SILVER: Self Data Augmentation for Out-of-Scope Detection in Dialogues

Chunpeng Ma and **Takuya Makino**
Megagon Labs, Tokyo, Recruit Co., LTD.
1-9-2 Marunouchi Chiyoda-ku, Tokyo, 100-6640, Japan
{ma.chunpeng, makino}@megagon.ai

Abstract

Detecting out-of-scope (OOS) utterances is crucial in task-oriented dialogue systems, but obtaining enough annotated OOS dialogues to train a binary classifier directly is difficult in practice. Existing data augmentation methods generate OOS dialogues automatically, but their performance usually depends on an external corpus. This dependence not only induces uncertainty, but also reduces the quality of generated dialogues. Specifically, all of them are out-of-domain (OOD).

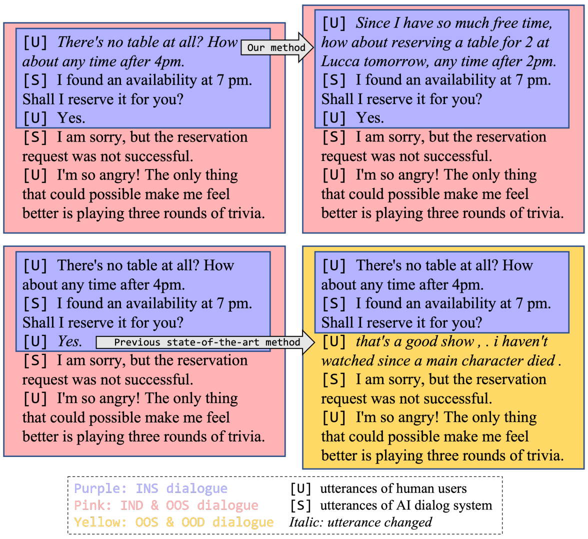
Herein we propose SILVER, a *self* data augmentation method that does not use external data. It addresses issues of previous research and improves the accuracy of OOS detection (false positive rate: 90.5% → 47.4%). Furthermore, SILVER successfully generates high-quality *in-domain* (IND) OOS dialogues in terms of naturalness (percentage: 8% → 68%) and OOS correctness (percentage: 74% → 88%), as evaluated by human workers.

1 Introduction

Task-oriented dialogue systems are ubiquitous (Budzianowski et al., 2018; Chiu et al., 2022). However, they require human operators to deal with complicated intentions that are beyond their capacities. Thus, out-of-scope (OOS) detection remains a serious issue.

Due to the lack of OOS annotations in open-world settings, previous research usually detects OOS samples *indirectly* such as resorting to in-scope (INS) samples. Recently, data augmentation methods (Ng et al., 2020; Razumovskaia et al., 2022) have made it possible to detect OOS *directly* using a binary classifier.

One such method is GOLD (Chen and Yu, 2021). GOLD uses simple rules to replace utterances in known OOS dialogues with sentences selected from a large pool, making it possible to train a binary classifier to decide OOS dialogues *directly*.



`[U]` *There's no table at all? How about any time after 4pm.*
`[S]` I found an availability at 7 pm. Shall I reserve it for you?
`[U]` Yes.
`[S]` I am sorry, but the reservation request was not successful.
`[U]` I'm so angry! The only thing that could possible make me feel better is playing three rounds of trivia.

`[U]` *Since I have so much free time, how about reserving a table for 2 at Lucca tomorrow, any time after 2pm.*
`[S]` I found an availability at 7 pm. Shall I reserve it for you?
`[U]` Yes.
`[S]` I am sorry, but the reservation request was not successful.
`[U]` I'm so angry! The only thing that could possible make me feel better is playing three rounds of trivia.

`[U]` There's no table at all? How about any time after 4pm.
`[S]` I found an availability at 7 pm. Shall I reserve it for you?
`[U]` *Yes.*
`[S]` I am sorry, but the reservation request was not successful.
`[U]` I'm so angry! The only thing that could possible make me feel better is playing three rounds of trivia.

`[U]` There's no table at all? How about any time after 4pm.
`[S]` I found an availability at 7 pm. Shall I reserve it for you?
`[U]` *that's a good show , . i haven't watched since a main character died .*
`[S]` I am sorry, but the reservation request was not successful.
`[U]` I'm so angry! The only thing that could possible make me feel better is playing three rounds of trivia.

Our method
Previous state-of-the-art method
Purple: INS dialogue
Pink: IND & OOS dialogue
Yellow: OOS & OOD dialogue
[U] utterances of human users
[S] utterances of AI dialog system
Italic: utterance changed

Figure 1: Comparison of GOLD and SILVER. To automatically generate an OOS dialogue, SILVER replaces the first utterance with an IND utterance, while GOLD replaces the third utterance with an OOD utterance, making the dialogue become OOD and incomprehensible.

However, three issues (`[I1]` to `[I3]`) have prevented the realization of GOLD's full potential:

- `[I1]` (ref. §3.1) Because GOLD depends on an external utterance pool, the generated dialogues are generally OOD.[1] Furthermore, an external pool is sometimes difficult to obtain in practice.
- `[I2]` (ref. §3.2) GOLD elects OOS using simple rules built upon the outputs of supporting model (e.g. $\max(\text{output probability}) >$ threshold) and combines these results by majority voting. However, the accuracy is insufficient.
- `[I3]` (ref. §3.3) To ensure sufficient OOS dialogues are generated, GOLD elects from a large candidate list.[2] A large candidate list

[1] See below for the discussion about "scope" and "domain."

[2] Each utterance corresponds to 1,024 candidates in official implementation: `https://github.com/asappresearch/`

Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), pages 26–38
November 1–4, 2023. ©2023 Association for Computational Linguistics

is a tradeoff with the performance due to the existence of low-quality dialogues.

To overcome these issues, we propose a method called **S**elf **I**terative OOS **L**abeling **v**ia **E**nsembling T**r**ees (**SILVER**). SILVER consists of solutions ([S1] to [S3]), which correspond to [I1] to [I3], respectively.

- [S1] (ref. §4.1) Build pools from training data and elect candidates in a novel way.
- [S2] (ref. §4.2) Detect OOS using an ensemble of decision trees (Mason et al., 1999).
- [S3] (ref. §4.3) Generate OOS dialogues iteratively.

Figure 1 compares dialogues generated by GOLD and SILVER. The effectiveness of each solution is verified in Section 5, and follow-up experiments (Section 6) deepen the understanding of SILVER's behavior.

1.1 Preliminary: Domain and Scope

Throughout this paper, we distinguish two concepts: *domain* and *scope*. We define them as follows.

Definition 1 (domain)**.** *Domain* is the subject or topic of a dialogue. Given a set of predefined domains, if the domain of a dialogue belongs to this set, then the dialogue is *in-domain* (IND). Otherwise, the dialogue is *out-of-domain* (OOD).

Definition 2 (scope)**.** *Scope* is the capability of a chatbot system. Given a chatbot system, if a dialogue can be understood by this chatbot, then the dialogue is *in-score* (INS). Otherwise, the dialogue is *out-of-scope* (OOS).

To clarify their distinction, the following keypoints should be emphasized.

First, *domains* (e.g., movie, finance, travel, etc.) reflect the characteristics of *dialogues*, while *scopes* (e.g., hotel reservation is beyond the scope of bank chatbot) reflect the characteristics of *chatbots*.

Second, when we say a dialogue is OOD or OOS, we consider the dialogue *as a whole*. This means that even if a dialogue is OOD or OOS, it is possible that the first few utterances are IND or INS.

Third, the relationships of IND, OOD, INS and OOS are as follows in general:

$$\text{INS} \subseteq \text{IND},$$

and

$$\text{OOD} \subseteq \text{OOS}.$$

The reason is that when we design a chatbot (e.g. finance chatbot), we expect that the chatbot should be able to understand *all* dialogues in some domains (e.g. finance domain), i.e., INS = IND for a perfect chatbot. However, generally the real chatbot can only understand *some* dialogues in that domain, i.e., INS ⊂ IND. This indicates the existence of IND & OOS dialogues (e.g. pink dialogues in Figure 1).

2 Related Works

2.1 OOS detection

Different natural language processing (NLP) tasks employ OOS detection. Examples include text classification (Fumera et al., 2003; Tan et al., 2019) and question answering (Rajpurkar et al., 2018; Kamath et al., 2020). Various methods have been proposed to detect OOS such as extrapolating to OOS samples (Daumé III, 2007; Yogatama et al., 2019), deciding whether to predict or abstain on test examples (Dong et al., 2018; Feng et al., 2019), etc. Below, three methods, which are the building blocks of both GOLD and SILVER, are reviewed.

(1) MaxProb (Hendrycks and Gimpel, 2017). A supporting model for a classification task (e.g., intent classification) is trained in advance. If the maximum value of the output probability distribution is below a predetermined threshold, then the input is classified as OOS.

(2) BertEmbed (Podolskiy et al., 2021). For each category, embeddings of all its samples are calculated by fine-tuned BERT (Devlin et al., 2019). If an input's embedding is sufficiently far (measured by cosine distance), then the input is classified as OOS.

(3) Dropout (Gal and Ghahramani, 2016). If the predictions of models whose nodes are dropped out randomly agree with each other, then the input is classified as INS.

SILVER builds a strong classifier by combining these methods to accurately detect OOS dialogues.

2.2 Data augmentation

Typical data augmentation methods in NLP modify training data by perturbing text directly (Wei and Zou, 2019), perturbing latent embedding space (Liu et al., 2019), or paraphrasing (Zhang et al., 2019).

gold/blob/master/app.py.

In the context of dialogue, data augmentation methods have been proposed for natural language understanding (Hou et al., 2018) or intent detection (Niu and Bansal, 2019). New dialogues are created by utilizing neural networks such as generative adversarial networks (Marek et al., 2021) or designing training strategies (Cubuk et al., 2018).

Unlike typical methods above, SILVER generates OOS samples via ensemble learning. Its implementation is straightforward with the help of off-the-shelf libraries, while the diverse features for classification provide a flexible architecture.

2.3 GOLD: *G*enerating *O*ut-of-scope *L*abels with *D*ata Augmentation

GOLD (Chen and Yu, 2021) is the data augmentation method most closely related to this work. Given a small set of annotated OOS dialogues (1% of the size of INS), GOLD replaces utterances with sentences selected from an external pool to generate new OOS dialogues. Selected sentences should be in the neighborhood of the original utterances. Then GOLD defers to the methods described in Section 2.1 to filter the generated OOS. Majority voting combines predictions of different methods. Filtered OOS dialogues are concatenated with the original annotated OOS dialogues and are used to train a binary classifier.

GOLD has a practical appeal. Labor-intensive data collection and annotation of OOS are unnecessary, and the data augmentation method is orthogonal to the classification improvements. Both advantages extend its applicability to real scenarios. However, the issues detailed in the next section limit its performance.

3 Empirical Investigation of GOLD

Experiments in this section are conducted on the STAR dataset (Mosig et al., 2020). To ensure comparability with Chen and Yu (2021), we followed their configurations and used the same split for train/dev/test.

3.1 Issue 1: Dependence on an external pool

Although an external pool of utterances is an essential component of GOLD, it is not always available for real applications. For example, to make a dialogue system of low-resource languages, collecting enough utterances from native speakers is difficult, let alone generating good dialogues.

Even assuming the accessibility of external pool of utterance, as reported by Chen and Yu (2021), GOLD's performance largely depends on the choice of the external pool. We argue that there is a second issue: the generated dialogues are all OOD in general. They differ significantly from the IND OOS dialogues in the original training data (see Figure 1 for an example). This deviation limits the effect of generated dialogues on improving the classifier's performance.

Method	OOS	Naturalness
GOLD	74%	8%
SILVER	88%	68%

Table 1: Human evaluation results of 50 dialogues generated by GOLD and SILVER. Numbers are the percentages of real OOS/natural dialogues.

To verify this argument quantitatively, we asked 3 human workers to evaluate 50 randomly sampled dialogues generated by GOLD to decide (1) whether the generated dialogue is really an OOS and (2) whether the generated dialogue is natural enough for comprehension. The first row of Table 1 shows the evaluation results. Although most generated dialogues ($> 50\%$) are OOS, only a small number are natural.

3.2 Issue 2: Simple election rules

Election, the stage where OOS dialogues are selected and INS dialogues are removed, is a core component of GOLD. If an INS dialogue mistakenly remains and is used to train the classifier, then it is no surprise that the classification fails.

Election in GOLD is built upon the three methods introduced in Section 2.1.[3] These methods are too simple to detect OOS dialogues accurately.

For simplicity, let's consider only MaxProb and BertEmbed. We collect (1) the maximum values from the output probability distributions generated by the supporting model, and (2) the minimum cosine distances of the embeddings between each dialogue and all dialogues from other categories. Figure 2 shows these distributions. The distributions of the first value are similar for both INS and OOS dialogues. Specifically, the modes are both in the interval "$0.95-$," which is consistent with observations in previous research (Yilmaz and Toraman, 2022). For the second value, the distributions for INS and OOS dialogues differ. However,

[3] To be precise, GOLD also considered Mahalanobis distances of representations calculated by RoBERTa. However, our discussions here are valid for both cases.

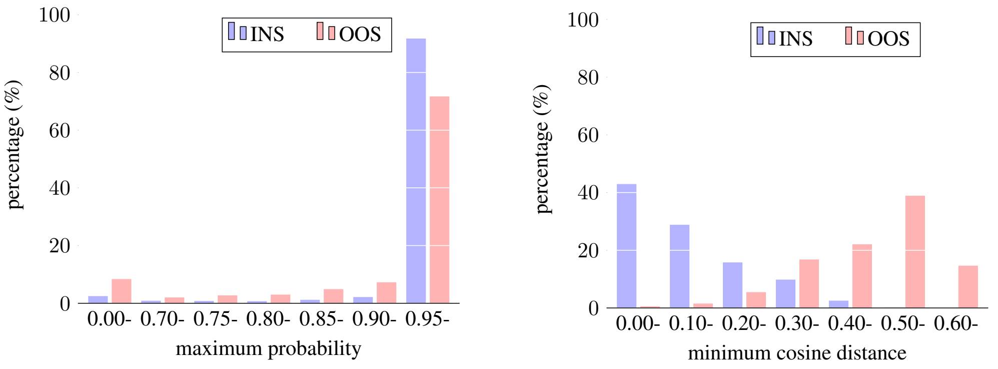
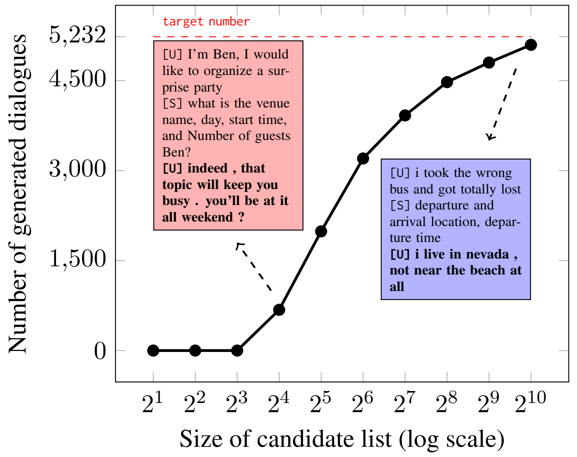


Figure 2: Left: Distribution of the maximum value of probabilities generated by supporting model. Right: Distribution of the minimum value of the cosine distances of the embeddings between each dialogue and all dialogues from other categories. INS and OOS dialogues are calculated separately.



[U] I'm Ben, I would like to organize a surprise party
[S] what is the venue name, day, start time, and Number of guests Ben?
[U] indeed , that topic will keep you busy . you'll be at it all weekend ?

[U] i took the wrong bus and got totally lost
[S] departure and arrival location, departure time
[U] i live in nevada , not near the beach at all

Figure 3: Candidate list must be enlarged to reach the target number. Dialogues tend to be real OOS in a small candidate list, but may be INS in a large candidate list. **Bold utterances** are from the pool.

there is a large overlap region between the two distributions.

To summarize, it is impossible to accurately separate INS and OOS for both MaxProb and BertEmbed, regardless of the set threshold value. Consequently, employing simple rules for election introduces a large amount of noise in the generated data, degrading the classification performance.

3.3 Issue 3: Low quality of generated data

GOLD must enlarge the candidate list size to generate enough dialogues. However, this strategy inevitably introduces a large amount of noise (i.e., INS dialogues) into the generated training data (Figure 3), which is intolerable for classifier training.

4 SILVER: Methodology

SILVER is proposed to overcome the aforementioned issues of GOLD. Figure 5 outlines the framework. First, we sample a small set of dialogues from the training data. These dialogues are known OOS. Then candidates are generated by randomly choosing one utterance from seed OOS dialogues and swapping it with an utterance extracted from INS (ref. §4.1). After generating numerous candidates, an ensemble classifier is used for election (ref. §4.2). Selected dialogues are concatenated to seed OOS samples, increasing the number of available OOS dialogues. Iterating this process several times provides sufficient data to train a binary classifier for OOS detection (ref. §4.3).

4.1 Self candidate generation

Candidates are generated by swapping utterances in seed OOS dialogues with those in the pool. To achieve this, two questions must be answered.

How should the utterance pool be built? All utterances of INS dialogues in the training data are used to build the utterance pool because we aim to generate candidates without using external corpora. Furthermore, for a task-oriented dialogue system, we assume that utterances from the user and system are in different clusters. Hence, two pools are built: (1) one for system utterances and (2) one for user utterances.

How should an *appropriate* utterance be selected? Two criteria are considered to determine appropriate utterances: (1) high similarity to the original utterance and (2) high divergence between each other. Figure 4 illustrates their trade-off.

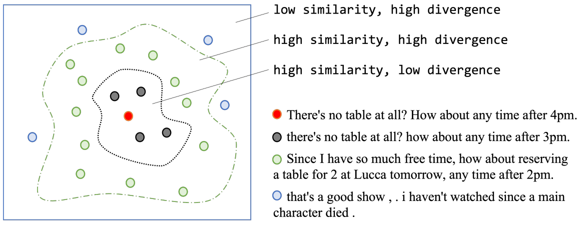

low similarity, high divergence
high similarity, high divergence
high similarity, low divergence
● There's no table at all? How about any time after 4pm.
● there's no table at all? how about any time after 3pm.
○ Since I have so much free time, how about reserving a table for 2 at Lucca tomorrow, any time after 2pm.
○ that's a good show , . i haven't watched since a main character died .

Figure 4: Trade-off between similarity and divergence when selecting appropriate utterances from an utterance pool.

(1) *High similarity*: By selecting utterances similar to the original one, the naturalness of the original OOS dialogue is kept. This means that the blue points in Figure 4, which were selected by GOLD, will not be selected by SILVER.

(2) *High divergence*: Dialogues generated by simply modifying some words in the original utterances do not improve the classification performance. We hope the generated dialogues differ from each other. This means that selecting the black points in Figure 4 should be avoided.

Therefore, only *appropriate* utterances in the sense of high similarity and high divergence (i.e., green points in Figure 4) should be selected. In practice, utterances are selected from the set $\mathcal{N}(16) - \mathcal{N}(4)$, where $\mathcal{N}(k)$ is the set of k-nearest utterances from the original utterance.

4.2 Tree ensemble

Ideally, SILVER should be simple yet flexible, and orthogonal to other studies on OOS detection. Therefore, rather than learning from scratch, we combine the outputs of weak models and build a classifier above them by ensemble learning.

We observed that each simple rule can be abstracted as a decision tree. For example, given the output probability distribution of supporting model $[p_1, \ldots, p_l]$, MaxProb can be decomposed as a decision tree with nodes "$p_i > p_j$? p_i : p_j" and leaves "$p_m > \theta$? INS : OOS" (represented in the format of the ternary conditional operator), where $m \in [1, l]$ is the index of the maximum value.

This inspired us to build a strong classifier via tree ensemble methods (Mason et al., 1999). Specifically, we adopted the gradient tree boosting algorithm (Chen and Guestrin, 2016) to assemble simple rules (decision trees). The feature sets consist of three parts, which correspond to the OOS detection methods introduced in Section 2.1.

(1) Probability-based feature. An intent classifier is trained as the supporting model. Then, given a dialogue d, the supporting model outputs the probability distribution over all intent labels: $[p_1(d), \ldots, p_l(d)]$, where l is the number of possible intent labels, and $\forall i \in [1, l], 0 \leq p_i(d) \leq 1$. Based on this probability distribution, the probability-based feature is calculated as:

$$X_{prob}(d) = [\sigma^{-1}(p_1(d)), \ldots, \sigma^{-1}(p_l(d))], \quad (1)$$

where $\sigma(\cdot)$ is the standard logistic function.

(2) Distance-based feature. Given a dialogue d, the distance-based feature is calculated as below:

$$\begin{aligned} X_{dist}(d) = [&\mathrm{Dist}(h_{\mathrm{BERT}}(d), \overline{h_{\mathrm{BERT}}(\mathcal{D}_1)}), \\ &\ldots, \mathrm{Dist}(h_{\mathrm{BERT}}(d), \overline{h_{\mathrm{BERT}}(\mathcal{D}_l)})], \end{aligned} \quad (2)$$

where $\mathrm{Dist}(\cdot, \cdot)$ is the cosine distance between two vectors, and $h_{\mathrm{BERT}}(d)$ is the representation of the last hidden layer given input d. $\mathcal{D}_i$ is the collection of all dialogues with intent label i, and $\overline{h_{\mathrm{BERT}}(\mathcal{D}_i)}$ is the average of their representations.

(3) Ensemble-based feature. This is the average of the output probability distributions of three different runs by randomly dropping out different nodes of the baseline intent classifier, which is given as:

$$X_{drop}(d) = \left[\frac{1}{3}\sum_{k=1}^{3} \sigma^{-1}(p_1^k(d)), \ldots, \frac{1}{3}\sum_{k=1}^{3} \sigma^{-1}(p_l^k(d))\right], \quad (3)$$

where $p_i^k(d)$ is the probability of intent label i at the k-th run given dialogue d, after dropping out some nodes of the neural network. The dropout percentage is 10%.

The final feature set is the concatenation of X_{prob}, X_{dist} and X_{drop}. It is trained on *sampled* training data. Thus, no extra annotation is needed.

4.3 Iterative data augmentation

To make sure enough OOS dialogues are generated, SILVER augments data in an iterative manner. After each iteration, newly generated dialogues are aggregated and considered known OOS dialogues. Then these are used to generate more dialogues in the next iteration. This iterative process generates *high-quality* dialogues with *high efficiency*.

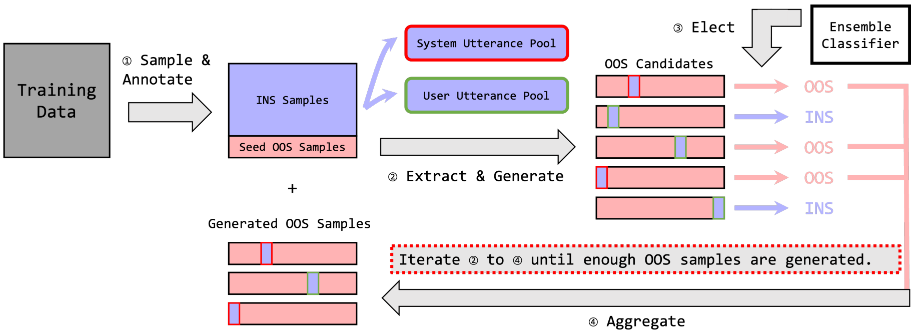


Figure 5: Framework of SILVER.

Modules			STAR				FLOW				ROSTD			
Pool	Elect	Iter.	AUROC	AUPR	FPR@.95	FPR@.90	AUROC	AUPR	FPR@.95	FPR@.90	AUROC	AUPR	FPR@.95	FPR@.90
Ext.	Rnd.	✗	0.7827	0.3618	90.5%	77.8%	0.6692	0.1503	89.1%	80.3%	0.9918	0.9224	1.96%	1.28%
Ext.	MV	✗	0.8456	0.4501	75.4%	59.7%	0.7111	0.1789	83.7%	76.4%	0.9967	0.9613	0.30%	0.30%
Ext.	TE	✗	0.8632	0.4721	63.9%	48.4%	0.7287	0.2183	79.2%	72.1%	0.9985	0.9805	0.15%	0.13%
Ext.	TE	✓	0.8858	**0.4906**	56.9%	38.3%	0.7373	0.2299	76.9%	69.3%	**0.9991**	**0.9910**	**0.09%**	**0.09%**
Int.	Rnd.	✗	0.7843	0.2623	82.7%	74.3%	0.7825	0.2608	79.7%	72.4%	0.8594	0.2967	31.7%	15.0%
Int.	MV	✗	0.8363	0.3618	71.5%	49.3%	0.8030	0.2995	71.3%	60.6%	0.9966	0.9744	0.25%	0.09%
Int.	TE	✗	0.8643	0.3952	60.7%	40.4%	0.8215	0.3368	56.9%	45.1%	0.9971	0.9680	0.22%	0.13%
Int.	TE	✓	**0.8992**	0.4212	**47.4%**	**33.9%**	**0.8319**	**0.3379**	**55.0%**	**43.9%**	0.9971	0.9680	0.22%	0.13%
		GOLD	0.8683	0.4450	56.0%	40.9%	0.8022	0.3243	60.6%	49.5%	0.9990	0.9933	0.17%	0.09%

Table 2: Results of OOS detection. Column "Pool" means whether *external* (Ext.) data (PersonaChat (Zhang et al., 2018)) or *internal* (Int.) data (i.e., training data) is used to generate utterance pool. Column "Elect" gives different election methods: random selection (Rnd.), majority voting (MV), or tree ensemble (TE). Column "Iter." indicates whether dialogues are generated iteratively (✓) or not (✗). Therefore, Int. + TE + ✓means all components of SILVER are applied. **Best** and runner-up of different configurations are denoted by **bold** and underlined texts, respectively. Last line is copied from Chen and Yu (2021).

High quality. The candidate list is kept small, and contains only appropriate (i.e., high similarity & divergence) dialogues, which are rarely INS dialogues. When combined with a powerful ensemble classifier, the generated dialogues have a satisfactory quality.

High efficiency. Because INS dialogues rarely exist in the candidate list, many generated dialogues remain after election. Consequently, the number of available OOS dialogues increases rapidly, reaching the target number in only a few iterations.

5 Experiments

5.1 Datasets and configurations

Besides STAR used in Section 3, we also conducted experiments on FLOW (Andreas et al., 2020) and ROSTD (Gangal et al., 2020) data, with the same split for train/dev/test as Chen and Yu (2021).

The supporting model was a classifier finetuned on the task of intent classification, consisting of a pretrained BERT model[4], with two feed-forward layers above. The model inputs were the first 256 words of the dialogues. The model was optimized using the Adam algorithm (Kingma and Ba, 2014).

We forced the size of seed OOS dialogues to be 1% of INS, and the target number of generated OOS dialogues was 24 times the seed size.

Experiment results were evaluated using the following metrics: (1) AUROC, area under receiver operating characteristic curve, (2) AUPR, area under precision-recall curve, and (3) FPR@θ, false positive rate with threshold θ.

5.2 Results on OOS detection

Table 2 shows the key experiment results of SILVER for OOS detection. To verify the effectiveness of the three key components of SILVER corresponding to §4.1 to §4.3, we modified or removed some of these components. The effectiveness of each component is summarized below.

Effectiveness of self candidate generation. The performance of SILVER depends on the characteristics of the dataset. Hence, it performed differently on the three datasets.

For STAR, "Int." and "Ext." performed similarly.

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

As noted by Chen and Yu (2021), PersonaChat is particularly suitable for OOS detection of STAR data because it is a rich source of OOS dialogues.

For FLOW, "Int." performed better, indicating that PersonaChat is not well suited to build utterance pools. By building pools and generating candidates from training data, SILVER consistently performed well.

For ROSTD, "Ext." performed better. This is because all dialogues in ROSTD contained only one utterance. By replacing this utterance with utterances selected from the INS samples in the training data, the generated dialogues contained only INS dialogues and were filtered during the election. For the case of 'Int. + Rnd.," the generated INS dialogues were randomly labeled as OOS, importing a large amount of noise into training data, which significantly decreased the classification performance. For cases of "Int. + MV" and "Int. + TE," almost no new OOS dialogues were generated. In contrast, for the case of "Ext.," OOS dialogues were generated successfully, improving the classification performance.

Effectiveness of tree ensemble. Tree ensembles outperformed majority voting, and both outperformed random selection. This was verified by empirical investigations in Section 3.2. Although three kinds of features provided sufficient information for classification, the simple rules adopted by GOLD are too coarse to elect OOS dialogues accurately, even with majority voting. However, tree ensembles are an ideal substitute.

Effectiveness of iterative data augmentation. In all cases, iterative data augmentation improved the performance of OOS detection. Note that for cases without iterative data augmentation, we enlarged the candidate list to ensure enough OOS dialogues were generated. Hence, the performance gain was due to the improvement of *quality*, rather than *quantity* of the candidate lists.

In summary, except for special cases (e.g., length-1 dialogues in ROSTD), combining all three components of SILVER achieves the best performance for OOS detection.

5.3 Evaluation of data quality

Next, we analyzed the quality of the generated data. It should be emphasized that the quality of generated dialogues is evaluated *intrinsically* not extrinsically. Specifically, we focus on evaluating (1) the quality of the generated dialogue *itself* and (2) the generated data *as a whole*. Herein the gain of the classification performance contributed by generated data is *not* considered. Although *intrinsic* high-quality does not necessarily contribute to extrinsic tasks directly, it is indispensable in real applications.

	# Unique utterances in generated data	#Unique utterances in utterance pool
GOLD	$4,153$	$93,472$
SILVER	$5,289$	$32,320$

Table 3: Numbers of unique utterances.

To evaluate the quality of generated dialogue *itself*, we evaluate whether the generated dialogues are (1) OOS and (2) natural. The second row of Table 1 shows the human evaluation results of dialogues generated by SILVER. Compared with the first row, SILVER outperformed GOLD on both evaluation metrics.

To evaluate the quality of generated data *as a whole*, we compared the generated data of GOLD and SILVER with the original data. This resulted in the following observations:

SILVER-generated data has a larger diversity. It is possible that one utterance was selected twice during generation. This reduces the diversity of the generated data. Table 3 shows the numbers of unique utterances in the generated data and the utterance pools. Although SILVER utilized a much smaller pool (built on training data), the generated data contains more unique utterances, indicating a larger diversity.

SILVER generated IND OOS data. An advantage of SILVER is its ability to generate INS OOS dialogues. We calculated the representations of the original OOS dialogues and the dialogues generated by GOLD or SILVER using vanilla RoBERTa (Liu et al., 2019). Figure 6 shows the 2-dim t-SNE visualization (Van der Maaten and Hinton, 2008) of these representations along with the average Mahalanobis distances between clusters. The OOS dialogues generated by GOLD differed from the original ones, indicating that these dialogues are OOD. In contrast, the overlap between the original OOS and SILVER-generated dialogues is large, implying that SILVER generates IND data.

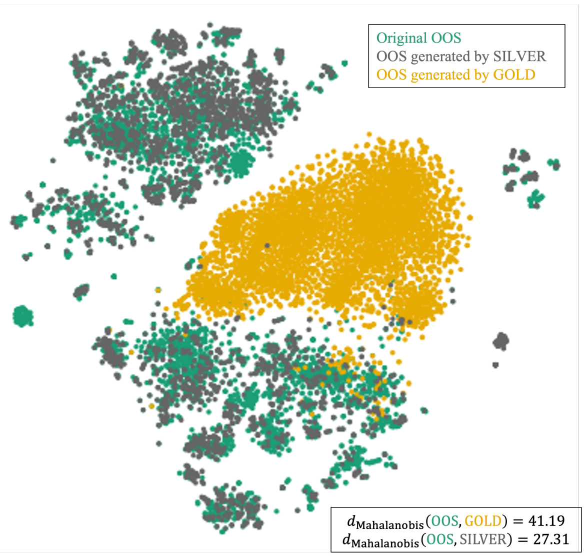


Figure 6: Visualization of the original and generated OOS dialogues.

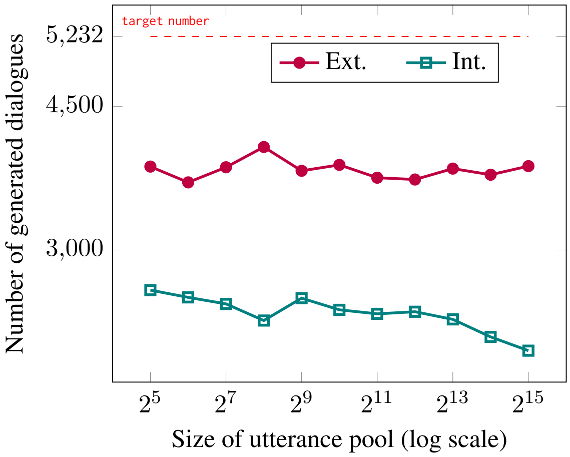


Figure 7: Number of generated dialogues after one iteration. To control the size of the utterance pool, a different number of utterances is randomly selected.

6 Discussion and Analysis

We conducted follow-up experiments to demonstrate the necessity of the components in SILVER and to deepen the understanding of its behavior.

Stability on small utterance pools. It is possible that SILVER fails when there is insufficient training data to build the utterance pool. We claim that SILVER performs stably even when the pool contains only a few utterances. Figure 7 shows the number of generated dialogues after one iteration with different pool sizes. Given only 2^5 utterances in the pool, SILVER successfully generated a reasonable number of OOS dialogues no matter how the pools are built.

Source	Pool size	AUROC	AUPR	FPR@.95	FPR@.90
Ext.	2^5	0.8960	0.4334	39.9%	28.6%
Ext.	2^{15}	0.8716	0.3867	59.2%	39.8%
Int.	2^5	0.8634	0.3504	57.1%	36.2%
Int.	2^{15}	0.8762	0.3823	52.5%	32.4%

Table 4: Performances of OOS detection on development data of STAR with different pool sizes. Tree ensemble and iterative data augmentation are applied. For each case, experiments are repeated five times, and the average results are used to remove fluctuations.

Utterance source	AUROC	AUPR	FPR@.95	FPR@.90
$\mathcal{N}(4)$	0.8222	0.2535	68.8%	48.0%
$\mathcal{N}(16)$	0.8705	0.3099	57.7%	37.2%
$\mathcal{N}(16) - \mathcal{N}(4)$	0.8743	0.4215	50.3%	35.4%

Table 5: Classification results on the development data of STAR. Utterance sources differ for self candidate generation. $\mathcal{N}(k)$ is the set of k-nearest utterances from the original utterance.

Using these generated dialogues to train a classifier for OOS detection, we obtain results shown in Table 4. With only 2^5 internal utterances, OOS detection performs sufficiently well. Surprisingly, increasing pool sizes is not necessarily beneficial to OOS detection, due to the increment of the possibility of bringing in noise.

Trade-off of similarity and divergence for candidate generation. Table 5 highlights the requirement for two criteria to select the appropriate utterances during the candidate generation introduced in Section 4.1. Because utterances in $\mathcal{N}(4)$ are almost the same as those in the training data, the generated dialogues do not improve the classification performance. Meanwhile, $\mathcal{N}(16)$ provides more choices for candidate generation, and removing utterances from $\mathcal{N}(4)$ further improves the performance.

Feature set for tree ensemble. Tree ensembles use three types of features. To verify their necessities, we evaluated them in terms of three metrics.

- Precision on INS. This is crucial to data augmentation. If this is low, an INS dialogue will be mistakenly included in the generated data. This negatively impacts the classifier training process due to the existence of a large amount of noise.
- Recall on OOS. This measures the efficiency of data augmentation. If this is low, OOS dialogues will rarely be selected during election, resulting in an endless generation process.

Features	Precision (INS)	Recall (OOS)	Overall F1
Probability-based	0.9693	0.5169	0.9551
Distance-based	0.9686	0.5056	0.9554
Ensemble-based	0.9679	0.4944	0.9528
All	0.9700	0.5281	0.9551

Table 6: Comparison of feature sets (ref. §4.2) used for the tree ensemble, evaluated on development data of STAR.

		Epochs of iteration		
Data	Goal	0	1	2
STAR	5,450	218	3,937	8,851
FLOW	15,500	620	9,963	23,349

Table 7: Number of OOS dialogues after each iteration.

- Overall F1. This is a total evaluation of the ensemble classifier that considers the issue of data unbalance.

Table 6 shows the results of these metrics for different feature sets. Precision on INS is equally good for all three feature sets, indicating the feasibility of learning a classifier using these features.[5] Combining all feature sets slightly increased both precision on INS and recall on OOS due to the power of the tree ensemble algorithm.

High-efficiency iterative data augmentation. As introduced in Section 4.3, an advantage of iterative data augmentation is high efficiency, which means that sufficient data can be obtained quickly. Table 7 verifies this advantage. Initially (Iteration 0), there is only a small seed of OOS dialogues. However, the number of OOS dialogues exceeded the goal after two epochs.

7 Conclusion

We proposed SILVER to generate OOS dialogues without using external data. The components in SILVER are designed to overcome issues and realize the full potential of state-of-the-art augmentation methods. Using only training data, SILVER successfully generated high-quality IND OOS dialogues, which not only contributed to the improved performance of *extrinsic* tasks such as OOS detection, but are also natural enough *intrinsically*, indicating the potential for future applications.

Limitations

Follow-up experiments to investigate situations where SILVER fails were conducted to demonstrate the limitations of the proposed method. SILVER failed on the ROSTD data using utterance pools built on training data (ref. Table 2). Thus, we investigated the characteristics of the target data (e.g., ROSTD), which may hinder SILVER's success.

Figure 8 investigates the dialogue length distribution of the target data. SILVER tends to filter out short dialogues while keeping long dialogues. This is within our expectations. A generated dialogue of length 3 contains $\frac{1}{3} \approx 33\%$ utterances selected from the INS part in the training data. In contrast, this is reduced to $\frac{1}{9} \approx 11\%$ for a dialogue of length 9. The limiting case has only 1 utterance in each dialogue. Consequently, all generated dialogues contain 100% INS utterances. This is exactly the case for the ROSTD data.

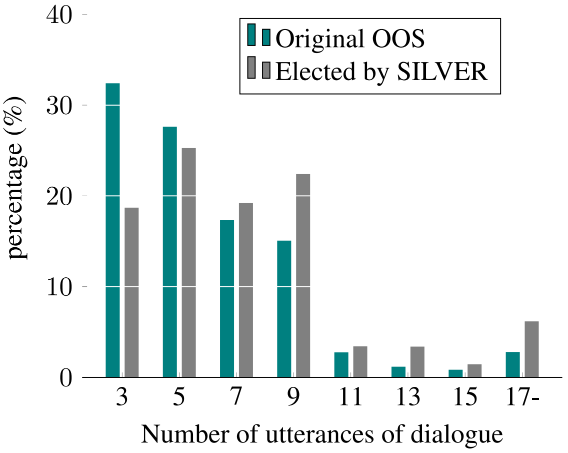


Figure 8: Distributions of dialogue length (i.e., number of utterances). Compared with short dialogues, long dialogues tend to be elected.

Ethics Statement

The proposed method aims to generate dialogues to aid in training a classifier for OOS detection. Instead of generating dialogues from scratch, utterances are extracted from existing benchmark datasets, including STAR, FLOW, ROSTD, and PersonaChat. To our knowledge, these datasets have been collected in a legal manner and do not contain sentences with ethics issues. They are widely used in previous studies in the area of NLP. Therefore, our proposed method is unable to generate data that can be used for unethical or illegal purposes. We comply with the ACL Ethics Policy.

[5] Recall that a single feature set (e.g., probability-based feature) consists of many components (e.g., the probability distribution over all possible categories). Thus, the tree ensemble algorithm still works.

References

Jacob Andreas, John Bufe, David Burkett, Charles Chen, Josh Clausman, Jean Crawford, Kate Crim, Jordan DeLoach, Leah Dorner, Jason Eisner, Hao Fang, Alan Guo, David Hall, Kristin Hayes, Kellie Hill, Diana Ho, Wendy Iwaszuk, Smriti Jha, Dan Klein, Jayant Krishnamurthy, Theo Lanman, Percy Liang, Christopher H. Lin, Ilya Lintsbakh, Andy McGovern, Aleksandr Nisnevich, Adam Pauls, Dmitrij Petters, Brent Read, Dan Roth, Subhro Roy, Jesse Rusak, Beth Short, Div Slomin, Ben Snyder, Stephon Striplin, Yu Su, Zachary Tellman, Sam Thomson, Andrei Vorobev, Izabela Witoszko, Jason Wolfe, Abby Wray, Yuchen Zhang, and Alexander Zotov. 2020. Task-Oriented Dialogue as Dataflow Synthesis. *Transactions of the Association for Computational Linguistics*, 8:556–571.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 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, Brussels, Belgium. Association for Computational Linguistics.

Derek Chen and Zhou Yu. 2021. GOLD: Improving out-of-scope detection in dialogues using data augmentation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 429–442, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Ssu Chiu, Maolin Li, Yen-Ting Lin, and Yun-Nung Chen. 2022. SalesBot: Transitioning from chit-chat to task-oriented dialogues. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6143–6158, Dublin, Ireland. Association for Computational Linguistics.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. 2018. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, Prague, Czech Republic. Association for Computational Linguistics.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Li Dong, Chris Quirk, and Mirella Lapata. 2018. Confidence modeling for neural semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 743–753, Melbourne, Australia. Association for Computational Linguistics.

Jean Feng, Arjun Sondhi, Jessica Perry, and Noah Simon. 2019. Selective prediction-set models with coverage guarantees. *arXiv preprint arXiv:1906.05473*.

Giorgio Fumera, Ignazio Pillai, and Fabio Roli. 2003. Classification with reject option in text categorisation systems. In *12th International Conference on Image Analysis and Processing, 2003. Proceedings.*, pages 582–587. IEEE.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

Varun Gangal, Abhinav Arora, Arash Einolghozati, and Sonal Gupta. 2020. Likelihood ratios and generative classifiers for unsupervised out-of-domain detection in task oriented dialog. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7764–7771.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *Proceedings of International Conference on Learning Representations*.

Yutai Hou, Yijia Liu, Wanxiang Che, and Ting Liu. 2018. Sequence-to-sequence data augmentation for dialogue language understanding. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1234–1245, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Amita Kamath, Robin Jia, and Percy Liang. 2020. Selective question answering under domain shift. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5684–5696, Online. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Petr Marek, Vishal Ishwar Naik, Vincent Auvray, and Anuj Goyal. 2021. Oodgan: Generative adversarial network for out-of-domain data generation. *arXiv preprint arXiv:2104.02484*.

Llew Mason, Jonathan Baxter, Peter Bartlett, and Marcus Frean. 1999. Boosting algorithms as gradient descent. *Advances in neural information processing systems*, 12.

Johannes EM Mosig, Shikib Mehri, and Thomas Kober. 2020. Star: A schema-guided dialog dataset for transfer learning. *arXiv preprint arXiv:2010.11853*.

Nathan Ng, Kyunghyun Cho, and Marzyeh Ghassemi. 2020. SSMBA: Self-supervised manifold based data augmentation for improving out-of-domain robustness. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1268–1283, Online. Association for Computational Linguistics.

Tong Niu and Mohit Bansal. 2019. Automatically learning data augmentation policies for dialogue 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 1317–1323, Hong Kong, China. Association for Computational Linguistics.

Alexander Podolskiy, Dmitry Lipin, Andrey Bout, Ekaterina Artemova, and Irina Piontkovskaya. 2021. Revisiting mahalanobis distance for transformer-based out-of-domain detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13675–13682.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Evgeniia Razumovskaia, Ivan Vulić, and Anna Korhonen. 2022. Data augmentation and learned layer aggregation for improved multilingual language understanding in dialogue. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2017–2033, Dublin, Ireland. Association for Computational Linguistics.

Ming Tan, Yang Yu, Haoyu Wang, Dakuo Wang, Saloni Potdar, Shiyu Chang, and Mo Yu. 2019. Out-of-domain detection for low-resource 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 3566–3572, Hong Kong, China. Association for Computational Linguistics.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

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 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Eyup Yilmaz and Cagri Toraman. 2022. D2U: Distance-to-uniform learning for out-of-scope detection. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2093–2108, Seattle, United States. Association for Computational Linguistics.

Dani Yogatama, Cyprien de Masson d'Autume, Jerome Connor, Tomas Kocisky, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, et al. 2019. Learning and evaluating general linguistic intelligence. *arXiv preprint arXiv:1901.11373*.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. 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 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

A Details of Datasets

Throughout this paper, we conducted experiments on three datasets: STAR (Mosig et al., 2020), FLOW (Andreas et al., 2020) and ROSTD (Gangal et al., 2020). Statistics of each dataset is shown in Table 8. All these datasets consist of task-oriented dialogues. Each dialogue consists of one or several utterances between human and chatbot/human. All utterances are in English.

All these datasets follow the MIT license. Copyrights belong to their creators. Our use of these datasets was consistent with their intended use, i.e., for the research on dialogues of natural languages. All datasets are sufficiently anonymized to make

split	STAR	FLOW	ROSTD
train	22,051/1,248	60,119/4,499	30,521/3,200
dev	2,751/178	3,239/228	4,181/453
test	2,708/168	3,227/239	8,621/937

Table 8: Numbers of INS/OOS dialogues in each dataset.

identification of individuals impossible. We randomly sampled 100 dialogues, and asked human workers to check these dialogues. We found that these dialogues do *not* contain any information that names or uniquely identifies individual people, and do *not* contain offensive content.

B Details of Experiments

We have reported key configurations in Section 5.1. In this section, we report more details of experiments to reproduce reported results.

All experiments were conducted on Google Cloud Platform.[6] The instance used for experiments contains one GPU (Nvidia T4).

For data augmentation, we implement the tree ensemble module using XGBoost library.[7] Grid search is used for searching the best hyperparameters for tree ensemble. For STAR dataset, if we use all three types of features, it takes about 70 minutes for tree ensemble.

After data augmentation, we train a binary classifier to detect OOS dialogues. The classifier consists of a `bert-base-uncased` model (109 million parameters), and two feed-forward layers (231 thousand parameters). We resort to libraries (e.g., `pytorchlightning`,[8] `transformers`,[9] etc.) to simplify implementation. For STAR dataset, it takes about 10 minutes for each epoch. We stop training after 13 epochs, and select the model with the largest AUROC on the development data as the final model.

For evaluation, we resort to `scikit-learn` library.[10] Specifically, we use `roc_auc_score` and `average_precision_score` functions to calculate AUROC and AUPR, respectively. Calculation of FPR@θ is implemented by ourselves, by simply combining `roc_curve` function with binary search.

C Details of Human Annotation

To evaluate the quality of automatically generated dialogues, we randomly sampled 50 dialogues and ask human workers to check them manually. Human evaluation results have been reported in Table 1. In this section, we report more details of human annotation.

Figure 9 is the screenshot of user-interface for annotators. We aimed at evaluating the quality in two aspects: OOS correctness and dialogue naturalness.

For OOS correctness, we gave annotators the following instruction.

> Is this dialogue really out-of-scope? For example, the chatbot can only deal with hotel reservation, but the customer asks today's weather. Another example is that the customer becomes angry because the chatbot cannot understand his/her intention.

For naturalness, we gave annotators the following instruction.

> Does the replaced utterance make the whole dialogue strange? Specifically, if the dialogue remains to be natural after replacing by the new utterance, then label this dialogue as "natural," otherwise, label this dialogue as "unnatural."

All annotators are full-time employees affiliated in the same team as the authors. They all have high levels of English proficiency, and are able to annotate dialogues correctly. Annotation was done in an in-house environment, and all dialogues are used only for the purpose of research. After annotation, no ethics issues were reported.

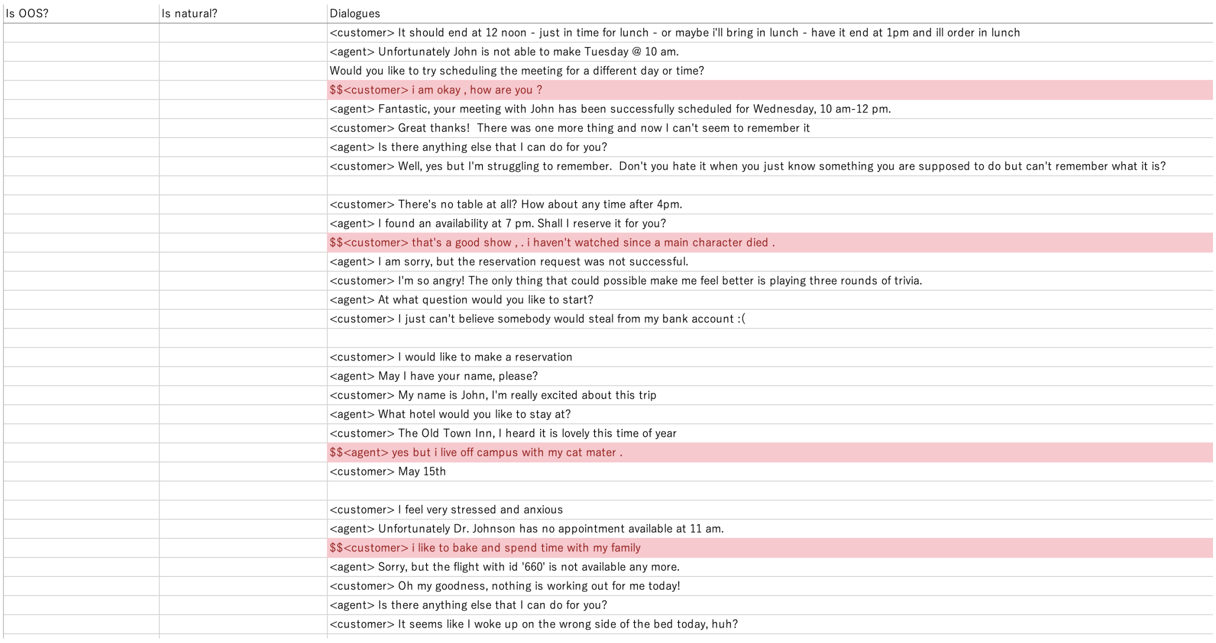
[6] https://console.cloud.google.com/

[7] https://xgboost.ai

[8] https://www.pytorchlightning.ai

[9] https://github.com/huggingface/transformers

[10] https://scikit-learn.org/



Is OOS?	Is natural?	Dialogues
		<customer> It should end at 12 noon - just in time for lunch - or maybe i'll bring in lunch - have it end at 1pm and ill order in lunch
		<agent> Unfortunately John is not able to make Tuesday @ 10 am.
		Would you like to try scheduling the meeting for a different day or time?
		$$<customer> i am okay , how are you ?
		<agent> Fantastic, your meeting with John has been successfully scheduled for Wednesday, 10 am-12 pm.
		<customer> Great thanks! There was one more thing and now I can't seem to remember it
		<agent> Is there anything else that I can do for you?
		<customer> Well, yes but I'm struggling to remember. Don't you hate it when you just know something you are supposed to do but can't remember what it is?
		<customer> There's no table at all? How about any time after 4pm.
		<agent> I found an availability at 7 pm. Shall I reserve it for you?
		$$<customer> that's a good show , . i haven't watched since a main character died .
		<agent> I am sorry, but the reservation request was not successful.
		<customer> I'm so angry! The only thing that could possible make me feel better is playing three rounds of trivia.
		<agent> At what question would you like to start?
		<customer> I just can't believe somebody would steal from my bank account :(
		<customer> I would like to make a reservation
		<agent> May I have your name, please?
		<customer> My name is John, I'm really excited about this trip
		<agent> What hotel would you like to stay at?
		<customer> The Old Town Inn, I heard it is lovely this time of year
		$$<agent> yes but i live off campus with my cat mater .
		<customer> May 15th
		<customer> I feel very stressed and anxious
		<agent> Unfortunately Dr. Johnson has no appointment available at 11 am.
		$$<customer> i like to bake and spend time with my family
		<agent> Sorry, but the flight with id '660' is not available any more.
		<customer> Oh my goodness, nothing is working out for me today!
		<agent> Is there anything else that I can do for you?
		<customer> It seems like I woke up on the wrong side of the bed today, huh?

Figure 9: Screenshot of user-interface for annotators. Replaced utterances are marked in red.