CONTEXTUAL OUT-OF-DOMAIN UTTERANCE HANDLING WITH COUNTERFEIT DATA AUGMENTATION

Sungjin Lee

Microsoft Research
One Microsoft Way
Redmond, WA 98052, USA

Igor Shalyminov[*]

School of Mathematical and Computer Sciences
Heriot-Watt University
Edinburgh, EH14 4AS, UK

ABSTRACT

Neural dialog models often lack robustness to anomalous user input and produce inappropriate responses which leads to frustrating user experience. Although there are a set of prior approaches to out-of-domain (OOD) utterance detection, they share a few restrictions: they rely on OOD data or multiple sub-domains, and their OOD detection is context-independent which leads to suboptimal performance in a dialog. The goal of this paper is to propose a novel OOD detection method that does not require OOD data by utilizing counterfeit OOD turns in the context of a dialog. For the sake of fostering further research, we also release new dialog datasets which are 3 publicly available dialog corpora augmented with OOD turns in a controllable way. Our method outperforms state-of-the-art dialog models equipped with a conventional OOD detection mechanism by a large margin in the presence of OOD utterances.

Index Terms— Out-of-domain utterance, Neural dialog model, Counterfeit data augmentation

1. INTRODUCTION

Recently, there has been a surge of excitement in developing chatbots for various purposes in research and enterprise. Data-driven approaches offered by common bot building platforms (e.g. Google Dialogflow, Amazon Alexa Skills Kit, Microsoft Bot Framework) make it possible for a wide range of users to easily create dialog systems with a limited amount of data in their domain of interest. Although most task-oriented dialog systems are built for a closed set of target domains, any failure to detect *out-of-domain* (OOD) utterances and respond with an appropriate fallback action can lead to frustrating user experience. There have been a set of prior approaches for OOD detection which require both *in-domain* (IND) and OOD data [1, 2]. However, it is a formidable task to collect sufficient data to cover in theory unbounded variety of OOD utterances. In contrast, [3] introduced an in-domain verification method that requires only IND utterances. Later, with the rise of deep neural networks, [4] proposed an autoencoder-based OOD detection method which surpasses prior approaches without access to OOD data. However, those approaches still have some restrictions such that there must be multiple sub-domains to learn utterance representation and one must set a decision threshold for OOD detection. This can prohibit these methods from being used for most bots that focus on a single task.

1	*sys:*	Hello, welcome to the Cambridge restaurant system. How may I help you?
2	*usr:*	**What time is the next bus from squirrel hill to morewood?**
3	*sys:*	Sorry I didn't catch that. Could you please repeat?
4	*usr:*	**Bring me to the nearest grocery store**
5	*sys:*	Sorry I didn't catch that. Could you please repeat?
6	*usr:*	*So sorry man* I want a moderately priced restaurant in the west part of town
7	*sys:*	<api call>

Table 1. Augmented dialog example (OOD utterances in bold and segment-level OOD content in italics.)

The goal of this paper is to propose a novel OOD detection method that does not require OOD data by utilizing counterfeit OOD turns in the context of a dialog. Most prior approaches do not consider dialog context and make predictions for each utterance independently. We will show that this independent decision leads to suboptimal performance even when actual OOD utterances are given to optimize the model and that the use of dialog context helps reduce OOD detection errors. To consider dialog context, we need to connect the OOD detection task with the overall dialog task. Thus, for this work, we build upon *Hybrid Code Networks* (HCN) [5] since HCNs achieve state-of-the-art performance in a data-efficient way for task-oriented dialogs, and propose AE-HCNs which extend HCNs with an autoencoder (Figure 1). Furthermore, we release new dialog datasets which are three publicly available dialog corpora augmented with OOD turns in a controlled way (exemplified in Table 1) to foster further research. [1]

[*]The work was done during an internship at Microsoft Research

[1]https://github.com/sungjinl/icassp2019-ood-dataset.git

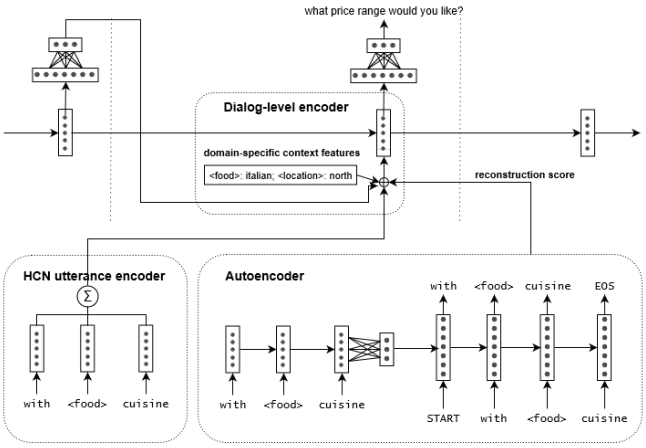

Fig. 1. The architecture of AE-HCN which is the same as HCN except for the autoencoder component.

2. METHODS

In this section, we first present the standard HCN model. Then we introduce the proposed AE-HCN(-CNN) model, consisting of an autoencoder and a reconstruction score-aware HCN model. Finally, we describe the counterfeit data augmentation method for training the proposed model.

2.1. HCN

As shown in Figure 1, HCN considers a dialog as a sequence of turns. At each turn, HCN takes a tuple, (U_t, a_{t-1}, s_t), as input to produce the next system action [2] a_t, where U_t is a user utterance consisting of N tokens, i.e., $U_t = \{u_1, ..., u_N\}$, a_{t-1} a one-hot vector encoding the previous system action and s_t a contextual feature vector generated by domain-specific code. The user utterance is encoded as a concatenation of a bag-of-words representation and an average of word embeddings of the user utterance:

$$x_t = [bow(U_t); average(e(u_1), ..., e(u_N))] \quad (1)$$

where $e(\cdot)$ denotes a word embedding layer initialized with GloVe [6] with 100 dimensions. HCN then considers the input tuple, (x_t, a_{t-1}, s_t), to update the dialog state through an LSTM [7] with 200 hidden units:

$$h_t = LSTM(h_{t-1}, [x_t; a_{t-1}; s_t]) \quad (2)$$

Finally, a distribution over system actions is calculated by a dense layer with a softmax activation:

$$P(a_t) = softmax(Wh_t + b) \quad (3)$$

[2] A system action can be either a text output or an api call.

2.2. AE-HCN

On top of HCN, AE-HCN additionally takes as input an autoencoder's reconstruction score r_t for the user utterance for dialog state update (Figure 1):

$$h_t = LSTM(h_{t-1}, [x_t; a_{t-1}; s_t; r_t]) \quad (4)$$

The autoencoder is a standard seq2seq model which projects a user utterance into a latent vector and reconstructs the user utterance. Specifically, the encoder reads U_t using a GRU [8] to produce a 512-dimensional hidden vector $h_N^{enc} (1 < n < N)$ which in turn gets linearly projected to a 200-dimensional latent vector z:

$$h_n^{enc} = GRU_{enc}(h_{n-1}^{enc}, e(u_n)) \quad (5)$$

$$z = W_z h_N^{enc} + b_z \quad (6)$$

The output of the decoder at step n is a distribution over words:

$$P_{dec}(y_n) = softmax(W_{dec} h_n^{dec} + b_{dec}) \quad (7)$$

$$h_n^{dec} = GRU_{dec}(h_{n-1}^{dec}, e(y_{n-1})) \quad (8)$$

$$h_0^{dec} = W_{dec} z + b_{dec} \quad (9)$$

where GRU_{dec} has 512 hidden units. The reconstruction score r_t is the normalized generation probability of U_t:

$$r_t = \frac{\sum_{n=0}^{N} \log P_{dec}(u_n)}{N} \quad (10)$$

2.3. AE-HCN-CNN

AE-HCN-CNN is a variant of AE-HCN where user utterances are encoded using a CNN layer with max-pooling (following [9]) rather than equation 1:

$$x_t = Pooling_{max}(CNN(e(u_1), ..., e(u_n))) \quad (11)$$

The CNN layer considers two kernel sizes (2 and 3) and has 100 filters for each kernel size.

2.4. Counterfeit Data Augmentation

To endow an AE-HCN(-CNN) model with a capability of detecting OOD utterances and producing fallback actions without requiring real OOD data, we augment training data with counterfeit turns. We first select arbitrary turns in a dialog at random according to a counterfeit OOD probability ρ, and insert counterfeit turns before the selected turns. A counterfeit turn consists of a tuple (U_t, a_{t-1}, s_t, r_t) as input and a fallback action a_t as output. We copy a_{t-1} and s_t of each selected turn to the corresponding counterfeit turns since OOD utterances do not affect previous system action and feature vectors generated by domain-specific code. Now we generate a counterfeit U_t and r_t. Since we don't know OOD utterances a priori, we randomly choose one of the user utterances of the

same dialog to be U_t. This helps the model learn to detect OOD utterances because a random user utterance is contextually inappropriate just like OOD utterances are. We generate r_t by drawing a sample from a uniform distribution, $U[\alpha, \beta]$, where α is the maximum reconstruction score of training data and β is an arbitrary large number. The rationale is that the reconstruction scores of OOD utterances are likely to be larger than α but we don't know what distribution the reconstruction scores of OOD turns would follow. Thus we choose the most uninformed distribution, i.e., a uniform distribution so that the model may be encouraged to consider not only reconstruction score but also other contextual features such as the appropriateness of the user utterance given the context, changes in the domain-specific feature vector, and what action the system previously took.

3. DATASETS

To study the effect of OOD input on dialog system's performance, we use three task-oriented dialog datasets: bAbI6 [10] initially collected for Dialog State Tracking Challenge 2 [11]; GR and GM taken from Google multi-domain dialog datasets [12]. Basic statistics of the datasets are shown in Table 2. bAbI6 deals with restaurant finding tasks, GM buying a movie ticket, and GR reserving a restaurant table, respectively. We generated distinct action templates by replacing entities with slot types and consolidating based on dialog act annotations.

We augment test datasets (denoted as Test-OOD in Table 2) with real user utterances from other domains in a controlled way. Our OOD augmentations are as follows:

- *OOD utterances*: user requests from a foreign domain — the desired system behavior for such input is a fallback action,
- *segment-level OOD content*: interjections in the user in-domain requests — treated as valid user input and is supposed to be handled by the system in a regular way.

These two augmentation types reflect a specific dialog pattern of interest (see Table 1): first, the user utters a request from another domain at an arbitrary point in the dialog (each turn is augmented with the probability p_{ood_start}, which is set to 0.2 for this study), and the system answers accordingly. This may go on for several turns in a row —each following turn is augmented with the probability p_{ood_cont}, which is set to 0.4 for this study. Eventually, the OOD sequence ends up and the dialog continues as usual, with a segment-level OOD content of the user affirming their mistake. While we introduce the OOD augmentations in a controlled programmatic way, the actual OOD content is natural. The OOD utterances are taken from dialog datasets in several foreign domains: 1) Frames dataset [13] — travel booking (1198 utterances); 2) Stanford Key-Value Retrieval Network Dataset [14] — calendar scheduling, weather information retrieval, city navigation (3030 utterances); 3) Dialog State Tracking Challenge 1 [15] — bus information (968 utterances).

In order to avoid incomplete/elliptical phrases, we only took the first user's utterances from the dialogs. For segment-level OOD content, we mined utterances with the explicit affirmation of a mistake from Twitter and Reddit conversations datasets — 701 and 500 utterances respectively.

bAbI6	**Train**	**Dev**	**Test**	**Test-OOD**
# dialogs	1618	500	1117	1117
Avg. turns per dialog	20.08	19.30	22.07	27.27
GR	**Train**	**Dev**	**Test**	**Test-OOD**
# dialogs	1116	349	775	775
Avg. turns per dialog	9.07	6.53	6.87	9.01
GM	**Train**	**Dev**	**Test**	**Test-OOD**
# dialogs	362	111	252	252
Avg. turns per dialog	8.78	9.14	8.73	11.25

Table 2. Data statistics. The numbers of distinct system actions are 58, 247, and 194 for bAbI6, GR, and GM, respectively.

4. EXPERIMENTAL SETUP AND EVALUATION

We comparatively evaluate four different models: 1) an HCN model trained on in-domain training data; 2) an AE-HCN-Indep model which is the same as the HCN model except that it deals with OOD utterances using an independent autoencoder-based rule to mimic [4] – when the reconstruction score is greater than a threshold, the fallback action is chosen; we set the threshold to the maximum reconstruction score of training data; 3) an AE-HCN(-CNN) model trained on training data augmented with counterfeit OOD turns – the counterfeit OOD probability ρ is set to 15% and β to 30. We apply dropout to the user utterance encoding with the probability 0.3. We use the Adam optimizer [16], with gradients computed on mini-batches of size 1 and clipped with norm value 5. The learning rate was set to 1×10^{-3} throughout the training and all the other hyperparameters were left as suggested in [16]. We performed early stopping based on the performance of the evaluation data to avoid overfitting. We first pretrain the autoencoder on in-domain training data and keep it fixed while training other components.

The result is shown in Table 3. Since there are multiple actions that are appropriate for a given dialog context, we use per-utterance *Precision@K* as performance metric. We also report f1-score for OOD detection to measure the balance between precision and recall. The performances of HCN on Test-OOD are about 15 points down on average from those on Test, showing the detrimental impact of OOD utterances to such models only trained on in-domain training data. AE-HCN(-CNN) outperforms HCN on Test-OOD by a large margin about 17(20) points on average while keeping the minimum performance trade-off compared to Test. Interestingly, AE-HCN-CNN has even better performance than HCN on Test, indicating that, with the CNN encoder, counterfeit OOD augmentation acts as

Domain	bAbI6			GR			GM		
Test Data	**Test**	**Test-OOD**		**Test**	**Test-OOD**		**Test**	**Test-OOD**	
Metrics	**P@1**	**P@1**	**OOD F1**	**P@3**	**P@3**	**OOD F1**	**P@3**	**P@3**	**OOD F1**
HCN	53.41	41.95	0	**58.89**	41.65	0	41.18	27.08	0
AE-HCN-Indep	31.29	41.06	48.68	51.90	55.42	71.52	31.12	42.78	64.35
AE-HCN	53.58	55.04	**73.41**	56.97	58.90	74.67	40.61	48.59	**69.31**
AE-HCN-CNN	**55.04**	**55.35**	70.38	58.32	**64.51**	**81.33**	**45.12**	**52.79**	68.59

Table 3. Evaluation results. P@K means Precision@K. OOD F1 denotes f1-score for OOD detection over utterances.

an effective regularization. In contrast, AE-HCN-Indep failed to robustly detect OOD utterances, resulting in much lower numbers for both metrics on Test-OOD as well as hurting the performance on Test. This result indicates two crucial points: 1) the inherent difficulty of finding an appropriate threshold value without actually seeing OOD data; 2) the limitation of the models which do not consider context. For the first point, Figure 2 plots histograms of reconstruction scores for IND and OOD utterances of bAbI6 Test-OOD. If OOD utterances had been known a priori, the threshold should have been set to a much higher value than the maximum reconstruction score of IND training data (6.16 in this case).

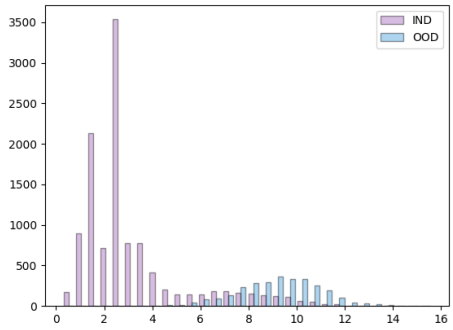

Fig. 2. Histograms of AE reconstruction scores for the bAbI6 test data. The histograms for other datasets follow similar trends.

For the second point, Table 4 shows the search for the best threshold value for AE-HCN-Indep on the bAbI6 task when given actual OOD utterances (which is highly unrealistic for the real-world scenario). Note that the best performance achieved at 9 is still not as good as that of AE-HCN(-CNN). This implies that we can perform better OOD detection by jointly considering other context features.

Finally, we conduct a sensitivity analysis by varying counterfeit OOD probabilities. Table 5 shows performances of AE-HCN-CNN on bAbI6 Test-OOD with different ρ values, ranging from 5% to 30%. The result indicates that our method manages to produce good performance without regard to the ρ value. This superior stability nicely contrasts with the high sensitivity of AE-HCN-Indep with regard to threshold values as shown in Table 4.

Threshold	Precision@1	OOD F1
6	40.39	48.38
7	42.56	50.46
8	43.69	51.08
9	52.21	63.86
10	47.27	44.44

Table 4. Performances of AE-HCN-Indep on bAbI6 Test-OOD with different thresholds.

Test Data	Test	Test-OOD	
Counterfeit OOD Rate	**Precision@1**	**Precision@1**	**OOD F1**
5%	55.25	55.48	69.72
10%	55.08	57.29	74.73
15%	55.04	55.35	70.38
20%	53.48	56.53	75.55
25%	53.72	56.66	73.13
30%	54.87	56.02	71.44

Table 5. Performances of AE-HCN-CNN on bAbI6 Test-OOD with varying counterfeit OOD rates.

5. CONCLUSION

We proposed a novel OOD detection method that does not require OOD data without any restrictions by utilizing counterfeit OOD turns in the context of a dialog. We also release new dialog datasets which are three publicly available dialog corpora augmented with natural OOD turns to foster further research. In the presence of OOD utterances, our method outperforms state-of-the-art dialog models equipped with an OOD detection mechanism by a large margin — more than 17 points in Precision@K on average — while minimizing performance trade-off on in-domain test data. The detailed analysis sheds light on the difficulty of optimizing context-independent OOD detection and justifies the necessity of context-aware OOD handling models. We plan to explore other ways of scoring OOD utterances than autoencoders. For example, variational autoencoders or generative adversarial networks have great potential. We are also interested in using generative models to produce more realistic counterfeit user utterances.

6. REFERENCES

[1] Mikio Nakano, Shun Sato, Kazunori Komatani, Kyoko Matsuyama, Kotaro Funakoshi, and Hiroshi G Okuno, "A two-stage domain selection framework for extensible multi-domain spoken dialogue systems," in *Proceedings of the SIGDIAL 2011 Conference*. Association for Computational Linguistics, 2011, pp. 18–29.

[2] Gokhan Tur, Anoop Deoras, and Dilek Hakkani-Tür, "Detecting out-of-domain utterances addressed to a virtual personal assistant," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[3] Ian Lane, Tatsuya Kawahara, Tomoko Matsui, and Satoshi Nakamura, "Out-of-domain utterance detection using classification confidences of multiple topics," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 150–161, 2007.

[4] Seonghan Ryu, Seokhwan Kim, Junhwi Choi, Hwanjo Yu, and Gary Geunbae Lee, "Neural sentence embedding using only in-domain sentences for out-of-domain sentence detection in dialog systems," *Pattern Recognition Letters*, vol. 88, pp. 26–32, 2017.

[5] Jason D Williams, Kavosh Asadi, and Geoffrey Zweig, "Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning," *arXiv preprint arXiv:1702.03274*, 2017.

[6] Jeffrey Pennington, Richard Socher, and Christopher D Manning, "Glove: Global vectors for word representation," *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, vol. 12, pp. 1532–1543, 2014.

[7] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[8] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," *Syntax, Semantics and Structure in Statistical Translation*, p. 103, 2014.

[9] Yoon Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1746–1751.

[10] Antoine Bordes and Jason Weston, "Learning end-to-end goal-oriented dialog," *arXiv preprint arXiv:1605.07683*, 2016.

[11] Matthew Henderson, Blaise Thomson, and Jason D. Williams, "The second dialog state tracking challenge," in *Proceedings of the SIGDIAL 2014 Conference, The 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 18-20 June 2014, Philadelphia, PA, USA*, 2014, pp. 263–272.

[12] Pararth Shah, Dilek Hakkani-Tür, Gokhan Tür, Abhinav Rastogi, Ankur Bapna, Neha Nayak, and Larry Heck, "Building a conversational agent overnight with dialogue self-play," *arXiv preprint arXiv:1801.04871*, 2018.

[13] Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman, "Frames: a corpus for adding memory to goal-oriented dialogue systems," in *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, Saarbrücken, Germany, August 15-17, 2017*, 2017, pp. 207–219.

[14] Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D. Manning, "Key-value retrieval networks for task-oriented dialogue," in *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, Saarbrücken, Germany, August 15-17, 2017*, 2017, pp. 37–49.

[15] Jason D. Williams, Antoine Raux, Deepak Ramachandran, and Alan W. Black, "The dialog state tracking challenge," in *Proceedings of the SIGDIAL 2013 Conference, The 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 22-24 August 2013, SUPELEC, Metz, France*, 2013, pp. 404–413.

[16] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *The International Conference on Learning Representations (ICLR).*, 2015.