Improving Limited Labeled Dialogue State Tracking with Self-Supervision

Chien-Sheng Wu, Steven Hoi, and Caiming Xiong
Salesforce Research
[wu.jason, shoi, cxiong]@salesforce.com

Abstract

Existing dialogue state tracking (DST) models require plenty of labeled data. However, collecting high-quality labels is costly, especially when the number of domains increases. In this paper, we address a practical DST problem that is rarely discussed, i.e., learning efficiently with limited labeled data. We present and investigate two self-supervised objectives: preserving latent consistency and modeling conversational behavior. We encourage a DST model to have consistent latent distributions given a perturbed input, making it more robust to an unseen scenario. We also add an auxiliary utterance generation task, modeling a potential correlation between conversational behavior and dialogue states. The experimental results show that our proposed self-supervised signals can improve joint goal accuracy by 8.95% when only 1% labeled data is used on the MultiWOZ dataset. We can achieve an additional 1.76% improvement if some unlabeled data is jointly trained as semi-supervised learning. We analyze and visualize how our proposed self-supervised signals help the DST task and hope to stimulate future data-efficient DST research.

1 Introduction

Dialogue state tracking is an essential component in task-oriented dialogue systems designed to extract user goals/intentions expressed during a conversation. Accurate DST performance can facilitate downstream applications such as dialogue management. However, collecting dialogue state labels is very expensive and time-consuming (Budzianowski et al., 2018), requiring dialogue experts or trained turkers to indicate all *(domain, slot, value)* information for each turn in dialogues. This problem becomes important from single-domain to multi-domain scenarios. It will be more severe for a massive-multi-domain setting, making DST models less scalable to a new domain.

Existing DST models require plenty of state labels, especially those ontology-based DST approaches (Henderson et al., 2014; Mrkšić et al., 2017; Zhong et al., 2018). They assume a predefined ontology that lists all possible values is available, but an ontology requires complete state annotation and is hard to get in real scenario (Xu and Hu, 2018). They also cannot track unseen slot values that are not predefined. Ontology-free approaches (Xu and Hu, 2018; Chao and Lane, 2019), on the other hand, are proposed to generate slot values from dialogue history directly. They achieve good performance on multi-domain DST by copy-attention mechanism but still observe a significant performance drop under limited labeled data scenario (Wu et al., 2019a).

In this paper, we approach the DST problem using copy-augmented ontology-free models from a rarely discussed perspective, assuming that only a few dialogues in a dataset have annotated state labels. We present two self-supervised learning (SSL) solutions: 1) Preserving latent consistency: We encourage a DST model to have similar latent distributions (e.g., attention weights and hidden states) for a set of slightly perturbed inputs. This assumption is known as consistency assumption (Zhou et al., 2004; Chapelle et al., 2009; Berthelot et al., 2019) in semi-supervised learning, making distributions sufficiently smooth for the intrinsic structure collectively. 2) Modeling conversational behavior: We train a DST model to generate user utterances and system responses, hoping that this auxiliary generation task can capture intrinsic dialogue structure information and benefit the DST performance. This training only needs dialogue transcripts and does not require any further annotation. We hypothesize that modeling this potential correlation between utterances and states is helpful for generalization, making a DST model more

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 4462–4472
November 16 - 20, 2020.

Usr	Can you help me find a **nightclub** in **south** Cambridge?
Sys	**The Night** is located at 22 Sidney St. Their phone number is 01223324600. You will need to call for their entry fee.
Usr	Can you schedule me a taxi to take me there?
Sys	Can book you a taxi. Can you tell me the arrival or departure time ?
Usr	Also, I need a hotel with **parking** and **2 stars**.
Annotated State	(attraction, type, nightclub), (attraction, area, south), (attraction, name, The Night), (hotel, parking, yes), (hotel, stars, 2)

Table 1: A multi-domain dialogue example in Multi-WOZ.

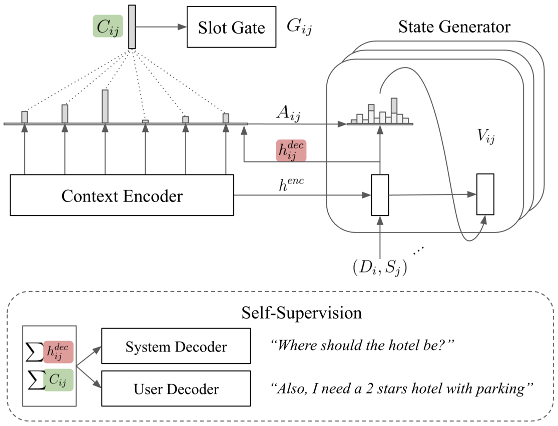


Figure 1: The block diagram of copy-attention ontology-free framework for dialogue state tracking. The self-supervised modules (dotted parts) are discarded during inference time.

robust to unseen scenarios.

We simulate limited labeled data using Multi-WOZ (Budzianowski et al., 2018), one of the task-oriented dialogue benchmark datasets, with 1%, 5%, 10%, and 25% labeled data scenarios. The experimental results of 1% data setting show that we can improve joint goal accuracy by 4.5% with the proposed consistency objective and with an additional 4.43% improvement if we add the behavior modeling objective. Furthermore, we found that a DST model can also benefit from those remaining unlabeled data if we joint train with their self-supervised signals, suggesting a promising research direction of semi-supervised learning. Lastly, we visualize the learned latent variables and conduct an ablation study to analyze our approaches.

2 Background

Let us define $X_{1:T} = \{(U_1, R_1), \dots, (U_T, R_T)\}$ as the set of user utterance and system response pairs in T turns of a dialogue, and $B = \{B_1, \dots, B_T\}$ are the annotated dialogue states. Each B_t contains a set of *(domain, slot, value)* tuples accumulated from turn 1 to turn t, therefore, the number of tuples usually grows with turn t. Note that it is possible to have multiple domains triggered in the same state B_t. A dialogue example and its labeled states are shown in Table 1.

We briefly introduce a common approach for ontology-free DST in the following. As shown in Figure 1, a context encoder encodes dialogue history $X_{1:t}$, and a state generator decodes slot values V_{ij} for each *(domain, slot)* pair $\{(D_i, S_j)\}$, where i denotes the domain index and j is the slot index. The context encoder and the state generator can be either a pre-trained language model or a simple recurrent neural network. During the decoding stage for each V_{ij}, a copy-attention mechanism such as text span extraction (Vinyals et al., 2015) or pointer generator (See et al., 2017) approach is added to the state generator and strengthen its value generation process.

Moreover, many ontology-free DST models are also equipped with a slot gate mechanism (Xu and Hu, 2018; Rastogi et al., 2019; Zhang et al., 2019), which is a classifier that predicts whether a *(domain, slot)* pair is mentioned, not mentioned, or a user does not care about it. In this pipeline setting, they can add additional supervision to their models and ignore the not mentioned pairs' prediction. More specifically, the *(domain, slot)* pair $\{(D_i, S_j)\}$ obtains its context vector C_{ij} to predict a slot gate distribution G_{ij}. The context vector C_{ij} is the weighted-sum of encoder hidden states using the attention distribution A_{ij}, and G_{ij} is a three-way classification distribution mapping from the context vector:

$$\begin{aligned} G_{ij} &= \text{FFN}(C_{ij}) \in \mathbb{R}^3, \\ C_{ij} &= A_{ij} h^{enc} \in \mathbb{R}^{d_{emb}}, \\ A_{ij} &= \text{Softmax}(\text{Dist}(h^{dec}_{ij}, h^{enc})) \in \mathbb{R}^M, \end{aligned} \tag{1}$$

where d_{emb} is the hidden size, $h^{enc} \in \mathbb{R}^{M \times d_{emb}}$ is hidden states of the context encoder for M input words, and $h^{dec}_{ij} \in \mathbb{R}^{d_{emb}}$ is the first hidden state of the state generator. The Dist function can be any vector similarity metric, and FFN can be any kind of classifier.

Such model is usually trained end-to-end with two loss functions, one for slot values generation and the other for slot gate prediction. The overall supervised learning objective from the annotated state labels is

$$L_{sl} = \sum^{|ij|} H(V_{ij}, \hat{V}_{ij}) + H(G_{ij}, \hat{G}_{ij}), \quad (2)$$

where H is the cross-entropy function. The total number of *(domain, slot)* pairs is $|ij|$, and there are 30 pairs in MultiWOZ.

3 Self-Supervised Approaches

This section introduces how to leverage dialogue history X, which is easy to collect, to boost DST performance without annotated dialogue state labels implicitly. We first show how we preserve latent consistency using stochastic word dropout, and we discuss our design for utterance generation.

3.1 Latent Consistency

The goal of preserving latent consistency is that DST models should be robust to a small perturbation of input dialogue history. As shown in Figure 2, we first randomly mask out a small number of input words into unknown words for N_{drop} times. Then we use N_{drop} dialogue history together with the one without dropping any word as input to the base model and obtain $N_{drop}+1$ model predictions.

Masking words into unknown words can also strengthen the representation learning because when important words are masked, a model needs to rely on its contextual information to obtain a meaningful representation for the masked word. For example, "I want a cheap restaurant that does not spend much." becomes "I want a [UNK] restaurant that [UNK] not spend much." This idea is motivated by the masked language model learning (Devlin et al., 2019). We randomly mask words instead of only hiding slot values because it is not easy to recognize the slot values without ontology.

Afterward, we produce a "gues" for its latent variables: the attention distribution and the slot gate distribution in our setting. Using the $N_{drop}+1$ model's predictions, we follow the label guessing process in MixMatch algorithm (Berthelot et al., 2019) to obtain a smooth latent distribution. We compute the average of the model's predicted distributions by

$$\hat{A}^*_{ij}, \hat{G}^*_{ij} = \frac{\sum_{d=1}^{N_{drop}+1} P(A_{ij}, G_{ij} | X^d_{1:t}, \theta)}{N_{drop}+1}, \quad (3)$$

where θ is the model parameters. A_{ij} and G_{ij} are the smooth latent distribution that we would like a DST model to follow. We include the original input without word masking input the average. During the early stage of training, we may not have a good latent distribution even if it has labeled supervision.

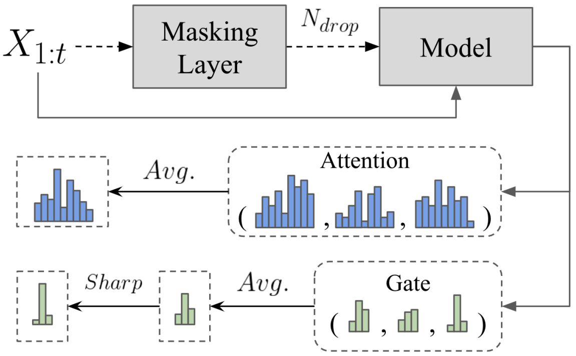


Figure 2: The block diagram of preserving latent consistency. $N_{drop}+1$ attention and slot gate distributions are averaged (and sharpened) to be the guessed distribution.

Furthermore, inspired by the common usage of entropy minimization (Grandvalet and Bengio, 2005), we perform one more step for the gate distribution. We apply a sharpening function, adjusting the temperature T of the categorical distribution, to reduce the entropy of slot gate prediction.

$$\begin{aligned} \hat{G}^{**}_{ij} &= \text{Sharp}(\hat{G}^*_{ij}, T), \\ \text{Sharp}(p, T)_i &= p_i^{\frac{1}{T}} / \sum p_i^{\frac{1}{T}}. \end{aligned} \quad (4)$$

In this way, we encourage a DST model to be more confident to its gate prediction as T decreases, since the sharpen $\hat{G}^{**}_{ij}$ will approach a one-hot distribution when $T = 0$. The sharpening function is not applied to the predicted attention distribution because we do not expect and force attention distribution to be a sharp categorical distribution.

We use the two guessed distributions to train a DST model to be consistent for the attention and slot gate given noise inputs. The following consistency loss is added:

$$\begin{aligned} L_{cons} = \sum^{|ij|} \sum_{d}^{N_{drop}+1} (&\text{MSE}(\hat{G}^{**}_{ij}, \hat{G}^d_{ij}) \\ &+ \text{MSE}(\hat{A}^*_{ij}, \hat{A}^d_{ij})). \end{aligned} \quad (5)$$

We follow Berthelot et al. (2019) to apply the mean-squared error function as our loss function.

We train a model to be consistent in terms of latent distributions because it is hard to guarantee the quality of generated values in different perturbed input, especially when we do not have much labeled data. Also, each perturbed sample may generate slot values that have different number of words, and maintaining consistency of sequential distributions could be challenging. As a result, we use slot gate distribution and attention distribution as intermediate targets since the former is the first stage for the whole prediction process, and the latter directly influences the copy mechanism.

3.2 Conversational Behavior Modeling

We hypothesize that with similar dialogue states, a system will reply also similar responses. For example, when a system asks "What is your taxi destination from Palo Alto?", then we can infer that system's state may include *(taxi, departure, Palo Alto)*. In this way, we can potentially model the correlation between dialogue states and dialogue behavior. In practice, we use two decoders, one modeling user and one modeling system behavior, to generate utterances based on the learned representations from a DST model.

We use a gated recurrent unit (GRU) to generate the next system response based on the dialogue history $X_{1:t}$ and current predicted dialogue states B_t, and use another GRU to generate/recover user utterance based on last dialogue history $X_{1:t-1}$ and current predicted dialogue states B_t. Intuitively, we expect the system GRU to capture correlation between R_{t+1} and B_t, and the user GRU to learn for U_t and B_t. GRUs generate a sequence of words during training and compute cross-entropy losses between generated sentences and target sentences. We do not use the attention mechanism intentionally because 1) our goal is not to have an outstanding performance on sentence generation, and 2) we expect the model can generate sentences by solely aligning its initial states from a DST model.

As shown in Figure 1, we initial our system and user GRUs using latent variables from an ontology-free DST model. The initial state h_{init} to be aligned is defined by

$$h_{init} = \sum^{|ij|} [h_{ij}^{dec}; C_{ij}], \tag{6}$$

where $[;]$ denotes vector concatenation and we sum representations from all *(domain, slot)* pairs. We use the context vector C_{ij} to represent dialogue history, and h_{ij}^{dec} to represent dialogue state. The overall self-supervised loss function for modeling conversational behavior is

$$L_{cb} = H(R_{t+1}, \hat{R}_{t+1}) + H(U_t, \hat{U}_t), \tag{7}$$

where $\hat{R}_{t+1}$ and $\hat{U}_t$ are predicted response and user utterance initialized by the h_{init} vector.

3.3 Overall Objectives

During training, we optimize both supervised signal and self-supervised signal using the labeled data. The overall loss function is

$$L_{label} = L_{sl} + \alpha L_{cb} + \beta L_{cons}, \tag{8}$$

where α and β are hyper-parameters.

Other than labeled data, we can also sample unlabeled data to perform self-supervision as a regularization term. This strategy can be considered as a semi-supervised approach, leveraging unlabeled data to learn a smooth prediction. For unlabeled data, we use only the self-supervised signal to update the model,

$$L_{unlabel} = L_{cb} + \beta L_{cons}. \tag{9}$$

In practice, we first draw a batch of samples from labeled data to update the model's parameters and then draw another batch of samples from unlabeled data. We find that taking turns to train unlabeled data with labeled data works better than pre-training with unlabeled data then fine-tuning on labeled data.

4 Experiments

4.1 Base Model

In this paper, we focus on applying self-supervision for ontology-free DST approaches. We select TRADE (Wu et al., 2019a) model as the base model. We select TRADE because 1) it is a pointer-generator based dialogue state tracker with a copy-attention mechanism that can generate unseen slot values, and 2) it is one of the best ontology-free models that show good domain generalization ability in its zero-shot and few-shot experiments, and it is open-source [1]. Note that our proposed self-supervised training objectives are not limited to one DST model. For example, the BERTQA-based span extraction methods (Chao and Lane, 2019; Gao et al., 2019) can be applied with slight modification, viewing [CLS] token as the encoded vector and the span distributions as the slot contextual representations.

[1] github.com/jasonwu0731/trade-dst

	1%	**5%**	**10%**	**25%**
TRADE (w/o Ont.) (Wu et al., 2019a)	9.70 (11.74)	29.38 (32.41)	34.07 (37.42)	41.41 (44.01)
+ Consistency	14.22 (15.77)	30.18 (33.59)	36.14 (39.03)	41.38 (44.33)
+ Behavior	18.31 (20.59)	31.13 (34.38)	36.90 (40.70)	42.48 (45.12)
Consistency + Behavior	18.65 (21.21)	31.61 (35.67)	37.05 (40.29)	**42.71 (45.21)**
Consistency + Behavior + Unlabeled Data	**20.41 (23.0)**	**33.67 (37.82)**	**37.16 (40.65)**	42.69 (45.14)
SUMBT (w/ Ont.) (Lee et al., 2019)	4.30 (-)	30.56 (-)	38.31 (-)	42.59 (-)
TOD-BERT (w/ Ont.) (Wu et al., 2020)	10.3 (-)	27.8 (-)	38.8 (-)	44.3 (-)
DSDST-Span (w/o Ont.) (Zhang et al., 2019)	19.82 (-)	32.20 (-)	37.81 (-)	39.48 (-)

Table 2: Joint goal accuracy and its fuzzy matching version in parentheses on MultiWOZ test set from 1% to 25% labeled training data. As a reference, we test some other DST trackers that using the pre-trained language model BERT (Devlin et al., 2019) under limited labeled scenario, as shown in the last few rows.

	1%	**5%**	**10%**	**25%**	**100%**
Hotel	33	174	341	862	3381
Train	35	166	332	809	3103
Attraction	29	143	276	696	2717
Restaurant	36	181	377	928	3813
Taxi	11	71	150	395	1654
Total*	84	421	842	2105	8420

Table 3: Number of simulated labeled dialogues on MultiWOZ training set. (* Total number of dialogues is less than the summation of dialogues in each domain because each dialogue has multiple domains.)

4.2 Dataset

MultiWOZ (Budzianowski et al., 2018) is one of the largest existing human-human conversational corpus spanning over seven domains, containing around 8400 multi-turn dialogues, with each dialogue averaging 13.7 turns. We follow Wu et al. (2019a) to only use the five domains (*hotel*, *train*, *attraction*, *restaurant*, *taxi*) because the other two domains (*hospital*, *police*) have very few dialogues (10% compared to others) and only exist in the training set. In total, there are 30 *(domain, slot)* pairs. We also evaluate on its revised version 2.1 from Eric et al. (2019) in our experiments, due to the space limit, results on version 2.1 are reported in the Appendix.

We simulate a limited labeled data scenario by randomly selecting dialogues from the original corpus using a fixed random seed. The dataset statistics of each labeled ratio is shown in Table 3. For example, in 1% labeled data setting, there are 84 dialogues across five different domains. Note that the summation of dialogues from each domain is more than the number of total dialogues because each dialogue could have more than one domain, e.g., two domains are triggered in the Table 1.

4.3 Training Details

The model is trained end-to-end using Adam optimizer (Kingma and Ba, 2015) with a batch size of 8 or 32. A grid search is applied for α and β in the range of 0.1 to 1, and we find that models are sensitive to different α and β. The learning rate annealing is used with a 0.2 dropout ratio. All the word embeddings have 400 dimensions by concatenating 300 Glove embeddings (Pennington et al., 2014) and 100 character embeddings (Hashimoto et al., 2016). A greedy decoding strategy is used for the state generator because the slot values are usually short in length. We mask out 20%-50% of input tokens to strengthen prediction consistency. The temperature T for sharpening is set to 0.5, and augmentation number N_{drop} is 4.

4.4 Results

Joint goal accuracy and its fuzzy matching [2] version are used to evaluate the performance on multi-domain DST. The joint goal accuracy compares the predicted dialogue states to the ground truth B_t at each dialogue turn t, and the output is considered correct if and only if all the *(domain, slot, value)* tuples exactly match the ground truth values in B_t, which is a very strict metric. The fuzzy joint goal accuracy is used to reward partial matches with the ground truth (Rastogi et al., 2019). For example, two similar values "Palo Alto" and "Palo Alto city" have a fuzzy score of 0.78.

In Table 2, we evaluate four different limited labeled data scenarios: 1%, 5%, 10%, and 25%. We test our proposed self-supervised signals by only adding latent consistency objective (row 2), only adding conversational behavior objective (row 3), using both of them (row 4), and using both

[2] github.com/seatgeek/fuzzywuzzy

	Gate Acc (↑)	Attention KL (↓)
100% Data	97.61	-
1% Data w/o SSL	91.38	10.58
1% Data w/ SSL	**94.30**	**6.19**

Table 4: Gate accuracy on 1% data improves 2.92% and KL divergence between 1% and 100% data decreases 4.39 with self-supervision.

of them together with unlabeled data (row 5). In general, we find that each self-supervision signal we presented is useful in its degree, especially for 1% and 5% labeled data scenarios. Modeling conversational behavior seems to be more effective than preserving prediction consistency, which is not surprising because the latter is a point-wise self-supervised objective function. We also found that self-supervision becomes less dominant and less effective as the number of labeled data increases. We try 100% labeled data with self-supervision, and it only achieves slight improvement, 48.72% joint goal accuracy compared to the original reported 48.62%.

Taking a closer look to the results in Table 2, preserving consistency has 4.52% (or 4.03% fuzzy) improvement for 1% scenario. Once the labeled data increases to 25% (2105 dialogues), there is no difference with or without the consistency objective. Meanwhile, modeling conversational behavior objective seems to be more effective than the consistency objective, as it has 8.61% (or 8.85% fuzzy) improvement. A small improvement can be further observed if we combine both of them and jointly train end-to-end. When we also leverage those remaining dialogue data and conduct semi-supervised learning, we can achieve the highest joint goal accuracy, 20.41% in 1% setting, and 33.67% in 5% setting. In these experiments, we simply use the remaining dialogues in the dataset as unlabeled data, e.g., 1% labeled with 99% unlabeled, 5% labeled with 95% unlabeled, etc.

We also test some other DST trackers in the last few rows in Table 2, which all of them are replied on the pre-trained language model BERT (Devlin et al., 2019). SUMBT (Lee et al., 2019) and TOD-BERT (Wu et al., 2020) are ontology-based approaches. The former uses BERT to encode each utterance and builds an RNN tracker on top of BERT. The latter uses its pre-trained task-oriented dialogue BERT to encode dialogue history and adds simple slot-dependent classifiers. Note that we still assume they have a full ontology in this setting even though it is not a fair comparison under a limited labeled scenario. DSDST-Span (Zhang et al., 2019) is an ontology-free DST tracker, it uses BERT to encode dialogue history together with each *(domain, slot)* pair separately and extract a corresponding text span as its slot values.

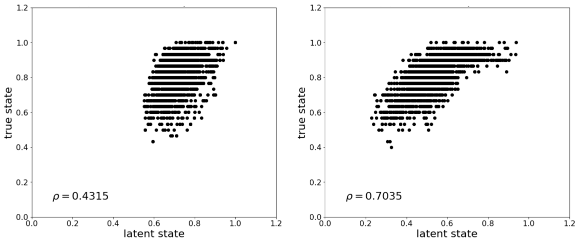


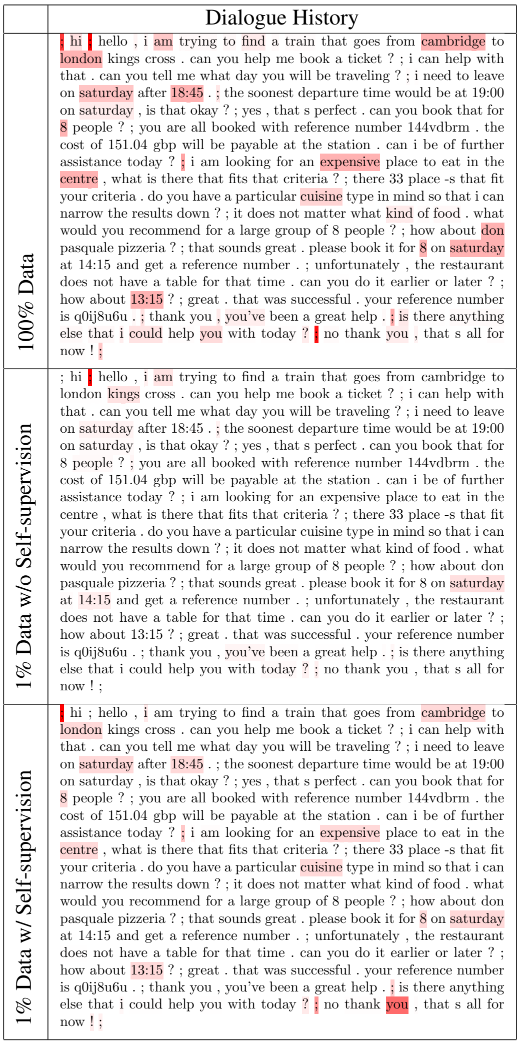
Figure 3: The correlation on test set between latent dialogue states and true dialogue states on 1% labeled data. Left-hand side is without self-supervision and right-hand side is with self-supervision.

5 Analysis and Visualization

We would interpret how self-supervised signals help to learn better DST performance. The first interesting observation is that the key improvement comes from the slot-dependent context vectors C_{ij}. If we remove the context vector C_{ij} from Eq (6), the performance of 1% labeled data setting drops from 18.31% to 11.07%. The next question is: what do these contextual vectors influence? First, context vectors are the weighted-sum of encoder hidden states, which means they correlate with the learned attention distribution. Also, context vectors are used to predict slot gates, which is essential to be able to trigger the state generator. Therefore, using self-supervision to align contextual slot vectors may help get better attention distributions and better slot gate prediction.

Slot Gate As shown in Table 4, gate accuracy of 1% labeled data improves by around 3% with self-supervision. We also compare attention distributions among a model trained with 1% labeled data, a model trained with 1% labeled data and self-supervision, and a model trained with 100% labeled data. We observe a smaller value of KL divergence with self-supervision (the lower, the better), i.e., the attention distribution becomes more similar to the one learned from 100% labeled data, which we assume that it is supposed to be a better attention distribution.

We randomly pick up 2,000 dialogue turns on the test set to compute the correlation between latent



	Dialogue History
100% Data	; hi ; hello , i am trying to find a train that goes from cambridge to london kings cross . can you help me book a ticket ? ; i can help with that . can you tell me what day you will be traveling ? ; i need to leave on saturday after 18:45 . ; the soonest departure time would be at 19:00 on saturday , is that okay ? ; yes , that s perfect . can you book that for 8 people ? ; you are all booked with reference number 144vdbrm . the cost of 151.04 gbp will be payable at the station . can i be of further assistance today ? ; i am looking for an expensive place to eat in the centre , what is there that fits that criteria ? ; there 33 place -s that fit your criteria . do you have a particular cuisine type in mind so that i can narrow the results down ? ; it does not matter what kind of food . what would you recommend for a large group of 8 people ? ; how about don pasquale pizzeria ? ; that sounds great . please book it for 8 on saturday at 14:15 and get a reference number . ; unfortunately , the restaurant does not have a table for that time . can you do it earlier or later ? ; how about 13:15 ? ; great . that was successful . your reference number is q0ij8u6u . ; thank you , you've been a great help . ; is there anything else that i could help you with today ? ; no thank you , that s all for now ! ;
1% Data w/o Self-supervision	; hi ; hello , i am trying to find a train that goes from cambridge to london kings cross . can you help me book a ticket ? ; i can help with that . can you tell me what day you will be traveling ? ; i need to leave on saturday after 18:45 . ; the soonest departure time would be at 19:00 on saturday , is that okay ? ; yes , that s perfect . can you book that for 8 people ? ; you are all booked with reference number 144vdbrm . the cost of 151.04 gbp will be payable at the station . can i be of further assistance today ? ; i am looking for an expensive place to eat in the centre , what is there that fits that criteria ? ; there 33 place -s that fit your criteria . do you have a particular cuisine type in mind so that i can narrow the results down ? ; it does not matter what kind of food . what would you recommend for a large group of 8 people ? ; how about don pasquale pizzeria ? ; that sounds great . please book it for 8 on saturday at 14:15 and get a reference number . ; unfortunately , the restaurant does not have a table for that time . can you do it earlier or later ? ; how about 13:15 ? ; great . that was successful . your reference number is q0ij8u6u . ; thank you , you've been a great help . ; is there anything else that i could help you with today ? ; no thank you , that s all for now ! ;
1% Data w/ Self-supervision	; hi ; hello , i am trying to find a train that goes from cambridge to london kings cross . can you help me book a ticket ? ; i can help with that . can you tell me what day you will be traveling ? ; i need to leave on saturday after 18:45 . ; the soonest departure time would be at 19:00 on saturday , is that okay ? ; yes , that s perfect . can you book that for 8 people ? ; you are all booked with reference number 144vdbrm . the cost of 151.04 gbp will be payable at the station . can i be of further assistance today ? ; i am looking for an expensive place to eat in the centre , what is there that fits that criteria ? ; there 33 place -s that fit your criteria . do you have a particular cuisine type in mind so that i can narrow the results down ? ; it does not matter what kind of food . what would you recommend for a large group of 8 people ? ; how about don pasquale pizzeria ? ; that sounds great . please book it for 8 on saturday at 14:15 and get a reference number . ; unfortunately , the restaurant does not have a table for that time . can you do it earlier or later ? ; how about 13:15 ? ; great . that was successful . your reference number is q0ij8u6u . ; thank you , you've been a great help . ; is there anything else that i could help you with today ? ; no thank you , that s all for now ! ;

Figure 4: Attention visualization for a dialogue history. The darker color means higher attention weight. The 1% labeled data model with self-supervision learns attention distribution more similar to the one using 100% labeled data.

learned states (h_{init}) of 1% labeled data and the true gating status (G) of the *(domain, slot)* pairs. As shown in Figure 3, the x-axis is the cosine similarity score between two latent dialogue states the model learned, and the y-axis is the cosine similarity score of their true gating status. Ideally, when the slot gate status is similar, then the learned representations should also have a high similarity score. We find the model trained with self-supervision (right) has a higher Pearson correlation coefficient than the one without (left), increasing from 0.4315 to 0.7035, implying that with self-supervision, models can learn better state representations.

Copy Attention We also visualize the attention distributions of a dialogue history in Figure 4. The darker red color means the higher attention weight and the higher copy probability. We sum attention distributions of A_{ij} for all *(domain, slot)* pairs and normalize it. The 1% labeled data model with self-supervision has an attention distribution similar to the one using 100% labeled data. For example, both of them focus on some useful slot information such as “Cambridge”, “London”, “Saturday”, and “18:45”. The results of attention distribution are crucial, especially in our limited labeled setting. The higher the attention weight, the higher the probability that such word will be copied from the dialogue history to the output slot values. More attention visualizations are shown in the Appendix.

Slot Accuracy Analysis We are interested in which domains and which slots are easier to be self-supervised learned. As shown in Figure 5, the x-axis is each *(domain, slot)* pair, and the y-axis is its slot accuracy (at each dialogue turn whether the pair is predicted correctly). The blue bar is the performance of 1% labeled data without self-supervision. The orange part is the improvement by using self-supervision. The green part can be viewed as the upper-bound of the base model using 100% labeled data.

The top three *(domain, slot)* pairs that is most effective with self-supervision are *(train, day)*, and *(train, departure)*, *(train, destination)*. On the other hand, self-supervision are less helpful to pairs such as *(hotel, parking)*, *(hotel, internet)*, *(restaurant, name)*, and all the pairs in the *taxi* domain. One possible reason is that self-supervision is sensitive to the unlabeled data size, i.e., the major domain is dominant in the overall performance. It is worth mentioning that in the *taxi* domain, all the slots perform relatively well with 1% labeled data. This could also explain why the zero-shot performance reported in Wu et al. (2019a) is much better than the other four domains.

6 Related Work

Dialogue State Tracking Traditional dialogue state tracking models combine semantics extracted by language understanding modules to estimate the current dialogue states (Williams and Young, 2007; Thomson and Young, 2010; Wang and Lemon, 2013; Williams, 2014), or to jointly learn speech understanding (Henderson et al., 2014; Zilka and Jurcicek, 2015). One drawback is that they rely on hand-crafted features and complex domain-specific lexicons besides the ontology, and are difficult to extend and scale to new domains. As the need

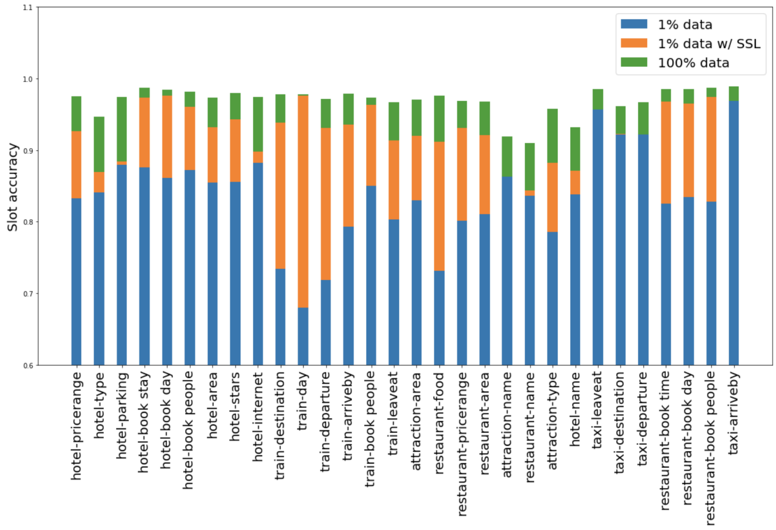


Figure 5: Slot accuracy visualization for each *(domain, slot)* pairs. Several slots such as *(train, day)* and *(hotel, book stay)* that using 1% data with self-supervision almost perform the same as using 100% data.

for domain expanding, research direction moves from single domain DST setting and datasets (Wen et al., 2017) to multi-domain DST setting and datasets (Budzianowski et al., 2018; Eric et al., 2019).

There are three main categories to perform DST, ontology-based, partial-ontology-based, and ontology-free approaches. Ontology-based methods (Mrkšić et al., 2017; Wen et al., 2017; Rastogi et al., 2017; Ren et al., 2018; Zhong et al., 2018; Ramadan et al., 2018; Lee et al., 2019; Chen et al.) train metric learning functions for context encoder and ontology encoder, and score over a predefined slot value candidates. Partial-ontology-based (Goel et al., 2019; Zhang et al., 2019; Rastogi et al., 2019) approaches only use part of an ontology to perform ranking and use generation techniques for the remaining slots. Ontology-free methods (Chao and Lane, 2019; Gao et al., 2019; Ren et al., 2019; Kumar et al., 2020; Wu et al., 2019a; Kumar et al., 2020; Kim et al., 2019) rely on generation with copy mechanism without predefined ontology, which has better generalization ability to unseen slot values. Our work is closer to ontology-free approaches because it is reasonable to assume that we cannot access an ontology under a limited labeled data scenario.

Self-Supervised Learning There is a wide literature on self-supervision (Barlow, 1989) and semi-supervised techniques (Chapelle et al., 2009). Swayamdipta et al. (2018) introduce a syntactic scaffold, an approach to incorporate syntactic information into semantic tasks. Sankar et al. (2019) found that Seq2Seq models are rarely sensitive to most perturbations, such as missing or reordering utterances. Shi et al. (2019) used variational RNN to extract latent dialogue structure and applied it to dialogue policy learning. Wu et al. (2019b) introduced a self-supervised learning task, inconsistent order detection, to explicitly capture the flow of conversation in dialogues. Jin et al. (2018) use unlabeled data to train probabilistic distributions over the vocabulary space as dialogue states for neural dialogue generation. Su et al. (2020) provide both supervised and unsupervised learning algorithms to train language understanding and generation models in a dual learning setting. Tseng et al. (2019) applied pseudo-labeling and $\prod$-model (Sajjadi et al., 2016) as additional semi-supervision to bootstrap state trackers. Our latent consistency comes from the consistency regularization (Sajjadi et al., 2016; Berthelot et al., 2019), leveraging the idea that a model should output the same class distribution for an unlabeled example even after it has been augmented.

7 Conclusion

We investigate the potential of using self-supervised approaches for label-efficient DST in task-oriented dialogue systems. We strengthen latent consistency by augmenting data with stochastic word dropout and label guessing. We model conversational behavior by the next response generation and turn utterance generation tasks. Experimental results show that we can significantly boost the joint goal accuracy with limited labeled data by exploiting self-supervision. We conduct comprehensive result analysis to cast light on and stimulate label-efficient DST.

References

Horace B Barlow. 1989. Unsupervised learning. *Neural computation*, 1(3):295–311.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. 2019. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*.

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.

Guan-Lin Chao and Ian Lane. 2019. Bert-dst: Scalable end-to-end dialogue state tracking with bidirectional encoder representations from transformer. *arXiv preprint arXiv:1907.03040*.

Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. 2009. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542.

Lu Chen, Boer Lv, Chi Wang, Su Zhu, Bowen Tan, and Kai Yu. Schema-guided multi-domain dialogue state tracking with graph attention neural networks.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

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*.

Shuyang Gao, Abhishek Sethi, Sanchit Aggarwal, Tagyoung Chung, and Dilek Hakkani-Tur. 2019. Dialog state tracking: A neural reading comprehension approach. *arXiv preprint arXiv:1908.01946*.

Rahul Goel, Shachi Paul, and Dilek Hakkani-Tür. 2019. Hyst: A hybrid approach for flexible and accurate dialogue state tracking. *arXiv preprint arXiv:1907.00883*.

Yves Grandvalet and Yoshua Bengio. 2005. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536.

Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2016. A joint many-task model: Growing a neural network for multiple nlp tasks. *arXiv preprint arXiv:1611.01587*.

Matthew Henderson, Blaise Thomson, and Steve Young. 2014. Word-based dialog state tracking with recurrent neural networks. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 292–299.

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. ACM.

Sungdong Kim, Sohee Yang, Gyuwan Kim, and Sang-Woo Lee. 2019. Efficient dialogue state tracking by selectively overwriting memory. *arXiv preprint arXiv:1911.03906*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Adarsh Kumar, Peter Ku, Anuj Kumar Goyal, Angeliki Metallinou, and Dilek Hakkani-Tur. 2020. Ma-dst: Multi-attention based scalable dialog state tracking. *arXiv preprint arXiv:2002.08898*.

Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019. SUMBT: Slot-utterance matching for universal and scalable belief tracking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5478–5483, Florence, Italy. Association for Computational Linguistics.

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, Vancouver, Canada. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher 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.

Osman Ramadan, Paweł Budzianowski, and Milica Gasic. 2018. Large-scale multi-domain belief tracking with knowledge sharing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 432–437. Association for Computational Linguistics.

Abhinav Rastogi, Dilek Hakkani-Tür, and Larry Heck. 2017. Scalable multi-domain dialogue state tracking. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 561–568. IEEE.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2019. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *arXiv preprint arXiv:1909.05855*.

Liliang Ren, Jianmo Ni, and Julian McAuley. 2019. Scalable and accurate dialogue state tracking via hierarchical sequence generation. *arXiv preprint arXiv:1909.00754*.

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.

Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. 2016. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 1163–1171.

Chinnadhurai Sankar, Sandeep Subramanian, Chris Pal, Sarath Chandar, and Yoshua Bengio. 2019. Do neural dialog systems use the conversation history effectively? an empirical study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 32–37, Florence, Italy. Association for Computational Linguistics.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1073–1083.

Weiyan Shi, Tiancheng Zhao, and Zhou Yu. 2019. Unsupervised dialog structure learning. *arXiv preprint arXiv:1904.03736*.

Shang-Yu Su, Chao-Wei Huang, and Yun-Nung Chen. 2020. Towards unsupervised language understanding and generation by joint dual learning. *arXiv preprint arXiv:2004.14710*.

Swabha Swayamdipta, Sam Thomson, Kenton Lee, Luke Zettlemoyer, Chris Dyer, and Noah A Smith. 2018. Syntactic scaffolds for semantic structures. *arXiv preprint arXiv:1808.10485*.

Blaise Thomson and Steve Young. 2010. Bayesian update of dialogue state: A pomdp framework for spoken dialogue systems. *Computer Speech & Language*, 24(4):562–588.

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 *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 1273–1278, Hong Kong, China. Association for Computational Linguistics.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700.

Zhuoran Wang and Oliver Lemon. 2013. A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information. In *Proceedings of the SIGDIAL 2013 Conference*, pages 423–432.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gasic, Lina M. Rojas Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. 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. Association for Computational Linguistics.

Jason D Williams. 2014. Web-style ranking and slu combination for dialog state tracking. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 282–291.

Jason D Williams and Steve Young. 2007. Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422.

Chien-Sheng Wu, Steven Hoi, Richard Socher, and Caiming Xiong. 2020. Tod-bert: Pre-trained natural language understanding for task-oriented dialogues. *arXiv preprint arXiv:2004.06871*.

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

Jiawei Wu, Xin Wang, and William Yang Wang. 2019b. Self-supervised dialogue learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3857–3867, Florence, Italy. Association for Computational Linguistics.

Puyang Xu and Qi Hu. 2018. An end-to-end approach for handling unknown slot values in dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1448–1457. Association for Computational Linguistics.

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*.

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-locally self-attentive encoder for dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1458–1467. Association for Computational Linguistics.

Dengyong Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, and Bernhard Schölkopf. 2004. Learning with local and global consistency. In *Advances in neural information processing systems*, pages 321–328.

Lukas Zilka and Filip Jurcicek. 2015. Incremental lstm-based dialog state tracker. In *2015 Ieee Workshop on Automatic Speech Recognition and Understanding (Asru)*, pages 757–762. IEEE.