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

Igor Shalyminov
Heriot-Watt University
Edinburgh, UK
is33@hw.ac.uk

Sungjin Lee
Microsoft Research
Redmond, USA
sule@microsoft.com

The problem

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 thus no access to OOD examples. In this paper, we explore the problem of robustness of such systems to anomalous input and the associated trade-off in accuracies on seen and unseen data.

A dataset for studying dialog systems' robustness

We created a corpus of dialogues to study OOD robustness in a controlled environment using `bAbI+ tools`, i.e. we augmented bAbI Dialog Task 6 (Dialog State Tracking Challenge 2) with user's utterances from several other domains:

- Frames dataset — travel booking (1198 utterances),
- Stanford Key-Value Retrieval Network Dataset — calendar scheduling, weather information retrieval, city navigation (3030 utterances),
- Dialog State Tracking Challenge 1 — bus information (968 utterances).

Our OOD augmentations are as follows:

- *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.

With these two augmentation types, we modelled a specific dialog pattern of interest: one of more user's OOD turns followed by self-correction and return to IND input (see Table 1).

For segment-level OOD, we mined utterances with the explicit affirmation of a mistake from Twitter and Reddit conversations datasets (e.g. ``my mistake'', ``I'm so sorry'') — 701 and 500 utterances respectively. Our datasets, as well as the tools for OOD-augmentation of arbitrary datasets of interest are openly available.

Example dialog

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 1: Augmented dialog example (OOD content in bold, segment-level in italics)

Model architecture

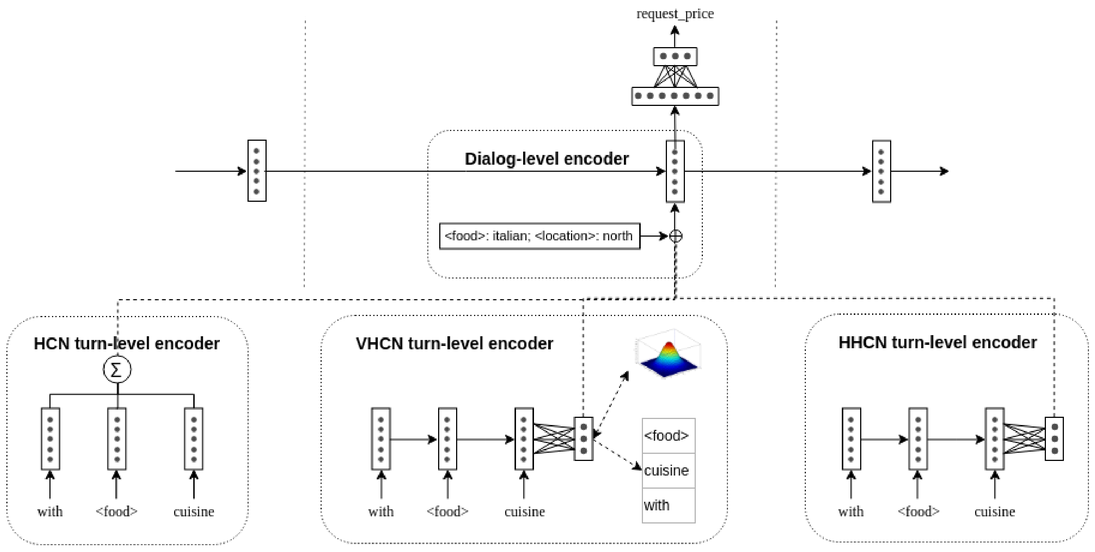

Figure 1: Hybrid Code Network-family models

Models

We experiment with *Hybrid Code Network* family of models. HCN is reported to be state-of-the-art for the original, IND-only bAbI Dialog Task 6 data — in this paper, we explore its robustness to OOD input.

Our models share dialog-level encoder and dialog act predictor, and they only differ on the turn level and in the overall optimization objective (see Figure 1). *The original* ***HCN*** encodes user's input turn x consisting of N tokens as follows:

$$HCN(x) = \frac{1}{N}\sum_i w2v(x_i) \quad (1)$$

where $w2v$ is the pre-trained Google News word2vec embeddings (frozen at the training time). HCN's optimization objective is categorical cross-entropy with respect to log-likelihood:

$$\mathcal{L}_{HCN} = \log p(a \mid x, c) \quad (2)$$

where a is the dialog action and c is dialog context.

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

$$HHCN(x) = LSTM(x) \quad (3)$$

The optimization objective is the same as of HCN.

Variational HCN (VHCN) — the model we are introducing here — uses a Variational Autoencoder for generating turn-level latent code z:

$$VHCN(x) = \mu(LSTM(x)) + \sigma(LSTM(x)) * N(0,1) \quad (4)$$

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.

VHCN optimization objective is as follows:

$$\mathcal{L}_{VHCN} = \mathbb{E}_{q(z)}[log(p(a \mid z, c))] + \mathbb{E}_{q(z)}[p(x_{BoW} \mid z)] - KL(q(z \mid x) \mid\mid p(z)) \quad (5)$$

In this objective, bag-of-words representation of the input x_{BoW} is used for the secondary task (input reconstruction, 2nd term) in order to lower the task's complexity. It also helps keep the variational properties of the model (i.e. non-zero KL-term) without employing KL-annealing.

Turn dropout

In the absence of real OOD examples, we employ a negative sampling-based approach and generate them synthetically from available IND data — namely, we replace random dialog turns with synthetic ones, and assign the fallback action to them. 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[i, j]` with a sequence of random words sampled uniformly from the the vocabulary, and corresponding `a[i, j]` with the fallback action, and leave all other features intact. In this way, we're simulating anomalous turns given usual contexts while putting minimum assumptions on their structure.

Results

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 2: Evaluation results

Discussion and future work

Among the models we evaluated, the original HCN trained with turn dropout demonstrated the best performance as an OOD detector and thus overall IND + OOD accuracy on the augmented dataset. The reason for the model's superior performance may be turn-level averaging instead of recurrent encoding (the case of HHCN and VHCN) which makes HCN less dependent on specific word sequences. In turn, VHCN trained with turn dropout achieves more than **56%** on clean data thus outperforming initial HCN's result in the original paper.

The next step in our research is to explore how our techniques apply to the few-shot setup and to achieve OOD robustness with maximum data efficiency in this setting.

bit.ly/ood_robust_hcn