A Probabilistic End-To-End Task-Oriented Dialog Model with Latent Belief States towards Semi-Supervised Learning

Yichi Zhang[1], **Zhijian Ou**[1*], **Huixin Wang**[2], **Junlan Feng**[2]
[1] Speech Processing and Machine Intelligence Lab, Tsinghua University, Beijing, China
[2] China Mobile Research Institute, Beijing, China
zhangyic17@tsinghua.org.cn, ozj@tsinghua.edu.cn

Abstract

Structured belief states are crucial for user goal tracking and database query in task-oriented dialog systems. However, training belief trackers often requires expensive turn-level annotations of every user utterance. In this paper we aim at alleviating the reliance on belief state labels in building end-to-end dialog systems, by leveraging unlabeled dialog data towards semi-supervised learning. We propose a probabilistic dialog model, called the LAtent BElief State (LABES) model, where belief states are represented as discrete latent variables and jointly modeled with system responses given user inputs. Such latent variable modeling enables us to develop semi-supervised learning under the principled variational learning framework. Furthermore, we introduce LABES-S2S, which is a copy-augmented Seq2Seq model instantiation of LABES[1]. In supervised experiments, LABES-S2S obtains strong results on three benchmark datasets of different scales. In utilizing unlabeled dialog data, semi-supervised LABES-S2S significantly outperforms both supervised-only and semi-supervised baselines. Remarkably, we can reduce the annotation demands to 50% without performance loss on MultiWOZ.

1 Introduction

Belief tracking (also known as dialog state tracking) is an important component in task-oriented dialog systems. The system tracks user goals through multiple dialog turns, i.e. infers structured *belief states* expressed in terms of slots and values (e.g. in Figure 1), to query an external database (Henderson et al., 2014). Different belief tracking models have been proposed in recent years, either trained independently (Mrkšić et al., 2017; Ren et al., 2018; Wu et al., 2019) or within end-to-end (E2E) trainable dialog systems (Wen et al., 2017a,b; Liu and Lane, 2017; Lei et al., 2018; Shu et al., 2019; Liang et al., 2020; Zhang et al., 2020).

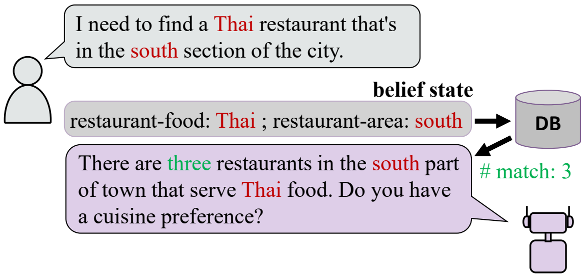

I need to find a Thai restaurant that's in the south section of the city.
belief state
restaurant-food: Thai ; restaurant-area: south
DB
# match: 3
There are three restaurants in the south part of town that serve Thai food. Do you have a cuisine preference?

Figure 1: The cues for inferring belief states from user inputs and system responses. The system response reveals the belief state either directly in the form of word repetition (red), or indirectly in the form of the database query result (green) determined by the belief state.

Existing belief trackers mainly depend on supervised learning with human annotations of belief states for every user utterance. However, collecting these turn-level annotations is labor-intensive and time-consuming, and often requires domain knowledge to identify slots correctly. Building E2E trainable dialog systems, called E2E dialog systems for short, even further magnifies the demand for increased amounts of labeled data (Gao et al., 2020; Zhang et al., 2020).

Notably, there are often easily-available unlabeled dialog data such as between customers and trained human agents accumulated in real-world customer services. In this paper, we are interested in reducing the reliance on belief state annotations in building E2E task-oriented dialog systems, by leveraging unlabeled dialog data towards semi-supervised learning. Intuitively, the dialog data, even unlabeled, can be used to enhance the performance of belief tracking and thus benefit the whole dialog system, because there are cues from

[*] Corresponding author.
[1] Code available at https://github.com/thu-spmi/LABES

user inputs and system responses which reveal the belief states, as shown in Figure 1.

Technically, we propose a latent variable model for task-oriented dialogs, called the **LA**tent **BE**lief **S**tate (LABES) dialog model. The model generally consists of multiple (e.g. T) turns of user inputs $u_{1:T}$ and system responses $r_{1:T}$ which are observations, and belief states $b_{1:T}$ which are latent variables. Basically, LABES is a conditional generative model of belief states and system responses given user inputs, i.e. $p_\theta(b_{1:T}, r_{1:T}|u_{1:T})$. Once built, the model can be used to infer belief states and generate responses. More importantly, such latent variable modeling enables us to develop semi-supervised learning on a mix of labeled and unlabeled data under the principled variational learning framework (Kingma and Welling, 2014; Sohn et al., 2015). In this manner, we hope that the LABES model can exploit the cues for belief tracking from user inputs and system responses. Furthermore, we develop LABES-S2S, which is a specific model instantiation of LABES, employing copy-augmented Seq2Seq (Gu et al., 2016) based conditional distributions in implementing $p_\theta(b_{1:T}, r_{1:T}|u_{1:T})$.

We show the advantage of our model compared to other E2E task-oriented dialog models, and demonstrate the effectiveness of our semi-supervised learning scheme on three benchmark task-oriented datasets: CamRest676 (Wen et al., 2017b), In-Car (Eric et al., 2017) and MultiWOZ (Budzianowski et al., 2018) across various scales and domains. In supervised experiments, LABES-S2S obtains state-of-the-art results on CamRest676 and In-Car, and outperforms all the existing models which do not leverage large pretrained language models on MultiWOZ. In utilizing unlabeled dialog data, semi-supervised LABES-S2S significantly outperforms both supervised-only and prior semi-supervised baselines. Remarkably, we can reduce the annotation requirements to 50% without performance loss on MultiWOZ, which is equivalent to saving around 30,000 annotations.

2 Related Work

On use of unlabeled data for belief tracking. Classic methods such as self-training (Rosenberg et al., 2005), also known as pseudo-labeling (Lee, 2013), has been applied to belief tracking (Tseng et al., 2019). Recently, the pretraining-and-fine-tuning approach has received increasing interests (Heck et al., 2020; Peng et al., 2020; Hosseini-Asl et al., 2020). The generative model based semi-supervised learning approach, which blends unsupervised and supervised learning, has also been studied (Wen et al., 2017a; Jin et al., 2018). Notably, the two approaches are orthogonal and could be jointly used. Our work belongs to the second approach, aiming to leverage unlabeled dialog data beyond of using general text corpus. A related work close to ours is SEDST (Jin et al., 2018), which also perform semi-supervised learning for belief tracking. Remarkably, our model is optimized under the principled variational learning framework, while SEDST is trained with an ad-hoc combination of posterior regularization and auto-encoding. Experimental in §6.2 show the superiority of our model over SEDST. See Appendix A for differences in model structures between SEDST and LABES-S2S.

End-to-end task-oriented dialog systems. Our model belongs to the family of E2E task-oriented dialog models (Wen et al., 2017a,b; Li et al., 2017; Lei et al., 2018; Mehri et al., 2019; Wu et al., 2019; Peng et al., 2020; Hosseini-Asl et al., 2020). We borrow some elements from the Sequicity (Lei et al., 2018) model, such as representing the belief state as a natural language sequence (a text span), and using copy-augmented Seq2Seq learning (Gu et al., 2016). But compared to Sequicity and all its follow-up works (Jin et al., 2018; Shu et al., 2019; Zhang et al., 2020; Liang et al., 2020), a feature in our LABES-S2S model is that the transition between belief states across turns and the dependency between system responses and belief states are well statistically modeled. This new design results in a completely different graphical model structure, which enables rigorous probabilistic variational learning. See Appendix A for details.

Latent variable models for dialog. Latent variables have been used in dialog models. For non task-oriented dialogs, latent variables are introduced to improve diversity (Serban et al., 2017; Zhao et al., 2017; Gao et al., 2019), control language styles (Gao et al., 2019) or incorporate knowledge (Kim et al., 2020) in dialog generation. For task-oriented dialogs, there are prior studies which use latent internal states via hidden Markov models (Zhai and Williams, 2014) or variational autoencoders (Shi et al., 2019) to discover the underlying dialog structures. In Wen et al. (2017a) and Zhao et al. (2019), dialog acts are treated as latent variables, together with variational learning and reinforcement learning, aiming to improve response generation. To the best of our knowledge, we are the first to model belief state as discrete latent variables, and propose to learn these structured representations via the variational principle.

3 Latent Belief State Dialog Models

We first introduce LABES as a general dialog modeling framework in this section. For dialog turn t, let u_t be the user utterance, b_t be the current belief state after observed u_t and r_t be the corresponding system response. In addition, denote c_t as the dialog context or model input at turn t, such as $c_t \triangleq \{r_{t-1}, u_t\}$ as in this work. Note that c_t can include longer dialog history depending on specific implementations. Let d_t be the database query result which can be obtained through a database-lookup operation given the belief state b_t.

Our goal is to model the joint distribution of belief states and system responses given the user inputs, $p_\theta(b_{1:T}, r_{1:T}|u_{1:T})$, where T is the total number of turns and θ denotes the model parameters. In LABES, we assume the joint distribution follows the directed probabilistic graphical model illustrated in Figure 2, which can be formulated as:

$$\begin{aligned} p_\theta(b_{1:T}, r_{1:T}|u_{1:T}) &= p_\theta(b_{1:T}|u_{1:T})p_\theta(r_{1:T}|b_{1:T}, u_{1:T}) \\ &= \prod_{t=1}^{T} p_\theta(b_t|b_{t-1},c_t)p_\theta(r_t|c_t,b_t,d_t) \end{aligned}$$

where b_0 is an empty state. Intuitively, we refer the conditional distribution $p_\theta(b_t|b_{t-1},c_t)$ as the belief state decoder, and $p_\theta(r_t|c_t,b_t,d_t)$ the response decoder in the above decomposition. Note that the probability $p(d_t|b_t)$ is omitted as database result d_t is deterministically obtained given b_t. Thus the system response can be generated as a three-step process: first predict the belief state b_t, then use b_t to query the database and obtain d_t, finally generate the system response r_t based on all the conditions.

Unsupervised Learning

We introduce an inference model $q_\phi(b_t|b_{t-1}, c_t, r_t)$ (described by dash arrows in Figure 2) to approximate the true posterior $p_\theta(b_t|b_{t-1}, c_t, r_t)$. Then we can derive the variational evidence lower bound (ELBO) for unsupervised learning as follows:

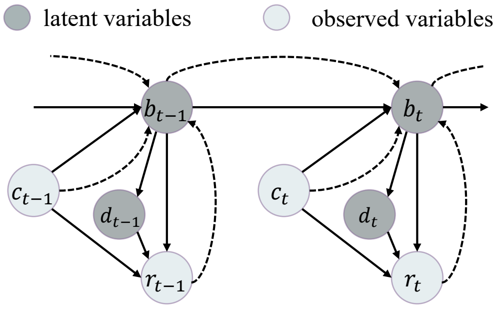


Figure 2: The probabilistic graphical model of LABES. Solid arrows describe the conditional generative model p_θ, and dash arrows describe the approximate posterior model q_ϕ. Note that we set $c_t \triangleq \{r_{t-1}, u_t\}$ in our model, and omit u_t from the graph for simplicity.

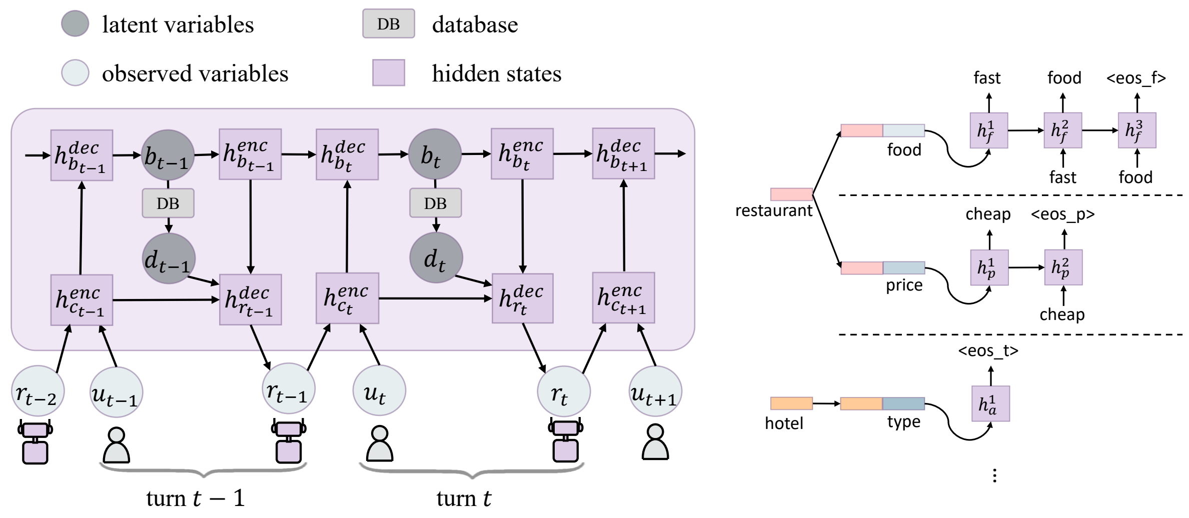
$$\begin{aligned} \mathcal{J}_{un} &= \mathbb{E}_{q_\phi(b_{1:T})}\left[\log \frac{p_\theta(b_{1:T}, r_{1:T}|u_{1:T})}{q_\phi(b_{1:T}|u_{1:T}, r_{1:T})}\right] \\ &= \sum_{t=1}^{T} \mathbb{E}_{q_\phi(b_{1:T})}\left[\log p_\theta(r_t|c_t, b_t, d_t)\right] \\ &\quad - \alpha \mathrm{KL}\left[q_\phi(b_t|b_{t-1}, c_t, r_t) \| p_\theta(b_t|b_{t-1}, c_t)\right] \end{aligned}$$

where

$$q_\phi(b_{1:T}) \triangleq \prod_{t=1}^{T} q_\phi(b_t|b_{t-1}, c_t, r_t)$$

and α is a hyperparameter to control the weight of the KL term introduced by Higgins et al. (2017).

Optimizing $\mathcal{J}_{un}$ requires drawing posterior belief state samples $b_{1:T} \sim q_\phi(b_{1:T}|u_{1:T}, r_{1:T})$ to estimate the expectations. Here we use a sequential sampling strategy similar to Kim et al. (2020), where each b_t sampled from $q_\phi(b_t|b_{t-1}, c_t, r_t)$ at turn t is used as the condition to generate the next turn's belief state b_{t+1}. For calculating gradients with discrete latent variables, which is non-trivial, some methods have been proposed such as using a score function estimator (Williams, 1992) or categorical reparameterization trick (Jang et al., 2017). In this paper, we employ the simple Straight-Through estimator (Bengio et al., 2013), where the sampled discrete token indexes are used for forward computation, and the continuous softmax probability of each token is used for backward gradient calculation. Although the Straight-Through estimator is biased, we find it works pretty well in our experiments, therefore leave the exploration of other optimization methods as future work.



(a) Overview of LABES-S2S.

(b) Structure of the belief state decoder.

Figure 3: (a) shows the computational graph of LABES-S2S. In (b), rectangles in different colors denote different word embeddings, and the embedding of domain names and slot names are concatenated as the initial input. Note that the same (i.e. weight-tied) decoder is shared across all slots. Decoding stops when a slot-specific end-of-sentence symbol is generated, which is possible to be the first output if the slot does not appear in the dialog.

Semi-Supervised Learning

When b_t labels are available, we can easily train the generative model p_θ and inference model q_ϕ via supervised maximum likelihoods:

$$\mathcal{J}_{sup} = \sum_{t=1}^{T} \Big[\log p_\theta(b_t | b_{t-1}, c_t) + \log p_\theta(r_t | c_t, b_t, d_t) + \log q_\phi(b_t | b_{t-1}, c_t, r_t) \Big]$$

When a mix of labeled and unlabeled data is available, we perform semi-supervised learning using a combination of the supervised objective $\mathcal{J}_{sup}$ and the unsupervised objective $\mathcal{J}_{un}$. Specifically, we first pretrain p_θ and q_ϕ on small-sized labeled data until convergence. Then we draw supervised and unsupervised minibatches from labeled and unlabeled data and perform stochastic gradient ascent over $\mathcal{J}_{sup}$ and $\mathcal{J}_{un}$, respectively. We use supervised pretraining first because training $q_\phi(b_t|b_{t-1}, c_t, r_t)$ to correctly generate slot values and special outputs such as “dontcare” and end-of-sentence tokens as much as possible is important to improve sample efficiency in subsequent semi-supervised learning.

4 LABES-S2S: A Copy-Augmented Seq2Seq Instantiation

In the above probabilistic dialog model LABES, the belief state decoder $p_\theta(b_t|b_{t-1},c_t)$ and the response decoder $p_\theta(r_t|c_t,b_t,d_t)$ can be flexibly implemented. In this section we introduce LABES-S2S as an instantiation of the general LABES model based on copy-augmented Seq2Seq conditional distributions (Gu et al., 2016), which is shown in Figure 3(a) and described in the following. The responses are generated through two Seq2Seq processes: 1) decode the belief state given dialog context and last turn’s belief state and 2) decode the system response given dialog context, the decoded belief state and database query result.

Belief State Decoder

The belief state decoder is implemented via a Seq2Seq process, as shown in Figure 3(b). Inspired by Shu et al. (2019), we use a single GRU decoder to decode the value for each informable slot separately, feeding the embedding of each slot name as the initial input. In multi-domain setting, the domain name embedding is concatenated with the slot name embedding to distinguish slots with identical names in different domains (Wu et al., 2019).

We use two bi-directional GRUs (Cho et al., 2014) to encode the dialog context c_t and previous belief state b_{t-1} into a sequence of hidden vectors $h^{enc}_{c_t}$ and $h^{enc}_{b_{t-1}}$ respectively, which are the inputs to the belief state decoder. As there are multiple slots, and their values can also consist of multiple tokens, we denote the i-th token of slot s by $b^{s,i}_t$. To decode each token $b^{s,i}_t$, we first compute an attention vector over the encoder vectors. Then the attention vector and the embedding of the last decoded token $e(b_t^{s,i-1})$ are concatenated and fed into the decoder GRU to get the decoder hidden state $h_{b_t^{s,i}}^{dec}$, denoted as $h_{s,i}^{dec}$ for simplicity:

$$\begin{aligned} a_t^{s,i} &= \mathrm{Attn}(h_{c_t}^{enc} \circ h_{b_{t-1}}^{enc}, h_{s,i}^{dec}) \\ h_{s,i}^{dec} &= \mathrm{GRU}(a_t^{s,i} \circ e(b_t^{s,i-1}), h_{s,i-1}^{dec}) \\ \hat{h}_{s,i}^{dec} &= \mathrm{dropout}\big(h_{s,i}^{dec} \circ e(b_t^{s,i-1})\big) \end{aligned}$$

where $\circ$ denotes vector concatenation. We use the last hidden state of the dialog context encoder as $h_{s,0}^{dec}$, and the slot name embedding as $e(b_t^{s,0})$. We reuse $e(b_t^{s,i-1})$ to form $\hat{h}_{s,i}^{dec}$ to give more emphasis on the slot name embedding and add a dropout layer to reduce overfitting. $\hat{h}_{s,i}^{dec}$ is then used to compute a generative score ψ_{gen} for each token w in the vocabulary $\mathcal{V}$, and a copy score ψ_{cp} for words appeared in c_t and b_{t-1}. Finally, these two scores are combined and normalized to form the final decoding probability following:

$$\begin{aligned} \psi_{gen}(b_t^{s,i} = w) &= v_w^{\mathsf{T}} \mathrm{W}_{\mathrm{gen}} \hat{h}_{s,i}^{dec}, \quad w \in \mathcal{V} \\ \psi_{cp}(b_t^{s,i} = x_j) &= h_{x_j}^{enc\mathsf{T}} \mathrm{W}_{\mathrm{cp}} \hat{h}_{s,i}^{dec}, \quad x_j \in c_t \cup b_{t-1} \\ p(b_t^{s,i} = w) &= \frac{1}{Z}\left(e^{\psi_{gen}(w)} + \sum_{j:x_j=w} e^{\psi_{cp}(x_j)}\right) \end{aligned}$$

where $\mathrm{W}_{\mathrm{gen}}$ and W_{cp} are trainable parameters, v_w is the one-hot representation of w, x_j is the j-th token in $c_t \cup b_{t-1}$ and Z is the normalization term.

With copy mechanism, it is easier for the model to extract words mentioned by the user and keep the unchanged values from previous belief state. Meanwhile, the decoder can also generate tokens not appeared in input sequences, e.g. the special token “dontcare” or end-of-sentence symbols. Since the decoding for each slot is independent with each other, all the slots can be decoded in parallel to speed up.

The posterior network $q_\phi(b_t|b_{t-1}, c_t, r_t)$ is constructed through a similar process, where the only difference is that the system response r_t is also encoded and used as an additional input to the decoder. Note that the posterior network is separately parameterized with ϕ.

Response Decoder

The response decoder is implemented via another Seq2Seq process. After obtaining the belief state b_t, we use it to query a database to find entities that meet user's need, e.g. Thai restaurants in the south area. The query result d_t is represented as a 5-dimension one-hot vector to indicate 0, 1, 2, 3 and >3 matched results respectively. We only need the number of matched entities instead of their specific information as the input to the response decoder, because we generate delexicalized responses with placeholders for specific slot values (as shown in Table 4) to improve data efficiency (Wen et al., 2015). The values can be filled through simple rule-based post-processing afterwards.

Instead of directly decoding the response from the belief state decoder's hidden states (Lei et al., 2018), we again use the bi-directional GRU (the one used to encode b_{t-1}) to encode the current belief state b_t into hidden vectors $h_{b_t}^{enc}$. Then for each token r_t^i in the response, the decoder state $h_{r_{t,i}}^{dec}$ can be computed as follows:

$$\begin{aligned} a_t^{i} &= \mathrm{Attn}(h_{c_t}^{enc} \circ h_{b_t}^{enc}, h_{r_{t,i}}^{dec}) \\ h_{r_{t,i}}^{dec} &= \mathrm{GRU}(a_t^{i} \circ e(r_t^{i-1}) \circ d_t, h_{r_{t,i-1}}^{dec}) \\ \hat{h}_{r_{t,i}}^{dec} &= h_{r_{t,i}}^{dec} \circ a_t^{i} \circ d_t \end{aligned}$$

Note that dropout is not used for $\hat{h}_{r_{t,i}}^{dec}$, since response generation is not likely to overfit, compared to belief tracking in practice. We omit the probability formulas because they are almost the same as in the belief state decoder, except for changing the copy source from $c_t \cup b_{t-1}$ to $c_t \cup b_t$.

5 Experimental Settings

5.1 Datasets

We evaluate the proposed model on three benchmark task-oriented dialog datasets: the Cambridge Restaurant (CamRest676) (Wen et al., 2017b), Stanford In-Car Assistant (In-Car) (Eric et al., 2017) and MultiWOZ (Budzianowski et al., 2018), with 676/3031/10438 dialogs respectively. In particular, MultiWOZ is one of the most challenging dataset up-to-date given its multi-domain setting, complex ontology and diverse language styles. As there are some belief state annotation errors in MultiWOZ, we use the corrected version MultiWOZ 2.1 (Eric et al., 2019) in our experiments. See Appendix B for more detailed introductions and statistics.

5.2 Evaluation Metrics

We evaluate the model performance under the end-to-end setting, i.e. the model needs to first predict belief states and then generate response based on its own belief predictions. For evaluating belief tracking performance, we use the commonly used `joint goal accuracy`, which is the proportion of dialog turns where all slot values are correctly predicted. For evaluating response generation, we use `BLEU` (Papineni et al., 2002) to measure the general language quality. The response quality towards task completion is measured by dataset-specific metrics to facilitate comparison with prior works. For CamRest676 and In-Car, we use `Match` and `SuccF1` following Lei et al. (2018). For MultiWOZ, we use `Inform` and `Success` as in Budzianowski et al. (2018), and also a combined score computed through (`Inform`+`Success`)×0.5+`BLEU` as the overall response quality suggested by Mehri et al. (2019).

5.3 Baselines

In our experiments, we compare our model to various Dialog State Tracking (DST) and End-to-End (E2E) baseline models. Recently, large-scale pretrained language models (LM) such as BERT (Devlin et al., 2019) and GPT-2 (Radford et al., 2019) are used to improve the performance of dialog models, however in the cost of tens-fold larger model sizes and computations. We distinguish them from light-weighted models trained from scratch in our comparison.

Independent DST Models: For CamRest676, we compare to StateNet (Ren et al., 2018) and TripPy (Heck et al., 2020), which are the SOTA model without/with BERT respectively. For MultiWOZ, we compare to BERT-free models TRADE (Wu et al., 2019), NADST (Le et al., 2020b) and CSFN-DST (Zhu et al., 2020), and BERT-based models including TripPy, the BERT version of CSFN and DST-Picklist (Zhang et al., 2019).

E2E Models: E2E models can be divided into three sub-categories. The TSCP (Lei et al., 2018), SEDST (Jin et al., 2018), FSDM (Shu et al., 2019), MOSS (Liang et al., 2020) and DAMD (Zhang et al., 2020) are based on the copy-augmented Seq2Seq learning framework proposed by Lei et al. (2018). LIDM (Wen et al., 2017a), SFN (Mehri et al., 2019) and UniConv (Le et al., 2020a) are modular designed, connected through neural states and trained end-to-end. SimpleTOD (Hosseini-Asl et al., 2020) and SOLOLIST (Peng et al., 2020) are two recent models, which both use a single auto-regressive language model, initialized from GPT-2, to build the entire system.

Semi-Supervised Methods: First, we compare with SEDST (Jin et al., 2018) for semi-supervised belief tracking performance. SEDST is also a E2E dialog model based on copy-augmented Seq2Seq learning (see Appendix A for more details). Over unlabled dialog data, SEDST is trained through posterior regularization (PR), where a posterior network is used to model the posterior belief distribution given system responses, and then guide the learning of prior belief tracker through minimizing the KL divergence between them. Second, based on the LABES-S2S model, we compare our variational learning (VL) method to a classic semi-supervised learning baseline, self-training (ST), which performs as its name suggests. Specifically, after supervised pretraining over small-sized labeled dialogs, we run the system to generate pseudo belief states b_t over unlabeled dialogs, and then train the response decoder $p_\theta(r_t|b_t, c_t, d_t)$ in a supervised manner. The gradients will propagate through the discrete belief states by the Straight Through gradient estimator (Bengio et al., 2013) over the computational graph, thus also adjusting the belief state decoder $p_\theta(b_t|b_{t-1}, c_t)$.

6 Results and Analysis

In our experiments, we report both the best result and the statistical result obtained from multiple independent runs with different random seeds. Details are described in the caption of each table. The implementation details of our model is available in Appendix C. Results are organized to show the advantage of our proposed LABES-S2S model over existing models (§6.1) and the effectiveness of our semi-supervised learning method (§6.2).

6.1 Benchmark Performance

We first train our LABES-S2S model under full supervision and compare with other baseline models on the benchmarks. The results are given in Table 1 and Table 2.

As shown in Table 1, LABES-S2S obtains new SOTA joint goal accuracy on CamRest676 and the highest match scores on both CamRest676 and In-Car datasets. Its BLEU scores are also beyond or close to the previous SOTA models. The relatively low SuccF1 is due to that in LABES-S2S, we do not apply additional dialog act modeling and reinforcement fine-tuning to encourage slot token generation as in other E2E models.

Table 2 shows the MultiWOZ results. Among

Type	Model	CamRest676				In-Car		
		Joint Goal	Match	SuccF1	BLEU	Match	SuccF1	BLEU
DST	StateNet (Ren et al., 2018)	88.9	-	-	-	-	-	-
	TripPy (Heck et al., 2020)	92.7±0.2	-	-	-	-	-	-
E2E	LIDM (Wen et al., 2017a)	84.2*	91.2	84.0	24.6	72.1	76.2	17.3
	TSCP (Lei et al., 2018)	87.4*	92.7	85.4	25.3	84.5	81.1	21.9
	SEDST (Jin et al., 2018)	88.1*	92.7	75.4	23.6	84.5	**82.9**	19.3
	FSDM (Shu et al., 2019)	-	93.5	**86.2**	25.8	84.8	82.1	21.5
	MOSS (Liang et al., 2020)	88.4*	95.1	86.0	**25.9**	-	-	-
	LABES-S2S (best)	**93.5**	**96.4**	82.3	25.6	**86.6**	78.0	**23.2**
	LABES-S2S (statistical)	91.7±1.5	96.4±0.5	83.0±1.0	25.5±0.4	85.8±1.7	77.0±1.7	22.8±1.1

Table 1: Results on CamRest676 and In-Car. The model with the highest joint goal accuracy on the development set of CamRest676 is shown as the best result, as similarly reported in prior work. Statistical results are reported as the mean and standard deviation of 5 runs. * denotes results obtained by our run of the open-source code.

Model Configure				Belief Tracking	Response Generation			
Type	Model	Size	Pretrained LM	Joint Goal	Inform	Success	BLEU	Combined
DST	TRADE (Wu et al., 2019)	10.2M	no	45.60	-	-	-	-
	NADST (Le et al., 2020b)	12.9M	no	49.04	-	-	-	-
	CSFN-DST (Zhu et al., 2020)	63M	no	50.81	-	-	-	-
E2E	TSCP (Lei et al., 2018)	1.4M	no	37.53	66.41	45.32	15.54	71.41
	SFN + RL (Mehri et al., 2019)	1.4M	no	21.17*	73.80	58.60	16.88	83.04
	DAMD (Zhang et al., 2020)	2.0M	no	35.40*	76.40	60.40	16.60	85.00
	UniConv (Le et al., 2020a)	16M	no	50.14	72.60	62.90	**19.80**	87.55
	LABES-S2S (best)	3.8M	no	**51.45**	**78.07**	**67.06**	18.13	**90.69**
	LABES-S2S (statistical)	3.8M	no	50.05	76.89	63.30	17.92	88.01
DST	CSFN-DST + BERT (Zhu et al., 2020)	115M	BERT	52.88	-	-	-	-
	DST-Picklist (Zhang et al., 2019)	220M	BERT	53.30	-	-	-	-
	TripPy (Heck et al., 2020)	110M	BERT	55.29	-	-	-	-
E2E	SimpleTOD (Hosseini-Asl et al., 2020)	81M	DistilGPT-2	56.45	85.00	70.05	15.23	92.98
	SOLOLIST (Peng et al., 2020)	117M	GPT-2	-	85.50	72.90	16.54	95.74

Table 2: Results on MultiWOZ 2.1. The model with the highest validation joint goal accuracy is shown as the best result, as similarly reported in prior work. The standard deviations for the statistical results are in Table 5 in the appendix. * denotes results obtained by our run of the open-source code.

all the models without using large pretrained LMs, LABES-S2S performs the best in belief tracking joint goal accuracy and 3 out of the 4 response generation metrics. Although the response generation performance is not as good as recent GPT-2 based SimpleTOD and SOLOLIST, our model is much smaller and thus computational cheaper.

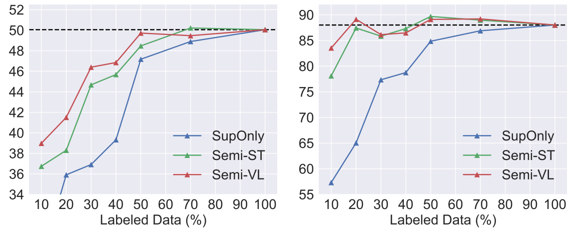
6.2 Semi-Supervised Experiments

In our semi-supervised experiments, we first split the data according to a fixed proportion, then train the models using only labeled data (SupOnly), or using both labeled and unlabeled data (Semi) with the proposed variational learning method (Semi-VL), self-training (Semi-ST) and posterior regularization (Semi-PR) introduced in §5.3 respectively. We conduct experiments with 50% and 25% labeled data on CamRest676 and In-Car following Jin et al. (2018), and change the labeled data proportion from 10% to 100% on MultiWOZ. The results are shown in Table 3 and Figure 4.

In Table 3, we can see that semi-supervised learning methods outperform the supervised learning baseline consistently in all experiments for the two datasets. In particular, the improvement of Semi-VL over SupOnly on our model is significantly larger than Semi-PR over SupOnly on SEDST in most metrics, and Semi-VL obtains a joint goal accuracy of 1.3%∼3.9% higher over Semi-ST. These results indicate the superiority of our LABES modeling framework in utilizing unlabeled data over other semi-supervised baselines. Since LABES mainly improves modeling of belief states, it is more relevant to examine the belief tracking metrics such as joint goal accuracy and match rate (partly determined by the belief tracking accuracy).

Labeled Data	Model & Method	CamRest676				In-Car			
		Joint Goal	Match	SuccF1	BLEU	Joint Goal	Match	SuccF1	BLEU
50%	LABES-S2S + SupOnly	83.3	91.8	80.5	23.8	77.9	81.0	74.5	20.4
	LABES-S2S + Semi-ST	86.3	93.1	83.1	25.3	79.8	83.4	74.8	22.1
	LABES-S2S + Semi-VL	89.7	94.4	83.1	25.3	81.1	84.1	77.5	22.6
	SEDST + SupOnly	78.5	89.1	65.0	18.6	74.4	74.1	69.2	16.9
	SEDST + Semi-PR	79.5	91.1	71.2	21.4	77.2	77.8	75.0	19.4
25%	LABES-S2S + SupOnly	68.8	85.9	75.3	21.7	74.3	73.7	62.8	15.8
	LABES-S2S + Semi-ST	74.1	91.1	82.5	25.4	74.9	74.4	76.9	22.5
	LABES-S2S + Semi-VL	77.5	93.6	81.4	25.5	78.8	79.3	76.6	22.4
	SEDST + SupOnly	64.2	80.3	66.8	16.9	57.8	51.0	50.4	14.1
	SEDST + Semi-PR	65.1	83.0	71.7	22.1	63.6	59.9	70.4	19.3

Table 3: *SupOnly* denotes training with only labeled data, and *Semi* denotes training with both labeled and unlabeled data in each dataset. ST, VL and PR denote self-training, variational learning and posterior regularization (Jin et al., 2018) respectively. Results of SEDST are obtained by our run of the open-source code. All the scores in this table are the mean from 5 runs.



(a) Joint Goal Accuracy (b) Combined Score

Figure 4: Performance of different methods w.r.t labeling proportion on MultiWOZ 2.1. The dash line corresponds to the baseline trained with 100% labeled data.

Note that Semi-VL and Semi-ST are fed with the same set of system responses, thus they obtain similar SuccF1 and BLEU scores in Table 3, which mainly measure the response quality.

The results on MultiWOZ shown in Figure 4 also support the above conclusions. From the plot of metric scores w.r.t labeling proportions, we can see how many labels can be reduced clearly. Our LABES-S2S model trained with Semi-VL obtains a joint goal accuracy of 49.47% and a combined score of 89.21 on only 50% of labeled data, which is very close to 50.05% and 88.01 obtained under 100% supervision. This indicates that we can reduce 50% of labels without losing performance, which results in reducing around 30,000 belief state annotations given the size of MultiWOZ. Moreover, it can be seen from Figure 4 that our Semi-VI can improve the belief tracking and response generation performance when labeling only 10% of dialogues, and the smaller amount of labels, the larger gain obtained by Semi-VI.

Dialog #586 in CamRest676	
u_1: I am looking for an expensive restaurant that serves Russian food. b_1: {food: Russian, pricerange: expensive} r_1: There is no expensive restaurant that serves Russian food. Can I help you with anything else? u_2: Yes, do you have British type food ? b_2: {food: British, pricerange: expensive} r_2: Yes, there are 6 options. Does the part of town matter? u_3: I don't care.	
Sup Only	b_3: {food: British, pricerange: expensive} r_3: [v.name] is located at [v.addr]. How else I can help you with?
Semi -VL	b_3: {food: British, pricerange: expensive, area: dontcare} r_3: [v.name] is a nice British restaurant in the [v.area] of town. Would you like their address?
Dialog #PMUL1329 in MultiWOZ	
u_1: I need a train going to Bishops Stortford. b_1: {destination: Bishops Stortford} r_1: OK. Are you departing from Cambridge? u_2: Yes, that's why I'm requesting help from Cambridge towninfo.	
Sup Only	b_3: {destination: Bishops Stortford} r_3: Where will you be departing from?
Semi -VL	b_3: {destination: Bishops Stortford, departure: Cambridge} r_3: I have [v.choice] trains that meet your criteria. What day would you like to travel?

Table 4: Comparison of two example turns generated by our model with supervised learning only (SupOnly) and semi-supervised variational learning (Semi-VL).

6.3 Case Study

We give two examples where the model trained with Semi-VL improves over the supervised-training-only baseline. In both examples, the user indicates his/her goal implicitly with a short reply. These rarely occurred corner cases are missed by the baseline model, but successfully captured after semi-supervised learning. Moreover, we can see that Semi-VL helps our model learn the cue word "British" which contributes to a more informative response in the first dialog, and in the second dialog, avoid the incoherent error caused by error propagation, thus improve the response generation quality.

7 Conclusion and Future Work

In this paper we are interested in reducing belief state annotation cost for building E2E task-oriented dialog systems. We propose a conditional generative model of dialogs - LABES, where belief states are modeled as latent variables, and unlabeled dialog data can be effectively leveraged to improve belief tracking through semi-supervised variational learning. Furthermore, we develop LABES-S2S, which is a copy-augmented Seq2Seq model instantiation of LABES. We show the strong benchmark performance of LABES-S2S and the effectiveness of our semi-supervised learning method on three benchmark datasets. In our experiments on MultiWOZ, we can save around 50%, i.e. around 30,000 belief state annotations without performance loss.

There are some interesting directions for future work. First, the LABES model is general and can be enhanced by, e.g. incorporating large-scale pre-trained language models, allowing other options for the belief state decoder and the response decoder such as Transformer based. Second, we can analogously introduce dialog acts $a_{1:T}$ as latent variables to define the joint distribution $p_\theta(b_{1:T}, a_{1:T}, r_{1:T}|u_{1:T})$, which can be trained with semi-supervised learning and reinforcement learning as well.

Acknowledgments

This work is supported by NSFC 61976122, Ministry of Education and China Mobile joint funding MCM20170301.

References

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyag Gao, and Dilek Hakkani-Tur. 2019. Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. *arXiv preprint arXiv:1907.01669*.

Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D. Manning. 2017. Key-value retrieval networks for task-oriented dialogue. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 37–49.

Jun Gao, Wei Bi, Xiaojiang Liu, Junhui Li, Guodong Zhou, and Shuming Shi. 2019. A discrete cvae for response generation on short-text conversation. In *2019 Conference on Empirical Methods in Natural Language Processing*, pages 1898–1908.

Silin Gao, Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020. Paraphrase augmented task-oriented dialog generation. *arXiv preprint arXiv:2004.07462*.

Xiang Gao, Yizhe Zhang, Sungjin Lee, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2019. Structuring latent spaces for stylized response generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1814–1823.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gašić. 2020. Trippy: A triple copy strategy for value independent neural dialog state tracking. *arXiv preprint arXiv:2005.02877*.

Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014. The second dialog state tracking challenge. In *Proceedings of the 15th annual meeting of the special interest group on discourse and dialogue (SIGDIAL)*, pages 263–272.

Irina Higgins, Loïc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-vae: Learning basic visual concepts with a constrained variational framework. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *arXiv preprint arXiv:2005.00796*.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Xisen Jin, Wenqiang Lei, Zhaochun Ren, Hongshen Chen, Shangsong Liang, Yihong Zhao, and Dawei Yin. 2018. Explicit state tracking with semi-supervisionfor neural dialogue generation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1403–1412.

Byeongchang Kim, Jaewoo Ahn, and Gunhee Kim. 2020. Sequential latent knowledge selection for knowledge-grounded dialogue. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Hung Le, Doyen Sahoo, Chenghao Liu, Nancy F Chen, and Steven CH Hoi. 2020a. Uniconv: A unified conversational neural architecture for multi-domain task-oriented dialogues. *arXiv preprint arXiv:2004.14307*.

Hung Le, Richard Socher, and Steven C. H. Hoi. 2020b. Non-autoregressive dialog state tracking. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Dong-Hyun Lee. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 2.

Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *ACL 2018: 56th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1437–1447.

Xiujun Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Celikyilmaz. 2017. End-to-end task-completion neural dialogue systems. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 733–743.

Weixin Liang, Youzhi Tian, Chengcai Cheng, and Zhou Yu. 2020. Moss: End-to-end dialog system framework with modular supervision. In *AAAI 2020 : The Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Bing Liu and Ian Lane. 2017. An end-to-end trainable neural network model with belief tracking for task-oriented dialog. *Proc. Interspeech 2017*, pages 2506–2510.

Shikib Mehri, Tejas Srinivasan, and Maxine Eskenazi. 2019. Structured fusion networks for dialog. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2017. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1777–1788.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2020. Soloist: Few-shot task-oriented dialog with a single pretrained auto-regressive model. *arXiv preprint arXiv:2005.05298*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Liliang Ren, Kaige Xie, Lu Chen, and Kai Yu. 2018. Towards universal dialogue state tracking. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2780–2786.

Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. 2005. Semi-supervised self-training of object detection models. *WACV/MOTION*, 2.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Weiyan Shi, Tiancheng Zhao, and Zhou Yu. 2019. Unsupervised dialog structure learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1797–1807.

Lei Shu, Piero Molino, Mahdi Namazifar, Hu Xu, Bing Liu, Huaixiu Zheng, and Gökhan Tür. 2019. Flexibly-structured model for task-oriented dialogues. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, pages 3483–3491.

Bo-Hsiang Tseng, Marek Rei, Paweł Budzianowski, Richard Turner, Bill Byrne, and Anna Korhonen. 2019. Semi-supervised bootstrapping of dialogue state trackers for task-oriented modelling. In *2019 Conference on Empirical Methods in Natural Language Processing*, pages 1273–1278.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721.

Tsung-Hsien Wen, Yishu Miao, Phil Blunsom, and Steve J. Young. 2017a. Latent intention dialogue models. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3732–3741. PMLR.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gasic, Lina M Rojas Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017b. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 438–449.

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

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *ACL 2019 : The 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819.

Qingyang Wu, Yichi Zhang, Yu Li, and Zhou Yu. 2019. Alternating recurrent dialog model with large-scale pre-trained language models. *arXiv preprint arXiv:1910.03756*.

Ke Zhai and Jason D Williams. 2014. Discovering latent structure in task-oriented dialogues. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36–46.

Jian-Guo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Yao Wan, Philip S Yu, Richard Socher, and Caiming Xiong. 2019. Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. *arXiv preprint arXiv:1910.03544*.

Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020. Task-oriented dialog systems that consider multiple appropriate responses under the same context. In *AAAI 2020 : The Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Tiancheng Zhao, Kaige Xie, and Maxine Eskenazi. 2019. Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. In *NAACL-HLT 2019: Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1208–1218.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. 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 (Volume 1: Long Papers)*, pages 654–664.

Su Zhu, Jieyu Li, Lu Chen, and Kai Yu. 2020. Efficient context and schema fusion networks for multi-domain dialogue state tracking. *arXiv preprint arXiv:2004.03386*.

A Model Comparisons with Prior Work

In this section, we comment on the differences between our LABES-S2S model and Sequicity (Lei et al., 2018) in both models and learning methods. Note that SEDST (Jin et al., 2018) employs the same model structure as Sequicity. First, Figure 5 shows the difference in computational graphs between Sequicity/SEDST and LABES-S2S. For Sequicity/SEDST, b_t and r_t are decoded directly from the belief state decoder's hidden states $h^{dec}_{b_t}$, thus the conditional probability of r_t given b_t and the state transition probability between b_{t-1} and b_t are not considered[2]. In contrast, LABES-S2S

[2] Strictly speaking, the transition between belief states across turns and the dependency between system responses

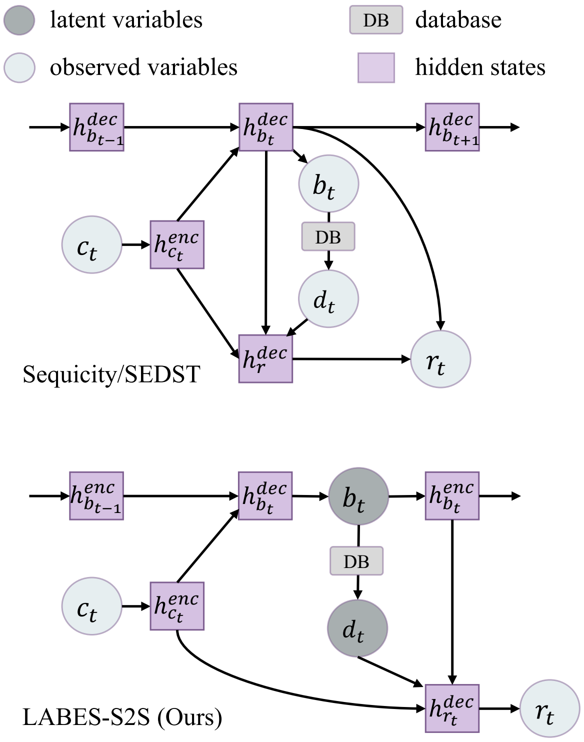


Figure 5: Comparison of computational graphs.

model introduces an additional b_t encoder and uses the encoder hidden states $h^{enc}_{b_t}$ to generate system response and next turn's belief state, thus the conditional probability $p_\theta(r_t|b_t, c_t)$ and state transition probability $p_\theta(b_t|b_{t-1}, c_t)$ are well defined by two complete Seq2Seq processes.

Second, the difference in models can also be clearly seen from the probabilistic graphical model structures as shown in Figure 6. LABES-S2S is a conditional generative model where the belief states are latent variables. In contrast, Sequicity/SEDST do not treat the belief states as latent variables.

Third, the above differences in models lead to differences in learning methods for Sequicity/SEDST and LABES-S2S. Sequicity can only be trained on labeled data via multi-task supervised learning. SEDST resorts to an ad-hoc combination of posterior regularization and auto-encoding for semi-supervised learning. Remarkably, LABES-S2S is optimized under the principled variational learning framework.

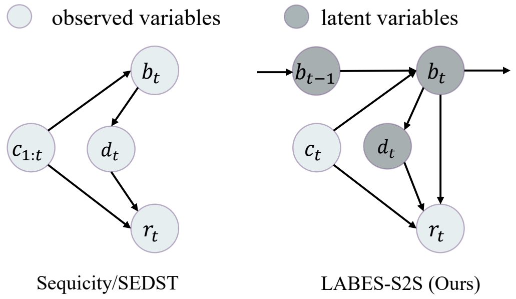


Figure 6: Comparison of probabilistic graphical model structures.

B Datasets

In our experiments, we evaluate different models on three benchmark task-oriented datasets with different scales and ontology complexities (Table 6). The Cambridge Restaurant (CamRest676) dataset (Wen et al., 2017b) contains single-domain dialogs where the system assists users to find a restaurant. The Stanford In-Car Assistant (In-Car) dataset (Eric et al., 2017) consists of dialogs between a user and a in-car assistant system covering three tasks: calendar scheduling, weather information retrieval and point-of-interest navigation. The MultiWOZ (Budzianowski et al., 2018) dataset is a large-scale human-human multi-domain dataset containing dialogs in seven domains including attraction, hotel, hospital, police, restaurant, train, and taxi. It is more challenging due to its multi-domain setting, complex ontology and diverse language styles. As there are some belief state annotation errors in MultiWOZ, we use the corrected version MultiWOZ 2.1 (Eric et al., 2019) in our experiments. We follow the data preprocessing setting in Zhang et al. (2020), whose data cleaning is developed based on Wu et al. (2019).

C Implementation Details

In our implementation of LABES-S2S, we use 1-layer bi-directinonal GRU as encoders and standard GRU as decoders. The hidden sizes are 100/100/200, vocabulary sizes are 800/1400/3000, and learning rates of Adam optimizer are $3e^{-3}/3e^{-3}/5e^{-5}$ for CamRest676/In-Car/MultiWOZ respectively. In all experiments, the embedding size is 50 and we use GloVe (Pennington et al., 2014) to initialize the embedding matrix. Dropout rate is 0.35 and λ for variational infer-ece is 0.5, which are selected via grid search from $\{0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5\}$ and $\{0.1, 0.3, 0.5, 0.7, 1.0, 1.5\}$ respectively. The

and belief states are modeled very weakly in Sequicity/SEDST, only owing to the copy mechanism. For simpliciy, we omit such relations in both Figure 5 and 6.

Model	Belief Tracking	Response Generation			
	Joint Goal	Inform	Success	BLEU	Combined
LABES-S2S (statistical)	50.05±0.92	76.89±1.51	63.30±2.35	17.92±0.35	88.01±2.10

Table 5: Statistical results of our LABES-S2S model with standard deviations on MultiWOZ 2.1.

	CamRest676	In-Car	MultiWOZ
#Dialog	676	3031	10438
Avg. #Turn	4.1	5.2	6.9
#Domain	1	3	7
#Info. Slot	3	11	31
#Req. Slot	7	11	38
#Values	99	284	4510

Table 6: Statistics of dialog datasets. Info and Req are shorthands for informable and requestable respectively.

learning rate decays by half every 2 epochs if no improvement is observed on development set. Training early stops when no improvement is observed on development set for 4 epochs. We use 10-width beam search for CamRest676 and greedy decoding for other datasets. All the models are trained on a NVIDIA Tesla-P100 GPU.