Improving Robustness of Neural Dialog Systems in a Data-Efficient Way with Turn Dropout

Anonymous Author(s)
Affiliation
Address
email

Abstract

Neural network-based dialog models often lack robustness to anomalous, out-of-domain (OOD) user input which leads to unexpected dialog behavior and thus considerably limits such models' usage in mission-critical production environments. The problem is especially relevant in the setting of dialog system bootstrapping with limited training data and no access to OOD examples. In this paper, we explore the problem of robustness of such systems to anomalous input and the associated to it trade-off in accuracies on seen and unseen data. We present a new dataset for studying the robustness of dialog systems to OOD input, which is bAbI Dialog Task 6 augmented with OOD content in a controlled way. We then present turn dropout, a simple yet efficient negative sampling-based technique for improving robustness of neural dialog models. We demonstrate its effectiveness applied to Hybrid Code Networks (HCNs) on our data. Specifically, an HCN trained with turn dropout achieves more than **75%** per-utterance accuracy on the augmented dataset's OOD turns and **74%** F1-score as an OOD detector. Furthermore, we introduce a Variational HCN enhanced with turn dropout which achieves more than **56.5%** accuracy on the original bAbI Task 6 dataset, thus outperforming the initially reported HCN's result.

1 Introduction

Data-driven approaches for building dialog systems have recently passed the stage of open-ended academic research and are adopted in platforms like *Google Dialogflow*, *Apple SiriKit*, *Amazon Alexa Skills Kit*, and *Microsoft Cognitive Services*. However, most of those platforms' data-driven functionality is limited to Natural Language Understanding: user intent detection, named entity recognition, and slot filling. A more unified approach to dialog system bootstrapping — end-to-end dialog learning — is still only emerging as a commercial service, e.g. *Microsoft Conversation Learner*. Although still in its early age, end-to-end dialog learning from examples offers great potential: it doesn't require advanced programming skills and thus it makes it possible for a wider range of users to create dialog systems for their purposes. In turn, in the enterprise environment, end-to-end dialog learning bridges the gap between user experience designers and the actual working systems thus making product cycles and overall workflow faster.

From the technical point of view, the key issue in end-to-end training is the lack of robustness of the resulting systems. In the real-world setting of rapid dialog system prototyping, it is common to have only in-domain (IND) data for a closed target domain. This leads to a significant overfitting of machine learning methods and unpredictable behavior in the cases outside of what was seen during training. For a closed-domain dialog system, it's extremely important to maintain predictable behavior on anomalous, OOD user input.

Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.

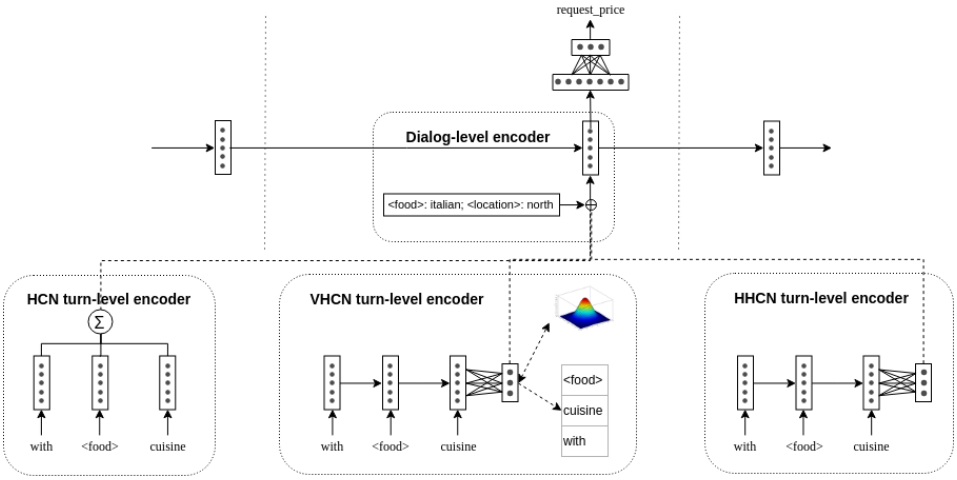

Figure 1: Hybrid Code Network model family

In this paper, we focus on studying the effect of OOD input on end-to-end goal-oriented dialog models' performance and propose a simple and efficient solution to improving robustness only using IND data. Our contribution is thus two-fold:

- We present a dataset for studying the effect of OOD input on dialog models.
- We present turn dropout, an efficient negative sampling-technique for training dialog models that are capable of OOD handling using only the IND data.

We show that HCN-based models enhanced with turn dropout show superior performance on OOD input, as well as surpass original HCN's result on IND-only data.

2 Related work

Detection of anomalous input is a key research problem in machine learning. In the area of dialog systems, there is a series of approaches to detecting and processing of OOD input. If treated as a classification problem, this problem require both IND and OOD data [10, 13]. Although for the real-world scenario of end-to-end dialog system learning the task of collecting data covering potentially unbounded variety of OOD input is impractical. In contrast, there are also approaches like an in-domain verification method [7] and an autoencoder-based OOD detection [11] which do not require OOD data. However, they still have restrictions such that there must be multiple sub-domains to learn utterance representation and one must set a decision threshold for OOD detection. For a dialog system that is supposed to work in a single closed domain, these methods are not a viable solution.

In contrast to those approaches, we present a simple and efficient technique for training dialog systems robust to OOD input in an end-to-end way, which allows the model to leverage the dialog context information to avoid the necessity of using IND data.

3 Dataset for studying robustness of dialog systems

In order to study the effect of OOD input on end-to-end dialog system's performance, we used a dataset of real human-computer goal-oriented dialogs and augmented it with real user utterances from other domains in a controlled way using the open-source toolkit bAbI tools[1] [12].

As our main dataset, we use bAbI Dialog Task 6 [2], real human-computer conversations in the restaurant serach domain initially collected for Dialog State Tracking Challenge 2 [5].

Our OOD augmentations are as follows:

[1] `https://bit.ly/babi_tools`

- *turn-level OOD*: user requests from a foreign domain — the desired system behavior for such input is the fallback action,
- *segment-level OOD*: 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 2): 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}), 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}. Eventually, the OOD sequence ends up and the dialog continues as usual, with a segment-level OOD of the user affirming their mistake. For this study, we set p_{ood_start} to 0.2 and p_{ood_cont} to 0.4[2].

While we introduce the OOD augmentations in a controlled programmatic way, the actual OOD content is natural. The turn-level OOD utterances are taken from dialog datasets in several foreign domains:

- Frames dataset [1] — travel booking (1198 utterances),
- Stanford Key-Value Retrieval Network Dataset [4] — calendar scheduling, weather information retrieval, city navigation (3030 utterances),
- 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, we mined utterances with the explicit affirmation of a mistake from Twitter and Reddit conversations datasets — 701 and 500 utterances respectively. Our datasets, as well as the tools for OOD-augmentation of arbitrary datasets of interest are openly available[3].

4 A data-efficient technique for training robust dialogue systems

4.1 Models

In this paper, we experiment with Hybrid Code Network family of models [14]. HCN is reported to be state-of-the-art for the original, IND-only bAbI Dialog Task 6 data. Thus, in this paper we experiment with it and explore its robustness to OOD input.

HCN is a hierarchical dialog control model with a turn-level and a dialog-level components (we will call them both encoders). The turn-level encoder produces a latent representation of a single dialog turn, and the dialog-level one augments it with additional dialog-level features and produces a latent representation of the entire dialog with an RNN. The resulting representation is then fed into the predictor MLP which outputs the final dialog actions (restricted by expert-provided binary action masks). Our models are described below — they share the same dialog-level encoder and predictor. The differences are on the turn level and in the overall optimization objective (see Figure 1 for an illustration).

HCN — the original model introduced by [14]. Its encoding of the turn t of N tokens is as follows:

$$HCN_t = \frac{1}{N}\sum_i word2vec(t_i)$$

Where $word2vec$ is the pre-trained Google News word2vec embeddings (frozen at the training time). HCN's optimization objective is catecogical cross-entropy with respect to negative log-likelihood (NLL) of resulting actions.

Hierarchical HCN (HHCN) uses an RNN (in our case an LSTM cell [6]) for encoding each utterance:

$$HHCN_t = LSTM(t)$$

[2]We experimented with other values of p_{ood_start} and p_{ood_cont} but didn't see significant differences in the results. Further experiments for different domains are encouraged using the tools provided

[3]See <link anonymized>

The optimization objective is the same as of HCN. Variants of this model were described by [8] and [9].

Variational HCN (VHCN) which, to the best of our knowledge, is presented here for the first time — uses a Variational Autoencoder as the turn-level encoder, so that the resulting turn encoding is VAE's latent variable (normally referred to as z):

$$VHCN_t = \mu(LSTM(t)) + \sigma(LSTM(t)) * N(0, 1)$$

Where μ and σ are MLPs for predicting z's posterior distribution parameters, and $N(0, 1)$ is a sample from its prior distribution, a standard Gaussian [3].

This model differs from the previous two in that it learns dialog control and autoencoding jointly. In order to keep the secondary task less complex than the main one, we represent VAE's reconstruction targets as bags of words (BoW). Thus, VHCN loss function is as follows:

$$L_{VHCN} = L_{NLL(a_i)} + L_{BoW(t_i)} + L_{KL(q_z||p_z)}$$

In the above formula, the first term is the main task's NLL loss for the dialog action a_i, the second one is the VAE's BoW reconstruction loss for the input turn t_i, and the last turn is KL-divergence between the prior and posterior distribution of the VAE's latent variable z — following [3], we compute it in a closed form.

Another benefit of the BoW loss is, as reported in [16], it helps keep the variational properties of the model (i.e. non-zero KL-term) without the necessity of using the KL-term annealing trick [3] which is itself challenging to control in practice. Unlike the authors of the original BoW loss approach, we don't stack softmax cross-entropy losses for each token and instead use a single sigmoid cross-entropy loss for the entire BoW vector.

All the models above use the same dialog-level LSTM encoder with additional features concatenated to the turn representations: BoW turn features, dialog context features, and previous system action[4].

4.2 Turn dropout

In order to train a system robust to OOD in the absence of real OOD examples, we employ a negative sampling-based approach and generate them synthetically from available IND data with a technique we call *turn dropout*. Namely, we replace random dialog turns with synthetic ones, and assign them the fallback action.

More formally, our dialog features are as follows: $< f\_turn, f\_ctx, f\_mask, a >$, i.e. turn features (token sequences), dialog context features, action masks, and target actions respectively.

Under turn dropout, for a randomly selected dialog i and its turn j, we replace f_turn_{ij} with a sequence of random vocabulary words (drawn from a uniform distribution over the vocabulary) and UNK tokens, and corresponding a_{ij} with the fallback action, and leave all other features intact. In this way, we're simulating anomalous turns for the system given usual contexts (as stored in the dialog RNN's state), and we put minimum assumptions on the synthesized turns' structure (we only limit their lengths to be within the bounds of the real utterances).

5 Experimental setup and evaluation

We train our models only using the original bAbI Dialog Task 6 dataset, and evaluate them on our OOD-augmented versions of it: we use the per-utterance accuracy as our main evaluation metric; the models are trained with the same hyperparameters (where applicable) listed in Table 3. The models use the common unified vocabulary including all words from our datasets (including OOD content): the intuition behind this is as follows: production dialog models often use word embedding matrices with vocabularies significantly exceeding that of the training data in order to take advantage of additional generalization power via relations like synonymy, hyponymy, or hypernymy normally efficiently handled by distributed word representations. Therefore, mapping every unseen word to an 'UNK' doesn't quite reflect that setting.

[4]Without the loss of the architecture generality, we have action mask vectors as additional features for the dialog-level LSTM [14], but they don't convey any information and are always set to 1's

Model	bAbI Dialog Task 6	bAbI Dialog Task 6 + OOD			
	Overall acc.	Overall acc.	Seg. OOD acc.	OOD acc.	OOD F1
HCN	0.557	0.438	**0.455**	0.0	0.0
HHCN	0.531	0.418	0.424	0.0	0.0
VHCN	0.533	0.413	0.413	0.0	0.0
TD-HCN	0.563	**0.575**	0.257	**0.754**	**0.743**
TD-HHCN	0.505	0.455	0.435	0.274	0.418
TD-VHCN	**0.565**	0.545	0.407	0.530	0.667

Table 1: Evaluation results

We tuned our models' hyperparameters using 2-stage grid search, tracking the development set accuracy. At the first stage, we adjusted the embedding dimensionality of our models (and the latent variable size in case of VHCN). Then, given the values found, at the second stage we adjusted turn dropout ratio at the interval $[0.05 - 0.7]$. Exact hyperparameter values are detailed in Table 3.

The results are shown in Table 1 — please note, apart from the accuracies we report OOD F1-measure, a metric showing the model's performance as a conventional OOD detector, with positive class being the fallback action, and negative — all the IND classes actions.

Finally, given the stochastic nature of VHCN, we reported its mean accuracy scores over 3 runs (we used the same criterion for selecting the best model during the training procedure).

6 Discussion and future work

In this paper, we explored the problem of robustness of neural dialog systems to OOD input. Specifically, we presented a dataset for studying this problem along with a general procedure for augmenting arbitrary datasets of interest for such purpose. Secondly, we introduced turn dropout, a simple yet efficient technique for improving OOD robustness of dialog control models and evaluated its effect on several Hybrid Code Network-family models.

As our experiments showed, while learning to handle both IND and OOD input with access to IND-only data at the training time, there appears the following trade-off: a model performing better on the 'clean' test turns is prone to lower accuracy on OOD — it can be said that it slightly overfits to its devset. On the other hand, a model regularized with turn dropout during training naturally performs better on unseen OOD turns, but with not as high accuracy on its 'clean', IND test data. Another side of the trade-off is the accuracy of OOD detection vs robust handling of IND input with segment-level noise. As our results showed, models specifically trained for OOD detection all demonstrate lower accuracy on the noisy IND.

Among the models we evaluated, it's worth noting that the original HCN demonstrated the best performance as an OOD detector (more than **74%** F1-score) and thus overall IND + OOD accuracy on the augmented dataset — more than **57%**. While some parts of its architecture (e.g. mean vector-based turn encoding or bag-of-words feature vector at the utterance level) may not seem to be the most robust solution, the model demonstrate superior overall performance. Averaging at the turn level instead of recurrent encoding (the case of HHCN and VHCN) makes the model less dependent on actual word sequences seen during training but on the keywords themselves.

In turn, VHCN demonstrated superior performance on IND data when trained with turn dropout, more than **56%** — it benefited in terms of both overall accuracy and the absence of false-positive OODs thus outperforming the original HCN as reported by [14]. An additional challenge was to train it while keeping its variational properties (i.e. reasonably high KL term) — the BoW reconstruction loss which we used in order to simplify the secondary task, helped with this as well [16]. On the other hand, while achieving superior performance on clean data, VHCN's properties didn't result in OOD handling improvements.

The question which is still unanswered is how these techniques apply to the setting of few-shot training. In the practical setup of training dialog systems from minimal data, having access to even medium-sized datasets like bAbI Dialog Task 6 isn't realistic, and all the initial requirements for the models have to be met only using the minimal training data available. It's the next step in our

research to explore how our techniques apply to this setup and what needs to be done in order to achieve OOD robustness with maximum few-shot data efficiency.

References

[1] 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*, pages 207–219, 2017.

[2] Antoine Bordes, Y-Lan Boureau, and Jason Weston. Learning end-to-end goal-oriented dialog. *ICLR*, 2017.

[3] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 10–21, 2016.

[4] 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*, pages 37–49, 2017.

[5] 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*, pages 263–272, 2014.

[6] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[7] 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*, 15(1):150–161, 2007.

[8] Sungjin Lee. Toward continual learning for conversational agents. *CoRR*, abs/1712.09943, 2017.

[9] Weiri Liang and Meng Yang. Hierarchical hybrid code networks for task-oriented dialogue. In De-Shuang Huang, Kang-Hyun Jo, and Xiao-Long Zhang, editors, *Intelligent Computing Theories and Application*, pages 194–204, Cham, 2018. Springer International Publishing.

[10] 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*, pages 18–29. Association for Computational Linguistics, 2011.

[11] 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*, 88:26–32, 2017.

[12] Igor Shalyminov, Arash Eshghi, and Oliver Lemon. Challenging Neural Dialogue Models with Natural Data: Memory Networks Fail on Incremental Phenomena. In *Proceedings of the 21st Workshop on the Semantics and Pragmatics of Dialogue (SemDial 2017 - SaarDial)*, 2017.

[13] 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.

[14] Jason D. Williams, Kavosh Asadi, and Geoffrey Zweig. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 665–677, 2017.

[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*, pages 404–413, 2013.

[16] Tiancheng Zhao, Ran Zhao, and Maxine Eskénazi. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 654–664, 2017.

Appendix A

1	*sys:*	Hello, welcome to the Cambridge restaurant system. You can ask for restaurants by area, price range or food type. How may I help you?
2	*usr:*	**I'd like to get away from Monday, August 15, 2016 to Wednesday, August 31, 2016. My budget is $3200 and I would leave from Detroit.**
3	*sys:*	Sorry I didn't catch that. Could you please repeat?
4	*usr:*	**Will there be frost in Durham next week?**
3	*sys:*	Sorry I didn't catch that. Could you please repeat?
4	*usr:*	**So sorry man** I want a moderately priced restaurant in the west part of town
5	*sys:*	<api call>

Table 2: Augmented dialog example (OOD content in bold)

Hyperparameter	**HCN**	**HHCN**	**VHCN**
Embedding size	64	128	128
Latent variable size	—	—	8
Learning rate	0.001		
Optimizer	Adam		
Early stopping threshold (epochs)	20		
Turn dropout ratio	0.4	0.6	0.3
Word dropout ratio	0.2		

Table 3: Model hyperparameters