Unsupervised Dialogue Spectrum Generation for Log Dialogue Ranking

Xinnuo Xu†, Yizhe Zhang‡, Lars Liden‡, Sungjin Lee‡
†The Interaction Lab, Heriot-Watt University, Edinburgh
`xx6@hw.ac.uk`
‡Microsoft Research, Redmond, WA, USA
`Yizhe.Zhang, Lars.Liden, sule@microsoft.com`

Abstract

Although the data-driven approaches of some recent bot building platforms make it possible for a wide range of users to easily create dialogue systems, those platforms don't offer tools for quickly identifying which log dialogues contain problems. This is important since corrections to log dialogues provide a means to improve performance after deployment. A log dialogue ranker, which ranks problematic dialogues higher, is an essential tool due to the sheer volume of log dialogues that could be generated. However, training a ranker typically requires labelling a substantial amount of data, which is not feasible for most users. In this paper, we present a novel unsupervised approach for dialogue ranking using GANs and release a corpus of labelled dialogues for evaluation and comparison with supervised methods. The evaluation result shows that our method compares favorably to supervised methods without any labelled data.

1 Introduction

Task-oriented dialogue systems provide a natural interface to accomplish various daily-life tasks such as restaurant finding and flight booking. 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 dialogue systems with a limited amount of data in their domain of interest. Typically, the development process of a dialogue system based on data-driven approaches (Williams et al., 2017; Bordes et al., 2016) goes around an operational loop in Figure 1: (1) The cycle begins with a developer creating a training dataset with seed dialogues. (2) A dialogue system is trained and deployed. (3) Real users interact with the system and generate log dialogues. (4) The developer reviews the logs to identify which log dialogues contain problems. (5) The developer updates the training dataset to fix the problems. (6) The cycle repeats from step 2). Of all steps, (4) is the most significant in slowing down the loop, because of the sheer volume of log dialogues that can be generated and the need to manually inspect each. Thus, it is essential to support tools that help developers quickly identify problematic log dialogues. To achieve this goal, we propose a neural dialog ranker whose goal is to place problematic dialogues higher in the rank.

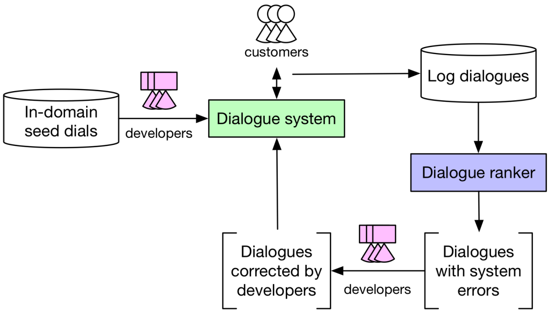

Figure 1: Operational loop of data-driven dialogue system development.

However, training a ranker typically requires labelling a substantial amount of data, which is not feasible for most developers. Furthermore, one might have to repeat this process whenever a significant change is made to the system's behavior. This motivates us to explore a set of unsupervised approaches to reduce the prohibitive cost. The core idea of these methods is that we learn a generative model to produce problematic dialogue examples as positive examples and train a ranker with seed dialogues used as negative examples. Specifically, we propose a novel dialogue generator using Generative Adversarial Networks (GANs) and train the generator with a curriculum learning scheme. Another possible avenue is to leverage off-the-shelf dialogue quality classifiers which are trained on open-domain corpora such as

Proceedings of the SIGDial 2019 Conference, pages 143–154
Stockholm, Sweden, 11-13 September 2019. ©2019 Association for Computational Linguistics

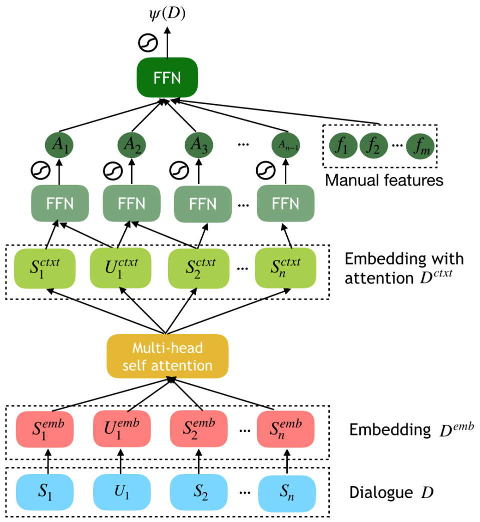

Figure 2: Overall architecture of our dialogue ranker.

dialogue breakdown detection challenge systems in DSTC6 (Higashinaka et al., 2017). In the experiment on the labelled dialogue corpus that we collected via Amazon Mechanical Turk, we show that our approach outperformes the off-the-shelf model by a significant margin thanks to the capability of generating domain-relevant problematic dialogues. The evaluation result also shows that our unsupervised method compares favorably to supervised methods without any labelled data.

The rest of this paper is organized as follows. In Section 2, we formalize the dialogue ranking task and describe our neural ranking model. In Section 3, we introduce a set of unsupervised methods for problematic dialogue example generation. Section 4 describes the datasets we used for this study. Section 5 explains our experiments. In Section 6, we discuss our experimental results. Section 7 provides a survey of related work. We finish with conclusions and future work in Section 8.

2 Dialogue Ranking

A dialogue ranker aims to assign higher scores to problematic dialogues than normal ones so that developers may quickly identify problematic dialogues in the ranked list of log dialogues. Formally, given a dialogue $\mathbf{D} = \{S_1, U_1, S_2, \cdots, U_{n-1}, S_n\}$, a dialogue ranker ψ produces a score of $\mathbf{D}$ being problematic where S_i and U_i are the system and user utterance in i^{th} turn, respectively.[1] To train the dialogue ranker ψ, we formulate the ranking task as binary classification where problematic and normal dialogues correspond to positive and negative classes, respectively. We optimize the cross-entropy objective:

$$\mathcal{L}_{\texttt{xent}} = \frac{1}{N}\sum_{i=1}^{N}[y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)]$$

where $\hat{y}_i = 1/(1 + \exp(-\psi(\mathbf{D}_i))$ and $y_i = 1$ for positive class and 0 otherwise.

We design a deep neural ranking model for ψ to automatically learn salient features as illustrated in Figure 2. We first use a bi-directional LSTM (Bi-LSTM) to encode each utterance in dialogue $\mathbf{D}$:

$$\mathbf{D}^{emb} = \{S_1^{emb}, U_1^{emb}, S_2^{emb}, \cdots, U_{n-1}^{emb}, S_n^{emb}\}$$

where $S_i^{emb} = \textit{Bi-LSTM}(S_i)$ and $U_i^{emb} = \textit{Bi-LSTM}(U_i)$. Then, we calculate attention embeddings for each utterance with a multi-head self-attention mechanism (Vaswani et al., 2017):

$$\mathbf{D}^{attn} = \text{softmax}\left(\frac{\mathbf{D}^{emb}\mathbf{D}^{emb\top}}{\sqrt{d}}\right)\mathbf{D}^{emb}$$

where d is the dimension of the embedding and $\mathbf{D}^{attn} = \{S_1^{attn}, U_1^{attn}, \cdots, U_{n-1}^{attn}, S_n^{attn}\}$. Now, we apply a feed-forward network (FFN) to the concatenation of utterance embeddings $\mathbf{D}^{emb}$ and their corresponding attentive embeddings $\mathbf{D}^{attn}$ to yield context-sensitive utterance embeddings:

$$\mathbf{D}^{ctxt} = \{S_1^{ctxt}, U_1^{ctxt}, , \cdots, U_{n-1}^{ctxt}, S_n^{ctxt}\}$$

where $S(U)_i^{ctxt} = FFN([S(U)_i^{emb}, S(U)_i^{attn}])$ and $[\cdot, \cdot]$ denotes a concatenation operator. After that, we apply another FFN followed by a sigmoid activation to each pair of utterances to measure the consistency of adjacency pairs:

$$A_i = \text{sigmoid}(FFN([X_i^{ctxt}, Y_i^{ctxt}]))$$

where (X_i^{ctxt}, Y_i^{ctxt}) is either (S_i^{ctxt}, U_i^{ctxt}) or $(U_i^{ctxt}, S_{i+1}^{ctxt})$. Finally, the ranker ψ produces a ranking score for the dialogue based on the consistency scores and a set of manually crafted features:

$$\psi(\mathbf{D}) = FFN([A_1, \cdots, A_{n-1}, f_1, \cdots, f_m])$$

where f_i denotes a set of manual features. In this study, we use a single manual feature to consider redundant turns:

$$f = \frac{\text{Num (distinct utterances)}}{\text{Num (all utterances)}}$$

---

[1] One turn consists of a pair of system and user utterances.

Each instance of FFNs has separate parameters and consists of two linear layers with a ReLU activation in between:

$$FFN(x) = \max(0, xW_1 + b_1) W_2 + b_2$$

3 Unsupervised Approach

Training a ranker typically requires labelling a substantial amount of data and one might have to repeat this process whenever a significant change is made to the system's behavior. This is not feasible for most developers and motivates us to explore a set of unsupervised approaches. The core idea is that we learn a generative user simulator and have it talk with the bot to produce problematic dialogues. We then train a ranker with seed dialogues used as normal examples. A straightforward approach for problematic dialogue generation is to train the generative user simulator on a dialogue corpus collected from a wide range of different domains, dubbed as `MultiDomain`. However, this approach can only produce obvious problematic dialogues where the simulated user mimics users who barely know what the bot is for.

To generate more relevant dialogues, one can fine-tune the `MultiDomain` model on the seed dialogues, dubbed as `FineTune`. But this approach gains an improved relevance at the cost of decreased diversity and it is a formidable task to adjust fine-tuning to strike the right balance between relevance and diversity.

We observe that, in most cases, a dialogue follows a natural course until a problem occurs and the dialogue subsequently gets off track. Table 1 shows a problematic dialogue. To bring this to our problematic dialogue generation, we introduce a novel stepwise fine-tuning approach, called `StepFineTune`. The idea is that we fine-tune the `MultiDomain` model only up to l-th turn to generate dialogues in which it normally unfolds up to l-th turn and starts seeing problems afterward. As we fine-tune the model in this stepwise fashion from $l = 1$ to n, we accumulate all the dialogues that we generate at each step. This allows us to produce a spectrum of diverse problematic dialogues while controlling relevance.

However, it is widely known that the typical MLE training scheme often generates bland and generic responses (Li et al., 2016). To alleviate this problem and generate naturally diverse dialogues, we propose a novel stepwise GAN training scheme, dubbed as `StepGAN`. `StepGAN` differs from `StepFineTune` in that it conducts GAN training instead of the simple MLE fine-tuning for each step. As we cast the dialogue ranking task as binary classification in Section 2, the dialogue ranking model ψ lends itself well to discriminating real dialogues from simulated ones. In the rest of this section, we describe `StepGAN` in detail.

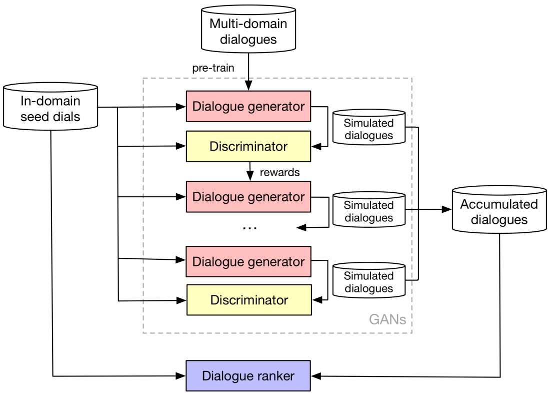

Figure 3: The overall pipeline of the StepGAN approach.

3.1 StepGAN overview

Figure 3 shows the overall pipeline of the `StepGAN` approach. A dialogue generator consists of a user simulator and the bot, and have them talk with each other. We start off by pre-training a generative user simulator on a large corpus of dialogues collected from multiple domains which teaches the simulator basic language skills and helps learn diverse out-of-domain behavior. We use the pre-trained user simulator to produce problematic dialogues and pre-train a discriminator with seed dialogues used as normal dialogues.

We then begin stepwise GAN training. At each step, starting from the user simulator of turn $l-1$ and the pre-trrained discriminator, we further train them using GAN to make the first l turns of a generated dialogue less distinguishable from the seed dialogues, as listed in Algorithm 1. To achieve this goal, we truncate both seed and generated dialogues up to length l when we train the user simulator and discriminator. Once the GAN training is done, we generate a set of problematic dialogues $\mathcal{D}^l_{pos}$ using the trained user simulator. Note that we don't truncate these dialogues so that they may follow a normal course up to l-th turn and start seeing problems afterward[2]. When we finish the final step L, we collect all the problematic dialogues generated from each step and construct a

---

[2] A dialogue ends either when the system or user terminates or when the pre-set maximum length is reached.

Algorithm 1 GAN training at step l

```
Require: Problematic dialogues $\mathcal{D}_{pos}$
$\mathcal{D}^l_{seed} \leftarrow$ seed dialogues truncated at turn $l$
$g^l \leftarrow$ user simulator from turn $l-1$
$d^l \leftarrow$ pre-trained discriminator
while not convergent do
    $\mathcal{D}^l_{gen} \leftarrow$ Truncate(Generate($g^l$), $l$)
    $d^l \leftarrow$ Train_d($\mathcal{D}^l_{seed}, \mathcal{D}^l_{gen}$)
    $g^l \leftarrow$ Train_g($\mathcal{D}^l_{gen}, d^l$)
end while
$\mathcal{D}^l_{pos} \leftarrow$ Generate($g^l$)
```

accumulated dataset:

$$\mathcal{D}_{pos} = \{\mathcal{D}^1_{pos}, \cdots, \mathcal{D}^L_{pos}\}$$

Finally, we train the dialogue ranker ψ on the accumulated problematic data $\mathcal{D}_{pos}$ and seed dialogues $\mathcal{D}_{seed}$.

3.2 GAN training details

Based on our empirical study, we choose to use a sequence-to-sequence model with attention for our user simulator. With GAN, at the l-th iteration, we optimize the following objective which basically adjusts the user simulator to fool the discriminator:

$$\begin{aligned} J(\theta) &= \mathbb{E}_{p_\theta(\mathbf{D})}\left[d^l([\mathbf{D}^{<l}, U_l])\right] \\ &+ \lambda \sum_{i=0}^{l} H_\theta(U_i \mid \mathbf{D}^{<i}) \end{aligned} \tag{1}$$

where $\mathbf{D}$ denotes a generated dialogue and $\mathbf{D}^{<i} = [S_1, U_1, \cdots, U_{i-1}, S_i]$. $d^l(\cdot)$ denotes the discriminator being trained at the l-th iteration and returns the probability of $\mathbf{D}^{<l}$ being real, as reward for training the generator. θ is the parameters for the user simulator and $H(\cdot)$ is the entropy penalty (Pereyra et al., 2017) for increasing the generation diversity:

$$H_\theta\left(U_i \mid \mathbf{D}^{<i}\right) = \sum_{j=0}^{N_u} H_\theta\left(u_j \mid \mathbf{D}^{<i}, u^{<j}\right) \tag{2}$$

where,

$$\begin{aligned} H_\theta\left(u_j \mid \mathbf{D}^{<i}, u^{<j}\right) = -\sum_{m=0}^{M} p_\theta\left(u_j^m \mid \mathbf{D}^{<i}, u^{<j}\right) \\ \cdot \log p_\theta\left(u_j^m \mid \mathbf{D}^{<i}, u^{<j}\right) \end{aligned} \tag{3}$$

In Eq 2 and 3, N_u is the number of tokens in U_i and M is the vocabulary size. $p_\theta(u_j^m \mid \mathbf{D}^{<i}, u^{<j})$ is the conditional distribution over the vocabulary at time step j in the generation of U_i. Since Eq 1 is not differentiable, we adopt the REINFORCE algorithm (Williams, 1992) for gradient updates:

$$\begin{aligned} \nabla_\theta J(\theta) \propto d^l(U_l, \mathbf{D}^{<l}) \sum_{i=0}^{l} \nabla_\theta \log p_\theta\left(U_i \mid \mathbf{D}^{<i}\right) \\ + \sum_{i=0}^{l} \nabla_\theta H_\theta\left(U_i \mid \mathbf{D}^{<i}\right) \end{aligned}$$

To stabilize the learning process, we employ two common techniques: 1) a baseline: we take the average of rewards in each training batch 2) teacher forcing: we occasionally draw a random dialogue from the seed dialogues with $d^l(\cdot)$ set to return 1. To increase the diversity of the output of the user simulator, during inference, we combine sampling with beam search. At each time step j, instead of choosing the top $beam\_size$ terms, we sample $beam\_size$ terms according to the probability distribution $p_\theta\left(\mathbf{u}_j \mid \mathbf{D}^{<i}, u^{<j}\right)$.

Since we cast the dialogue ranking task as binary classification, we use the same architecture as the dialogue ranking model in Section 2 to discriminate seed dialogues from simulated ones. The only difference is that seed and generated dialogues now correspond to positive and negative classes, respectively.

4 Datasets

In this work, we build a log dialogue ranker for the restaurant inquiry bot offered by the PyDial platform.[3] The task for the bot is to search for restaurants based on user's requirements in a multi-turn natural language communication. Three main corpora are introduced: (1) log dialogues with labels, (2) seed dialogues for the restaurant domain, (3) a large corpus of dialogues collected from multiple domains (Lee et al., 2019).

Log dialogues with labels

To collect log dialogues, we deployed the Pydial restaurant bot via the Amazon Mechanical Turk (AMT) platorm.[4] We ask turkers to find

---

[3] `http://www.camdial.org/pydial/`

[4] We use the data collection toolkit offered by ParlAI `http://www.parl.ai/static/docs/tutorial_mturk.html`.

restaurants that satisfy automatically generated requirements, such as food type, location and price range, by chatting with the restaurant bot. To make the conversation natural, we encourage turkers to speak in natural utterances and do not allow any turkers to carry out more than 20 dialogues in total. At the end of each task, turkers are required to answer a questionnaire whether they found restaurants meeting their requirements, and whether they experienced contextually unnatural turns in the conversation. We control the quality of a turker's judgements by checking if a turker judges correctly for some obvious cases that we can automatically identify.

From the collected dialogues, we label successful dialogues without any contextually unnatural turns as 0 (normal dialogue), and the rest as 1 (problematic dialogue). Table 2 shows the number and average length of log dialogues. Examples are shown in Table 1.

We split the corpus as shown in Table 3.

Note that, the training and validation sets are used only for supervised training, whereas the test set is used for evaluating all approaches.

Seed dialogues

The corpus of seed dialogues has two use cases: 1) we use it to fine-tune the user simulator for the `FineTune` and `StepFineTune` approaches, 2) `StepGAN` takes it as input to the discriminator training and teacher forcing process. Since the restaurant bot does not have associated seed dialogues, we collect 100 seed dialogues by having the bot talk with the agenda-based user simulator that Pydial offers. [5]

Multi-domain dialogues

The multi-domain corpus[6] has two use cases: 1) we use it for training the user simulator for the `MultiDomain` approach, 2) we pretrain the simulator for the `StepFineTune` and `StepGAN` approaches. The multi-domain corpus consists of around 40,000 dialogues with 11 turns on average. Each dialogue is a task-oriented conversational interaction between two real speakers over 51 domains and 242 tasks, collected by crowd-sourcing in which one turker is simulating a user and the other one is simulating a chatbot. We preprocess dialogues into training pairs for the sequence-to-sequence model learning. A training pair consists of a dialogue context and the corresponding response. We consider three consecutive turns as dialogue context and the following turn as response. The number of dialogues in train/val/test sets are 38000/1000/1000 and the training pairs are 394821/10411/10370.

5 Experiments

We describe a set of experimental settings in Section 5.1 and evaluation metrics in Section 5.2.

5.1 Experimental settings

We present three experiments to demonstrate the efficacy of our approaches.

- *`StepGAN` vs. other unsupervised approaches:* To compare the `StepGAN` against other unsupervised approaches introduced in Section 3:
  - *`MultiDomain`:* We train the user simulator on the multi-domain corpus.
  - *`FineTune`:* We fine-tune the user simulator on the seed dialogues.
  - *`StepFineTune`:* We fine-tune the user simulator in a stepwise fashion.

  We train dialogue rankers for each approach on a balanced dataset, consisting of 1000 problematic dialogues generated by each approach and 1000 normal dialogues oversampled from the seed dialogues to balance the class distribution. We evaluate two additional models:
  - *`DBDC3`:* We use the off-the-shelf CRF model trained on open-domain dialogue corpora[7] for the dialogue breakdown detection challenge (Higashinaka et al., 2017). Since the DBDC3 model operates at the utterance level, we aggregate the utterance scores to obtain a dialogue-level score. Based on our experiments, we choose to use a model trained on the collection of four challenge datasets (CIC_115, IRIS_100, TKTK_100, YI_100) and aggregate the utterance scores through summation.
  - *`PlainMultiDomain`:* Instead of training a user simulator, we simply take the entire multi-domain corpus (38,000 dialogues) as problematic dialogues and train the ranker with the same number of normal dialogues oversampled from the seed dialogues.

---

[5] We collected 100 seed dialogues based on our observations that most developers start training their bots with a seed dialogue corpus on a similar scale.

[6] The multi-domain corpus (MetaLWOz) will be made available through a DSTC8 track (Lee et al., 2019).

[7] https://github.com/dbd-challenge/dbdc3

One normal example in the labelled corpus
SYS: hello, welcome to the restaurant finding system. how may i help you?
USER: hi i ' m looking for a restaurant.
SYS: what part of town do you have in mind?
USER: the east part of town please.
SYS: what kind of food would you like?
USER: i'd like chinese food please.
SYS: yu garden is a nice place, it serves chinese food and it is in the east part of town.
USER: what is the phone number?
SYS: yu garden is a nice place their phone number is 0 1223 248882.
USER: what is the address?
SYS: yu garden is a nice place their address is 529 newmarket road fen ditton.
USER: thank you
SYS: thank you, goodbye.
One problematic example in the labelled corpus
SYS: hello, welcome to the restaurant finding system. how may i help you?
USER: hello i would really like to eat some cheap indian tonight
SYS: what part of town do you have in mind?
USER: i would really like to eat in the centre of town
SYS: i am sorry but there is no place, in the expensive price range, serving corsica food and in the area of centre.
USER: i would really like so indian thats cheap in the centre of town
SYS: the gandhi is a nice place, it is in the cheap price range, it is in the centre of town and it serves indian food.
USER: thats great, have you got a contact number?
SYS: what part of town do you have in mind?
USER: the centre of town
SYS: efes restaurant is a nice place, it is in the centre of town, it is in the moderate price range and it serves turkish food.
USER: i want to eat at the gandhi have you got the address?
SYS: the gandhi is a nice place their address is 72 regent street city centre.
USER: thank you, have you got a phone number for the gandhi?
SYS: the gandhi is a nice place their phone number is 0 1223 353942.
USER: thanks
SYS: chiquito bar is a nice place, it serves mexican food in the expensive price range and it is in the south part of town.
USER: quit

Table 1: One normal and one problematic examples in the labelled corpus.

	Normal dials	**Problematic dials**
Num of dials	1621	1329
Avg length	5.87 turns	8.28 turns

Table 2: The number of dialogues and average dialogue length in the log dialogue corpus.

	Train	**Val**	**Test**
Normal dialogues	1321	100	200
Problematic dialogues	1029	100	200

Table 3: Log dialogues are randomly sampled and split into training, validation and test sets.

- *`StepGAN` vs. supervised approach:* One advantage for the unsupervised approach is to generate a larger training set without human efforts. To compare the best performance of the `StepGAN` model against the supervised model, we train `StepGAN` models on three balanced datasets with 2000, 4000 and 6000 dialogues each and compare them to a supervised dialogue ranker trained on a balanced dataset of 2000 labelled dialogues randomly sampled from the training set described in Section 4.
- *Semi-supervised learning attempts:* On top of the labelled data, we can employ unsupervised approaches for data augmentation. For the evaluation of such a semi-supervised setting, we compare the performance of supervised models with 500 and 2000 labelled examples and that of their counterparts which leverage additional 6000 examples generated by `StepGAN`.

Note that, all dialogue rankers are tested on the 400-instance balanced test set described in Table 3. We train 10 models on randomly sampled training sets and report average performance.

5.2 Evaluation metrics

We use ranking metrics for evaluation:

- *P@K* – Precision at k, corresponds to the number of problematic dialogues in the top k ranked options.
- *R@K* – Recall at k, corresponds to the number of problematic dialogues in the top k ranked options against the number of all problematic dialogues in the test set (i.e. 200). Note that we modified the standard of Recall at k to get monotonic increase with respect to k.

6 Results and Discussion

In this section, we first present the results for the experimental settings in Section 5.1 that we de-

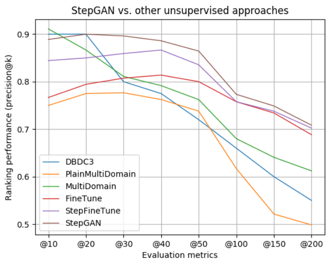

Figure 4: Precision@k StepGAN vs. other unsupervised approaches.

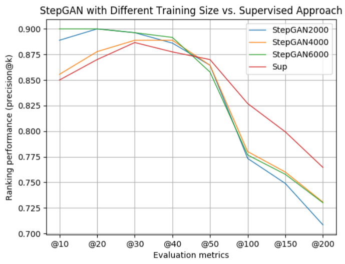

Figure 5: Precision@k StepGAN with different training size vs. supervised approach.

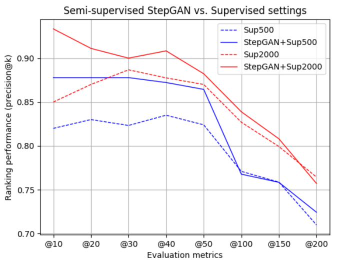

Figure 6: Precision@k of semi-supervised StepGAN vs. supervised settings.

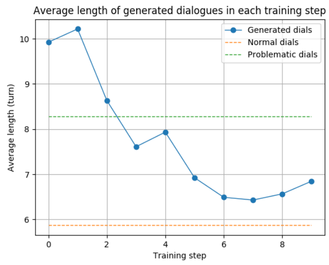

Figure 7: Average length of generated dialogues in each training step.

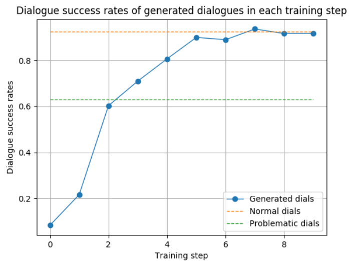

Figure 8: Success rates of generated dialogues in each training step.

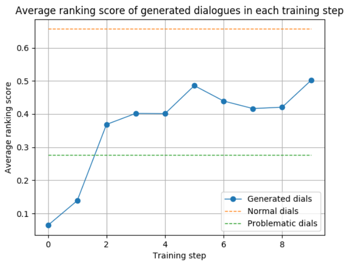

Figure 9: Average score of generated dialogues in each training step.

Model	DB	PM	MD	FT	SF	SG	Sup
P@10	.700	.750	**.911**	.767	.844	.889	.850
P@20	.800	.775	.867	.794	.850	**.900**	.870
P@30	.800	.777	.811	.807	.859	**.896**	.887
P@40	.825	.763	.792	.814	.867	**.886**	.878
P@50	.800	.738	.762	.800	.836	.864	**.870**
P@100	.720	.617	.680	.758	.758	.773	**.827**
P@150	.713	.521	.641	.734	.738	.749	**.799**
P@200	.655	.499	.612	.688	.702	.708	**.765**
R@10	.035	.038	**.046**	.038	.042	.044	.043
R@20	.080	.077	.087	.079	.085	**.090**	.087
R@30	.120	.117	.122	.121	.129	**.134**	.133
R@40	.165	.153	.158	.163	.173	**.177**	.176
R@50	.200	.185	.191	.200	.209	.216	**.218**
R@100	.360	.308	.340	.379	.379	.387	**.414**
R@150	.535	.391	.481	.551	.553	.562	**.600**
R@200	.655	.499	.612	.688	.702	.708	**.765**

Table 4: Evaluation results. DB, PM, MD, FT, SF and SG stand for the `DBDC3`, `PlainMultiDomain`, `MultiDomain`, `FineTune`, `StepFineTune` and `StepGAN` approach, respectively. The Sup denotes the supervised approach trained on the balanced labelled dialogues.

signed to study the efficacy of our unsupervised approaches. Then, we analyze the generated dialogues to test if `StepGAN` can generate reasonably problematic dialogues. Table 4 shows the overall results in Precision@k and Recall@k.

6.1 Comparative results

`StepGAN` vs. other unsupervised approaches: Figure 4 shows that `StepGAN` outperforms other unsupervised approaches by a large margin. The only exception is `MultiDomain` when $k = 10$. One noteworthy observation is made by comparing `MultiDomain` with `FineTune` – `MultiDomain` is more effective than `FineTune` when k is small, less than 30 in this case. This is because most turns are contextually wrong or unnatural when we look into the most problematic dialogues (e.g. $k < 10$) and `MultiDomain` generates exactly such dialogues. On the contrary, as k increases, generated dialogues gradually resemble normal ones with less wrong turns and `FineTune` essentially generates this type of dialogues.

This observation indicates that a high-quality model should be able to generate a spectrum of dialogues, ranging from obvious dialogues to subtle ones. That is why we introduced a stepwise training scheme and thus `StepFineTune` shows a significantly improved performance. Furthermore, `StepGAN` employs the GAN training procedure to generate more natural and diverse dialogues and almost always outperforms all other unsupervised approaches. The fact that `StepGAN` outperformes `DBDC3` and `PlainMultiDomain` highlightes that the `StepGAN`'s capability of generating domain-relevant problematic dialogues is crucial in obtaining high performance.

- ***`StepGAN` vs. supervised approach:*** In Figure 5, *StepGAN2000*, *StepGAN4000* and *StepGAN6000* denote ranking models trained on 2000, 4000, 6000 balanced datasets generated by `StepGAN` respectively. *Sup* stands for a ranker trained on 2000 balanced labelled dialogues. Interestingly, `StepGAN` performs even better than the supervised approach when $k < 50$. Even though the supervised approach yields higher performance when k is large, `StepGAN` still compares favorably and the gap is narrower if more dialogues are generated. Note that having developers review a large number of log dialogues (over 100) induces a significant cognitive load. Thus, the higher performance of `StepGAN` in the small k regime can offer more practical value.
- ***Semi-supervised learning attempts:*** In Figure 6, *Sup500* and *Sup2000* denote supervised dialogue rankers trained on randomly sampled 500 and 2000 balanced labelled dialogues, respectively. *StepGAN+Sup500* and *StepGAN+Sup2000* denote semi-supervised approaches trained on the 500 and 2000 labelled datasets plus 6000 simulated dialogues generated by `StepGAN`, respectively. The higher performance of the semi-supervised approaches compared to the supervised counterparts highlights that our unsupervised approach can bring additional generalization by simulating a wide range of dialogues that are not covered by labelled data. As expected, the performance gain increases as we move to a smaller data regime, e.g. 500 labelled dialogues.

6.2 Analysis on generated dialogues

To investigate how generated dialogues move toward normal dialogues, we examine dialogues generated at each step of `StepGAN` training in terms of three quantitative metrics: average dialogue length, task success rate and ranking score produced by the supervised ranker. Figure 7, 8 and 9 clearly show that as training progresses, the characteristics of generated dialogues are getting more similar to normal dialogues – average dialogue length of generated dialogues is decreasing while task success rate and average ranking score are increasing. In Table 5, we show example dialogues generated by `StepGAN` after two different training steps to qualitatively examine generated dialogues. The generated dialogues clearly show that `StepGAN` manages to follow a normal course up to the training-step turn and then start to get off track. For the full dialogue examples generated by different generative models, refer to Supplemental Material A.

7 Related Work

In contrast to conventional approaches (Jokinen and McTear, 2009), end-to-end neural models have been recently investigated for task-oriented dialogue systems which allows for directly learning dialogue systems from human-human dialogue data (Wen et al., 2016; Bordes et al., 2016; Williams et al., 2017). For such approaches, our proposed dialogue ranking method can help developers quickly go over log dialogues to improve their dialogue systems. There are prior studies on dialogue quality assessment. Walker et al. (1997); Engelbrecht et al. (2009) investigate automatic ways of dialogue evaluation. Walker et al. (2002); Herm et al. (2008); Kim (2007); Higashinaka et al. (2010, 2016) focuse on dialogue breakdown detection during the dialogue. Ultes et al. (2012); Su et al. (2016) estimate dialogue quality to improve dialogue policy. However, none has directly dealt with log dialogue ranking task and offered unsupervised approaches to reduce labelling cost. For automatic error handling, Clark (1996); Bohus and Rudnicky (2005); Higashinaka et al. (2015) examined different types of errors. Whereas, for our work, it suffices to rank problematic dialogues higher than normal ones so that developers may take appropriate measures based on the goal of their bots, e.g., adding new slots or system actions to handle previously uncovered scenarios.

8 Conclusions and Future Work

We proposed unsupervised approaches for the task of *log dialogue ranking* and provided comprehensive experimental results. `StepGAN` outperforms other unsupervised approaches and compares favorably with the supervised approach. Moreover, `StepGAN` improves upon the supervised approach in semi-supervised settings. Finally, we release the labelled dialogue corpus and seed dialogues. Future work includes 1) in-depth human evaluation on the quality of generated dialogues, 2) testing `StepGAN` on a broader range of tasks 3) performing dialogue clustering before ranking to reduce redundant workload for developers.

Dialogues simulated by the StepGAN approach in different training steps
StepGAN training at step 1
SYS: hello, welcome to the restaurant finding system. how may i help you?
USER: hi, i want to book a table
SYS: what part of town do you have in mind?
USER: boston,.
SYS: what part of town do you have in mind?
USER: part is going up in colorado.
SYS: what part of town do you have in mind?
USER: i do not want to book a movie price range and in the usa in the west of town.
SYS: i am sorry but there is no place, serving north american food and in the area of west.
USER: alright, that's understandable, can you tell me more?
SYS: i am sorry but there is no place, in the area of west and serving north american food.
USER: that's sound great. oh that's the limit. that is all i need as well though, can you help
SYS: thank you, goodbye.
StepGAN training at step 3
SYS: hello, welcome to the restaurant finding system. how may i help you?
USER: i need another restaurant.
SYS: what part of town do you have in mind?
USER: i want a venue in the moderate price range and in the north of town.
SYS: what kind of food would you like?
USER: i need a indian venue.
SYS: the nirala is a nice place, it is in the moderate price range, it serves indian food and it is in the north part of town.
USER: i am not looking for this. please change.
SYS: the nirala is a nice place, it is in the moderate price range, it serves indian food and it is in the north part of town.
USER: the account 555 - venue is great. i am in the moderate price range and i am going at town.
SYS: the nirala is a nice place, it serves indian food, it is in the moderate price range and it is in the north part of town.
USER: well, i want a venue in the business site. please do, i am going to get a brief globe.
SYS: the nirala is a nice place, it serves indian food, it is in the moderate price range and it is in the north part of town.
USER: that doesn't matter.
SYS: the nirala is a nice place, it is in the north part of town, it is in the moderate price range and it serves indian food.
USER: okay, thanks for the help me book
SYS: thank you, goodbye.
StepGAN training at step 6
SYS: hello, welcome to the restaurant finding system. how may i help you?
USER: i need another restaurant.
SYS: what part of town do you have in mind?
USER: it doesn't matter.
SYS: what part of town do you have in mind?
USER: it doesn't matter.
SYS: what part of town do you have in mind?
USER: i want a venue in the expensive price range and in the centre of town.
SYS: what kind of food would you like?
USER: i need a european venue.
SYS: michaelhouse cafe is a nice place, it is in the centre of town, it is in the expensive price range and it serves european food.
USER: what is the phone number of the restaurant?
SYS: michaelhouse cafe is a nice place their phone number is 0 1223 309147.
USER: thanks for your help, goodbye.
SYS: thank you, goodbye.
StepGAN training at step 9
SYS: hello, welcome to the restaurant finding system. how may i help you?
USER: i need a european restaurant.
SYS: what part of town do you have in mind?
USER: i prefer a venue in centre.
SYS: would you like something in the cheap, moderate, or expensive price range?
USER: i want a venue in the expensive price range.
SYS: michaelhouse cafe is a nice place, it is in the expensive price range, it is in the centre of town and it serves european food.
USER: what is the address of michaelhouse cafe?
SYS: michaelhouse cafe is a nice place their address is st. michael's church trinity street city centre.
USER: thanks for your help, goodbye.
SYS: thank you, goodbye.

Table 5: Dialogues simulated by the StepGAN approach in different training steps.

References

Dan Bohus and Alexander I Rudnicky. 2005. Sorry, i didn't catch that!-an investigation of non-understanding errors and recovery strategies. In *6th SIGdial workshop on discourse and dialogue*.

Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2016. Learning end-to-end goal-oriented dialog. *arXiv preprint arXiv:1605.07683*.

Herbert H Clark. 1996. *Using language*. Cambridge university press.

Klaus-Peter Engelbrecht, Florian Gödde, Felix Hartard, Hamed Ketabdar, and Sebastian Möller. 2009. Modeling user satisfaction with hidden markov models. In *Proceedings of the SIGDIAL 2009 Conference*, pages 170–177.

Ota Herm, Alexander Schmitt, and Jackson Liscombe. 2008. When calls go wrong: How to detect problematic calls based on log-files and emotions? In *Ninth Annual Conference of the International Speech Communication Association*.

Ryuichiro Higashinaka, Kotaro Funakoshi, Masahiro Araki, Hiroshi Tsukahara, Yuka Kobayashi, and Masahiro Mizukami. 2015. Towards taxonomy of errors in chat-oriented dialogue systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 87–95.

Ryuichiro Higashinaka, Kotaro Funakoshi, Michimasa Inaba, Yuiko Tsunomori, Tetsuro Takahashi, and Nobuhiro Kaji. 2017. Overview of dialogue breakdown detection challenge 3. *Proceedings of Dialog System Technology Challenge*, 6.

Ryuichiro Higashinaka, Kotaro Funakoshi, Yuka Kobayashi, and Michimasa Inaba. 2016. The dialogue breakdown detection challenge: Task description, datasets, and evaluation metrics. In *LREC*.

Ryuichiro Higashinaka, Yasuhiro Minami, Kohji Dohsaka, and Toyomi Meguro. 2010. Modeling user satisfaction transitions in dialogues from overall ratings. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 18–27. Association for Computational Linguistics.

Kristiina Jokinen and Michael McTear. 2009. Spoken dialogue systems. *Synthesis Lectures on Human Language Technologies*, 2(1):1–151.

Woosung Kim. 2007. Online call quality monitoring for automating agent-based call centers. In *Eighth Annual Conference of the International Speech Communication Association*.

Sungjin Lee, Hannes Schulz, Adam Atkinson, Jianfeng Gao, Kaheer Suleman, Layla El Asri, Mahmoud Adada, Minlie Huang, Shikhar Sharma, Wendy Tay, and Xiujun Li. 2019. Multi-domain task-completion dialog challenge.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119. Association for Computational Linguistics.

Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*.

Pei-Hao Su, Milica Gasic, Nikola Mrksic, Lina Rojas-Barahona, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. On-line active reward learning for policy optimisation in spoken dialogue systems. *arXiv preprint arXiv:1605.07669*.

Stefan Ultes, Alexander Schmitt, and Wolfgang Minker. 2012. Towards quality-adaptive spoken dialogue management. In *NAACL-HLT Workshop on Future Directions and Needs in the Spoken Dialog Community: Tools and Data*, pages 49–52. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Marilyn A Walker, Irene Langkilde-Geary, H Wright Hastie, Jerry Wright, and Allen Gorin. 2002. Automatically training a problematic dialogue predictor for a spoken dialogue system. *Journal of Artificial Intelligence Research*, 16:293–319.

Marilyn A Walker, Diane J Litman, Candace A Kamm, and Alicia Abella. 1997. Paradise: A framework for evaluating spoken dialogue agents. *arXiv preprint cmp-lg/9704004*.

Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2016. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*.

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

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Supplemental Material

A Dialogues simulated by unsupervised approaches

Dialogues simulated by the MultiDomain approach
SYS: hello, welcome to the restaurant finding system. how may i help you?
USER: i am looking for cheap french cuisine
SYS: i'm having trouble understanding what you want. could you please rephrase it?
USER: i want a champagne without a different tables
SYS: midsummer house restaurant is nice
USER: where is the restaurant?
SYS: rajmahal is nice
USER: how about champagne's?
SYS: the good luck chinese food takeaway is nice
USER: ohh okay. that's the problem! i really'm craving midsummer vegas.
SYS: what part of town do you have in mind?
USER: the galleria vegas adventure
SYS: galleria is a nice place it is in the centre of town.
USER: awesome, i'll go with that topic to serve, need else
SYS: galleria is a nice place it is in the moderate price range.
USER: , my account is very good i guess
SYS: galleria is a nice place it serves european food.
USER: i will thanks anyways
SYS: thank you, goodbye.
Dialogues simulated by the FineTune approach
SYS: hello, welcome to the restaurant finding system. how may i help you?
USER: hi, i want to book a table
SYS: what part of town do you have in mind?
USER: i prefer a venue in south.
SYS: what kind of food would you like?
USER: i need a chinese venue.
SYS: would you like something in the cheap, moderate, or expensive price range?
USER: it doesn't matter.
SYS: would you like something in the cheap, moderate, or expensive price range?
USER: i want a venue in the expensive price range.
SYS: the good luck chinese food takeaway is a nice place, it is in the south part of town, it is in the expensive price range and it serves chinese food.
USER: thanks for your help, goodbye.
SYS: thank you, goodbye.

Table 6: Dialogues simulated by the MultiDomain approach/FineTune approach.

Dialogues simulated by the StepFineTune approach in different training steps
StepFineTune training at step 1
SYS: hello, welcome to the restaurant finding system. how may i help you?
USER: i need another restaurant.
SYS: what part of town do you have in mind?
USER: i need another restaurant.
SYS: what part of town do you have in mind?
USER: washington dc.
SYS: what part of town do you have in mind?
USER: in the west of town.
SYS: what kind of food would you like?
USER: i want a table
SYS: what kind of food would you like?
USER: i need another restaurant.
SYS: what kind of food would you like?
USER: i want restaurant in centre.
SYS: what kind of food would you like?
USER: i want a restaurant in east.
SYS: what kind of food would you like?
USER: i want a restaurant in the moderate price range.
SYS: what kind of food would you like?
USER: i want a restaurant in the cheap price range and in the centre of town.
SYS: what kind of food would you like?
USER: i want a restaurant in the moderate price range.
SYS: what kind of food would you like?
StepFineTune training at step 3
SYS: hello, welcome to the restaurant finding system. how may i help you?
USER: i want a restaurant in the moderate price range.
SYS: what part of town do you have in mind?
USER: i prefer a venue in west.
SYS: what kind of food would you like?
USER: i need a venue in south.
SYS: what kind of food would you like?
USER: i need a international venue.
SYS: i am sorry but there is no place, serving international food, in the area of south and in the moderate price range.
USER: no, i am not looking for this. please show me another one.
SYS: i am sorry but there is no place, serving international food, in the moderate price range and in the area of south.
USER: no, i need a venue in south.
SYS: i am sorry but there is no place, serving international food, in the area of south and in the moderate price range.
USER: what about the venue in south?
SYS: i am sorry but there is no place, in the moderate price range, in the area of south and serving international food.
USER: thanks for the help, goodbye.
SYS: thank you, goodbye.
StepFineTune training at step 6
SYS: hello, welcome to the restaurant finding system. how may i help you?
USER: hi, i want to book a table
SYS: what part of town do you have in mind?
USER: i want a venue in any part of town.
SYS: what kind of food would you like?
USER: i need a asian oriental venue.
SYS: would you like something in the cheap, moderate, or expensive price range?
USER: i want a venue in the expensive price range.
SYS: saigon city is a nice place, it serves asian oriental food, it is in the north part of town and it is in the expensive price range.
USER: what is the phone number of the restaurant?
SYS: saigon city is a nice place their phone number is 0 1223 356555.
USER: thanks for your help, goodbye.
SYS: thank you, goodbye.
StepFineTune training at step 9
SYS: hello, welcome to the restaurant finding system. how may i help you?
USER: hi, i want to book a table
SYS: what part of town do you have in mind?
USER: i prefer a venue in south.
SYS: what kind of food would you like?
USER: i need a chinese venue.
SYS: would you like something in the cheap, moderate, or expensive price range?
USER: i need a restaurant in the expensive price range.
SYS: the good luck chinese food takeaway is a nice place, it is in the south part of town, it serves chinese food and it is in the expensive price range.
USER: thanks for your help, goodbye.
SYS: thank you, goodbye.

Table 7: Dialogues simulated by the StepFineTune approach in different training steps.