

USING DOMAIN KNOWLEDGE TO GUIDE DIALOG STRUCTURE INDUCTION VIA NEURAL PROBABILISTIC SOFT LOGIC

Anonymous authors

Paper under double-blind review

ABSTRACT

Dialog Structure Induction (DSI) is the task of inferring the latent dialog structure (i.e., a set of dialog states and their temporal transitions) of a given goal-oriented dialog. It is a critical component for modern dialogue system design and discourse analysis. Existing DSI approaches are often purely data-driven, deploy models that infer latent states without access to domain knowledge, underperform when the training corpus is limited/noisy, or have difficulty when test dialogs exhibit distributional shifts from the training domain. This work explores a neural-symbolic approach as a potential solution to these problems. We introduce *Neural Probabilistic Soft Logic Dialog Structure Induction* (NEUPSL DSI), a principled approach that injects symbolic knowledge into the latent space of a generative neural model. Over three unsupervised dialog structure induction datasets the injection of symbolic knowledge using NEUPSL DSI provides a consistent boost in performance over the canonical baselines.

1 INTRODUCTION

The seamless integration of prior domain knowledge into the neural learning of language structure has been an open challenge in the machine learning and natural language processing communities. In this work, we inject symbolic knowledge into the neural learning process of a two-party *dialog structure induction* (DSI) task (Zhai & Williams, 2014; Shi et al., 2019). This task aims to learn a graph, known as the *dialog structure*, capturing the potential flow of states occurring in a dialog dataset for a specific task-oriented domain, e.g., Figure 1 represents a possible dialog structure for the goal-oriented task of booking a hotel. Nodes in the dialog structure represent conversational topics or *dialog acts* that abstract the intent of individual utterances, and edges represent transitions between dialog acts over successive turns of the dialog.

Traditionally, dialog structure is hand-crafted by human domain experts. This process is labor-intensive and, in most situations, does not generalize easily to new domains. Previous work has used supervised methods to learn this dialog structure from labeled data, starting from Jurafsky (1997). However, since structure annotation is expensive and subject to low inter-rater agreements, these methods are constrained by small training datasets, and low label quality Zhai & Williams (2014). Therefore, recent work studies unsupervised DSI; e.g., *hidden Markov models* Chotimongkol (2008); Ian Ritter et al. (2010); Zhai & Williams (2014) and more recently *Variational Recurrent Neural Networks* (VRNN) Chung et al. (2015); Shi et al. (2019). Being purely data-driven, these approaches have difficulty with limited/noisy data and cannot easily exploit domain-specific or domain-independent constraints on dialog (e.g. *Greet* utterances are typically made in the first couple of turns) that may be readily provided by human experts.

In this work, we propose *Neural Probabilistic Soft Logic Dialog Structure Induction* (NEUPSL DSI). This practical neuro-symbolic approach improves the quality of learned dialog structure by infusing domain knowledge into the end-to-end, gradient-based learning of a neural model. We leverage *Probabilistic Soft Logic* (PSL), a well-studied soft logic formalism, to express domain knowledge as soft rules in succinct and interpretable first-order logic statements that can be incorporated easily into differentiable learning (Bach et al., 2017; Pryor et al., 2022). This leads to a simple method for knowledge injection with minimal change to the SGD-based training pipeline of an existing neural generative model.

Our key contributions are: 1) We propose NEUPSL DSI, which introduces a novel smooth relaxation of PSL constraints tailored to ensure a rich gradient signal during back-propagation; and 2) We evaluate NEUPSL DSI over synthetic and realistic dialog datasets under three settings: standard generalization, domain generalization, and domain adaptation.

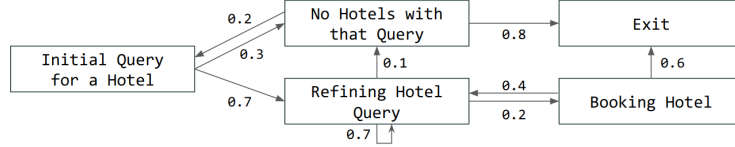


Figure 1: Example dialog structure for the goal-oriented task booking a hotel.

2 BACKGROUND ON PROBABILISTIC SOFT LOGIC

This work introduces soft constraints in a declarative fashion, similar to Probabilistic Soft Logic (PSL) Bach et al. (2017). PSL models relational dependencies and structural constraints using first-order logical rules, referred to as *templates* with arguments known as *atoms*. For example, the statement “the first utterance in a dialog is likely to belong to the *greet* state” can be expressed as:

$$\text{FIRSTUTT}(U) \rightarrow \text{STATE}(U, \text{greet}) \quad (1)$$

Where $(\text{FIRSTUTT}(U), \text{STATE}(U, \text{greet}))$ are the *atoms* (i.e., atomic boolean statements) indicating, respectively, whether an utterance U is the first utterance of the dialog, or if it belongs to the state *greet*. The atoms in a PSL rule are *grounded* by replacing the free variables (such as U above) with concrete instances from a domain of interest (e.g., the concrete utterance ‘Hello!’) to create *ground atoms*. The observed variables and target/decision variables of the probabilistic model correspond to ground atoms constructed from the domain, e.g., $\text{FIRSTUTT}('Hello!')$ is an observed variable and $\text{STATE}('Hello!', \text{greet})$ is a target variable. PSL allows the originally Boolean-valued atoms to take continuous truth values in the interval $[0, 1]$. In doing so, PSL replaces logical operations with a form of soft logic called *Lukasiewicz logic* Klir & Yuan (1995): 1) $A \wedge B = \max(0.0, A+B-1.0)$, 2) $A \vee B = \min(1.0, A+B)$, and 3) $\neg A = 1.0 - A$. A and B represent either ground atoms or logical expressions over atoms and take values in $[0, 1]$. For example, PSL converts Equation 1 into:

$$\phi = \min\{1, 1 - \text{FIRSTUTT}(U) + \text{STATE}(U, \text{greet})\} \quad (2)$$

PSL creates a collection of functions $\{\phi_i\}_{i=1}^m$, called *potentials*, that map data to $[0, 1]$. PSL defines a conditional probability density function over the unobserved random variables \mathbf{y} given the observed data \mathbf{x} and non-negative weights λ known as the *Hinge-Loss Markov Random Field* (HL-MRF):

$$P(\mathbf{y}|\mathbf{x}) \propto \exp\left(-\sum_{i=1}^m \lambda_i \cdot \phi_i(\mathbf{y}, \mathbf{x})\right) \quad (3)$$

3 NEURAL PROBABILISTIC SOFT LOGIC DIALOG STRUCTURE INDUCTION

Our neuro-symbolic approach to dialog structure induction combines the principled formulation of probabilistic soft logic (PSL) (Bach et al., 2017) rules with the state-of-the-art Direct-Discrete Variational Recurrent Neural Network (DD-VRNN) (Shi et al., 2019). We refer to our approach as *Neural Probabilistic Soft Logic Dialog Structure Induction* (NEUPSL DSI). Throughout this section, we define the dialog structure learning problem, describe how to integrate the neural and symbolic losses, and introduce an improvement to the neuro-symbolic gradient.

Problem Formulation Given a goal-oriented dialog corpus \mathcal{D} , we consider the DSI problem of learning a graph G underlying the corpus. More formally, a *dialog structure* is defined as a directed graph $G = (S, P)$, where $S = \{s_1, \dots, s_m\}$ encodes a set of dialog states, and P a probability distribution $p(s_t|s_{<t})$ representing the likelihood of transition between states (see Figure 1 for an example). Given the underlying dialog structure G , a dialog $d_i = \{x_1, \dots, x_T\} \in \mathcal{D}$ is a temporally-ordered set of utterances x_t . Assume x_t is defined according to an utterance distribution conditional on past history $p(x_t|s_{\leq t}, x_{<t})$, and the state s_t is defined according to $p(s_t|s_{<t})$. Given a dialog corpus $\mathcal{D} = \{d_i\}_{i=1}^n$, the task of DSI is to learn a directed graphical model $G = (S, P)$ as close to the underlying graph as possible.

3.1 INTEGRATING NEURAL AND SYMBOLIC LEARNING UNDER NEUPSL DSI

We now introduce how the NEUPSL DSI approach formally integrates the DD-VRNN with the soft symbolic constraints to allow for end-to-end gradient training. To do this, we build upon the foundations developed by Pryor et al. (2022) on Neural Probabilistic Soft Logic (NeuPSL) by augmenting the standard unsupervised DD-VRNN loss (Shi et al., 2019) with a constraint loss. Figure 2 provides a graphical representation of this integration of the DD-VRNN and the symbolic constraints. Intuitively, NEUPSL DSI can be described in three parts: instantiation, inference, and learning.

Instantiation of a NEUPSL DSI model uses a set of first-order logic templates to create a set of potentials that define a loss used for learning and evaluation. Let p_w be the DD-VRNN’s predictive

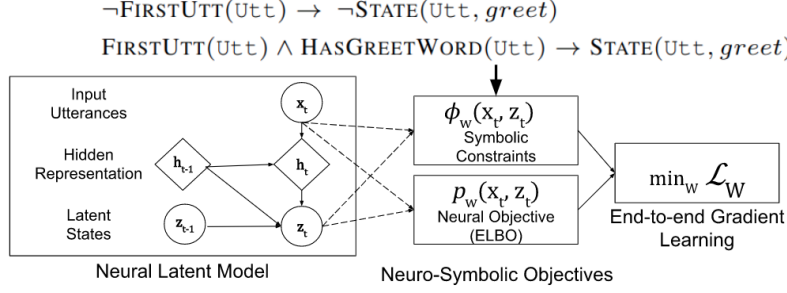


Figure 2: The high-level pipeline of the NEUPSL DSI learning procedure.

function of latent states with hidden parameters \mathbf{w} and input utterances \mathbf{x} . The output of this function, defined as $p_{\mathbf{w}}(\mathbf{x})$, will be the probability distribution representing the likelihood of each latent class for a given utterance. Given a first-order symbolic rule $\ell_i(\mathbf{y}, \mathbf{x})$ where the decision variable $\mathbf{y} = p_{\mathbf{w}}(\mathbf{x})$ is the latent state prediction from the neural model $p_{\mathbf{w}}(\mathbf{x})$, we can instantiate a set of **deep hinge-loss potentials** of the form:

$$\phi_{\mathbf{w},i}(\mathbf{x}) = \max(0, \ell_i(p_{\mathbf{w}}(\mathbf{x}), \mathbf{x})) \quad (4)$$

For example, in reference to Equation 2, the decision variable $\mathbf{y} = p_{\mathbf{w}}(\mathbf{x})$ is associated with the $\text{STATE}(\mathbf{x}, \text{greet})$ random variables, leading to:

$$\ell_i(p_{\mathbf{w}}(\mathbf{x}), \mathbf{x}) = \min\{1, 1 - \text{FIRSTUTT}(\mathbf{U}) + p_{\mathbf{w}}(\mathbf{x})\} \quad (5)$$

With the instantiated model described above, the NEUPSL DSI inference objective is broken into a *neural inference* objective and a *symbolic inference* objective. The neural inference objective is computed by evaluating the DD-VRNN model predictions with respect to the standard loss function for DSI. Given the deep hinge-loss potentials $\{\phi_{\mathbf{w},i}\}_{i=1}^m$, the symbolic inference objective is the HL-MRF likelihood (Equation 3) evaluated at the decision variables $\mathbf{y} = p_{\mathbf{w}}(\mathbf{x})$: $P_{\mathbf{w}}(\mathbf{y}|\mathbf{x}, \lambda) = \exp(-\sum_{i=1}^m \lambda_i \cdot \phi_{\mathbf{w},i}(\mathbf{x}))$. Under NEUPSL DSI, the decision variables $\mathbf{y} = p_{\mathbf{w}}(\mathbf{x})$ are implicitly controlled by neural network weights \mathbf{w} , therefore the conventional MAP inference in symbolic learning for decision variables $\mathbf{y}^* = \arg \min_{\mathbf{y}} P(\mathbf{y}|\mathbf{x})$ can be done simply via neural weight minimization $\arg \min_{\mathbf{w}} P_{\mathbf{w}}(\mathbf{y}|\mathbf{x})$. As a result, NEUPSL DSI learning minimizes a constrained optimization objective:

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} [\mathcal{L}_{DD-VRNN} + \lambda * \mathcal{L}_{constraint}] \quad (6)$$

where $\mathcal{L}_{constraint}$ is the log-likelihood of the hinge loss: $\mathcal{L}_{constraint} = -\log P_{\mathbf{w}}(\mathbf{y}|\mathbf{x}, \lambda)$.

3.2 IMPROVING SOFT LOGIC CONSTRAINTS FOR GRADIENT LEARNING

The straightforward linear soft constraints used by the classic Lukasiewicz relaxation fail to pass back gradients with a magnitude and instead pass back a direction (e.g., ± 1). Formally, the gradient of a potential $\phi_{\mathbf{w}}(\mathbf{x}) = \max(0, \ell(p_{\mathbf{w}}(\mathbf{x}), \mathbf{x}))$ with respect to \mathbf{w} is:

$$\frac{\partial}{\partial \mathbf{w}} \phi_{\mathbf{w}} = \frac{\partial}{\partial \mathbf{w}} \ell(p_{\mathbf{w}}, \mathbf{x}) \cdot 1_{\phi_{\mathbf{w}} > 0} = \left[\frac{\partial}{\partial p_{\mathbf{w}}} \ell(p_{\mathbf{w}}, \mathbf{x}) \right] \cdot \frac{\partial}{\partial \mathbf{w}} p_{\mathbf{w}} \cdot 1_{\phi_{\mathbf{w}} > 0} \quad (7)$$

Here $\ell(p_{\mathbf{w}}(\mathbf{x}), \mathbf{x}) = a \cdot p_{\mathbf{w}}(\mathbf{x}) + b$ where $a, b \in \mathbb{R}$ and $p_{\mathbf{w}}(\mathbf{x}) \in [0, 1]$, which leads to the gradient $\frac{\partial}{\partial p_{\mathbf{w}}} \ell(p_{\mathbf{w}}, \mathbf{x}) = a$. Observing the three Lukasiewicz operations described in Section 2, it is clear that a will always result in ± 1 unless there are multiple $p_{\mathbf{w}}(\mathbf{x})$ per constraint. As a result, this classic soft relaxation leads to a naive, non-smooth gradient ($\frac{\partial}{\partial \mathbf{w}} \phi_{\mathbf{w}} = [a 1_{\phi_{\mathbf{w}} > 0}] \cdot \frac{\partial}{\partial \mathbf{w}} p_{\mathbf{w}}$) that is mostly consists of the predictive probability gradient $\frac{\partial}{\partial \mathbf{w}} p_{\mathbf{w}}$. It barely informs the model of the degree to which $p_{\mathbf{w}}$ satisfies the symbolic constraint $\phi_{\mathbf{w}}$ (other than the non-smooth step function $1_{\phi_{\mathbf{w}} > 0}$), thereby creating challenges in gradient-based learning. In this work, we propose a novel log-based relaxation that provides smoother and more informative gradient information for the symbolic constraints:

$$\psi_{\mathbf{w}}(\mathbf{x}) = \log(\phi_{\mathbf{w}}(\mathbf{x})) = \log(\max(0, \ell(p_{\mathbf{w}}(\mathbf{x}), \mathbf{x}))) \quad (8)$$

This seemingly simple transformation brings a non-trivial change to the gradient behavior:

$$\frac{\partial}{\partial \mathbf{w}} \psi_{\mathbf{w}} = \frac{1}{\phi_{\mathbf{w}}(\mathbf{x})} \cdot \frac{\partial}{\partial \mathbf{w}} \phi_{\mathbf{w}} = \left[\frac{a}{\phi_{\mathbf{w}}} 1_{\phi_{\mathbf{w}} > 0} \right] \cdot \frac{\partial}{\partial \mathbf{w}} p_{\mathbf{w}} \quad (9)$$

| Dataset | Setting | Method | Hidden Representation Learning | | Structure Induction (AMI) |
|---------------|-------------------------|------------|-------------------------------------|---|--------------------------------|
| | | | Full (Class-Balanced Accuracy) | Few-Shot (Class-Balanced Accuracy) | |
| MultiWoZ | Standard Generalization | DD-VRNN | 0.804 ± 0.037 | 0.643 ± 0.038 | 0.451 ± 0.042 |
| | | NEUPSL DSI | 0.806 ± 0.051 | 0.689 ± 0.038 | 0.618 ± 0.028 |
| SGD Synthetic | Standard Generalization | DD-VRNN | 0.949 ± 0.005 | 0.598 ± 0.019 | 0.553 ± 0.017 |
| | | NEUPSL DSI | 0.941 ± 0.009 | 0.765 ± 0.012 | 0.826 ± 0.006 |
| SGD Real | Standard Generalization | DD-VRNN | 0.661 ± 0.015 | 0.357 ± 0.015 | 0.448 ± 0.019 |
| | | NEUPSL DSI | 0.663 ± 0.015 | 0.517 ± 0.021 | 0.539 ± 0.048 |
| | Domain Generalization | DD-VRNN | 0.268 ± 0.012 | 0.320 ± 0.029 | 0.476 ± 0.029 |
| | | NEUPSL DSI | 0.299 ± 0.009 | 0.528 ± 0.026 | 0.541 ± 0.036 |
| | Domain Adaptation | DD-VRNN | 0.308 ± 0.011 | 0.505 ± 0.015 | 0.514 ± 0.028 |
| | | NEUPSL DSI | 0.297 ± 0.025 | 0.541 ± 0.023 | 0.559 ± 0.045 |

Table 1: Test set performance on all datasets. All reported results are averaged over 10 splits. The highest-performing methods per dataset and learning setting are bolded.

The gradient now contains $\frac{1}{\phi_{\mathbf{w}}(\mathbf{x})}$, which informs the model of the degree to which the prediction satisfies the symbolic constraint. As a result, when the satisfaction of a rule $\phi_{\mathbf{w}}$ is low (i.e., uncertain), the gradient magnitude will be high, and when the satisfaction of the rule is high, the gradient magnitude will be low. In this way, the gradient of the symbolic constraint guides the neural model to focus on learning the challenging examples that violate the symbolic rules.

4 EXPERIMENTAL EVALUATION

Datasets Experiments are conducted using three goal-oriented dialog datasets: MultiWoZ 2.1 synthetic Campagna et al. (2020) and two versions of the Schema Guided Dialog (SGD) dataset; *SGD-synthetic* (where the utterance is generated by a template-based dialog simulator) and *SGD-real* (which replaces the machine-generated utterances of SGD-synthetic with its human-paraphrased counterparts) (Rastogi et al., 2020). SGD-real dataset is evaluated over three unique settings: *standard generalization* (train and test over the same domain), *domain generalization* (train and test over different domains), and *domain adaptation* (train on (potentially labeled) data from the training domain and unlabeled data from the test domain, and test on evaluation data from the test domain).¹

Constraints In the synthetic MultiWoZ setting, we introduce a set of 11 structural domain agnostic dialog rules. An example of one of these rules can be seen in Equation 1. These rules are introduced to represent general facts about dialogs, with the goal of showing how the incorporation of a few expert-designed rules can drastically improve generalization performance. For SGD settings, a single dialog rule that encodes the concept that dialog acts should contain utterances with correlated tokens is used, e.g., utterances with ‘hello’ are likely to belong to the greet state. This rule demonstrates the boost in performance a model can achieve from a simple source of prior information.¹

Metrics and Methodology We assess the correctness of learned latent dialog structure and the quality of learned hidden representation using Adjusted Mutual Information (AMI) and linear probing, respectively. AMI allows for a comparison between ground truth labels² (e.g., “greet”, etc.) and latent state predictions (e.g., $State_1$, etc.). Linear probing trains a lightweight *probing* model on top of the frozen learned representation and evaluates the linear model’s generalization performance for supervised tasks Tenney et al. (2019). We train both a full supervision and few-shot supervision linear classifier on top of input features extracted from the penultimate layer of the DD-VRNN. Full supervision averages the class-balanced accuracy of two separate models that classify dialog acts (e.g., “greet”, etc.) and domains (“hotel”, etc.), respectively. Few-shot averages the class-balanced accuracy of models classifying dialog acts with one, five, and ten-shot settings.¹

Table 1 summarizes the results of NEUPSL DSI and DD-VRNN in unsupervised settings. NEUPSL DSI outperforms the strictly data-driven DD-VRNN on AMI by 4%-27% depending on the setting while maintaining or improving the hidden representation quality. To reiterate, this improvement is achieved without supervision in the form of labels, but rather a few structural constraints. Comparing AMI performance on SGD-real across different settings (standard generalization v.s. domain generalization/adaptation), we see the NEUPSL DSI consistently improves over DD-VRNN, albeit with the advantage slightly diminished in the non-standard generalization settings.

5 CONCLUSION

This paper introduces NEUPSL DSI, a novel neuro-symbolic learning framework that guides latent dialog structure learning using differentiable symbolic knowledge. Through extensive empirical evaluations, we illustrate how the injection of just a few domain knowledge rules significantly improves both correctness and hidden representation quality in this unsupervised NLP task.

¹Appendix describes further details.

²These labels were only used for final evaluation, not for training or hyperparameter tuning.

REFERENCES

- Stephen Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. Hinge-loss Markov random fields and probabilistic soft logic. *JMLR*, 18(1):1–67, 2017.
- Giovanni Campagna, Agata Foryciarz, Mehrad Moradshahi, and Monica S. Lam. Zero-shot transfer learning with synthesized data for multi-domain dialogue state tracking. In *Association for Computational Linguistics (ACL)*, 2020.
- Ananlada Chotimongkol. *Learning the Structure of Task-Oriented Conversations from the Corpus of In-Domain Dialogs*. PhD thesis, Carnegie Mellon University, Institute for Language Technologies, 2008.
- Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C. Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *Neural Information Processing Systems (NeurIPS)*, 2015.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pp. 4171–4186, Minneapolis, Minnesota, jun 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL <https://aclanthology.org/N19-1423>.
- Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Kumar Goyal, Peter Ku, and Dilek Hakkani-Tür. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *Language Resources and Evaluation Conference (LREC)*, 2019.
- Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997.
- Dan Jurafsky. Switchboard swbd-damsl shallow-discourse-function annotation coders manual. *Institute of Cognitive Science Technical Report*, 1997.
- Mihir Kale and Abhinav Rastogi. Template guided text generation for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6505–6520, Online, nov 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.527. URL <https://aclanthology.org/2020.emnlp-main.527>.
- George J. Klir and Bo Yuan. *Fuzzy Sets and Fuzzy Logic - Theory and Applications*. Prentice Hall, 1995.
- Ian Ritter, Colin Cherry, and Bill Dolan. Unsupervised modeling of twitter conversations. In *Association for Computational Linguistics (ACL)*, 2010.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2014.
- Connor Pryor, Charles Dickens, Eriq Augustine, Alon Albalak, William Wang, and Lise Getoor. Neupsl: Neural probabilistic soft logic. *arXiv*, 2022. URL <https://arxiv.org/abs/2205.14268>.
- Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8689–8696, 2020.
- Weiyan Shi, Tiancheng Zhao, and Zhou Yu. Unsupervised dialog structure learning. In *Association for Computational Linguistics (ACL)*, 2019.
- Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. What do you learn from context? probing for sentence structure in contextualized word representations, 2019. URL <https://arxiv.org/abs/1905.06316>.

Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research (JMLR)*, 11(95):2837–2854, 2010.

Ke Zhai and Jason D. Williams. Discovering latent structure in task-oriented dialogues. In *Association for Computational Linguistics (ACL)*, 2014.

```
# Token Constraint
 $w_1 : \text{HASWORD}(\text{Utt}, \text{Class}) \rightarrow \text{STATE}(\text{Utt}, \text{Class})$ 
```

Figure 3: SGD Structure Induction Constraint Model

A MODEL DETAILS

This section provides additional details on the NEUPSL DSI models for the Multi-WoZ and SGD settings. Throughout these subsections, we cover the symbolic constraints, evaluation metrics, and hyperparameters. The code is under the Apache 2.0 license.

A.1 SGD CONSTRAINTS

The NEUPSL DSI model uses a single constraint for all SGD settings (synthetic, standard, domain generalization, and domain adaptation). Figure 3 provides an overview of the constraint, which contains the following two predicates:

1. **STATE(Utt, Class)**

The STATE continuous valued predicate is the probability that an utterance, identified by the argument `Utt`, belongs to a dialog state, identified by the argument `Class`. For instance, the utterance *hello world !* for the *greet* dialog state would create a predicate with a value between zero and one, i.e., $\text{STATE}(\text{hello world !}, \text{greet}) = 0.7$.

2. **HASWORD(Utt, Class)**

The HASWORD binary predicate indicates if an utterance, identified by the argument `Utt`, contains a known token for a particular class, identified by the argument `Class`. For instance if a known token associated with the *greet* class is *hello*, then the utterance *hello world !* would create a predicate with value one, i.e. $\text{HASWORD}(\text{hello world !}, \text{greet}) = 1$.

This token constraint encodes the prior knowledge that utterances’ are likely to belong to dialog states when an utterance contains tokens representing that state. For example, if a known token associated with the *greet* class is *hello*, then the utterance *hello world !* is likely to belong to the *greet* state. The primary purpose of incorporating this constraint into the model is to show how even a small amount of prior knowledge can aid predictions. To get the set of tokens associated with each state, we trained a supervised linear classifier where the input is an utterance, and the label is the class. After training, every token is individually run through the trained model to get a set of logits over each class. These logits represent the relative importance that each token has over every class. Sparsity is introduced to this set of logits, leaving only the top 0.1% of values and replacing the others with zeros. This sparsity reduces the set of 261,651 logits to 262 non-zero logits.

A.2 MULTI-WoZ CONSTRAINTS

The NEUPSL DSI model for the Multi-WoZ setting uses a set of dialog constraints, which can be broken into dialog start, middle, and end. Figure 4 provides an overview of the constraints, which contains the following 11 predicates:

1. **STATE(Utt, Class)**

The STATE continuous valued predicate is the probability that an utterance, identified by the argument `Utt`, belongs to a dialog state, identified by the argument `Class`. For instance, the utterance *hello world !* for the *greet* dialog state would create a predicate with a value between zero and one, i.e., $\text{STATE}(\text{hello world !}, \text{greet}) = 0.7$.

2. **FIRSTUTT(Utt)**

The FIRSTUTT binary predicate indicates if an utterance, identified by the argument `Utt`, is the first utterance in a dialog.

```

# Dialog Start
w1 : ¬FIRSTUTT(Utt) → ¬STATE(Utt, greet)
w2 : FIRSTUTT(Utt) ∧ HASGREETWORD(Utt) → STATE(Utt, greet)
w3 : FIRSTUTT(Utt) ∧ ¬HASGREETWORD(Utt) → STATE(Utt, init_request)

# Dialog Middle
w4 : PREVUTT(Utt1, Utt2) ∧ STATE(Utt2, greet) → STATE(Utt1, init_request)
w5 : PREVUTT(Utt1, Utt2) ∧ ¬STATE(Utt2, greet) → ¬STATE(Utt1, init_request)
w6 : PREVUTT(Utt1, Utt2) ∧ STATE(Utt2, init_request) → STATE(Utt1, second_request)
w7 : PREVUTT(Utt1, Utt2) ∧ STATE(Utt2, second_request) ∧ HASINFOQUESTIONWORD(Utt1) → STATE(Utt1, info_question)
w8 : PREVUTT(Utt1, Utt2) ∧ STATE(Utt2, second_request) ∧ HASLOTQUESTIONWORD(Utt1) → STATE(Utt1, slot_question)
w9 : PREVUTT(Utt1, Utt2) ∧ STATE(Utt2, end) ∧ HASCANCELWORD(Utt1) → STATE(Utt1, cancel)

# Dialog End
w10 : LASTUTT(Utt) ∧ HASENDWORD(Utt) → STATE(Utt, end)
w11 : LASTUTT(Utt) ∧ HASACCEPTWORD(Utt) → STATE(Utt, accept)
w12 : LASTUTT(Utt) ∧ HASINSISTWORD(Utt) → STATE(Utt, insist)

```

Figure 4: MultiWoZ Structure Induction Constraint Model

3. **LASTUTT(Utt)**
The LASTUTT binary predicate indicates if an utterance, identified by the argument Utt, is the last utterance in a dialog.
4. **PREVUTT(Utt1, Utt2)**
The PREVUTT binary predicate indicates if an utterance, identified by the argument Utt2, is the previous utterance in a dialog of another utterance, identified by the argument Utt1.
5. **HASGREETWORD(Utt)**
The HASGREETWORD binary predicate indicates if an utterance, identified by the argument Utt, contains a known token for the greet class. The list of known greet words is ['hello', 'hi'].
6. **HASINFOQUESTIONWORD(Utt)**
The HASINFOQUESTIONWORD binary predicate indicates if an utterance, identified by the argument Utt, contains a known token for the info question class. The list of known info question words is ['address', 'phone'].
7. **HASLOTQUESTIONWORD(Utt)**
The HASLOTQUESTIONWORD binary predicate indicates if an utterance, identified by the argument Utt, contains a known token for the slot question class. The list of known slot question words is ['what', '?'].
8. **HASINSISTWORD(Utt)**
The HASINSISTWORD binary predicate indicates if an utterance, identified by the argument Utt, contains a known token for the insist class. The list of known insist words is ['sure', 'no'].
9. **HASCANCELWORD(Utt)**
The HASCANCELWORD binary predicate indicates if an utterance, identified by the argument Utt, contains a known token for the cancel class. The list of known cancel words is ['no'].
10. **HASACCEPTWORD(Utt)**
The HASACCEPTWORD binary predicate indicates if an utterance, identified by the argument Utt, contains a known token for the accept class. The list of known accept words is ['yes', 'great'].
11. **HASENDWORD(Utt)**
The HASENDWORD binary predicate indicates if an utterance, identified by the argument Utt, contains a known token for the end class. The list of known end words is ['thank', 'thanks'].

The dialog start constraints take advantage of the inherent structure built into the beginning of task-oriented dialogs. In the same order as the dialog start rules in Figure 4: 1) If the first turn utterance does not contain a known greet word, then it does not belong to the *greet* state. 2) If the first turn utterance contains a known greet word, then it belongs to the *greet* state. 3) If the first turn utterance does not contain a known greet word, then it belongs to the *initial request* state.

The dialog middle constraints exploit the temporal dependencies within the middle of a dialog. In the same order as the dialog middle rules in Figure 4: 1) If the previous utterance belongs to the *greet* state, then the current utterance belongs to the *initial request* state. 2) If the previous utterance does not belong to the *greet* state, then the current utterance does not belong to the *initial request* state. 3) If the previous utterance belongs to the *initial request* state, then the current utterance belongs to the *second request* state. 4) If the previous utterance belongs to the *second request* state and it has a known info question token, then the current utterance belongs to the *info question* state. 5) If the previous utterance belongs to the *second request* state and it has a known slot question token, then the current utterance belongs to the *slot question* state. 4) If the previous utterance belongs to the *end* state and it has a known cancel token, then the current utterance belongs to the *cancel* state.

The dialog end constraints take advantage of the inherent structure built into the end of task-oriented dialogs. In the same order as the dialog end rules in Figure 4: 1) If the last turn utterance contains a known end word, then it belongs to the *end* state. 2) If the last turn utterance contains a known accept word, then it belongs to the *accept* state. 3) If the last turn utterance contains a known insist word, then it belongs to the *insist* state.

A.3 EVALUATION METRICS

Adjusted Mutual Information (AMI) evaluates dialog structure prediction by evaluating the correctness of the dialog state assignments. Let $U^* = \{U_1^*, \dots, U_C^*\}$ be the ground-truth assignment of dialog states for all utterances in the corpus, and $U = \{U_1, \dots, U_C\}$ be the predicted assignment of dialog states based on the learned dialog structure model. U^* and U are not directly comparable because they draw from different base sets of states (U^* from the ground truth set of states and U from the collection of states induced by the DD-VRNN) that may even have different cardinalities. We address this problem using Adjusted Mutual Information (AMI), a metric developed initially to compare unsupervised clustering algorithms. Intuitively, AMI treats each assignment as a probability distribution over states and uses Mutual Information to measure their similarity, adjusting for the fact that larger clusters tend to have higher MI. AMI is defined as follows:

$$AMI(U, U^*) = \frac{MI(U, U^*) - \mathbb{E}(MI(U, U^*))}{Avg(H(U), H(U^*)) - \mathbb{E}(MI(U, U^*))}$$

Where $MI(U, U^*)$ is the mutual information score, $\mathbb{E}(MI(U, U^*))$ is the expected mutual information over all possible assignments, and $Avg(H(U), H(U^*))$ is the average entropy of the two clusters (Vinh et al., 2010).

A.4 HYPERPARAMETERS

The DD-VRNN uses an LSTM Hochreiter & Schmidhuber (1997) with 200-400 units for the RNNs, and fully-connected highly flexible feature extraction functions with a dropout of 0.4 for the input x , the latent vector z , the prior, the encoder and the decoder. The input to the DD-VRNN is the utterances with a 300-dimension word embedding created using a GloVe embedding Pennington et al. (2014) and a Bert embedding (Devlin et al., 2019). The maximum utterance word length was set to 40, the maximum length of a dialog was set to 10, and the tunable weight, γ (Equation ??), was set to 0.1. The total number of parameters is 26,033,659 for the model with GloVe embedding and 135,368,227 with Bert embedding. The experiments are run in Google TPU V4, and the total GPU hours for all finetuning are 326 GPU hours.

B DATASETS

This section provides additional information on the SGD, SGD synthetic, and MultiWoZ 2.1 synthetic datasets.

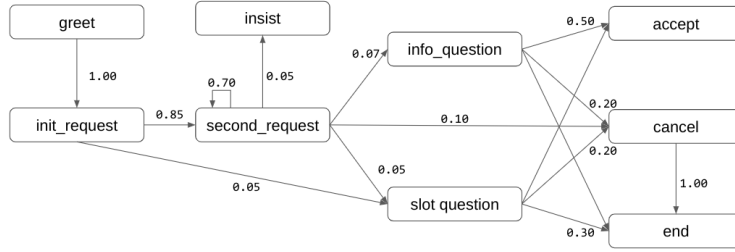


Figure 5: Ground truth dialog structure used to generate the MultiWoZ 2.1 dataset. The transition graph shows transitions over 0.05%.

B.1 SGD

The Schema-Guided Dialog (SGD) Rastogi et al. (2020) is a task-oriented conversation dataset involving interactions with services and APIs covering 20 domains. There are overlapping functionalities over many APIs, but their interfaces differ. One conversion may involve multiple domains. The train set contains conversions from 16 domains, with four held-out domains only present in test sets. This gives 34,308 in-domain and 5,441 out-of-domain test examples. To evaluate the model’s generalization, we evaluate the model performance on both test sets. In specific, we establish three different evaluation protocols.

- **SGD Standard Generalization** We train the model using the SGD train set and evaluate it on the in-domain test set.
- **SGD Domain Generalization** We train the model using the SGD train set and evaluate it on the out-of-domain test set.
- **SGD Domain Adaptation** We train the model using the SGD train set and label-wiped in-domain and out-of-domain test sets and evaluate it on the out-of-domain test set.

B.2 SGD SYNTHETIC

Using the template-based generator from the SGD developers Kale & Rastogi (2020), we generate 10,800 synthetic dialogs using the same APIs and dialog states as the official SGD data. We split the examples with 75% train and 25% test. The schema-guided generator code is under Apache 2.0 license: <https://github.com/google-research/task-oriented-dialogue/blob/main/LICENSE>.

B.3 MULITWoZ 2.1 SYNTHETIC

MultiWoZ 2.1 synthetic Campagna et al. (2020) is a multi-domain goal-oriented dataset covering five domains (Attraction, Hotel, Restaurant, Taxi, and Train) and nine dialog acts (*greet*, *initial request*, *second request*, *insist*, *info question*, *slot question*, *accept*, *cancel*, and *end*). Following Campagna et al. (2020), we generate 10^4 synthetic dialogs from a known ground-truth dialog structure. Figure 5 provides an overview of the ground truth dialog structure, which is based on the original MultiWoz 2.1 dataset Eric et al. (2019), used through the generative process. These 10^4 synthetic dialogs are randomly sampled without replacement to create ten splits with 80% train, 10% test, and 10% validation. The MultiWoZ 2.1 synthetic code is under the MIT License: <https://github.com/stanford-oval/zero-shot-multiwoz-acl2020>. The MultiWoZ 2.1 code uses genie under the MIT License: <https://github.com/stanford-oval/genie-k8s/blob/master/LICENSE>.