To achieve our goals, we propose a toolkit *LAUG*, for black-box evaluation of LU robustness. It is an ensemble of four data augmentation approaches, including Word Perturbation (WP), Text Paraphrasing (TP), Speech Recognition (SR), and Speech Disfluency (SD). Note that LAUG is model-agnostic and can be applied to any LU dataset theoretically. Each augmentation approach tests one

[1]The data, toolkit, and codes will be merged into `https://github.com/thu-coai/ConvLab-2` in the future.

or two proposed aspects of robustness as Table 1 shows. The intrinsic evaluation of the chosen approaches will be given in Sec. 4.

Capacity	LV	SC	NP
Word Perturbation (WP)	√		√
Text Paraphrasing (TP)	√		
Speech Recognition (SR)		√	√
Speech Disfluency (SD)		√	

Table 1: The capacity that each augmentation method evaluates, including Language Variety (LV), Speech Characteristics (SC) and Noise Perturbation (NP).

Task Formulation Given the paired data $\{X_t, y_t\}$ where $X_t = \{x_{2t-m}, \ldots, x_{2t-1}, x_{2t}\}$ is the dialog context at dialog turn t (each x is an utterance), and y_t is the dialog act (DA) of x_{2t}. m is the size of sliding window that controls the length of utilizing dialog history. Empirically, we set $m = 2$ in the experiment. Let $\mathcal{U}, \mathcal{S}$ denote the set of user/system utterances, respectively. Then, we have $x_{2t-2i} \in \mathcal{U}$ and $x_{2t-2i-1} \in \mathcal{S}$. The task of this paper is to examine different LU models whether they can predict y_t correctly given perturbed inputs $\tilde{X}_t$. The perturbation is only performed on user utterances.

Word Perturbation Inspired by EDA (*Easy Data Augmentation*) (Wei and Zou, 2019), we propose its semantically conditioned version, SC-EDA, which considers task-specific augmentation operations in LU. SC-EDA injects word-level perturbation into each utterance x and updates its corresponding semantic label y.

Table 2 shows an example of SC-EDA. EDA randomly performs one of the four operations, including *synonym replacement*, *random insertion*, *random swap* and *random deletion*[2]. Note that, to keep the label unchanged, SC-EDA does not modify words that are related to slot values of dialog acts in these four operations. Additionally, we design *slot value replacement*, which changes the utterance and the label at the same time. First of all, some slot values are randomly picked. Each picked slot value is replaced by another value in the database with the same slot name. For example in Table 2, "Cambridge" is replaced by "Liverpool", where both belong to the same slot name "dest" (destination).

Synonym replacement and *slot value replacement* aim at increasing the language variety, while *random word insertion/deletion/swap* test robustness of noise perturbation. From another perspective, four operations from EDA perform an Invariance test, while *slot value replacement* conducts a Directional Expectation test according to CheckList (Ribeiro et al., 2020).

Original	I want to go to Cambridge .
DA	attraction { inform (dest = Cambridge) }
Syno.	I wishing to go to Cambridge .
Insert	I need want to go to Cambridge .
Swap	I to want go to Cambridge .
Delete	I want ~~to~~ go to Cambridge .
SVR	I want to go to Liverpool .
DA	attraction { inform (dest = Liverpool) }

Table 2: A SC-EDA example. Syno., Insert, Swap and Delete are four operations described in EDA. The dialog act of the four utterances augmented with EDA, is identical to the original one. SVR denotes *slot value replacement*.

Text Paraphrasing The target of text paraphrasing is to generate a new sentence $x' \neq x$ while maintaining its dialog act unchanged, i.e. $y' = y$. We applied SC-GPT (Peng et al., 2020), a finetuned language model conditioned on the dialog acts, to paraphrase the sentences as data augmentation. Specifically, it characterizes the conditional probability

$$p_\theta(x|y) = \prod_{k=1}^{K} p(x_k|x_{<k}, y), \quad (1)$$

where $x_{<k}$ denotes all the tokens before the k-th position. The model parameters θ are trained by maximizing the log-likelihood of p_θ.

We observe that co-reference and ellipsis frequently occurs in user utterances. Therefore, we propose different encoding strategies during paraphrasing to further evaluate each model's capacity for context resolution. In particular, if the user mentions a certain domain *for the first time* in a dialog, we will insert a "*" mark into the sequential dialog act y' to indicate that the user tends to express without co-references or ellipsis, as shown in Table 3. Then SC-GPT is finetuned on the processed data so that it can be aware of dialog context when generating paraphrases. As a result, we find that the average token length of generated sentences with/without "*" is 15.96/12.67 respectively after SC-GPT's finetuning on MultiWOZ.

It should be noted that slot values of a sentence can be paraphrased by models, resulting in a different semantic meaning y'. To prevent generating

[2]See the EDA paper for details of each operation.

DA	train * { inform (dest = Cambridge ; arrive = 20:45) }
Text	Hi, I'm looking for a train that is going to Cambridge and arriving there by 20:45, is there anything like that?
DA	train { inform (dest = Cambridge ; arrive = 20:45) }
Text	Yes, to Cambridge, and I would like to arrive by 20:45.

Table 3: A text paraphrasing example that considers contextual resolution. The user omits to claim that he wants a train in the second sentence since he has mentioned this before.

irrelevant sentences, we apply automatic value detection in paraphrases with original slot values by fuzzy matching[3], and replace the detected values in bad paraphrases with original values. In addition, we filter out paraphrases that have missing or redundant information comparing to the original utterance.

Speech Recognition We simulate speech recognition (SR) process with a TTS-ASR pipeline (Park et al., 2019). First we transfer textual user utterance x to its audio form a using gTTS[4] (Oord et al., 2016), a Text-to-Speech (TTS) system. Then audio data is translated back into text x' by DeepSpeech2[5] (Amodei et al., 2016), an Automatic Speech Recognition (ASR) system. We directly use the released models in the repository as DeepSpeech2's configuration, where the speech model is trained on Baidu Internal English Dataset, and the language model is trained on CommonCrawl Data.

Type	Original	Augmented
Similar sounds	leicester	lester
Liaison	for 3 people	free people
Spoken numbers	13:45	thirteen forty five

Table 4: Examples of speech recognition perturbation.

Table 4 shows some typical examples of our SR augmentation. ASR sometimes wrongly identifies one word as another with similar pronunciation. Liaison constantly occurs between successive words. Expressions with numbers including time and price are written in numerical form but different in spoken language.

Since SR may modify the slot values in the translated utterances, fuzzy value detection is employed here to handle similar sounds and liaison problems when it extracts slot values to obtain a semantic label y'. However, we do not replace the noisy value with the original value as we encourage such misrecognition in SR, thus $y' \neq y$ is allowed. Moreover, numerical terms are normalized to deal with the spoken number problem. Most slot values could be relocated by our automatic value detection rules. The remainder slot values which vary too much to recognize are discarded along with their corresponding labels.

[3]https://pypi.org/project/fuzzywuzzy/

[4]https://pypi.org/project/gTTS/

[5]https://github.com/PaddlePaddle/DeepSpeech

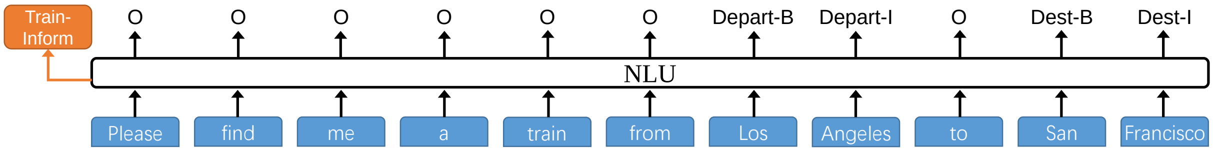
Speech Disfluency Disfluency is a common feature of real user speech. We follow the categorization of disfluency in previous works (Lickley, 1995; Wang et al., 2020b): filled pauses, repeats, restarts, and repairs.

Original	I want to go to Cambirdge.
Pauses	I want to um go to uh Cambirdge.
Repeats	I, I want to go to, go to Cambirdge.
Restarts	I just I want to go to Cambirdge.
Repairs	I want to go to Liverpool, sorry I mean Cambirdge.

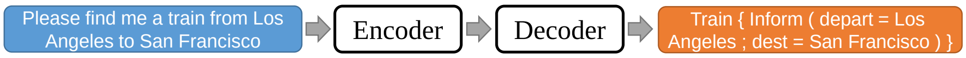
Table 5: Example of four types of speech disfluency.

Here we give a concrete illustration of SD by walking through the example in Table 5. Filler words ("um", "uh") are injected into the sentence to present pauses. Repeats are inserted by repeating the previous word. In order to approximate the real distribution of disfluency, the *interruption points* of filled pauses and repeats are predicted by a Bi-LSTM+CRF model (Zayats et al., 2016) trained on an annotated dataset SwitchBoard (Godfrey et al., 1992), which collected from real human talks. For restarts, we insert *false start terms* ("I just") as a prefix of the utterance to simulate self-correction. In LU task, we apply repairs on slot values to fool the models to predict wrong labels. We take the original slot value as *Repair* ("Cambridge") and take another value with the same slot name as *Reparandum* ("Liverpool"). An *edit term* ("sorry, I mean") is inserted between *Repair* and *Reparandum* to construct a correction. The filler words, restart terms, and edit terms and their occurrence frequency are all sampled from their distribution in SwitchBoard.

In order to keep the spans of slot values intact, each span is regarded as one whole word. No insertions are allowed to operate inside the span. Therefore, SD augmentation do not change the original semantic and labels of the utterance, i.e. $y' = y$.



(a) Classification-based language understanding



(b) Generation-based language understanding

Figure 2: An illustration of two categories of language understanding models. Dialog history is first encoded as conditions (not depicted here).

4 Experimental Setup

4.1 Data Preparation

MultiWOZ (Budzianowski et al., 2018) is a multi-domain task-oriented dialogue dataset which has been widely adopted. We conduct our experiments on the newest annotation-enhanced version MultiWOZ 2.3 (Han et al., 2020), which provides cleaned annotations of user dialog acts (i.e. semantic labels). The dialog act consists of four parts: domain, intent, slot names, and slot values. The statistics of the dataset are shown in Table 6. Following Takanobu et al. (2020), we calculate overall F1 scores as evaluation metrics due to the multi-intent setting in LU.

# Dialogs	10,438
# Turns	71,521
# Tokens	968,791
# DAs	118,534
# Domains	7
Avg. Turns / Dialog	6.85
Avg. Tokens / Turn	13.55
Avg. DAs / Turn	1.66

Table 6: Statistics of MultiWOZ 2.3. Here we only count user turns. 1,000 dialogs are respectively used for validation and test.

As for hyperparameters in LAUG, we set $\alpha = 0.1$ in EDA to perturb words. The learning rate used to finetune SC-GPT is 1e-4, the number of training epoch is 5, and the beam size during inference is 5. The beam size of the language model in DeepSpeech2 is set to 50. The learning rate of Bi-LSTM+CRF is 1e-3. The threshold of fuzzy matching in automatic value detection is set to 0.9.

The data are augmented by the addition of its copies, leading to a composite of all 4 augmentation types with equal proportion. Other setups are described in each experiment. Table 7 shows the change rates in different aspects by comparing our augmented utterances with the original counterparts.

Method	Change Rate			Human Annot.	
	Char	Word	Slot	Utter.	DA
WP	17.9	16.0	36.3	95.2	97.0
TP	60.3	74.4	13.3	97.1	97.7
SR	7.9	14.5	40.8	95.1	96.7
SD	22.7	30.4	0.4	98.8	99.2

Table 7: Statistics of augmented data. Automatic metrics include change rate of characters, words and slot values. Human evaluation includes appropriateness at utterance level (Utter.) and at dialog act level (DA).

To ensure the quality of our augmented test set, we conduct human annotation on 1,000 sampled utterances in each augmented test set. We ask annotators to check whether our augmented utterances are reasonable and our auto-detected value annotations are correct (two true-or-false questions). According to the feature of each augmentation method, different evaluation protocols are used. For TP and SD, annotators check whether the meaning of utterances and dialog acts are unchanged. For WP, changing slot values is allowed due to slot value replacement, but the slot name should be the same. For SR, annotators are asked to judge on the similarity of pronunciation rather than semantics. In summary, all the high scores in Table 7 demonstrate that LAUG makes proper approximation of natural perturbation in human inputs.

Model	Cls.	Gen.	Pre.
MILU (Hakkani-Tür et al., 2016)	√		
BERT (Devlin et al., 2019)	√		√
ToD-BERT (Wu et al., 2020)	√		√
CopyNet (Gu et al., 2016)		√	
GPT-2 (Radford et al., 2019)		√	√

Table 8: Features of base models. Cls./Gen. denotes classification/generation-based models. Pre. stands for pre-trained language models.

Model	Train	Ori.	WP	TP	SR	SD	Avg.	Drop	Recov.
MILU	Original	91.33	88.26	87.20	77.98	83.67	84.28	-7.05	/
	Augmented	91.39	90.01	88.04	86.97	89.54	88.64	-2.75	+4.36
BERT	Original	**93.40**	90.96	88.51	82.35	85.98	86.95	-6.45	/
	Augmented	93.32	92.23	89.45	89.86	92.71	91.06	-2.26	+4.11
ToD-BERT	Original	93.28	91.27	88.95	81.16	87.18	87.14	-6.14	/
	Augmented	93.29	**92.40**	89.71	**90.06**	**92.85**	**91.26**	-2.03	+4.12
CopyNet	Original	90.97	85.25	87.40	71.06	77.66	80.34	-10.63	/
	Augmented	90.49	89.19	89.53	85.69	89.83	88.56	-1.93	**+8.22**
GPT-2	Original	91.53	85.35	88.23	80.74	84.33	84.66	-6.87	/
	Augmented	91.59	90.26	**89.92**	86.55	90.55	89.32	-2.27	+4.66

Table 9: Robustness test results. Ori. stands for the original test set, WP, TP, SR, SD for 4 augmented test sets and Avg. for the average performance on 4 augmented test sets. The augmented training set is composed of 4 types of augmented data with equal proportion and the ratio of augmentation to original data is 1:1. Drop shows the performance decline between Avg. and Ori. Recov. denotes the performance recovery of Avg. between training on augmented/original data (e.g., 88.64%-84.28% for MILU).

4.2 Baselines

LU models roughly fall into two categories: classification-based and generation-based models. Classification based models (Hakkani-Tür et al., 2016; Goo et al., 2018) extract semantics by intent detection and slot tagging. Intent detection is commonly regarded as a multi-label classification task, and slot tagging is often treated as a sequence labeling task with *BIO format* (Ramshaw and Marcus, 1999), as shown in Fig. 2a. Generation-based models (Liu and Lane, 2016; Zhao and Feng, 2018) generate a dialog act containing intent and slot values. They treat LU as sequence-to-sequence models and transform the dialog acts into a sequential structure as shown in Fig. 2b. Five base models with different categories are used in the experiments, as shown in Table 8.

To support a multi-domain/multi-intent setting in classification-based models, we decouple the LU process into two steps: first perform both domain classification and intent detection, then concatenate two special tokens indicating the detected domain and intent at the beginning of the input sequence, and last encode the new sequence to predict the slot tags. In this way, the model can address *overlapping slot values* when slot values are shared in different dialog acts.

5 Evaluation Results

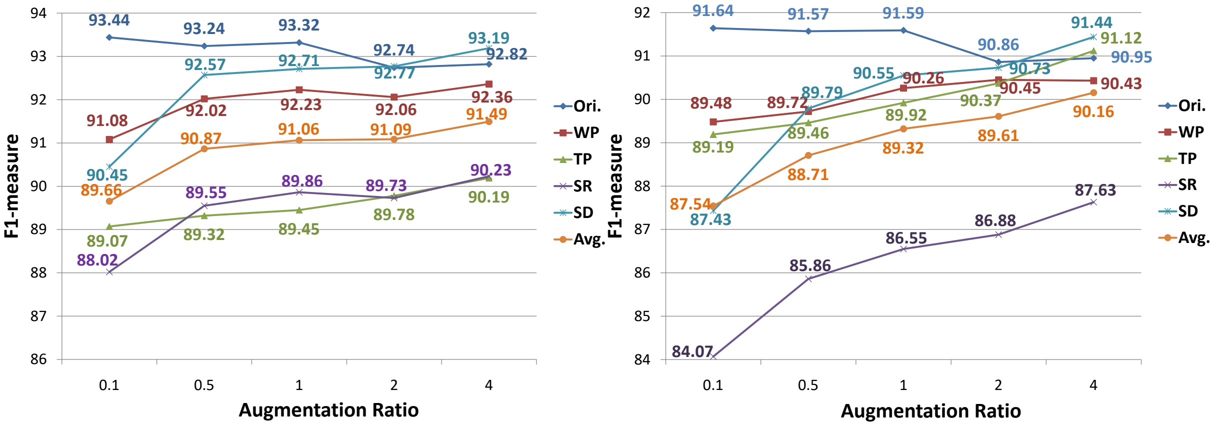
5.1 Main Results

We conduct robustness testing on all the three capacities for five base models using our four augmentation methods. Overall F1-measure performance is shown in Table 9. Robustness for each capacity can be measured by performance drops on the corresponding augmented test sets. All models achieve some performance recovery on augmented test sets after trained on the augmented data, while keeps a comparable result on the original test set. This indicates the effectiveness of LAUG in improving model's robustness.

We observe that pre-trained models outperform non-pre-trained ones on both original and augmented test sets. Classification-based models have better performance and are more robust than generation-based models, especially for Copynet, whose performance drops the most on augmented test sets. ToD-BERT, the state-of-the-art model which was further pre-trained on task-oriented dialog data, have comparable performance with BERT. Against most augmentation methods, ToD-BERT shows slightly better robustness than BERT.

Among the four augmentation methods, SR has the most impact on the models' performance and SD comes the second. The dramatic performance drop when testing on SR and SD data indicates that robustness for speech characteristics may be the most challenging issue, especially for CopyNet, which suffers a 19.91% decline in F1 on SR (90.97% - 71.06%). WP and TP make some difference on models' performance but far less than SR and SD.

Fig. 3 shows how the performance of BERT and GPT-2 changes when the ratio of augmented training data to the original data varies from 0.1 to 4.0. F1 scores on augmented test sets increase when there is more augmented data for training. The performance of BERT on augmented test sets is sharply improved when augmentation ratio is less than 0.5 but becomes almost unchanged after 0.5 while GPT-2 keeps increasing stably. This result shows the different characteristics between



(a) BERT (b) GPT-2

Figure 3: Performance with different ratios of augmented training data. The total amount of training data varies but they are always composed of 4 types of augmented data with equal proportion. Different test sets are shown with different colored lines.

classification-based models and generation-based models when finetuned with augmented data.

5.2 Ablation Study

Between augmentation approaches In order to study the influence of each augmentation approach, we test the performance changes when one augmentation approach is removed from constructing augmented training data. Experimental results on BERT and MILU are shown in Table 10.

Train	Ori.	WP	TP	SR	SD	Avg.
Aug.	91.39	90.01	88.04	86.97	89.54	88.64
-WP	91.29	**88.42**	88.43	86.98	89.20	88.26
-TP	91.55	90.15	**87.81**	86.82	89.42	88.55
-SR	91.23	90.13	88.30	**77.90**	89.51	86.46
-SD	91.56	90.24	88.60	86.78	**83.96**	87.40
Ori.	91.33	88.26	87.20	77.98	83.67	84.28

(a) MILU

Train	Ori.	WP	TP	SR	SD	Avg.
Aug.	93.32	92.23	89.45	89.86	92.71	91.06
-WP	93.23	**90.94**	89.42	89.93	92.82	90.78
-TP	93.08	92.24	**88.62**	89.80	92.62	90.82
-SR	93.43	92.30	89.50	**83.48**	93.07	89.59
-SD	93.11	92.15	89.44	90.00	**85.22**	89.20
Ori.	93.40	90.96	88.51	82.35	85.98	86.95

(b) BERT

Table 10: Ablation study between augmentation approaches for two models. Highlighted figures denote the most sharp decline for each augmented test set.

Large performance decline on each augmented test set is observed when the corresponding augmentation approach is removed in constructing training data. The performance after removing an augmentation method is comparable to the one without augmented training data. Only slight changes are observed without other approaches. These results indicates that our four augmentation approaches are relatively independent for the two based models.

Within augmentation approach Our implementation of WP and SD consist of several functional components. Ablation experiments here show how much performance is affected by each component in augmented test sets.

Test	MILU	Diff.	BERT	Diff.
WP	88.26	/	90.96	/
-Syno.	88.90	0.64	91.27	0.48
-Insert	88.90	0.64	91.30	0.51
-Delete	88.97	0.71	91.20	0.41
-Swap	89.15	0.89	91.33	**0.54**
-Slot	89.45	**1.19**	91.30	0.51
Ori.	91.33	3.05	93.40	2.61

(a) Word Perturbation

Test	MILU	Diff.	BERT	Diff.
SD	83.67	/	85.98	/
-Repair	89.47	**5.80**	91.05	**5.07**
-Pause	85.21	1.54	88.06	2.08
-Restart	84.03	0.36	86.22	0.24
-Repeat	83.64	-0.03	85.68	-0.30
Ori.	91.33	7.66	93.40	7.42

(b) Speech Disfluency

Table 11: Ablation study within two augmentation approaches. Models are trained on original training set. Highlight stands for the component with the most influence on model performance.

Original EDA consists of four functions as described in Table 2. Performance differences can reflect the influences of those components in Table 11a. The additional function of our SC-EDA is slot value replacement. We can also observe an increase in performance when it is removed.

Table 11b shows the results of ablation study on SD. Among the four types of disfluencies described in Table 5, *repairs* has the largest impact on models' performance. The performance is also affected by *pauses* but to a less extent. The influences of *repeats* and *restarts* are small, which indicates that neural models are robust to handle these two problems.

Model	Train	Scheme	Ori.	Avg.	Drop
MILU	Ori.	coupled	85.52	82.91	-2.61
		decoupled	**91.33**	**84.28**	-7.05
	Aug.	coupled	90.00	88.15	-1.85
		decoupled	**91.39**	**88.64**	-2.75
BERT	Ori.	coupled	88.94	80.33	-8.61
		decoupled	**93.40**	**86.95**	-6.45
	Aug.	coupled	88.84	88.63	-0.21
		decoupled	**93.32**	**91.06**	-2.26

Table 12: Robustness on different schemes. The coupled scheme predicts dialog acts with a joint tagging scheme; the decoupled scheme first detects domains and intents, then recognizes the slot tags.

5.3 Prediction Schemes

In this section, we study the influence of robustness on the training/prediction scheme. As described in Sec. 4.2, the process of classification-based LU models is decoupled into two steps to handle multiple labels: one for domain/intent classification and the other for slot tagging. Another strategy is to use the cartesian product of all the components of dialog acts, which yields a joint tagging scheme as presented in ConvLab (Lee et al., 2019). To give an intuitive illustration, the slot tag of the token "Los" becomes "Train-Inform-Depart-B" in the example of Fig. 2. The classification-based models can predict the dialog acts within a single step in this way.

Table 12 shows that MILU and BERT gain from the decoupled scheme on the original test set. This indicates that the decoupled scheme decreases the model complexity by decomposing the output space. Interestingly, there is no consistency between two models in terms of robustness. MILU via the coupled scheme behaves more robustly than the decoupled counterpart (-2.61 vs. -7.05), while BERT with the decoupled scheme outperforms its coupled version in robustness (-6.45 vs. -8.61). Meanwhile, BERT benefits from the decoupled scheme and still achieves 86.95% accuracy, but BERT training with the coupled scheme seems more susceptible. In addition, both MILU and BERT recover more performance by the proposed decoupled scheme. All these results demonstrate the superiority of the decoupled scheme in classification-based LU models.

5.4 Case Study and Error Analysis

We present some examples of augmented utterances in Table 13. In terms of model performance, MILU, BERT and GPT-2 perform well on WP and TP in the example. CopyNet misses some dialog acts when encountering WP and TP. For the SR, only BERT obtains all the correct labels. MILU is sensitive to the slot value so it fails to find "lester" and "thirteen forty five". Copynet's copy mechanism is fully confused by SR augmentation and predicts discontinuous slot values. GPT-2 successfully finds the non-numerical time but misses "leseter". In the SD utterance, the repair term fools all the models. Overall, in this example, BERT performs quite well while MILU and CopyNet expose some of their defects in robustness.

6 Related Work

Robustness in LU has always been a challenge in dialog systems. Several studies have investigated the model's sensitivity to the collected data distribution, in order to prevent models from overfitting to the training data and improve robustness in the real world. Kang et al. (2018) collected dialogs with templates and paraphrased with crowd-sourcing to achieve high coverage and diversity in training data, improving the generalization of dialog systems. Dinan et al. (2019) proposed a training schema that involves human in the loop in dialog systems to enhance the model's defense against human attack in an iterative way. Ganhotra et al. (2020) injected natural perturbation into the dialog history manually to refine over-controlled data generated through crowd-sourcing. All these methods require laborious human intervention. This paper aims to provide an effective and automatic way to test the robustness of LU in dialog systems.

Various textual adversarial attacks (Zhang et al., 2020a) have been proposed and received increasing attentions these years to measure the robustness of a victim model. Most attack methods perform white-box testing (Papernot et al., 2016; Li et al., 2019; Ebrahimi et al., 2018) based on the model's internal structure or gradient signals. Even some black-box attack models are not purely "black-box", which require the prediction scores (classification proba-

Ori.	I 'm leaving from Leicester and should arrive in Cambridge by 13:45
Golden	train { inform (dest = cambridge ; arrive = 13:45 ; depart = leicester) }
WP	I 'm leaving from Leicester and $\{$in$\}_{swap}$ arrive $\{$should$\}_{swap}$ Cambridge by $\{$06:54$\}_{replace}$.
Golden	train { inform (dest = cambridge ; arrive = 06:54 ; depart = leicester) }
MILU	train { inform (dest = cambridge ; arrive = 06:54 ; depart = leicester) }
BERT	train { inform (dest = cambridge ; arrive = 06:54 ; depart = leicester) }
Copy	train { inform (dest = cambridge ; depart = leicester) }
GPT-2	train { inform (dest = cambridge ; arrive = 06:54 ; depart = leicester) }
TP	Departing from Leicester and going to Cambridge. I need to arrive by 13:45.
Golden	train { inform (dest = cambridge ; arrive = 13:45 ; depart = leicester) }
MILU	train { inform (dest = cambridge ; arrive = 13:45 ; depart = leicester) }
BERT	train { inform (dest = cambridge ; arrive = 13:45 ; depart = leicester) }
Copy	train { inform (arrive = 13:45 ; depart = leicester) }
GPT-2	train { inform (dest = cambridge ; arrive = 13:45 ; depart = leicester) }
SR	I'm leaving from $\{$lester$\}_{similar}$ and should arrive in Cambridge by $\{$thirteen forty five$\}_{spoken}$.
Golden	train { inform (dest = cambridge ; arrive = thirteen forty five ; depart = lester) }
MILU	train { inform (dest = cambridge) }
BERT	train { inform (dest = cambridge ; arrive = thirteen forty five ; depart = lester) }
Copy	train { inform (dest = cambridge forty ; depart = lester) }
GPT-2	train { inform (dest = cambridge ; arrive = thirteen forty five) }
SD	$\{$Well, you know,$\}_{restart}$ I 'm leaving from Leicester and should arrive in $\{$King's College sorry, i mean$\}_{repair}$ Cambridge by 13:45.
Golden	train { inform (dest = cambridge ; arrive = 13:45 ; depart = leicester) }
MILU	train { inform (dest = king ; arrive = 13:45 ; depart = leicester) }
BERT	train { inform (dest = king 's college ; arrive = 13:45 ; depart = leicester) }
Copy	train { inform (arrive = 13:45 ; depart = leicester) }
GPT-2	train { inform (dest = king 's college ; arrive = 13:45 ; depart = leicester) }

Table 13: Augmented examples and corresponding model outputs. All models are trained on the original data only. Wrong values are colored in blue.

bilities) of the victim model (Jin et al., 2020; Ren et al., 2019; Alzantot et al., 2018). However, all these methods address random perturbation but do not consider linguistic phenomena to evaluate the real-life generalization of LU models.

While data augmentation can be an efficient method to address data sparsity, it can improve the generalization abilities and measure the model robustness as well (Eshghi et al., 2017). Paraphrasing that rewrites the sentences in dialog has been used to get diverse representation and thus enhancing robustness (Ray et al., 2018; Zhao et al., 2019; Iyyer et al., 2018). Word-level operations (Kolomiyets et al., 2011; Li and Qiu, 2020; Wei and Zou, 2019) including replacement, insertion, and deletion were also proposed to increase language variety. Other studies (Shah et al., 2019; Xu and Sarikaya, 2014) worked on the out-of-vocabulary problem when facing unseen user expression. Some other research focused on building robust spoken language understanding (Zhu et al., 2018; Henderson et al., 2012; Huang and Chen, 2019) from audio signals beyond text transcripts. Simulating ASR errors (Schatzmann et al., 2007; Park et al., 2019; Wang et al., 2020a) and speaker disfluency (Wang et al., 2020b; Qader et al., 2018) can be promising solutions to enhance robustness to voice inputs when only textual data are provided. As most work tackles LU robustness from only one perspective, we give a comprehensive study to reveal three critical issues in this paper, and shed light on a thorough robustness evaluation of LU in dialog systems.

7 Conclusion and Discussion

In this paper, we present a systematic robustness evaluation of language understanding in task-oriented dialog systems from three aspects: *language variety*, *speech characteristics*, and *noise perturbation*. And accordingly, we develop four data augmentation methods to approximate these language phenomena. In-depth experiments and analysis are conducted on MultiWOZ 2.3, with both classification- and generation-based LU models. The performance drop of all models on augmented test data indicates that these robustness issues are challenging and critical. Ablation studies are carried out to show the effect of each augmentation approach and that of each component within one approach. Additionally, we evaluate how the robustness performs differently between coupled and decoupled prediction schemes, showing different robustness behaviors for different models.

Existing and future dialog models can be evaluated in terms of robustness with our toolkit and data as our augmentation model does not depend on any particular LU models. Moreover, our proposed robustness evaluation scheme is extensible. In addition to the four approaches in LAUG, more methods to evaluate LU robustness can be considered in the future.

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896.

Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. 2016. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pages 173–182.

Timothy W Bickmore, Ha Trinh, Stefan Olafsson, Teresa K O'Leary, Reza Asadi, Nathaniel M Rickles, and Ricardo Cruz. 2018. Patient and consumer safety risks when using conversational assistants for medical information: an observational study of siri, alexa, and google assistant. *Journal of medical Internet research*, 20(9):e11510.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Emily Dinan, Samuel Humeau, Bharath Chintagunta, and Jason Weston. 2019. Build it break it fix it for dialogue safety: Robustness from adversarial human attack. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4529–4538.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. Hotflip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36.

Arash Eshghi, Igor Shalyminov, and Oliver Lemon. 2017. Bootstrapping incremental dialogue systems from minimal data: the generalisation power of dialogue grammars. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2220–2230.

Jatin Ganhotra, Robert C Moore, Sachindra Joshi, and Kahini Wadhawan. 2020. Effects of naturalistic variation in goal-oriented dialog. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 4013–4020.

Jianfeng Gao, Michel Galley, and Lihong Li. 2019. Neural approaches to conversational ai. *Foundations and Trends® in Information Retrieval*, 13(2-3):127–298.

John J Godfrey, Edward C Holliman, and Jane McDaniel. 1992. Switchboard: Telephone speech corpus for research and development. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, volume 1, pages 517–520. IEEE Computer Society.

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.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640.

Dilek Hakkani-Tür, Gokhan Tur, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. *Interspeech 2016*, pages 715–719.

Ting Han, Ximing Liu, Ryuichi Takanobu, Yixin Lian, Chongxuan Huang, Wei Peng, and Minlie Huang. 2020. Multiwoz 2.3: A multi-domain task-oriented dataset enhanced with annotation corrections and co-reference annotation. *arXiv preprint arXiv:2010.05594*.

Keqing He, Yuanmeng Yan, and XU Weiran. 2020. 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.

Matthew Henderson, Milica Gašić, Blaise Thomson, Pirros Tsiakoulis, Kai Yu, and Steve Young. 2012. Discriminative spoken language understanding using word confusion networks. In *2012 IEEE Spoken Language Technology Workshop (SLT)*, pages 176–181. IEEE.

Matthias Honal and Tanja Schultz. 2003. Correction of disfluencies in spontaneous speech using a noisy-channel approach. In *Eighth European Conference on Speech Communication and Technology*, pages 2781–2784.

Chao-Wei Huang and Yun-Nung Chen. 2019. Adapting pretrained transformer to lattices for spoken language understanding. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 845–852. IEEE.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. 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*, volume 34, pages 8018–8025.

Yiping Kang, Yunqi Zhang, Jonathan K Kummerfeld, Lingjia Tang, and Jason Mars. 2018. Data collection for dialogue system: A startup perspective. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 33–40.

Oleksandr Kolomiyets, Steven Bethard, and Marie-Francine Moens. 2011. Model-portability experiments for textual temporal analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, volume 2, pages 271–276.

Sungjin Lee, Qi Zhu, Ryuichi Takanobu, Zheng Zhang, Yaoqin Zhang, Xiang Li, Jinchao Li, Baolin Peng, Xiujun Li, Minlie Huang, et al. 2019. Convlab: Multi-domain end-to-end dialog system platform. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 64–69.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. Textbugger: Generating adversarial text against real-world applications. In *26th Annual Network and Distributed System Security Symposium*.

Linyang Li and Xipeng Qiu. 2020. Textat: Adversarial training for natural language understanding with token-level perturbation. *arXiv preprint arXiv:2004.14543*.

Robin J Lickley. 1995. Missing disfluencies. In *Proceedings of the international congress of phonetic sciences*, volume 4, pages 192–195.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. *Interspeech 2016*, pages 685–689.

Yijin Liu, Fandong Meng, Jinchao Zhang, Jie Zhou, Yufeng Chen, and Jinan Xu. 2019. Cm-net: A novel collaborative memory network for spoken language understanding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1050–1059.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.

Nicolas Papernot, Patrick McDaniel, Ananthram Swami, and Richard Harang. 2016. Crafting adversarial input sequences for recurrent neural networks. In *MILCOM 2016-2016 IEEE Military Communications Conference*, pages 49–54. IEEE.

Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. *Interspeech 2019*, pages 2613–2617.

Baolin Peng, Chenguang Zhu, Chunyuan Li, Xiujun Li, Jinchao Li, Michael Zeng, and Jianfeng Gao. 2020. Few-shot natural language generation for task-oriented dialog. *arXiv preprint arXiv:2002.12328*.

Raheel Qader, Gwénolé Lecorvé, Damien Lolive, and Pascale Sébillot. 2018. Disfluency insertion for spontaneous tts: Formalization and proof of concept. In *International Conference on Statistical Language and Speech Processing*, pages 32–44. Springer.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Lance A Ramshaw and Mitchell P Marcus. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer.

Avik Ray, Yilin Shen, and Hongxia Jin. 2018. Robust spoken language understanding via paraphrasing. *Interspeech 2018*, pages 3454–3458.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 1085–1097.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. *arXiv preprint arXiv:2005.04118*.

Jost Schatzmann, Blaise Thomson, and Steve Young. 2007. Error simulation for training statistical dialogue systems. In *2007 IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*, pages 526–531. IEEE.

Darsh Shah, Raghav Gupta, Amir Fayazi, and Dilek Hakkani-Tur. 2019. Robust zero-shot cross-domain slot filling with example values. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5484–5490.

Ryuichi Takanobu, Qi Zhu, Jinchao Li, Baolin Peng, Jianfeng Gao, and Minlie Huang. 2020. Is your goal-oriented dialog model performing really well? empirical analysis of system-wise evaluation. *arXiv preprint arXiv:2005.07362*.

Longshaokan Wang, Maryam Fazel-Zarandi, Aditya Tiwari, Spyros Matsoukas, and Lazaros Polymenakos. 2020a. Data augmentation for training dialog models robust to speech recognition errors. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 63–70.

Shaolei Wang, Wangxiang Che, Qi Liu, Pengda Qin, Ting Liu, and William Yang Wang. 2020b. Multitask self-supervised learning for disfluency detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9193–9200.

Yu Wang, Yilin Shen, and Hongxia Jin. 2018. A bimodel based rnn semantic frame parsing model for intent detection and slot filling. 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 309–314.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6383–6389.

Chien-Sheng Wu, Steven Hoi, Richard Socher, and Caiming Xiong. 2020. Tod-bert: Pre-trained natural language understanding for task-oriented dialogues. *arXiv preprint arXiv:2004.06871*.

Puyang Xu and Ruhi Sarikaya. 2014. Targeted feature dropout for robust slot filling in natural language understanding. In *Interspeech 2014*, pages 258–262.

Kang Min Yoo, Youhyun Shin, and Sang-goo Lee. 2019. Data augmentation for spoken language understanding via joint variational generation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7402–7409.

Vicky Zayats, Mari Ostendorf, and Hannaneh Hajishirzi. 2016. Disfluency detection using a bidirectional lstm. *Interspeech 2016*, pages 2523–2527.

Wei Emma Zhang, Quan Z Sheng, Ahoud Alhazmi, and Chenliang Li. 2020a. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(3):1–41.

Zheng Zhang, Ryuichi Takanobu, Qi Zhu, Minlie Huang, and Xiaoyan Zhu. 2020b. Recent advances and challenges in task-oriented dialog systems. *Science China Technological Sciences*, pages 1–17.

Lin Zhao and Zhe Feng. 2018. Improving slot filling in spoken language understanding with joint pointer and attention. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 426–431.

Zijian Zhao, Su Zhu, and Kai Yu. 2019. Data augmentation with atomic templates for spoken language understanding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3628–3634.

Su Zhu, Ouyu Lan, and Kai Yu. 2018. Robust spoken language understanding with unsupervised asr-error adaptation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6179–6183. IEEE.