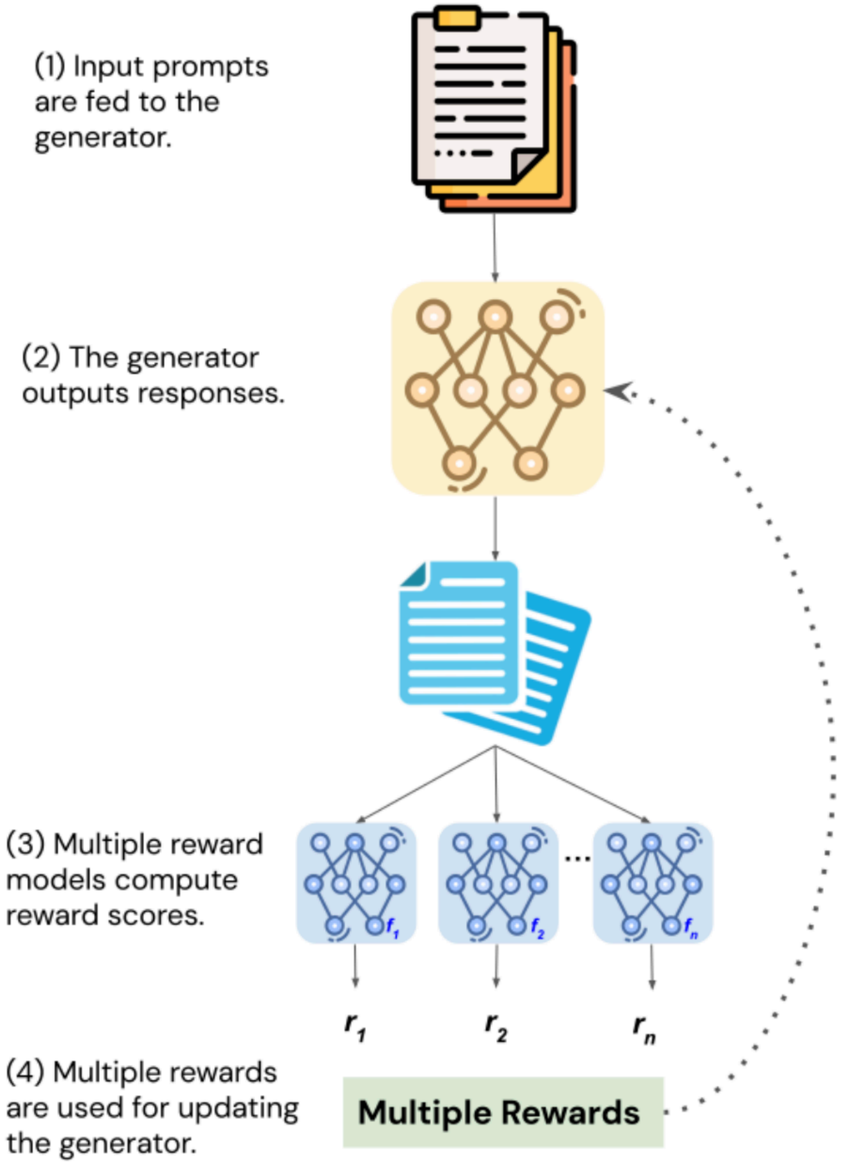
Dynamic Reward Adjustment in Multi-Reward Reinforcement Learning for Counselor Reflection Generation

Do June Min, Verónica Pérez-Rosas, Kenneth Resnicow, Rada Mihalcea
University of Michigan

Pre-recorded video for COLING 2024

Introduction

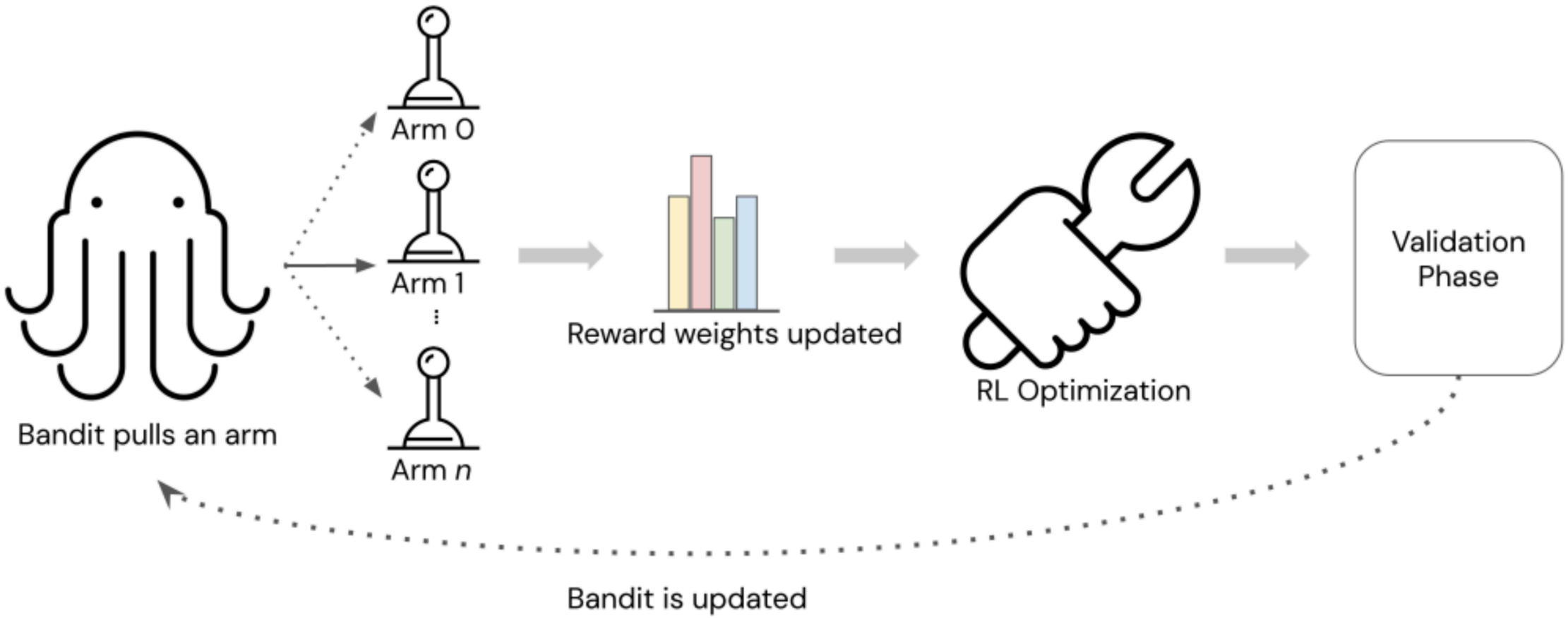
- Role of RL in NLP
  - Challenges of Multi-reward Optimization
  - Different classes of approaches have emerged
    - Combining vs Alternate Rewards
- New Multi-armed Bandit (MAB) Algorithms
  - DynaOpt and C-DynaOpt
  - Tested on counselor reflection generation.



Background

- Challenges in Automated Reflections: Accurately mimicking human emotional dynamics and language nuances in automated reflections requires a multi-reward optimization approach.
- Limitations of Current RL Approaches: Static reward systems may lead to suboptimal results because they fail to adapt and capture the evolving complexities of therapeutic dialogues during training.
- Multi-armed Bandits (MABs): A class of RL approaches that manages the exploration and exploitation trade-off, enabling dynamic adjustments to the training dynamics based on observed rewards.

Multi-armed Bandit Approach



Previous Work: DORB (Pasnuru et al, 2020)

- DORB operates by cycling through multiple rewards, optimizing one at a time.
- The key idea is to use multi-armed bandits (MABs) to model the relationship between each action (which reward to optimize) and the observed reward (after the reward is optimized, what rewards have been obtained?)
- Specifically, DORB employs the Exponential-weight algorithm for Exploration and Exploitation (Exp3), which tackles the adversarial bandit problem (Auer et al., 2002), to dynamically select from a pool of reward functions during training stages.

Algorithm: DynaOpt and C-DynaOpt

Algorithm 1 DynaOpt Optimization

Input: # of rewards N, # of train steps n_{train}, # of RL validation steps $\text{round}_{\text{bandit}}$ Initial policy p_0, Initial Distribution of reward weights, W (uniform distribution over N)

Make a copy p_θ of initial policy p_0.
Initialize Exp3 bandit B with $N+1$ arms.
Initialize weights $w_{0,i}$ over $i \in [1, 2, \cdots, N+1]$ as uniform distribution.
$a \leftarrow$ chooseArm(B) ▷ Eqn 1
$W \leftarrow$ UpdateRewardWeight($W, B_w, a, 1$) ▷ Eqns 2,3
$i \leftarrow 0$
while $i < n_{\text{train}}$ **do**
    train_responses $\leftarrow$ Sample(p_θ, train_data)
    $r_{\text{train}} \leftarrow$ ComputeReward(train_responses, W)
    Optimize p_θ with R_{train}, p_0 ▷ Eqn 5
    if i % $\text{round}_{\text{bandit}} == 0$ **then**
        dev_responses $\leftarrow$ Sample(p_θ, dev_data)
        $r_{\text{bandit}} \leftarrow$ ComputeReward(dev_responses, uniform weights)
        $r \leftarrow$ ComputeBanditReward(r_{bandit}) ▷ Eqn 4
        UpdateBandit(B, a, r) ▷ Eqns 2,3
        $a \leftarrow$ chooseArm(B)
        $W \leftarrow$ UpdateRewardWeight($W, B_w, a, 1$) ▷ Eqns 2,3
    end if
    $i \leftarrow i + 1$
end while

Proposed Methods: DynaOpt and C-DynaOpt

- Choosing an Action

$$p_t(i) = (1-\gamma)\frac{a_{t,i}}{\sum_{j=1}^{N} a_{t,j}} + \frac{\gamma}{N+1} \qquad (1)$$

- Updating the Weights (Exp3)

$$\hat{r}_{t,j}^{B^W} = \begin{cases} \frac{r_t^{B^W}}{p_t(i)} & \text{if } j = i \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

$$w_{t+1,i} = w_{t,i} \exp\left(\frac{\gamma \hat{r}_{t,i}^{B^W}}{K}\right) \qquad (3)$$

- Bandit Reward Computation

$$\hat{r}^t = \sum_i {r^t}_i$$
$$ {r^t}_i = \text{Mean}(R_{t,i}) - \text{Mean}(R_{t-1,i}) \qquad (4)$$

- Gamma is a hyperparmeter
- N+1 indexes N actions + No Update

Datasets

- Datasets Used: The research employs two key datasets, PAIR (Min et al, 2022) and CounselChat (Welivita and Pu, 2023), which contain real-world counseling dialogues. These datasets provide diverse scenarios for training and evaluating the models.

Statistics	PAIR	Welivita and Pu (2023)
# of Exchange Pairs	2544	1184
Avg # of Words	32.39	36.92
# of Complex Reflection	636	768
# of Simple Reflection	318	416
# of Non-Reflection	1590	0

Models & Algorithms

- Model: t5-base
- Baseline Algorithms
  - DORB (Pasnuru et al, 2020): Selectively chooses one reward function at a time from a set of available options based on their performance.
  - Cross Entropy: Supervised learning baseline
  - Round: Cycles through each reward function sequentially, giving each an equal opportunity to be tested and applied during training.
  - Uniform Weighted: Assigns equal weight to all reward functions

Reward Metrics

- Reflection Score (Min et al, 2022): Measures how accurately the model's generated responses reflect the emotional content and intent of the client's statements, assessing the quality of the reflections.
- Fluency: Evaluates the smoothness and naturalness of the language used in the model's responses by using perplexity.
- Coherence: Evaluates the logical flow and consistency of the generated counselor reflections. Implemented by training a RoBERTa classifier trained to detect coherent and incoherent client prompt and counselor response pairs.

Evaluation Metrics

- Automated Metrics
  - Diversity (Li et al., 2016): Gauges the linguistic diversity of the generated counselor reflections.
  - Levenshtein Edit Distance: Quantifies the extent to which the model successfully avoids verbatim repetition of client words.
- Human Metrics
  - Reflection, Coherence, Fluency annotated by two MI experts.

Automated Evaluation Results

Models	Reflection (↑)	Fluency (↑)	Coherence (↑)	Edit Rate (↑)	Diversity-2 (↑)
Round	-5.02%	11.36%	5.51%	-8.75%	-0.20%
Uniform Weighted	4.48%	8.13%	**5.36%**	-6.28%	**-0.23%**
DORB (Pasunuru et al., 2020)	-3.03%	9.54%	5.42%	-7.00%	-0.08%
DynaOpt	**7.80%**	7.03%	5.02%	**-4.90%**	-0.63%
C-DynaOpt	6.14%	**8.73%**	5.02%	-5.75%	-0.46%

- Performance Comparison: Combine methods outperform Alternate models, particularly in Reflection measurements, while both achieve similar Fluency and Coherence.
- Reflection Metrics: Notable improvement in Reflection scores with Combine methods over the Cross Entropy baseline; Alternate models show decreased performance in this area.
- Stability of Training: The Round class of models exhibits higher overall variance over random runs (reflection variance of 3.59 vs 1.43 & 1.29 of our methods), indicating less stability in the training process..

Human Evaluation Results

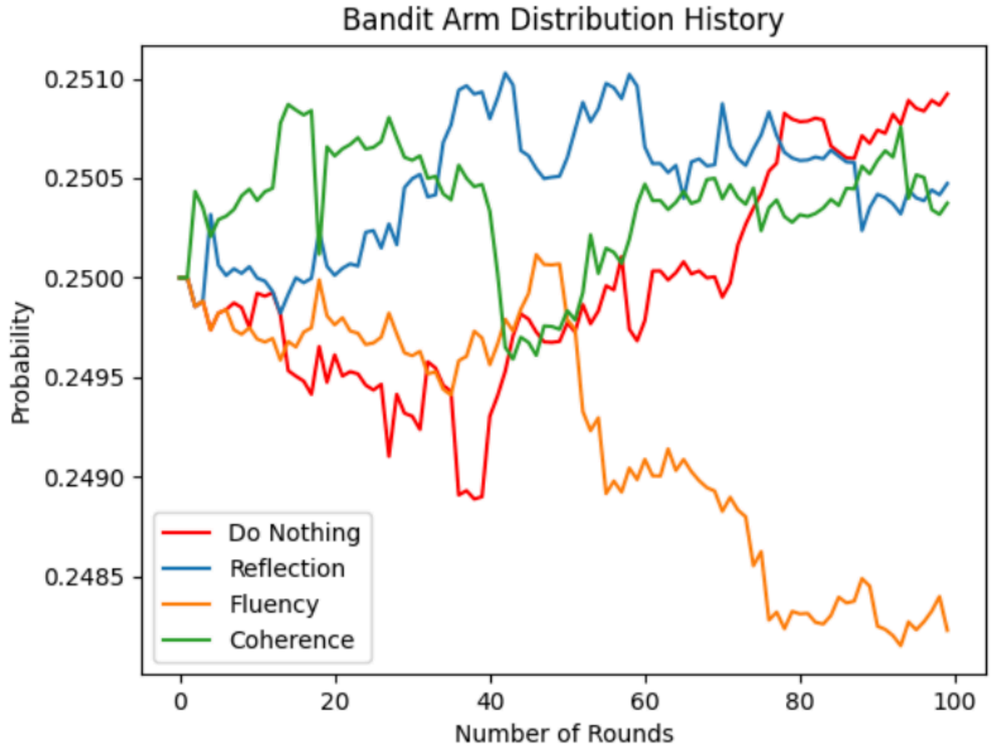
	Uniform Weighted	DORB (2020)	DynaOpt	C-DynaOpt
Reflection	28.29	25.30	**32.10**	29.93
Fluency	**60.31**	55.85	59.38	58.91
Coherence	62.48	62.79	63.62	**63.68**

- Comparison of Models: Human evaluations confirm that Combine models, specifically DynaOpt and C-DynaOpt, outperform the Uniform Weighted model in reflection quality.
- Fluency and Coherence: Despite lower scores in automated metrics, DynaOpt and C-DynaOpt exhibit superior fluency and coherence in human evaluations, suggesting that higher reflection levels enhance the naturalness of responses..

Overall Results

- Not All Multi-reward Optimization Methods Are Effective for Counselor Reflection Generation.
  - Methods that combine weights exhibit superior performance compared to the Alternate methods (Table 2).
- Comparative Advantage of Our Methods
  - DynaOpt and C-DynaOpt outperform not only the Alternate methods but also the Uniform Weighted baseline in terms of both automated and human reflection levels.

Bandit Visualization



Conclusion

- Problem Addressed: Optimizing multiple linguistic rewards in reinforcement learning for counselor reflection in motivational interviewing.
- Strategies Explored: Investigated Alternate and Combine approaches, enhancing them with bandit-augmented versions.
- Novel Methods: Introduced DynaOpt and C-DynaOpt, which dynamically adjust reward weights using multi-armed bandits during training.
- Empirical Findings: Demonstrated that previous approaches failed to improve response quality, while DynaOpt and C-DynaOpt surpassed existing baselines.

Acknowledgments, Resources, and Contact

- Acknowledgments: Special thanks to the research team and students from the University of Michigan School of Public Health for their invaluable feedback and participation. This work was partially supported by the National Science Foundation under award #2306372. The views expressed are those of the authors and do not necessarily reflect the views of the NSF.

- Our code is available at https://github.com/michigannlp/dynaopt

- Contact for Questions and Discussion: For further discussions and any questions, please email dojmin@umich.edu.