Feedback Attribution for Counterfactual Bandit Learning in Multi-Domain Spoken Language Understanding

Tobias Falke
Amazon Alexa AI
Berlin, Germany
falket@amazon.com

Patrick Lehnen
Amazon Alexa AI
Aachen, Germany
plehnen@amazon.com

Abstract

With counterfactual bandit learning, models can be trained based on positive and negative feedback received for historical predictions, with no labeled data needed. Such feedback is often available in real-world dialog systems, however, the modularized architecture commonly used in large-scale systems prevents the direct application of such algorithms. In this paper, we study the feedback attribution problem that arises when using counterfactual bandit learning for multi-domain spoken language understanding. We introduce an experimental setup to simulate the problem on small-scale public datasets, propose attribution methods inspired by multi-agent reinforcement learning and evaluate them against multiple baselines. We find that while directly using overall feedback leads to disastrous performance, our proposed attribution methods can allow training competitive models from user feedback.

1 Introduction

Spoken language understanding (SLU) is a key component of task-oriented dialog systems (Tur and De Mori, 2011). It is commonly modeled as two tasks: Intent classification (IC), which assigns an intent to an utterance, and slot labeling (SL), which recognizes boundaries and types of slots in the utterance's tokens. In recent years, neural models that jointly learn both tasks, in combination with pre-trained transformers (Chen et al., 2019; Zhang et al., 2019), have become the state-of-the-art approach to SLU (Louvan and Magnini, 2020; Weld et al., 2021). However, a sufficient amount of manually labeled data is needed for fine-tuning.

If dialog systems are actively used, a cost-efficient alternative to manually labeled data is to leverage user feedback, which can appear explicitly (e.g. "no stop", "thank you") and implicitly (e.g. retries, interruptions). Such positive and negative feedback in response to a prediction is known as *bandit feedback*. Compared to ground truth labels it is less informative, as negative feedback does not reveal which prediction would have been better, but on the other hand, it is available in large quantities without manual effort. Recent work proposed multiple possible ways in which such feedback can be used to improve dialog systems (Muralidharan et al., 2019; Ponnusamy et al., 2020; Kim and Kim, 2020; Falke et al., 2020). In this work, we focus on leveraging it via *counterfactual bandit learning*, a well-studied approach for training models with historical bandit feedback (Langford et al., 2008; Dudík et al., 2011; Joachims et al., 2018).

The key challenge addressed in this paper is the *feedback attribution* problem arising in large-scale multi-domain SLU systems (see Figure 1). In such systems, a common architecture is to combine domain-specific IC and SL models with a domain classifier (DC) (Jeong and Lee, 2009; Xu and Sarikaya, 2014; Hakkani-Tür et al., 2016), which allows to more easily update domain-specific behavior regularly without being bottlenecked by a single joint model. However, for the bandit learning setting, this introduces additional challenges: Given negative feedback for a combined prediction, it is unclear which individual model caused the error, and thus how to use the feedback for training. If it is given directly to all models, already correct predictions will be penalized. A three-fold attribution problem, consisting of *task-level*, *domain-level* and *token-level* attribution (see Section 3.2), needs to be solved to use the feedback effectively.

In this paper, we make several contributions to address the feedback attribution problem: First, we propose an experimental setup based on SNIPS (Coucke et al., 2018) and TOP (Schuster et al., 2019) to simulate the problem on small-scale public datasets. Second, we propose first attribution methods inspired by multi-agent reinforcement learning. And third, we evaluate them against multiple baselines on the two datasets. We find that while using the overall feedback without attribution leads to dis-

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 1190–1198
November 7–11, 2021.

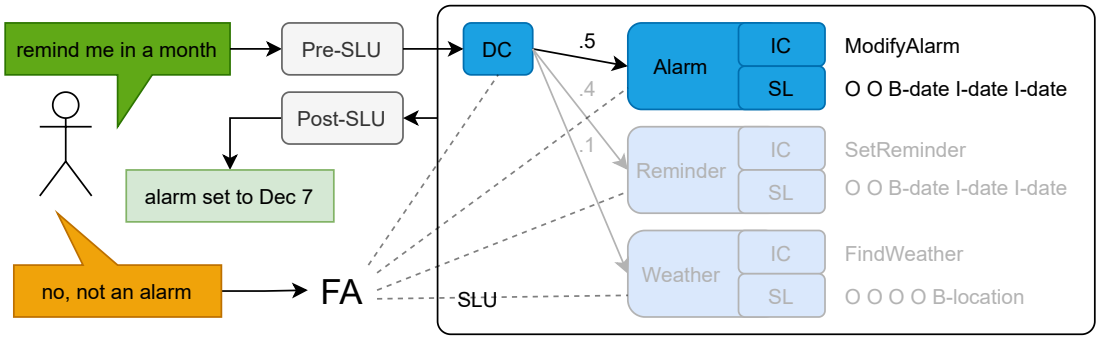

Figure 1: In modularized multi-domain SLU systems, a domain classifier (DC) is combined with multiple domain-specific intent classification (IC) and slot labeling (SL) models to interpret an utterance. Given utterance-level negative user feedback, it needs to be attributed (FA) to the individual models to effectively use it for training.

astrous performance, the proposed attribution methods allow learning competitive models if the logged feedback data contains sufficient exploration.

2 Counterfactual Bandit Learning

In counterfactual bandit learning, we have access to a dataset of n tuples of bandit feedback $(x_i, y_i, p_i, \delta_i)$ collected by a logging policy[1] π_0. In each tuple, y_i is the prediction made by π_0 for input x_i with probability $p_i = \pi_0(y_i|x_i)$, called propensity, and δ_i is the feedback received for that prediction. We assume $\delta_i \in \mathbb{R}$ and higher is better. The goal is to find a new policy π that maximizes the expected feedback of that policy:

$$R(\pi) = \mathop{\mathbb{E}}_{x\sim P(X)} \mathop{\mathbb{E}}_{y\sim\pi(Y|x)} \delta(x, y) \qquad (1)$$

Given the bandit dataset, we can estimate $R(\pi)$ via importance sampling with the inverse propensity scaling (IPS) estimator (Langford et al., 2008):

$$\hat{R}_{IPS}(\pi) = \frac{1}{n}\sum_{i=1}^{n} \delta_i \frac{\pi(y_i|x_i)}{p_i} \qquad (2)$$

Note that Eq. 2 relies only on the logged prediction y_i for which feedback was given. This makes it possible to use the partial bandit feedback directly to train a classifier. It is different from supervised learning, which instead requires access to the ground truth label y_i^* (see Eq. 3).

Several improvements to the IPS estimator have been proposed to learn more effectively, including adding self-normalization (Swaminathan and Joachims, 2015; Joachims et al., 2018) or by combining IPS with a direct feedback estimation models (Dudík et al., 2011; Wang et al., 2019). We rely on vanilla IPS for our experiments, but make no assumptions that would prevent using a more advanced bandit learning method instead.

A crucial assumption of most counterfactual bandit learning methods is that the logging policy is stochastic, i.e. feedback is for predictions that were sampled from π_0 and thus cover different labels for the same input. However, real-world systems, as Lawrence et al. (2017) points out, typically use argmax predictions to deliver the best possible user experience in each case, which limits the amount of exploration that is present in historical data. In their work, they find that counterfactual learning can be possible even with fully deterministic logging policies (Lawrence et al., 2017; Lawrence and Riezler, 2018). For the case of multi-domain SLU, we address that question in this work.

3 Application to Multi-Domain SLU

The goal of SLU is to predict, given an utterance x with T tokens, its domain y^D, intent y^I and token-wise slot labels $(y_t^S)_{t=1}^T$. In the modularized multi-domain setup considered here, such predictions are produced by a DC model π_D and domain-specific IC and SL models $\pi_{I,d}$ and $\pi_{S,d}$ (see Figure 1). At inference time, the IC and SL predictions of the top domain predicted by DC are used as the prediction.

3.1 Supervised Training

Given a dataset of examples with ground truth labels $D_S = \{(x_i, y_i^{*D}, y_i^{*I}, (y_{i,t}^{*S})_{t=1}^{T_i}\}_{i=1}^n$, the models in the multi-domain setting can be trained as follows: For the DC model, all examples of D_S are used, and for each domain-specific IC or SL model, the subset of examples of that domain, defined by y_i^{*D}, is used. The standard loss to train $\pi_D, \pi_{I,d}$ and $\pi_{S,d}$ with such data is cross entropy, applied at

[1] We use policy and model interchangeably in this paper.

the token level in the case of $\pi_{S,d}$:[2]

$$\mathcal{L}_{CE}(\pi) = -\frac{1}{n}\sum_{i=1}^{n}\log(\pi(y_i^*|x_i)) \quad (3)$$

3.2 Bandit Training and Attribution

In the bandit setting, we have instead a dataset $D_B = \{(x_i, y_i^D, y_i^I, (y_{i,t}^S)_{t=1}^{T_i}, \delta_i\}_{i=1}^n$ with overall feedback δ_i (omitting logging propensities for readability). It can be used to train the individual models $\pi_D, \pi_{I,d}$ and $\pi_{S,d}$ by minimizing the negative IPS estimate of Eq. 2 (at the token level for $\pi_{S,d}$):

$$\mathcal{L}_{IPS}(\pi) = -\hat{R}_{IPS}(\pi) \quad (4)$$

However, since δ_i is overall feedback, using it directly is problematic: If only some parts of the predicted labels are wrong, which is typically the case, using the negative feedback directly to train all models via Eq. 4 is detrimental for the already correct predictions. Three attribution challenges arise: Feedback should be attributed to the responsible model (*task-level attribution*) and sub-prediction (*token-level attribution* for SL). In addition, it is unclear which examples to use to train domain-specific IC/SL models, as the true domain is unknown (*domain-level attribution*). To cope with these challenges, we aim to attribute by mapping δ_i to fine-grained feedback δ_i^D, δ_i^I and $(\delta_{i,t}^S)_{t=1}^{T_i}$.

3.3 Attribution Methods

We propose and evaluate two methods for fine-grained feedback attribution:

Propensity-based FA The first method relies on the propensities of the logging policy, following the idea that the policy might be self-aware of some mistakes and reflects them in the propensities. Given overall feedback δ_i and propensities p_i^D, p_i^I and $(p_{i,t}^S)_{t=1}^{T_i}$, we attribute according to

$$\delta_i^c = \delta_i \frac{1 - p_i^c}{3 - p_i^D - p_i^I - p_i^S} \quad c \in \{D, I, S\} \quad (5)$$

$$\delta_{i,t}^S = \delta_i^S \frac{1 - p_{i,t}^S}{\sum\left(1 - p_{i,t'}^S\right)_{t'=1}^{T_i}} \quad t \in [1, T_i] \quad (6)$$

with p_i^S being the average of token-level propensities. This distributes the feedback proportional to the uncertainty reflected in the propensities.

[2]For SL, a CRF loss is an alternative to token-level cross entropy, but is not necessarily superior (Chen et al., 2019).

Advantage-based FA For the second method, we follow the credit assignment idea of COMA (Foerster et al., 2018, Eq. 4), a method for reinforcement learning in a multi-agent setting. It uses an advantage function for each agent that subtracts a baseline estimated based on a joint Q-function.

In our setup, this idea translates to using an *overall feedback estimator* $f(x, y^D, y^I, (y_t^S)) \rightarrow \delta$ and using it in an advantage function as follows for DC:

$$\delta_i^D = f(x_i, y_i^D, y_i^I, (y_{i,t}^S)_{t=1}^{T_i}) - \sum_{y' \in Y_D} w_{y'} \cdot f(x_i, y', y_i^I, (y_{i,t}^S)_{t=1}^{T_i}) \quad (7)$$

The second part of Eq. 7 computes the average estimated feedback across a set of alternative domain predictions Y_D by replacing the original y_i^D with the alternatives (and leaving the IC and SL predictions constant). By subtracting this baseline from the estimate for the originally predicted domain (first part of Eq. 7), we obtain the DC-specific feedback δ_i^D. If the original prediction was truly better or worse than the alternatives, that difference will be larger and can serve as a corresponding model-specific feedback signal. If it made no difference however, δ_i^D will be close to zero and the attribution thereby reflects this irrelevance.

We use variations of Eq. 7 to also attribute to IC and SL: While for DC, alternatives Y_D in Eq. 7 are all domains, we use for IC and SL the domain-specific label space of intents and slots. For SL, the equation is applied per token and we restrict alternatives Y_S to the two highest-propensity labels per token to limit computational complexity. We test different weights $w_{y'}$ (see Section 4).

Both FA methods are only applied if δ_i is negative feedback, as positive feedback does not require attribution. And finally, to address domain-level attribution, we only use those examples that have positive δ_i^D after FA for IC and SL, as for other examples it is unclear what domain to use them in.

4 Experiments

4.1 Setup

We use two English SLU benchmarks, SNIPS (Coucke et al., 2018) and the English part of multilingual TOP (Schuster et al., 2019). TOP has 34.7k train/dev and 8.6 test examples spanning 12 intents that are already grouped into 3 domains (*Alarms, Reminders, Weather*). For SNIPS, which has 13.8k

Method	SNIPS				TOP			
	DC-Acc	IC-Acc	SL-F1	Sem-Acc	DC-Acc	IC-Acc	SL-F1	Sem-Acc
Log. Policy	$97.48_{\pm 0.6}$	$96.97_{\pm 0.5}$	$79.14_{\pm 1.1}$	$57.87_{\pm 2.3}$	$99.57_{\pm 0.1}$	$97.30_{\pm 0.3}$	$87.36_{\pm 3.6}$	$76.07_{\pm 4.4}$
Bandit Learning, feedback for argmax prediction								
Overall	$76.70_{\pm 4.2}$	$60.01_{\pm 4.9}$	$39.06_{\pm 12.0}$	$49.60_{\pm 4.3}$	$92.29_{\pm 6.4}$	$79.40_{\pm 5.5}$	$62.49_{\pm 9.5}$	$74.78_{\pm 5.2}$
Positive	$\mathbf{98.66}_{\pm 0.2}$	$\mathbf{98.22}_{\pm 0.3}$	$\mathbf{89.51}_{\pm 0.5}$	$\mathbf{78.14}_{\pm 0.9}$	$\mathbf{99.88}_{\pm 0.0}$	$\mathbf{98.55}_{\pm 0.2}$	$\mathbf{93.48}_{\pm 0.7}$	$\mathbf{87.14}_{\pm 1.0}$
Propensity	$94.57_{\pm 3.0}$	$94.14_{\pm 3.0}$	$86.16_{\pm 1.6}$	$75.38_{\pm 1.8}$	$99.80_{\pm 0.1}$	$97.03_{\pm 3.6}$	$91.01_{\pm 5.4}$	$86.11_{\pm 3.6}$
Prop-All	$94.57_{\pm 3.0}$	$91.87_{\pm 3.1}$	$75.79_{\pm 6.5}$	$69.90_{\pm 3.8}$	$99.80_{\pm 0.1}$	$94.59_{\pm 5.2}$	$85.43_{\pm 6.0}$	$83.74_{\pm 4.4}$
Adva-Uni	$98.37_{\pm 0.6}$	$97.96_{\pm 0.8}$	$88.73_{\pm 1.6}$	$77.47_{\pm 1.9}$	$99.86_{\pm 0.0}$	$98.47_{\pm 0.4}$	$93.25_{\pm 1.1}$	$86.97_{\pm 1.4}$
Adva-Prop	$98.41_{\pm 0.7}$	$98.07_{\pm 0.6}$	$88.37_{\pm 0.8}$	$77.13_{\pm 1.4}$	$99.86_{\pm 0.1}$	$98.03_{\pm 1.9}$	$92.56_{\pm 3.1}$	$86.60_{\pm 2.9}$
Oracle	$99.36_{\pm 0.2}$	$99.10_{\pm 0.2}$	$95.43_{\pm 0.6}$	$89.09_{\pm 1.4}$	$99.93_{\pm 0.0}$	$99.04_{\pm 0.1}$	$95.41_{\pm 0.3}$	$91.06_{\pm 0.2}$
Bandit Learning, feedback for 5 sampled predictions								
Overall	$61.06_{\pm 7.7}$	$50.52_{\pm 5.1}$	$33.07_{\pm 10.6}$	$45.73_{\pm 4.2}$	$84.25_{\pm 8.1}$	$73.05_{\pm 7.6}$	$56.35_{\pm 14.9}$	$69.51_{\pm 7.4}$
Positive	$98.76_{\pm 0.4}$	$98.46_{\pm 0.4}$	$\mathbf{91.44}_{\pm 0.5}$	$\mathbf{82.53}_{\pm 0.8}$	$\mathbf{99.90}_{\pm 0.0}$	$98.84_{\pm 0.2}$	$94.66_{\pm 0.5}$	$89.51_{\pm 0.7}$
Propensity	$98.46_{\pm 0.2}$	$98.11_{\pm 0.3}$	$89.25_{\pm 1.2}$	$78.99_{\pm 2.0}$	$99.71_{\pm 0.3}$	$98.36_{\pm 0.5}$	$92.93_{\pm 2.2}$	$88.17_{\pm 1.7}$
Prop-All	$98.46_{\pm 0.2}$	$97.03_{\pm 1.0}$	$83.49_{\pm 3.2}$	$74.27_{\pm 1.7}$	$99.71_{\pm 0.3}$	$96.51_{\pm 1.5}$	$86.71_{\pm 4.0}$	$85.67_{\pm 1.5}$
Adva-Uni	$\mathbf{98.89}_{\pm 0.3}$	$\mathbf{98.57}_{\pm 0.3}$	$91.09_{\pm 0.7}$	$81.77_{\pm 1.1}$	$99.89_{\pm 0.0}$	$98.79_{\pm 0.3}$	$\mathbf{94.72}_{\pm 0.3}$	$\mathbf{89.55}_{\pm 0.5}$
Adva-Prop	$98.77_{\pm 0.3}$	$98.06_{\pm 0.7}$	$87.31_{\pm 1.5}$	$76.87_{\pm 2.0}$	$99.88_{\pm 0.0}$	$\mathbf{98.86}_{\pm 0.2}$	$94.19_{\pm 0.6}$	$89.33_{\pm 1.0}$
Oracle	$99.34_{\pm 0.2}$	$99.34_{\pm 0.2}$	$97.85_{\pm 0.4}$	$94.53_{\pm 0.7}$	$99.94_{\pm 0.0}$	$99.21_{\pm 0.0}$	$96.08_{\pm 0.1}$	$92.02_{\pm 0.2}$

Table 1: Test set performance for bandit learning with different FA methods (mean and std. dev. over 10 runs).

train/dev and 700 test examples, we introduce domains by grouping the 7 intents into 3 domains.[3]

We split the train/dev part of each dataset into *5% labeled data* (D_S), on which the logging policy π_0 is trained with full supervision, and use the rest to create *bandit data* D_B. For each utterance x_i in D_B, we take either the argmax prediction or sample from the predicted class distribution of π_0 and determine (perfect) overall feedback δ_i by comparing the prediction to the ground truth. If any of the domain, intent or slot labels are incorrect, we set $\delta_i = -1$, if not $\delta_i = 1$.

We compare seven methods to attribute δ_i:

- **Overall** No attribution, δ_i is used directly.
- **Positive** Only examples with positive feedback are used, which requires no attribution.
- **Propensity** Propensity-based FA following Eq. 5 and 6. Note that the method can change the magnitude of feedback per component, but it will always stay negative as the sign cannot change in the computation. As IC/SL models use only data with $\delta_i^D > 0$, they will use just positive feedback. We thus also include a variation **Prop-All** that instead uses all examples for IC/SL, at the risk of giving noisier signals.
- **Adva** Advantage-based FA following Eq. 7. The feedback estimator is trained on D_B minimizing mean squared error. We evaluate two variations, **Adva-Uni** using uniform weights $w_{y'}$ and **Adva-Prop** using π_0's propensities.
- **Oracle** Perfect attribution by setting -1 or 1 based on ground truth labels (upper bound).

Note that all methods use positive feedback right away and attribute only negative feedback (except for Positive which completely drops the negative feedback examples).

In our multi-domain setup, we train a DC and three IC/SL models. We rely on DistilBERT (Sanh et al., 2019) and fine-tune it with an utterance-level classifier for DC or, for IC/SL, with jointly trained utterance and token-level classifiers similar to Chen et al. (2019). In each case, we use a 256-d hidden ReLU layer. For π_0, all four models are trained with cross entropy (Eq. 3) on D_S. For bandit learning, we train the same models, using π_0 as initial weights, but use IPS loss (Eq. 4) and D_B. The overall feedback estimator f is trained with mean squared error on D_B. Additional details and hyperparameters, tuned per FA method, can be found in the Appendix. Evaluation is done on the standard test sets and we report accuracy for DC and IC, slot-based F1 for SL and overall accuracy (Sem-Acc), all averaged over 10 runs.

4.2 Results

Table 1 shows results for both datasets and feedback setups. We make the following observations:

[3] *(AddToPlaylist, PlayMusic), (RateBook, SearchCreativeWork), (BookRestaurant, SearchScreeningEvent, GetWeather).*

Multi-domain SLU can be learned from bandit feedback. The initial logging policy, trained with only 5% of the usual data, leaves substantial room for improvement on both datasets, especially in SL. All models trained with counterfactual bandit learning, except for the naive Overall baseline, effectively use the bandit feedback to improve over the logging policy. With perfect feedback assignment (Oracle), bandit learning can yield models that come even close to the performance of models trained supervised on 100% of the labels.

Feedback attribution is crucial. As the Overall baseline illustrates, using the overall feedback directly without any attribution hurts the models and decreases the performance below the logging policy, confirming the need to address this challenge. The perfect attribution (Oracle) as the upper bound on the other hand shows how well the learning can work with properly attributed feedback, resulting in large improvements over Overall. The assignment methods that we evaluated are capable of using some of that potential, but also still leave room to improve performance with better attribution.

Advantage-based FA beats propensity-based FA. The propensity-based approach does poorly compared to other methods, especially Prop-All. One shortcoming is that it is not equipped to solve the domain-level attribution problem and thus cannot use feedback effectively for IC/SL, but, more importantly, another shortcoming is that the attribution relies completely on the model's self-awareness of errors and does not leverage the feedback itself for attribution. The advantage-based methods leverage the feedback via the estimation model and show better performance, in particular the uniform-weighting variant.

Exploration is crucial. Side-stepping the attribution problem by not leveraging the negative feedback at all (Positive) shows strong results in both experimental settings, the one where feedback for only the argmax prediction is available and the one with feedback for 5 samples. Only in the 5-sample-setting, where the feedback estimator can be more effectively trained as the data contains some amount of exploration, Adva-Uni can beat Positive in some cases, demonstrating the potential of additionally using negative feedback. We would like to emphasize that even in this case the exploration in the dataset is limited, as the sampled predictions are still equal to the argmax in most cases (for DC/IC/SL on SNIPS 98.5%/98.2%/92.8% and TOP 99.7%/98.7%/96.9% of examples), and the setting is thus still far from the supervised setting where feedback for all possible predictions is known. We expect further improvements when using bandit data with higher exploration.

5 Conclusion

In this paper, we studied feedback attribution for bandit learning in multi-domain SLU. We proposed multiple attribution methods together with an evaluation setup and found that attribution is crucial to learning. Advantage-based FA helps if historical exploration is available. In future work, we plan to extend the setup to use true user feedback, which can be noisy and inconsistent, and to include components beyond SLU in the attribution scope.

Ethical Considerations

In this work, we study training SLU models directly from user feedback via counterfactual bandit learning. While making the model training more cost-efficient, this also has the benefit that the system behavior can be more in line with user expectations and as a result improve the user experience. However, optimizing models directly towards user feedback can also have negative consequences: The optimization could emphasize the feedback given by the largest user groups and thereby amplify model biases and undesired system behavior for minority groups. In addition, some users might even adapt their behavior to influence the system directly to their advantage or the disadvantage of other users. To mitigate such risks, we propose to verify the model performance on carefully curated offline test sets before deploying trained models and to try to filter out problematic user behavior from bandit data before training. As counterfactual bandit learning is fully offline, such measures can be implemented easily compared to the more challenging online learning setting.

Acknowledgements

We thank Quynh Do, Judith Gaspers, Daniil Sorokin and our anonymous reviewers for their thoughtful comments and suggestions that improved this paper. We would also like to thank Saleh Soltan for providing pretrained models and the DeepNLU team for making the experiments reported in this work easy to run.

References

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. *arXiv*, 1902.10909.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips Voice Platform: An Embedded Spoken Language Understanding System for Private-By-Design Voice Interfaces. *arXiv*, 1805.10190.

Miroslav Dudík, John Langford, and Lihong Li. 2011. Doubly robust policy evaluation and learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, page 1097–1104, Bellevue, WA, USA.

Tobias Falke, Markus Boese, Daniil Sorokin, Caglar Tirkaz, and Patrick Lehnen. 2020. Leveraging user paraphrasing behavior in dialog systems to automatically collect annotations for long-tail utterances. In *Proceedings of the 28th International Conference on Computational Linguistics: Industry Track*, pages 21–32, Online. International Committee on Computational Linguistics.

Jakob N. Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and S. Whiteson. 2018. Counterfactual multi-agent policy gradients. In *Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, LA, USA.

Dilek Hakkani-Tür, Gökhan Tür, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi-Domain Joint Semantic Frame Parsing Using Bi-Directional RNN-LSTM. In *Interspeech*, pages 715–719, San Francisco, CA, USA.

Minwoo Jeong and Gary Geunbae Lee. 2009. Multi-domain spoken language understanding with transfer learning. *Speech Communication*, 51(5):412–424.

Thorsten Joachims, Adith Swaminathan, and Maarten de Rijke. 2018. Deep learning with logged bandit feedback. In *International Conference on Learning Representations*.

Joo-Kyung Kim and Young-Bum Kim. 2020. Pseudo labeling and negative feedback learning for large-scale multi-label domain classification. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7964–7968.

John Langford, Alexander Strehl, and Jennifer Wortman. 2008. Exploration scavenging. In *Proceedings of the 25th International Conference on Machine Learning*, page 528–535, Helsinki, Finland.

Carolin Lawrence and Stefan Riezler. 2018. Improving a neural semantic parser by counterfactual learning from human bandit feedback. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1820–1830, Melbourne, Australia. Association for Computational Linguistics.

Carolin Lawrence, Artem Sokolov, and Stefan Riezler. 2017. Counterfactual learning from bandit feedback under deterministic logging : A case study in statistical machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2566–2576, Copenhagen, Denmark.

Samuel Louvan and Bernardo Magnini. 2020. Recent neural methods on slot filling and intent classification for task-oriented dialogue systems: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 480–496, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Deepak Muralidharan, Justine Kao, Xiao Yang, Lin Li, Lavanya Viswanathan, Mubarak Seyed Ibrahim, Kevin Luikens, Stephen Pulman, Ashish Garg, Atish Kothari, and Jason Williams. 2019. Leveraging User Engagement Signals For Entity Labeling in a Virtual Assistant. *arXiv*, 1909.09143.

Pragaash Ponnusamy, Alireza Roshan-Ghias, Chenlei Guo, and Ruhi Sarikaya. 2020. Feedback-based self-learning in large-scale conversational ai agents. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, New York, NY, USA.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, 1910.01108.

Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. 2019. Cross-lingual transfer learning for multilingual task oriented dialog. 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 3795–3805, Minneapolis, Minnesota.

Adith Swaminathan and Thorsten Joachims. 2015. The self-normalized estimator for counterfactual learning. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, page 3231–3239, Montreal, Canada. MIT Press.

Gokhan Tur and Renato De Mori. 2011. *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*. John Wiley and Sons.

Lequn Wang, Yiwei Bai, Arjun Bhalla, and Thorsten Joachims. 2019. Batch learning from bandit feedback through bias corrected reward imputation. In *ICML Workshop on Real-World Sequential Decision Making*.

H. Weld, X. Huang, S. Long, J. Poon, and S. C. Han. 2021. A survey of joint intent detection and slot-filling models in natural language understanding. *arXiv*, 2101.08091.

Puyang Xu and Ruhi Sarikaya. 2014. Contextual domain classification in spoken language understanding systems using recurrent neural network. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 136–140.

Zhichang Zhang, Zhenwen Zhang, Haoyuan Chen, and Zhiman Zhang. 2019. A joint learning framework with bert for spoken language understanding. *IEEE Access*, 7:168849–168858.

A Training Details

SLU Models For the logging policy, we fine-tune the *base uncased* version of DistilBERT (Sanh et al., 2019) with batch size 16, dropout 0.1, Adam with learning rates 1e-4 (DC) and 5e-4 (IC/SL) and gradually unfreeze the BERT layers over the first two epochs. All models are trained until the loss stops decreasing in the validation set. For bandit learning, we use the same architecture, initialized with the final weights of the logging policy models, and continue training with batch size 16. The model has between 66.6M (DC) and 66.8M (IC/SL) parameters (with slight differences depending on the size of a domain's label space).

Feedback Estimator For the feedback estimator $f(x, y^D, y^I, (y_t^S)) \rightarrow \delta$, we fine-tune the same pre-trained BERT model, but with a regression output on top of a 256-d hidden ReLU layer. As input, we provide a single sequence consisting of the predicted domain, intent and, token by token, the utterance and slot labels. To represent domain, intent and slot labels we use tokens reserved for such purposes in the pre-trained model's vocabulary. The model is trained to minimize mean squared error predicting the overall feedback in D_B. The batch size is 32. It has around 66.6M parameters.

Parameter Tuning To ensure a fair comparison, we tune the learning rates used for bandit learning (including for training the feedback estimator) for each feedback assignment method separately and pick the rate with best validation loss, for DC and IC/SL separately. That is especially important as feedback assigned with different methods can be of different magnitude. Table 2 shows the results.

Dataset	Feedback	FA	DC LR	IC/SL LR	f LR
Explored Learning Rates			5e-7	5e-6	1e-4
			1e-6	1e-5	5e-4
			5e-6	5e-5	1e-3
			1e-5	1e-4	5e-3
SNIPS	argmax	Overall	5e-6	5e-5	
SNIPS	argmax	Positive	5e-6	5e-5	
SNIPS	argmax	Propensity	1e-5	5e-5	
SNIPS	argmax	Prop-All	1e-5	1e-5	
SNIPS	argmax	Adva-Uni	5e-6	1e-5	1e-3
SNIPS	argmax	Adva-Prop	5e-6	5e-5	1e-3
SNIPS	argmax	Oracle	5e-6	5e-5	
SNIPS	5 samples	Overall	5e-6	1e-5	
SNIPS	5 samples	Positive	5e-6	1e-5	
SNIPS	5 samples	Propensity	5e-7	1e-5	
SNIPS	5 samples	Prop-All	5e-7	1e-5	
SNIPS	5 samples	Adva-Uni	5e-6	1e-5	1e-4
SNIPS	5 samples	Adva-Prop	5e-6	5e-5	1e-4
SNIPS	5 samples	Oracle	1e-6	1e-5	
TOP	argmax	Overall	1e-6	1e-5	
TOP	argmax	Positive	1e-5	5e-6	
TOP	argmax	Propensity	5e-7	5e-6	
TOP	argmax	Prop-All	5e-7	5e-6	
TOP	argmax	Adva-Uni	1e-5	1e-5	1e-3
TOP	argmax	Adva-Prop	5e-6	1e-5	1e-3
TOP	argmax	Oracle	1e-5	1e-5	
TOP	5 samples	Overall	1e-6	5e-6	
TOP	5 samples	Positive	1e-5	1e-5	
TOP	5 samples	Propensity	5e-7	1e-5	
TOP	5 samples	Prop-All	5e-7	1e-5	
TOP	5 samples	Adva-Uni	5e-7	5e-6	1e-4
TOP	5 samples	Adva-Prop	5e-7	1e-5	1e-4
TOP	5 samples	Oracle	1e-6	5e-6	

Table 2: Learning rates used for model training.