# Offline Reinforcement Learning from Human Feedback in Real-World Sequence-to-Sequence Tasks

**Julia Kreutzer[1]*, Stefan Riezler[2], Carolin Lawrence[3]**
[1]Google Research, Montreal, Canada
[2]Computational Linguistics & IWR, Heidelberg University, Germany
[3]NEC Laboratories Europe, Heidelberg, Germany
jkreutzer@google.com,
riezler@cl.uni-heidelberg.de,
carolin.lawrence@neclab.eu

## Abstract

Large volumes of interaction logs can be collected from NLP systems that are deployed in the real world. How can this wealth of information be leveraged? Using such interaction logs in an offline reinforcement learning (RL) setting is a promising approach. However, due to the nature of NLP tasks and the constraints of production systems, a series of challenges arise. We present a concise overview of these challenges and discuss possible solutions.

## 1 Introduction

When Natural Language Processing (NLP) systems are deployed in production, and interact with users ("the real world"), there are many potential ways of collecting feedback data or rich interaction logs. For example, one can ask for explicit user ratings (Kreutzer et al., 2018a), or collect user clicks (De Bona et al., 2010), or elicit user revisions (Trivedi et al., 2019) to get an estimate of how well the deployed system is doing. However, such user interaction logs are primarily used for an one-off assessment of the system, e.g., for spotting critical errors, detecting domain shifts, or identifying the most successful use cases of the system in production. This assessment can then be used to support the decision of keeping or replacing this system in production.

From a machine learning perspective, using interaction logs only for evaluation purposes is a lost opportunity for offline reinforcement learning (RL). Logs of user interactions are gold mines for off-policy learning, and they should be put to use, rather than being forgotten after a one-off evaluation purpose. To move towards the goal of using user interaction logs for learning, we will discuss which challenges have hindered RL from being employed in real-world interaction with users of NLP systems so far.

Concretely, our focus is on sequence-to-sequence learning for NLP applications (see § 2 for an overview). For example, many machine translation services provide the option for users to give feedback on the quality of the translation, e.g., by collecting post-edits. Similarly, industrial chatbots can easily collect vast amounts of interaction logs, which can be utilized with offline RL methods (Kandasamy et al., 2017; Zhou et al., 2017; Hancock et al., 2019). In the following, we will thus present challenges that are encountered in user-interactive RL for NLP systems. With this discussion, we aim to (1) encourage NLP practitioners to leverage their interaction logs through offline RL, and (2) inspire RL researchers to steel their algorithms for the challenging applications in NLP.

## 2 Offline Feedback for Seq2Seq in NLP

In sequence-to-sequence (Seq2Seq) learning, the task is to map an input sequence $\mathbf{x} = x_1, x_2, \ldots, x_{|\mathbf{x}|}, \forall x_i \in \mathcal{X}$ to an output sequence $\mathbf{y} = y_1, y_2, \ldots, y_{|\mathbf{y}|}, \forall y_j \in \mathcal{Y}$, where $\mathcal{X}, \mathcal{Y}$ denote the sets of input and output vocabularies, respectively. The conditional distribution of the output sequence given the input can be modeled with a policy $\pi_\theta$ with learnable parameters $\theta$. Assuming a left-to-right generation order, the output sequence $\mathbf{y}$ is generated by conditioning on previous output elements $\mathbf{y}_{<j}$ and the input sequence $\mathbf{x}$:

$$\pi_\theta(\mathbf{y} \mid \mathbf{x}) = \prod_{j=1}^{|\mathbf{y}|} \pi_\theta(y_j \mid \mathbf{y}_{<j}, \mathbf{x}). \qquad (1)$$

Mapping the sequence-to-sequence problem formulation to NLP tasks, we have for example:

- Machine translation: $\mathbf{x}$ is a source sentence and $\mathbf{y}$ the translation of $\mathbf{x}$ in a target language.

*All authors contributed equally, order has been randomized (see https://bit.ly/38PgRjm).

*Proceedings of the 5th Workshop on Structured Prediction for NLP*, pages 37–43
August 1–6, 2021. ©2021 Association for Computational Linguistics

- Semantic parsing: $\mathbf{x}$ is a sentence and $\mathbf{y}$ its semantic parse (e.g., in SQL).
- Summarization: $\mathbf{x}$ is the document that is to be summarized and $\mathbf{y}$ a corresponding summary.
- Dialogue generation: $\mathbf{x}$ is the conversation history and $\mathbf{y}$ an appropriate reply.

The most distinctive feature of Seq2Seq NLP tasks for RL are the extremely large, structured output spaces: given the output vocabulary of size $|\mathcal{Y}|$ and a maximum sequence length $M$, there are $|\mathcal{Y}|^M$ possible combinations of output sequences. For instance, in machine translation there might be as many as 30 000 output tokens in the vocabulary and the output sequence length could easily be 100, leading to a total of $30\,000^{100}$ possible outputs.

A successful policy identifies the few combination of tokens that form valid output sequences. In the most extreme case only one output sequence exists that will be correct. , e.g., in a semantic parsing setup, where potentially only one specific SQL query will return the correct answer when executed. To train a policy, supervised data can be used. There we assume a given dataset $\mathcal{D}_{sup} = \{(\mathbf{x}_t, \mathbf{y}_t)\}_{t=1}^{T}$ on which the parameters $\theta$ can be learnt with a maximum likelihood approach, aiming to maximize the model score for the given reference output.

In practice, it may be too expensive to collect correct, i.e., supervised, output sequences, since they require skilled annotators, e.g., trained translators for a machine translation task. Therefore, one option is to pre-train the policy on some available supervised data, which will allow the model to concentrate on reasonable areas in the output space (Choshen et al., 2020). The model can then be used to produce potentially imperfect output sequences and humans can judge an output $\tilde{\mathbf{y}}$ and a reward $\delta_t \in [0, 1]$ is assigned. Model parameters may be optimized by pairing the model outputs with their reward estimates. Depending on the use case, quality judgments may also exist for single elements in the structure, adding $\delta_{(t,j)}$ for every step in the output sequence. The core idea is that the weighting by $\delta$ enables learning from imperfect outputs while respecting their faults. In RL, these quality assessments are used to reward desirable model actions, here desirable sequence outputs.

When collecting quality judgments from human users in production systems, it would be risky to directly update the model online according to their feedback.[1] Some user feedback might be adversarial, inappropriate, or not representative when used for training without prior treatment (Rivas et al., 2018; Kreutzer et al., 2018a; Davis, 2016). [2] Furthermore, interpreting feedback wrongly (e.g., through incorrect credit assignment (Bahdanau et al., 2017)), or receiving misleading feedback (Nguyen et al., 2017; Kreutzer et al., 2018a), could easily push the policy into less favorable conditions.

Because updating systems online is too risky, quality judgments are instead stored in interaction logs, i.e., $\mathcal{D}_{log} = \{(\mathbf{x}_t, \tilde{\mathbf{y}}_t, \delta_t)\}_{t=1}^{T}$, and the system is updated offline. As a result, the imperfect output sequences are produced by a possibly different policy, the logging policy $\mu$, and updates to our learning policy are conducted offline, which is a classic *off-policy* RL scenario.

Due to the logging setup, the collected dataset is biased towards the choices of the deployed model, the logging policy $\mu$. This results in a counterfactual learning scenario (Bottou et al., 2013). The bias may be corrected via importance sampling. If the logging policy is known and $\mu(\hat{\mathbf{y}} \mid \mathbf{x})$ is logged as well, the policy can then be optimized for the Inverse Propensity Scoring (IPS) objective (Rosenbaum and Rubin, 1983):

$$\mathcal{L}_{\text{IPS}} = -\frac{1}{T}\sum_{t=1}^{T} \delta_t \frac{\pi_\theta(\tilde{\mathbf{y}}_t \mid \mathbf{x}_t)}{\mu(\tilde{\mathbf{y}}_t \mid \mathbf{x}_t)}. \qquad (2)$$

## 3 Challenges for Off-Policy RL in NLP

On top of the difficulties encountered in offline RL, additionally constraints arise in production scenarios. We address this and possible solutions in §3.1, while §3.2 focuses on how to obtain reliable data from which machine learning can succeed.

### 3.1 Deterministic Logging and Off-line Learning

In order to not show inferior outputs to users, production NLP systems show the most likely output, which disables the typically crucial exploration component of RL. This effectively results in deterministic logging policies that lack explicit exploration, which makes an application of standard off-

[1] The majority of RL research in NLP has focused on learning from online feedback (Sokolov et al., 2016; He et al., 2016; Li et al., 2016; Bahdanau et al., 2017; Nguyen et al., 2017; Nogueira and Cho, 2017; Lam et al., 2018).

[2] The chatbot Tay might be one of the most illustrative examples for what can go wrong (Davis, 2016).

policy methods for counterfactual learning questionable. For example, techniques such as inverse propensity scoring (Rosenbaum and Rubin, 1983) or weighted importance sampling (Precup et al., 2000; Jiang and Li, 2016; Thomas and Brunskill, 2016), rely on sufficient exploration of the output space by the logging system as a prerequisite for counterfactual learning. In fact, Langford et al. (2008) and Strehl et al. (2010) even give impossibility results for *exploration-free counterfactual learning*.

One option is to hope for *implicit exploration* due to input or context variability. This has been observed for the case of online advertising (Chapelle and Li, 2011) and investigated theoretically (Bastani et al., 2017). In NLP, output sequences may overlap in some of the words, so the learner could infer from rewards in which contexts specific words are more suitable than in others. This has been explored in the context of machine translation (Lawrence et al., 2017b), utilizing the Deterministic Propensity Matching (DPM) objective

$$\mathcal{L}_{\text{DPM}} = -\frac{1}{T}\sum_{t=1}^{T} \delta_t \pi_\theta(\tilde{\mathbf{y}}_t \mid \mathbf{x}_t), \tag{3}$$

which closely follows the IPS objective, however, due to the deterministic logging $\forall \tilde{\mathbf{y}}, \mu(\tilde{\mathbf{y}} \mid \mathbf{x}) = 1$. While this exploration is limited by the input data, solutions for safe exploration might be attractive to transfer to NLP applications to actively guide exploration while not sacrificing quality (Hans et al., 2008; Berkenkamp et al., 2017).

Another option is to consider concrete cases of *degenerate behavior* in estimation from logged data. We look at two such issues and possible solutions. Both problems occur irrespective of whether data is logged deterministically or not, but the effects of the degenerative behavior might be amplified in the case of deterministic logging.

The first form of degenerate behaviour occurs for a collected log $\mathcal{D}_{log}$ with $\delta \in [0, 1]$ because IPS and DPM can trivially be minimized by setting all probabilities in the dataset $\mathcal{D}$ to 1 for any $\delta_t > 0$ (Lawrence et al., 2017a). Concretely, this means, while the worst output sequences with $\delta_t = 0$ are simply ignored, all other sequences are encouraged, even if their reward is close to 0. However, it is clearly undesirable to increase the probability of low reward examples (Swaminathan and Joachims, 2015; Lawrence et al., 2017b,a).

There are two possible solutions to this problem: The first solution is to tune the learning rate and perform early stopping before the degenerate state can be reached. The second solution is to utilize a *multiplicative control variate* (Kong, 1992) for self-normalization (Swaminathan and Joachims, 2015). For efficient gradient calculation, batches of size $B$ can be reweighted one-step-late (OSL) (Lawrence and Riezler, 2018) using $\theta'$ from some previous iteration:

$$\mathcal{L}_{\text{OSL}} = -\frac{\frac{1}{B}\sum_{b=1}^{B} \delta_b \pi_\theta(\tilde{\mathbf{y}}_b \mid \mathbf{x}_b)}{\frac{1}{T}\sum_{t=1}^{T} \pi_{\theta'}(\tilde{\mathbf{y}}_t \mid \mathbf{x}_t)}. \tag{4}$$

Self-normalization discourages increasing the probability of low reward data because this would take away probability mass from higher reward outputs and as a result. This introduces a bias in the estimator (that decreases as $T$ increases), however, it makes learning under deterministic logging feasible, as has been shown for learning with real human feedback in a semantic parsing scenario (Lawrence and Riezler, 2018). This gives the RL agent an edge in learning in an environment that has been deemed impossible in the literature.

A second form of degenerate behavior occurs because the reward $\delta_t$ of an output sequence is typically measured with some non-negative value, e.g., $\delta_t \in [0, 1]$. For example, for machine translation, Kreutzer et al. (2018b) collect ratings for translations on a 5-point Likert scale and map the values linearly to $[0, 1]$. However, utilizing any of the above objectives means that bad output sequences with low rewards cannot actively be discouraged.

There are two possible solutions, both of which have been used as *additive control variates* to reduce variance in gradient estimators. First, low reward sequences can be discouraged by employing a reward baseline, where for example the average reward $\Delta = \frac{1}{t}\sum_{t'=1}^{t} \delta_{t'}$ is subtracted from each $\delta_t$. This will cause output sequences worse than the running average to be discouraged rather than encouraged. The second option is to use the logged data $\mathcal{D}_{log}$ to learn a *reward estimator* $\hat{\delta}$ that can return a reward estimate for any pair $(\mathbf{x}, \mathbf{y})$. This estimator together with the IPS objective leads to the Doubly Robust (DR) objective (Dudik et al., 2011),

$$\mathcal{L}_{\mathrm{DR}} = -\frac{1}{T}\sum_{t=1}^{T}\Big[\big(\delta_t - \hat{\delta}(\mathbf{x}_t, \tilde{\mathbf{y}}_t)\big)\ \pi_\theta(\tilde{\mathbf{y}}_t \mid \mathbf{x}_t) + \sum_{\tilde{\mathbf{y}}' \sim \pi_\theta(\tilde{\mathbf{y}}|\mathbf{x}_t)} \hat{\delta}(\mathbf{x}_t, \tilde{\mathbf{y}}')\ \pi_\theta(\tilde{\mathbf{y}}' \mid \mathbf{x}_t)\Big].$$

This objective enables the exploration of other outputs $\tilde{\mathbf{y}}'$ that are not part of the original log and encourages them based on the reward value returned by the estimator. For the task of machine translation, Lawrence et al. (2017b) show this objective to be the most successful in their setup, and Kreutzer et al. (2018a) report simulation results that show that this objective can significantly reduce the gap between offline and online policy learning, even if the reward estimator is not perfect. Zhou et al. (2017) present an alternating approach to integrating a reward estimator for exploration, by switching between learning offline from logged rewards and exploring online with the help of a reward estimator in phases.

### 3.2 Reliability and Learnability of Feedback

In interactive NLP, it is unrealistic to expect anything else than *bandit feedback* from a human user interacting with a chatbot, automatic summarization tool, or commercial machine translation system. That is, users of such systems will only provide a reward signal to the one output that is presented to them, and cannot be expected to rate a multitude of outputs for the same input. As a result, the feedback is very sparse in relation to the size of the output space.

Ideally, the user experience should not be disrupted through feedback collection. Non-intrusive interface options for example allow for corrections of the output ("post-edits" in the context of machine translation) as a negative signal, or recording whether the output is copied and/or shared without changes, which may be interpreted as a positive signal. However, the signal might be *noisy*, since the notion of output quality for natural language generation tasks is not a well-defined function to start with: Each input might have many possible valid outputs, each of which humans may judge differently, depending on many contextual and personal factors. In machine translation evaluation for instance, inter-rater agreements have traditionally been reported as low (Turian et al., 2003; Carl et al., 2011; Lommel et al., 2014), especially when quality estimates are collected from non-professional raters (Callison-Burch, 2009). Similar observations have been made for other text generation tasks (Godwin and Piwek, 2016; Verberne et al., 2018). Nguyen et al. (2017) illustrated how badly machine translation systems can handle human-level noise in direct feedback for online RL with simulations. The level of noise in real-world human feedback may be so high that it prevents learning completely, as for example experienced in e-commerce machine translation logs (Kreutzer et al., 2018a). The issue is even higher in dialogue generation where there are a plenitude of acceptable responses (Pang et al., 2020). To this aim, inverse RL has been proposed to infer reward functions from responses indirectly (Takanobu et al., 2019).

Surprisingly, the question of how to best improve an RL agent in the scenario of learning from real-world human feedback has been scarcely researched. This might originate from many RL research environments coming with fixed reward functions. In the real world, however, there is rarely a clearly defined single reward function for which it would suffice optimizing for. The suggestions in Dulac-Arnold et al. (2019) seem straightforward: warm-starting agents to decrease sample complexity or using inverse reinforcement learning to recover reward functions from demonstrations (Wang et al., 2020) — but they require additional supervision signals that RL was supposed to alleviate.

When it comes to the question *which type of human feedback is most beneficial* for training an RL agent, one finds a lot of blanket statements, e.g., referring to the advantages of pairwise comparisons (Thurstone, 1927). For instance, learning from human pairwise preferences from humans has been advertised for summarization (Christiano et al., 2017; Stiennon et al., 2020) and language modeling (Ziegler et al., 2019), but the reliability of the signal has not been evaluated. An exception is the work of Kreutzer et al. (2018b) which is the first to investigate two crucial questions. The first question addresses which type of human feedback — pairwise judgments or cardinal feedback on a 5-point scale — can be given most *reliably* by human teachers. The second question investigates which type of feedback allows to learn reward estimators that best approximate human rewards and can be best integrated into an end-to-end RL-NLP task.

Regarding the first question, Kreutzer et al. (2018b) found that the common assumption — that pairwise comparisons are easier to judge than a single output on a Likert scale (Thurstone, 1927) — turned out to be false for the task of machine translation. Inter-rater reliability proved to be higher for 5-point ratings (Krippendorff's $\alpha = 0.51$) than for pairwise judgments ($\alpha = 0.39$). (Kreutzer et al., 2018b) explain two advantages that the Likert scale setup offers: (1) it is possible to standardize cardinal judgments for each rater to remove individual biases, (2) they offer an absolute anchoring for quality, while a preference rankings leave the overall positioning of the pair of outputs on a quality scale open. For pairwise judgments it is difficult or even impossible to reliably choose between two outputs that are similarly good or bad, e.g., differing by only a few words. Therefore, filtering out raters with low intra-rater reliability proved effective for absolute ratings, while filtering outputs with a high variance in ratings was most effective for pairwise ratings, yielding the final inter-rater reliability given above. Discarding rated outputs, however, reduces the size of the log to learn from, which is undesirable in settings where rewards are scarce or costly.

To answer the second question, Kreutzer et al. (2018b) found a neural machine translation system can be significantly improved using a reward estimator trained on only a few hundred cardinal user judgments. This work highlights that future research in real-world RL might have to involve studies in *user interfaces* or user experience, since the interfaces for feedback collection influence the reward function that RL agents learn from – and thereby the downstream task success. Collecting implicit feedback (Kreutzer et al., 2018a; Jaques et al., 2020) might offer a better user experience.

For the challenges discussed in Sections 3.1 and 3.2, a promising approach is to tackle the arguably simpler problem of learning a reward estimator from human feedback first, then provide unlimited learned feedback to generalize to unseen outputs in off-policy RL. However, risks of bias introduction and potential benefits for noise reduction through replacing user feedback by reward estimators are yet to be quantified.

## 4 Conclusion

There is large potential in NLP to leverage user interaction logs for system improvement. We discussed how algorithms for offline RL can offer promising solutions for this learning problem. However, specific challenges in offline RL arise due to the particular nature of NLP systems that collect human feedback in real-world applications. We presented cases where such challenges have been found and offered solutions that have helped. So far, the solutions have mainly been explored in the context of machine translation and semantic parsing. In the future, it will be interesting to explore further tasks and additional real-world use cases to find out how to best learn from human feedback.

## References

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. An actor-critic algorithm for sequence prediction. In *5th International Conference on Learning Representations, ICLR*, Toulon, France.

H. Bastani, M. Bayati, and K. Khosravi. 2017. Exploiting the natural exploration in contextual bandits. *ArXiv e-prints*, 1704.09011.

Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. 2017. Safe model-based reinforcement learning with stability guarantees. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 908–918, Long Beach, California.

Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X. Charles, D. Max Chickering, Elon Portugaly, Dipanakar Ray, Patrice Simard, and Ed Snelson. 2013. Counterfactual Reasoning and Learning Systems: The Example of Computational Advertising. *Journal of Machine Learning Research*, 14.

Chris Callison-Burch. 2009. Fast, cheap, and creative: Evaluating translation quality using Amazon's Mechanical Turk. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Singapore.

Michael Carl, Barbara Dragsted, Jakob Elming, Daniel Hardt, and Arnt Lykke Jakobsen. 2011. The process of post-editing: a pilot study. *Copenhagen Studies in Language*, 41:131–142.

Olivier Chapelle and Lihong Li. 2011. An empirical evaluation of Thompson sampling. In *Advances in Neural Information Processing Systems (NeurIPS)*, Granada, Spain.

Leshem Choshen, Lior Fox, Zohar Aizenbud, and Omri Abend. 2020. On the weaknesses of reinforcement learning for neural machine translation. In *International Conference on Learning Representations (ICLR)*, Virtual.

Paul F. Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep Reinforcement Learning from Human Preferences. In

*Advances in Neural Information Processing Systems (NeurIPS)*, Long Beach, CA, USA.

Ernest Davis. 2016. AI amusements: the tragic tale of Tay the chatbot. *AI Matters*, 2(4):20–24.

Fabio De Bona, Stefan Riezler, Keith Hall, Massimiliano Ciaramita, Amaç Herdağdelen, and Maria Holmqvist. 2010. Learning dense models of query similarity from user click logs. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-ACL)*, Los Angeles, California.

Miroslav Dudik, John Langford, and Lihong Li. 2011. Doubly robust policy evaluation and learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, Bellevue, WA.

Gabriel Dulac-Arnold, Daniel J. Mankowitz, and Todd Hester. 2019. Challenges of real-world reinforcement learning. *CoRR*, abs/1904.12901.

Keith Godwin and Paul Piwek. 2016. Collecting reliable human judgements on machine-generated language: The case of the QG-STEC data. In *Proceedings of the 9th International Natural Language Generation conference (INLG)*, Edinburgh, UK.

Braden Hancock, Antoine Bordes, Pierre-Emmanuel Mazaré, and Jason Weston. 2019. Learning from dialogue after deployment: Feed yourself, chatbot!

Alexander Hans, Daniel Schneegaß, Anton Maximilian Schäfer, and Steffen Udluft. 2008. Safe exploration for reinforcement learning. In *ESANN*, pages 143–148.

Ji He, Jianshu Chen, Xiaodong He, Jianfeng Gao, Lihong Li, Li Deng, and Mari Ostendorf. 2016. Deep reinforcement learning with a natural language action space. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, Berlin, Germany.

Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. 2020. Way off-policy batch deep reinforcement learning of human preferences in dialog.

Nan Jiang and Lihong Li. 2016. Doubly robust off-policy value evaluation for reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, New York, NY.

Kirthevasan Kandasamy, Yoram Bachrach, Ryota Tomioka, Daniel Tarlow, and David Carter. 2017. Batch policy gradient methods for improving neural conversation models. In *5th International Conference on Learning Representations (ICLR)*.

Augustine Kong. 1992. A note on importance sampling using standardized weights. Technical Report 348, Department of Statistics, University of Chicago, Illinois.

Julia Kreutzer, Shahram Khadivi, Evgeny Matusov, and Stefan Riezler. 2018a. Can neural machine translation be improved with user feedback? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (ACL)*.

Julia Kreutzer, Joshua Uyheng, and Stefan Riezler. 2018b. Reliability and learnability of human bandit feedback for sequence-to-sequence reinforcement learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Tsz Kin Lam, Julia Kreutzer, and Stefan Riezler. 2018. A reinforcement learning approach to interactive-predictive neural machine translation. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation (EAMT)*, Alicante, Spain.

John Langford, Alexander Strehl, and Jennifer Wortman. 2008. Exploration scavenging. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, Helsinki, Finland.

Carolin Lawrence, Pratik Gajane, and Stefan Riezler. 2017a. Counterfactual Learning for Machine Translation: Degeneracies and Solutions. In *Proceedings of the NIPS WhatIf Workshop*, Long Beach, California, USA.

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 (ACL)*, Melbourne, Australia.

Carolin Lawrence, Artem Sokolov, and Stefan Riezler. 2017b. 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 (EMNLP)*, Copenhagen, Denmark.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Austin, Texas. Association for Computational Linguistics.

Arle Lommel, Maja Popovic, and Aljoscha Burchardt. 2014. Assessing inter-annotator agreement for translation error annotation. In *MTE: Workshop on Automatic and Manual Metrics for Operational Translation Evaluation*.

Khanh Nguyen, Hal Daumé III, and Jordan Boyd-Graber. 2017. Reinforcement learning for bandit neural machine translation with simulated human feedback. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Rodrigo Nogueira and Kyunghyun Cho. 2017. Task-oriented query reformulation with reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Copenhagen, Denmark.

Bo Pang, Erik Nijkamp, Wenjuan Han, Linqi Zhou, Yixian Liu, and Kewei Tu. 2020. Towards holistic and automatic evaluation of open-domain dialogue generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, Online.

Doina Precup, Richard S. Sutton, and Satinder P. Singh. 2000. Eligibility traces for off-policy policy evaluation. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*, San Francisco, CA.

Pablo Rivas, Kerstin Holzmayer, Cristian Hernandez, and Charles Grippaldi. 2018. Excitement and concerns about machine learning-based chatbots and talkbots: A survey. In *2018 IEEE International Symposium on Technology and Society (ISTAS)*, pages 156–162. IEEE.

Paul R. Rosenbaum and Donald B. Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1).

Artem Sokolov, Julia Kreutzer, Stefan Riezler, and Christopher Lo. 2016. Stochastic structured prediction under bandit feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, Barcelona, Spain.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020. Learning to summarize from human feedback.

Alexander L. Strehl, John Langford, Lihong Li, and Sham M. Kakade. 2010. Learning from logged implicit exploration data. In *Advances in Neural Information Processing Sytems (NIPS)*, Vancouver, Canada.

Adith Swaminathan and Thorsten Joachims. 2015. The self-normalized estimator for counterfactual learning. In *Advances in Neural Information Processing Systems (NIPS)*, Montreal, Canada.

Ryuichi Takanobu, Hanlin Zhu, and Minlie Huang. 2019. Guided dialog policy learning: Reward estimation for multi-domain task-oriented dialog. 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)*, Hong Kong, China.

Philip S. Thomas and Emma Brunskill. 2016. Data-efficient off-policy policy evaluation for reinforcement learning. In *Proceedings of the 33nd International Conference on Machine Learning (ICML)*, New York, NY.

Louis Leon Thurstone. 1927. A law of comparative judgement. *Psychological Review*, 34:278–286.

Gaurav Trivedi, Esmaeel R Dadashzadeh, Robert M Handzel, Wendy W Chapman, Shyam Visweswaran, and Harry Hochheiser. 2019. Interactive NLP in clinical care: Identifying incidental findings in radiology reports. *Applied clinical informatics*, 10(4):655.

Joseph P Turian, Luke Shea, and I Dan Melamed. 2003. Evaluation of machine translation and its evaluation. *Proceedings of MT Summit*, pages 386–393.

Suzan Verberne, Emiel Krahmer, Iris Hendrickx, Sander Wubben, and Antal van den Bosch. 2018. Creating a reference data set for the summarization of discussion forum threads. *Language Resources and Evaluation*, 52(2):461–483.

Jingkang Wang, Yang Liu, and Bo Li. 2020. Reinforcement learning with perturbed rewards. In *AAAI*, New York, New York.

Li Zhou, Kevin Small, O. Rokhlenko, and C. Elkan. 2017. End-to-end offline goal-oriented dialog policy learning via policy gradient. *ArXiv*, abs/1712.02838.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.