DORB: Dynamically Optimizing Multiple Rewards with Bandits

Ramakanth Pasunuru **Han Guo** **Mohit Bansal**
UNC Chapel Hill
{ram, hanguo, mbansal}@cs.unc.edu

Abstract

Policy gradients-based reinforcement learning has proven to be a promising approach for directly optimizing non-differentiable evaluation metrics for language generation tasks. However, optimizing for a specific metric reward leads to improvements in mostly that metric only, suggesting that the model is gaming the formulation of that metric in a particular way without often achieving real qualitative improvements. Hence, it is more beneficial to make the model optimize multiple diverse metric rewards jointly. While appealing, this is challenging because one needs to manually decide the importance and scaling weights of these metric rewards. Further, it is important to consider using a dynamic combination and curriculum of metric rewards that flexibly changes over time. Considering the above aspects, in our work, we automate the optimization of multiple metric rewards simultaneously via a multi-armed bandit approach (DORB), where at each round, the bandit chooses which metric reward to optimize next, based on expected arm gains. We use the Exp3 algorithm for bandits and formulate two approaches for bandit rewards: (1) Single Multi-reward Bandit (SM-Bandit); (2) Hierarchical Multi-reward Bandit (HM-Bandit). We empirically show the effectiveness of our approaches via various automatic metrics and human evaluation on two important NLG tasks: question generation and data-to-text generation. Finally, we present interpretable analyses of the learned bandit curriculum over the optimized rewards.

1 Introduction

Recent advancements in end-to-end neural networks-based approaches have shown wide success in various sequence generation tasks: machine translation (Sutskever et al., 2014; Luong et al., 2015), dialogue systems (Vinyals and Le, 2015; Serban et al., 2016), textual summarization (Rush et al., 2015; Nallapati et al., 2016; See et al., 2017), image/video captioning (Bahdanau et al., 2015; Venugopalan et al., 2015; Pasunuru and Bansal, 2017a), question generation (Du et al., 2017; Du and Cardie, 2018; Zhang and Bansal, 2019), etc. In all of these tasks, cross-entropy loss optimization has been widely used as a standard optimization approach (Sutskever et al., 2014), but this approach suffers from exposure-bias issue (Ranzato et al., 2016) and does not optimize for the non-differentiable automatic evaluation metrics that measure the quality of the generated sequence. Recent introduction of policy gradient-based reinforcement learning approaches address these issues for sequence generation tasks by directly optimizing the non-differentiable evaluation metrics (Zaremba and Sutskever, 2015; Ranzato et al., 2016; Rennie et al., 2017).

However, optimizing for a particular metric/reward via policy gradient-based approaches often leads to improvement in mostly that specific metric, suggesting that this approach is gaming the metrics (Paulus et al., 2018). The weighted average of multiple metrics or surrogate rewards have been explored (Liu et al., 2017), but these approaches have to deal with finding the optimal scale balance across different metrics. One can alternatively optimize multiple metrics via a mixing ratio (Pasunuru and Bansal, 2018), but this still needs careful tuning of the mixing ratio. Moreover, all these reward approaches are fixed and do not change over training, and all the metrics may not be important over every stage of the training. Thus, it might be useful to consider using a dynamic combination of metrics, which rewards to use early vs. later, or which rewards might be useful to come back later in training, and consider the context of the full history of rewards, as well as the models current state and the nature of the metric.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 7766–7780,
November 16–20, 2020.

To this end, we present a multi-armed bandit approach (which we name the DORB framework) where the arms of the bandit are the choices of the metrics that we want to optimize as rewards. At every round, the bandit chooses the next possible metric to optimize based on its previous performance history over these metrics, hence allowing the automatic learning of an optimal curriculum of rewards. We explore this approach in the context of exploration vs. exploitation via Exp3 algorithm (Auer et al., 2002b) with two novel approaches for bandit rewards: (1) Single Multi-reward Bandit (SM-Bandit); (2) Hierarchical Multi-reward Bandit (HM-Bandit). First, we present a reward scaling approach to maintain the metric rewards range in $[0, 1]$. Next, we present our SM-Bandit, where at each round, the bandit's reward is based on the performance improvement from multiple sources. Here, we use the average of all the scaled metric rewards from multiple sources as the final reward to the bandit. Finally, we present our HM-Bandit, which consists of a single first-level controller, as well as K second-level multi-armed bandits. The first-level controller's goal is to find the under-performing reward metric, while the second-level bandits' goal is to trigger the specific metric optimizer that will lead to a promising improvement in this specific metric.

We validate the effectiveness of our approaches on two important generation tasks: question generation and data-to-text generation, via both automatic evaluation metrics and human evaluation. For question generation, we present results on the SQuAD QG dataset (Du et al., 2017), and for data-to-text NLG, we choose the WebNLG dataset (Gardent et al., 2017). We show that our bandit-based approaches perform statistically significantly better (based on human evaluation) than strong single-reward based RL models as well as non-bandits based multi-reward methods such as the multi-task approach of Pasunuru and Bansal (2018). We further present various interpretable analyses of our bandit progress and learned rewards curriculum over different bandit approaches.

2 Related Works

Policy Gradient and Generative Models: Neural sequence to sequence models with cross-entropy optimization, potentially with attention mechanism (Bahdanau et al., 2015) and pointer-copy mechanism (See et al., 2017; Gulcehre et al., 2016; Vinyals et al., 2015a; Merity et al., 2018), are widely used in language generation tasks such as machine translation (Sutskever et al., 2014; Luong et al., 2015), abstractive summarization (Chopra et al., 2016; Nallapati et al., 2016), question generation (Du et al., 2017; Zhang and Bansal, 2019), video/image captioning (Xu et al., 2015; Vinyals et al., 2015b; Pasunuru and Bansal, 2017a; Zhou et al., 2018), as well as sentence simplification (Zhang and Lapata, 2017; Guo et al., 2018). However, often the final metrics of interest are not differentiable, and thus not compatible with the standard maximum-likelihood based training. Motivated by this, recently there has been a surge in applications of reinforcement learning techniques to language generation (Ranzato et al., 2016), in which the gradients of non-differentiable metrics are approximated using the scoring function (REINFORCE (Williams, 1992)). A few successful examples include image captioning (Rennie et al., 2017; Ren et al., 2017), abstractive summarization (Paulus et al., 2018; Chen and Bansal, 2018; Pasunuru and Bansal, 2018; Celikyilmaz et al., 2018), machine translation (Wu et al., 2016; Gu et al., 2017), sentence simplification (Zhang and Lapata, 2017), as well as video captioning (Pasunuru and Bansal, 2017b; Wang et al., 2018). Previous works have explored the problem of optimizing multiple rewards in the context of machine translation (Neubig and Watanabe, 2016). For example, the works of Duh et al. (2012) and Sankaran et al. (2013) are based on the theory of Pareto Optimality. Our approach, instead, dynamically decides the trade-off among metrics, rather than exploring the set of static Pareto-optimal hypotheses. The most related work on this line is Pasunuru and Bansal (2018), which simultaneously optimizes multiple rewards in alternate fashion for abstractive summarization. In our work, we use a multi-armed bandit framework to dynamically switch among multiple diverse reward optimizations in the context of policy-gradient-based generative models.[1]

Multi-Armed Bandit: Many control problems can be cast as multi-armed bandit problems, where the goal is to select a sequence of arms/actions in order to optimize certain objective (e.g., expected fu-

[1]When we say dynamic switching, we mean using one metric at a time (i.e., no explicit weighted combination of loss metrics' optimization for a single mini-batch), but to learn an implicit ratio/proportion of metrics' importance over the overall training trajectory (e.g., BLEU metric might be sampled three times more than ROUGE on average).

ture payoff) (Bubeck et al., 2012). One widely studied problem in the multi-armed bandit literature is finding the optimal trade-off between exploration and exploitation (Audibert et al., 2009; Macready and Wolpert, 1998; Auer et al., 2002a; Kveton et al., 2019; Bubeck et al., 2012). Some widely used bandit algorithms include ϵ-greedy (Sutton and Barto, 2018), Boltzmann exploration (Kaelbling et al., 1996), UCB (Auer et al., 2002a), Thompson sampling (Chapelle and Li, 2011), contextual bandit (Sharaf and Daumé III, 2019), as well as Exp3 adversarial bandit (Auer et al., 2002b). In this work, we use Exp3, and the hierarchical version of it, for the problem of optimizing multiple rewards.[2]

Multi-armed bandit algorithms have been used in a wide range of applications, such as online advertising (Chen et al., 2013), recommendation (Li et al., 2010), multi-task task selection (Guo et al., 2019a), and hyper-parameter optimization (Li et al., 2018; Merentitis et al., 2018). Recently, Graves et al. (2017) apply a non-stationary multi-armed bandit (in particular, the Exp3.S algorithm) to select an adaptive policy (curriculum) that a neural network follows to maximize the learning efficiency. Sharma and Ravindran (2017) use multi-armed bandit sampling to choose which domain data (harder vs. easier) to feed as input to a single model (using different Atari games). To our knowledge, we are the first ones to apply a multi-armed bandit to optimize multiple rewards in the context of text generation.

3 Multi-Reward Optimization

In this section, we first describe the policy gradients-based reinforcement learning (RL) approach for text generation tasks, and then discuss the need for a better multi-reward optimization approach for RL in the context of generation tasks. Lastly, we introduce our novel methods for multi-reward optimization via multi-armed bandits.

Glossary: Agent: RL policy gradients; Bandit: multi-armed bandit; Controller: controller in HM-Bandit (see Fig. 2).

Policy Gradient Background. Cross-entropy loss based optimization is traditionally used for the sequence generation tasks. However, recent policy gradient-based reinforcement learning approach has shown two advantages over the cross-entropy loss optimization approach: (1) avoiding *exposure bias* issue which is about the mismatch in the output distributions created by different train and test time decoding approaches in cross-entropy loss optimization; (2) able to directly optimize the non-differentiable evaluation metrics.

To this end, REINFORCE algorithm (Williams, 1992; Zaremba and Sutskever, 2015) is used to learn a policy p_θ defined by the model parameters θ to predict the next action (tokens in our setup). Specifically, instead of minimizing the negative log-likelihood, we minimize the following loss:

$$L_{\text{RL}} = -\mathbb{E}_{w^s \sim p_\theta}[r(w^s)] \quad (1)$$

where w^s is the sequence of sampled tokens and $r(\cdot)$ is the reward function that measures the quality of w^s. The derivative of this loss function can then be approximated using a single sample along with a bias estimator $\hat{b}$ to reduce variance:

$$\nabla_\theta L_{\text{RL}} = -(r(w^s) - \hat{b}) \nabla_\theta \log p_\theta(w^s) \quad (2)$$

There are several ways to calculate the baseline estimator, and in this work we use the SCST mechanism (Rennie et al., 2017).

Need for a better multi-reward optimization. Often, an RL agent can improve the policy p_θ via multiple reward sources. However, efficient ways of optimizing multiple rewards in a policy gradient-based reinforcement learning setup have been less explored. Previous works have either explored using a weighted combination of multiple rewards (Zhang and Lapata, 2017; Li et al., 2016) or alternate fashion of optimizing multiple rewards inspired via multi-task learning setup (Pasunuru and Bansal, 2018). However, these approaches have a disadvantage of tuning the weights of the rewards combination or using a static tunable mixing ratio while optimizing in an alternate fashion. To this end, we explore multi-reward optimization via a multi-armed bandit approach (Bubeck et al., 2012; Lattimore and Szepesvári, 2019; Burtini et al., 2015). During the training, the bandit explores/exploits the choice of reward functions in order to improve the overall performance of the model. In the remaining part of this section, we discuss various multi-armed bandit-based models for multi-reward optimization (Sec. 3.1), and reward settings (Sec. 3.2). Then, we present

[2] In our initial experiments, we experimented with a few other bandit approaches (UCB, contextual bandit, variants of Exp3, e.g., Exp3-S), but we ended up with our current Exp3 setting due to its performance and stability reasons within the scope of our methods and tasks.

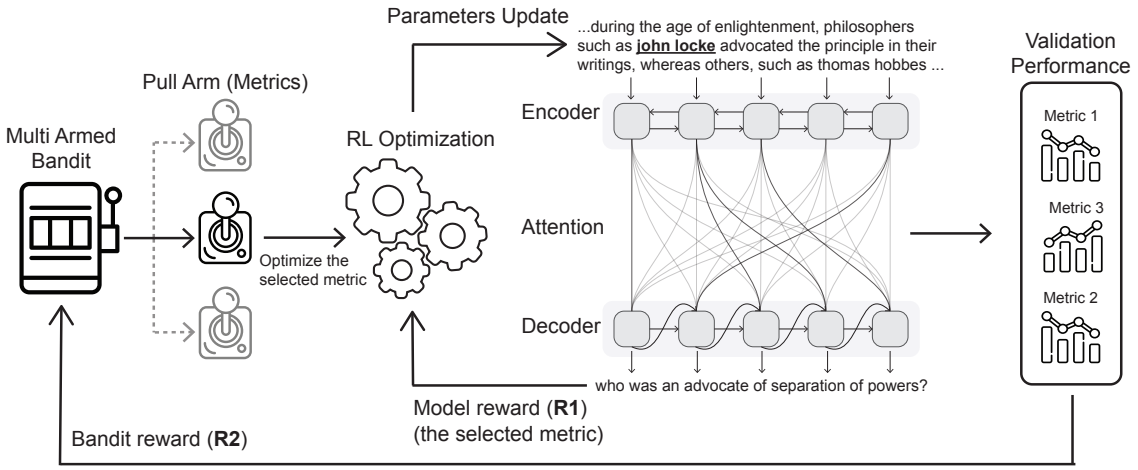

Figure 1: Overview of our multi-armed bandit reward selection framework DORB. At each step, the model outputs are scored based on a reward function (metric), where the choice of the reward function is dynamically controlled by the multi-armed bandit. Then the corresponding optimization is executed based on the chosen reward function. Finally, the observed validation performance metrics are given as feedback to the bandit.

the two novel approaches, namely Single Multi-reward Bandit (SM-Bandit, Sec. 3.3) and Hierarchical Multi-reward Bandit (HM-Bandit, Sec. 3.4).

3.1 Multi-Armed Bandit for Multi-Reward Optimization

Given a set of K candidate actions (arms) $\{a_1, a_2, ..., a_K\}$, the objective of a multi-armed bandit problem is to maximize rewards earned through a sequence of lever pulls (actions). We call this reward as bandit reward. We view the problem of optimizing multiple rewards as a sequential design of experiments (Robbins, 1952), where the bandit's goal is to decide the next arm (loss function) to pull after each round in order to maximize the rewards it earns.

Let $\{R_1, R_2, .., R_K\}$ be a set of different rewards from K sources which can measure the model/policy's performance. To directly maximize the performance of these K rewards, we need to use K different reinforcement learning-based loss functions. Let the loss function for R_i be:

$$L_{\text{RL}_i} = -\mathbb{E}_{w^s \sim p_\theta}[R_i(w^s)] \tag{3}$$

Each of these K loss functions is considered as an arm of the multi-armed bandit (i.e., the arms/joysticks in Fig. 1), where pulling the i^{th} arm will result in optimizing for reinforcement based loss function L_{RL_i} (i.e., in Fig. 1, main model parameters get updated). The goal of the bandit is to explore and exploit different loss functions and maximize its reward (the validation performance of the model, see Fig. 1). One widely studied problem is the trade-off between "exploitation" of the arm with the highest estimated payoff and "exploration" of less known arms. For this, we use the popular Exp3 bandit algorithm (Auer et al., 2002b) (see Appendix A for more details on Exp3).

3.2 Bandit Reward Settings

Note that in this work, we have two sets of rewards: rewards used for optimizing the sequence generation model via policy gradients-based reinforcement learning (R1 in Fig. 1, Sec. 3), and rewards used for the bandit (R2 in Fig. 1). The rewards for the generation model are used to optimize the model w.r.t. the metric of interest, while the rewards for the bandit help the bandit decide which "metric of interest" the generation model should optimize.

In order to maintain consistent magnitude/scale across metric rewards while using them for bandits, we use scaled rewards via the quantiles of rewards history following Graves et al. (2017). Let $\mathbf{R}^t = \{R^i\}_{i=1}^{t-1}$ be the history of unscaled rewards up to time step t. Let q_t^{lo} and q_t^{hi} be the lower and upper quantiles of $\mathbf{R}^t$, respectively.[3] Then, the scaled reward, $\hat{r}^t$ is defined as follows:

$$\hat{r}^t = \begin{cases} 0 & \text{if } R^t < q_t^{lo} \\ 1 & \text{if } R^t > q_t^{hi} \\ \frac{R^t - q_t^{lo}}{q_t^{hi} - q_t^{lo}}, & \text{otherwise} \end{cases} \tag{4}$$

Instead of keeping the entire history of rewards, we use past n rewards from the history.

[3] We set q_t^{lo} and q_t^{hi} to be 20^{th} and 80^{th} quantiles.

Algorithm 1 SM-Bandit Training

Inputs: #rewards: K, #train steps: n_{train}, #steps in bandit round: n_{bandit}
Initialize the Exp3 bandit B with K arms
$a \leftarrow \text{chooseArm}(B)$ ▷ Based on Eqn. 5
$i \leftarrow 0$
while $i < n_{\text{train}}$ **do**
  Sample word sequence w^s from model
  Calculate rewards R^{train} based on w^s
  Optimize model's L_{RL_a} loss using R_a^{train}
  if $i \mod n_{\text{bandit}} == 0$ **then**
    Evaluate model to get R^{val}
    $r \leftarrow \frac{1}{K}\sum_{k=1}^{K} \text{scaled}(R_k^{\text{val}})$ ▷ Based on Eqn. 4
    $\text{updateBandit}(B, a, r)$ ▷ Based on Eqn. 7
    $a \leftarrow \text{chooseArm}(B)$
  end if
  $i \leftarrow i + 1$
end while

3.3 Single Bandit with Multi-Reward

Often, we want to optimize multiple metrics in our RL approach. For this, we have to give a joint reward coming from multiple sources (metrics in our case) to the bandit as a bandit reward. One can easily give the weighted combination of these rewards coming from multiple sources as a reward to the bandit. However, tuning these weights is intractable if the number of reward sources is large. Here, we introduce a new approach called Single Multi-reward Bandit (SM-Bandit), which avoids tuning and uses rewards from multiple sources as feedback to the bandit. Let L_{RL_1}, L_{RL_2}, and L_{RL_3} be the reinforcement learning-based loss functions corresponding to three arms of the bandit: arm_1, arm_2, and arm_3, respectively. If arm_2 is selected at round t, then we optimize for L_{RL_2} and measure the performance of all the unscaled metric scores on the validation set and then calculate the corresponding scaled rewards for each metric. We average over these scaled rewards and give that as a reward to the bandit. The generalization of this reward for K-armed bandit is: $r^t = \frac{1}{K}\sum_{i=1}^{K} \hat{r}_i^t$, where r^t is the bandit reward at round t and $\hat{r}_i^t$ is the scaled reward (Eq. 4) for the metric corresponding to arm_i at round t. This approach allows us to avoid tuning the balancing weights across the metrics that we optimize, and ensure that the bandit is improving all metrics, as the bandit goal is to maximize the average of all metrics. A detailed procedure of SM-Bandit is described in Algorithm 1.

3.4 Hierarchical Bandit with Multi-Reward

The SM-bandit's goal in the previous approach described in Sec. 3.3 is to improve all metrics using a single bandit. In this section, we introduce another bandit-based variant to improve all metrics but by using multiple bandits which are controlled by a controller, called Hierarchical Multi-reward Bandits (HM-Bandit, Fig. 2). The HM-Bandit consists of a single first-level controller (not a bandit, top row in Fig. 2), and K second-level multi-armed bandits (middle row in Fig. 2). The first-level controller's goal is to find the under-performing reward metric, while the second-level bandits' goal is to trigger a specific metric optimizer that will lead to a promising improvement in this specific metric. More intuitively, the first-level controller sets the objective (e.g., ROUGE needs to be improved), while the second-level bandit decides which specific reward function can help accomplish the objective. A detailed procedure of our HM-bandit is described in Algorithm 2. This concept is also loosely related to Bayesian model selection, where it's common to use a hierarchical specification of models (Rasmussen and Williams, 2005).

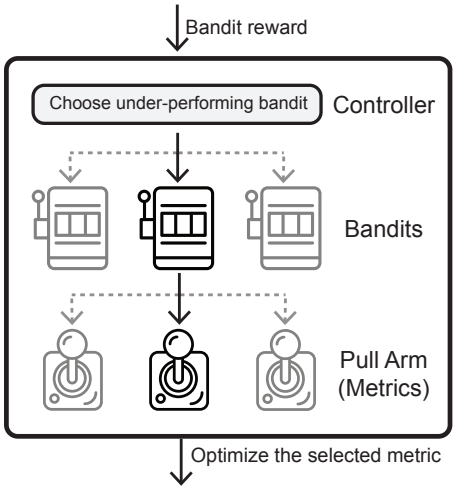

Figure 2: Overview of the hierarchical multi-armed bandit. The first-level has a controller and the second-level has bandits. The controller decides which bandit of the second-level will be pulled. The second-level bandits then decide which metric to use as the reward function during RL optimization.

4 Tasks and Setup

We use question generation and data-to-text generation tasks in our experiments. In this section, we discuss the details on these two tasks along with the experimental setup.

4.1 Question Generation

The goal of the question generation (QG) task is to generate a natural question that can be answered by the given answer span in a context. Recent works have applied seq2seq neural models for QG, e.g.,

Algorithm 2 HM-Bandit Training

Inputs: #rewards: K, #train steps: n_{train}, #steps in bandit round: n_{bandit}, #steps in controller round: $n_{\text{controller}}$
Create the controller C with K bandits
Initialize all bandits, and set $j \leftarrow 0$
$B \leftarrow \text{chooseBandit}(C, j)$ ▷ choose bandit at index j
$a \leftarrow \text{chooseArm}(B)$ ▷ Based on Eqn. 5
$i \leftarrow 0$
while $i < n_{\text{train}}$ **do**
  Sample word sequence w^s from model
  Calculate rewards R^{train} based on w^s
  Optimize model's L_{RL_a} loss using R_a^{train}
  if $i \mod n_{\text{bandit}} == 0$ **then**
    Evaluate model to get R^{val}
    $r \leftarrow \text{scaled}(R_j^{\text{val}})$
    $\text{updateBandit}(B, a, r)$ ▷ Based on Eqn. 7
    $a \leftarrow \text{chooseArm}(B)$
  end if
  if $i \mod n_{\text{controller}} == 0$ **then**
    Evaluate model to get R^{val}
    $j \leftarrow \text{argmin}_k\{\text{scale}(R_k^{\text{val}})\}_{k=1}^{K}$
    $B \leftarrow \text{chooseBandit}(C, j)$
    $a \leftarrow \text{chooseArm}(B)$
  end if
  $i \leftarrow i + 1$
end while

generating the question given answer sentence (Du et al., 2017; Zhou et al., 2017), or the whole paragraph (Du and Cardie, 2018; Song et al., 2018b; Liu et al., 2019a; Zhao et al., 2018; Kim et al., 2019; Sun et al., 2018). Many works also used RL to optimize specific metrics (Song et al., 2018a; Kumar et al., 2019; Yuan et al., 2017). Recently, Zhang and Bansal (2019) proposed semantics-enhanced rewards to improve the QG model, and also used the multi-reward approach proposed by Pasunuru and Bansal (2018) in their RL models.

Baseline. Given a paragraph p, and an answer span a, the goal of the QG model is to generate a question q answering a. We follow the encoder-attention-decoder style architecture (see Fig. 1). The encoder is a bi-directional LSTM-RNN (Hochreiter and Schmidhuber, 1997) with self-attention (Wang et al., 2017), and the decoder is a uni-directional LSTM-RNN with attention (Luong et al., 2015) and pointer (Gu et al., 2016) mechanism, similar to Zhang and Bansal (2019). The input to the model is a concatenation of contextualized word representations (BERT (Devlin et al., 2019)), answer tag embedding (BIO tagging scheme), Part-of-Speech (POS) tag embedding, and Named-Entity (NER) tag embedding.

Rewards. We use ROUGE-L, QPP, and QAP (Zhang and Bansal, 2019) as rewards for this task. QPP is calculated as the probability of the generated question being the paraphrase of the ground-truth question via a classifier trained on Quora Question Pairs. QAP is calculated as the probability of a pre-trained QA model to correctly answer the given generated question as input.

Dataset & Evaluation. We use the SQuAD QG English dataset from Du et al. (2017) for the QG task, derived from SQuAD v1.1 (Rajpurkar et al., 2016), and the test set consists of 10% sampled examples from the training set, as the SQuAD test set is not open. For pre-processing, we do standard tokenization. We report on evaluation metrics including BLEU-4, METEOR, ROUGE-L, Q-BLEU1 (Nema and Khapra, 2018), as well as QPP and QAP (Zhang and Bansal, 2019).

4.2 Data-to-Text Generation

Data-to-text is the task of expressing the components (attributes and values) of meaning representation (MR) as human-readable natural sentences. Previous work in this area include templates (Reiter, 1995), rules (Reiter et al., 2005), pipelines (Reiter, 2007; Reiter and Dale, 1997), probabilistic models (Liang et al., 2009) and more recently end-to-end as well as neural-based methods (Wen et al., 2015; Mei et al., 2016; Dušek and Jurcicek, 2016; Lampouras and Vlachos, 2016; Dušek et al., 2020; Wiseman et al., 2017; Gong, 2018; Chen and Mooney, 2008; Reiter, 2017; Lebret et al., 2016; Distiawan et al., 2018; Gehrmann et al., 2018; Marcheggiani and Perez-Beltrachini, 2018; Guo et al., 2019b; Zhao et al., 2020). In our work, we use the state-of-the-art model from Zhao et al. (2020) as our baseline.

Baseline. Given a set of Resource Description Framework (RDF) triples,[4] the task is to generate a natural language text describing the facts in the RDF data. Following Zhao et al. (2020), we serialize and reorder the RDF data as an intermediate planning setup, and feed the plan into a seq2seq model with attention and copy mechanism.

Rewards. We use BLEU, ROUGE-L, and Entailment-Score (Pasunuru and Bansal, 2018) as rewards. Entailment-Score is calculated based on the probability that the generated sentence is classified as an entailment w.r.t. the ground truth.[5]

[4] Each triple contains a subject, a predicate, and an object.
[5] We use a RoBERTa classifier (Liu et al., 2019b) trained on MultiNLI (Williams et al., 2018) as entailment scorer.

Models	BLEU-4	METEOR	ROUGE-L	Q-BLEU1	QPP	QAP
BASELINES						
Cross-Entropy (Zhang and Bansal, 2019)	17.88	22.38	46.39	49.01	28.83	54.25
ROUGE-RL	18.03	22.55	46.64	49.52	29.09	55.07
QPP-RL	17.90	22.55	46.68	49.50	30.10	55.50
QAP-RL	18.22	22.69	46.65	49.72	30.03	**57.60**
MULTI-REWARD MODELS						
Pasunuru and Bansal (2018)†	18.36	22.55	46.75	49.66	30.03	56.51
Our SM-Bandit†	**18.68**	**22.88**	46.80	**50.02**	**30.15**	56.92
Our HM-Bandit†	18.55	22.82	**46.84**	50.01	30.07	56.78

Table 1: Performance of our baselines and multi-armed bandit-based models on question generation task. † denotes that these models use ROUGE-L, QPP, and QAP rewards during the optimization.

Dataset & Evaluation. We use the WebNLG dataset (Gardent et al., 2017) - a widely used English benchmark for data-to-text generation which focuses on micro-planning involving several subtasks like referring expression generation, aggregation, lexicalization, sentence segmentation, and surface realization. It contains 9,674 unique RDF triple-sets and 25,298 text references, which is divided into train, dev, and test sets.[6] We report all our results on the 'seen' part of the test set. For each sample, the input is a set of up to 7 RDF triples from DBPedia, and the output is their text descriptions. The standard evaluation metrics for this dataset include METEOR[7] (Denkowski and Lavie, 2014), BLEU (Papineni et al., 2002), and TER[8] (Snover et al., 2006). We also report ROUGE-L (Lin, 2004) and Entailment-Score (Pasunuru and Bansal, 2018).

4.3 Training Details

All the hyperparameters are tuned on the validation set for both question generation and data-to-text tasks. We use TITAN X and GeForce GTX 1080 GPUs for all our experiments. For the question generation task, we use two layers for both encoder and decoder. We set the hidden size of LSTM-RNN to 600 and use BERT-based contextual embeddings as input. We use a batch size of 32, encoder maximum length of 512 and decoder maximum length of 50, and maximum gradient clipping of 5. We use Adam optimizer (Kingma and Ba, 2015) with a learning rate of 1e-3 and 1e-6 for the cross-entropy and RL models, respectively. For data-to-text task, we use the same hyperparameters as discussed in Zhao et al. (2020) for the cross-entropy model, e.g., we use Adam with a batch size of 64 and an initial learning rate of 0.001. All RL models are initialized with the best cross-entropy model checkpoint, and use Adam with a learning rate of 1e-6. We refer to Appendix B for full training details.

5 Results and Analysis

In this section, we present the performance of previous work, our cross-entropy baselines, our RL-based baselines, and finally our multi-arm bandit-based models. We start with results on automatic evaluation (Sec. 5.1-5.2). Next, we present results on human evaluation (Sec. 5.3). Finally, we present an interpretable analysis on the bandits (Sec. 5.4).

5.1 Results on Question Generation

Baselines. Table 1 presents results on the question generation dataset for our baselines. We use the previous state-of-the-art work (Zhang and Bansal, 2019) as our cross-entropy baseline. Next, we apply policy gradients-based reinforcement learning (RL) approach, and observe that all these models are better than the baseline in all metrics. Next, we will discuss the multi-reward RL models.

Multi-Armed Bandit Approaches. Finally, we evaluate our two bandit approaches: SM-Bandit and HM-Bandit as described in Sec. 3.3 and Sec. 3.4, respectively. Further, for a fair comparison of our multi-arm bandit-based models, we further implemented multi-reward alternate optimization approach introduced by Pasunuru and Bansal (2018) and considered it as baseline for our multi-reward models.[9,10] This model is slightly better

[6] https://webnlg-challenge.loria.fr/
[7] http://www.cs.cmu.edu/~alavie/METEOR/
[8] http://www.cs.umd.edu/~snover/tercom/

[9] We do not compare with fixed weighted combination of metrics during RL optimization, as finding the optimal weighted combination is exponential complexity (searching among 100 values for n metrics needs 100^n tuning experiments), which we want to avoid via our bandit approach.

[10] We also experimented with the random choice of metrics during optimization. The results on the question generation

Models	BLEU (↑)	METEOR (↑)	TER (↓)	ROUGE-L (↑)	Entailment (↑)
BASELINES					
Cross-Entropy (Zhao et al., 2020)	63.14	44.85	33.97	74.25	99.27
ROUGE-RL	63.35	44.84	33.85	74.29	99.11
BLEU-RL	63.24	44.82	33.94	74.26	99.30
Ent-RL	63.28	44.96	34.03	74.29	99.84
MULTI-REWARD MODELS					
Pasunuru and Bansal (2018)†	63.00	45.03	34.22	74.29	99.56
Our SM-Bandit†	**63.46**	**45.37**	33.59	74.38	100.13
Our HM-Bandit†	63.38	45.34	**33.58**	**74.39**	**100.21**

Table 2: Performance of our baselines and multi-arm bandit-based models on the 'seen' test set of WebNLG data-to-text task. † denotes that these models use ROUGE-L, BLEU, and Entailment rewards during the optimization. For TER metric, lower (↓) is better. For all other metrics, higher (↑) is better.

than single reward-based RL baselines. Table 1 presents the performance of the proposed two bandit models (SM-Bandit and HM-Bandit) on various automatic evaluation metrics, and we observe that on average these models perform much better than the cross-entropy and single reward RL baseline models. Further, our bandit models also perform better than the multi-reward approach proposed by Pasunuru and Bansal (2018), suggesting that our bandit-based models are able to dynamically select the reward to optimize for overall improvement in all the metrics that we want to optimize. Also see discussion of significant improvements in human evaluation in Sec 5.3.

5.2 Results on Data-to-Text Generation

Baselines. Table 2 presents our baselines on the WebNLG data-to-text task. Our cross-entropy model is comparable to the very recent state-of-the-art model (Zhao et al., 2020). Further, we present single reward based RL models with ROUGE-L, BLEU, and Entailment score as rewards, which again perform better than our cross-entropy model. Next, we will discuss multi-reward models.

Multi-Armed Bandit Approaches. Table 2 also presents our multi-armed bandit models (SM-Bandit and HM-Bandit) which simultaneously use ROUGE-L, BLEU, and Entailment score as rewards. Again, we consider the model proposed by Pasunuru and Bansal (2018) as a baseline for multi-reward models. On average, our bandit-based models perform better than all our baselines that are discussed in the above paragraph and also the model based on Pasunuru and Bansal (2018). Also see discussion of significant improvements in human evaluation in Sec 5.3.

task are very close to the baseline model (Pasunuru and Bansal, 2018): 18.31(BLEU), 22.50 (METEOR), 46.75 (ROUGE-L), 49.65 (Q-BLEU1), 30.04 (QPP), 56.56 (QAP). This is expected as the random choice baseline is same as uniform sampling of metrics, which is closer to alternate optimization.

Model	ROUGE	PB (2018)	SMB	HMB
QUESTION GENERATION TASK				
Relevance	4.28	4.40	4.56	4.55
Coherence	4.42	4.48	4.49	4.47
WEBNLG DATA-TO-TEXT TASK				
Relevance	4.61	4.68	4.79	4.81
Coherence	4.75	4.79	4.78	4.80

Table 3: Human evaluation results on QG and WebNLG tasks. ROUGE: ROUGE-L single-reward RL; PB (2018): Pasunuru and Bansal (2018). Our SM-Bandit and HM-Bandit are statistically significantly better than ROUGE and PB models (see Sec. 5.3).

5.3 Human Evaluation

It is shown that RL models can game the metric that we use as the objective function (Paulus et al., 2018). This motivated us to optimize the RL models on multiple metrics simultaneously, thus trying to improve all the metrics and making the RL model hard to game any particular metric. In this section, we validate the superiority of our bandit models via human evaluation studies.

We performed anonymous human evaluation studies using Amazon Mechanical Turk (MTurk). We chose human annotators such that they are located in the USA, have at least 10,000 approved HITs, and have an approval rate of greater than 98%. For both question generation and WebNLG data-to-text, we considered 200 samples for each, and compared ROUGE-L RL, Pasunuru and Bansal (2018), SM-Bandit, and HM-Bandit models by asking the annotators to rate the quality of the generated outputs based on relevance and coherence on 5-

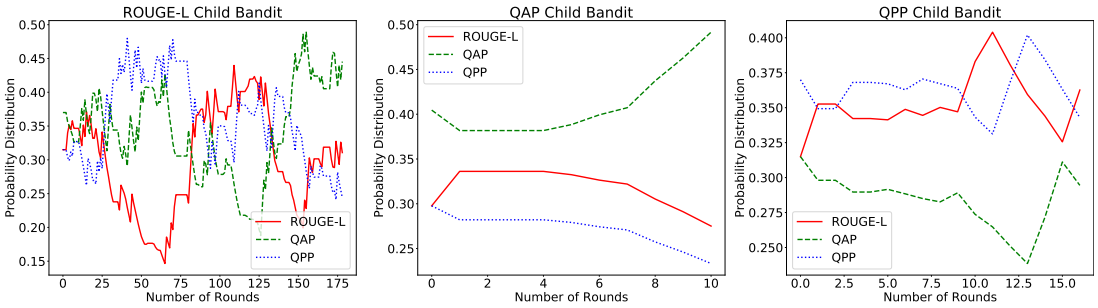

Figure 3: Plots showing the probability distribution of each child bandit of the HM-Bandit model on the QG task.

point Likert scale.[11] Table 3 presents these human evaluation studies. In terms of relevance, our SM-Bandit and HM-Bandit models are significantly better than Pasunuru and Bansal (2018) (p<0.01) and ROUGE-L RL models (p<0.01) on question generation.[12] On data-to-text, in terms of relevance, our SM-Bandit and HM-Bandit models are significantly better than Pasunuru and Bansal (2017a) with p<0.03 and p<0.02, respectively. Also, both bandit models are significantly better than ROUGE-L RL model with p<0.01.

5.4 Interpretable Bandit Analysis

Figure 4 presents the interpretable visualization of the probability distribution of each arm of the SM-Bandit as the training progresses. We observe that each metric has played an important role (as high probability arm) for at least a few rounds over the training trajectory. Also, there are multiple switchings of these metrics over the training trajectory, suggesting that this kind of automatic dynamic switching is important to improve the overall performance of RL models with multiple rewards.

Figure 3 presents the progress of child bandits of HM-Bandit during the training for question generation. As discussed in Sec. 3.4, these child bandits are controlled by a controller that selects the under-performing bandit. We observe that our HM-Bandit mostly used ROUGE-L child bandit for overall improvement in all metrics (as it is the under-performing metric). Further, each child bandit gave more importance to the metric that it wants to improve, e.g., the QAP child bandit gave more importance to the QAP arm. However, there is an exception for the ROUGE-L child bandit, where ROUGE-L arm is not the most important, suggesting that to improve the ROUGE-L metric other RL loss functions (QAP and QPP) are also useful.

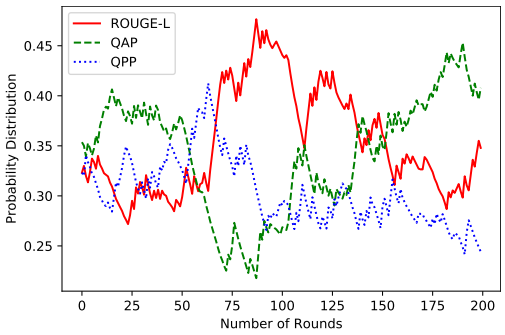

Figure 4: Plot showing the probability distribution of each arm of the SM-Bandit on question generation task.

6 Conclusion

We presented novel approaches for dynamically optimizing multiple reward metrics simultaneously via multi-armed bandit approach in the context of language generation. We described two such mechanisms, namely single bandit and hierarchical bandit with multiple rewards. We conducted experiments on two challenging language generation tasks: question generation and data-to-text generation, and our method achieved strong improvements based on human evaluation over previous approaches. We further presented interpretable analysis on our bandit methods.

Acknowledgments

We thank the reviewers for their helpful comments, Shiyue Zhang for the code to build QPP and QAP rewards, and Chao Zhao for the code of their SOTA WebNLG model. This work was supported by DARPA YFA17-D17AP00022, NSF-CAREER Award 1846185, ONR Grant N00014-18-1-2871, and Microsoft PhD Fellowship. The views contained in this article are those of the authors and not of the funding agency.

[11] For question generation, relevance is defined as how clearly the generated question will be able to point to the right answer, given an input paragraph as context. For WebNLG data-to-text, relevance is defined as how related is the generated description w.r.t. the given RDF data such as mentioning the facts. For both tasks, coherence is based on the logic, readability, and fluency of the generated question or description.

[12] We use bootstrap test (Efron and Tibshirani, 1994; Noreen, 1989) for calculating the statistical significance score.

References

Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. 2009. Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002a. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256.

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. 2002b. The nonstochastic multi-armed bandit problem. *SIAM journal on computing*, 32(1):48–77.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122.

Giuseppe Burtini, Jason Loeppky, and Ramon Lawrence. 2015. A survey of online experiment design with the stochastic multi-armed bandit. *arXiv preprint arXiv:1510.00757*.

Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. In *NAACL*, pages 1662–1675.

Olivier Chapelle and Lihong Li. 2011. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, pages 2249–2257.

David L Chen and Raymond J Mooney. 2008. Learning to sportscast: a test of grounded language acquisition. In *Proceedings of the 25th international conference on Machine learning*, pages 128–135. ACM.

Wei Chen, Yajun Wang, and Yang Yuan. 2013. Combinatorial multi-armed bandit: General framework and applications. In *International Conference on Machine Learning*, pages 151–159.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 675–686.

Sumit Chopra, Michael Auli, and Alexander M Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, pages 376–380.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Bayu Distiawan, Jianzhong Qi, Rui Zhang, and Wei Wang. 2018. Gtr-lstm: A triple encoder for sentence generation from rdf data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1627–1637.

Xinya Du and Claire Cardie. 2018. Harvesting paragraph-level question-answer pairs from wikipedia. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1907–1917.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1342–1352.

Kevin Duh, Katsuhito Sudoh, Xianchao Wu, Hajime Tsukada, and Masaaki Nagata. 2012. Learning to translate with multiple objectives. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–10.

Ondřej Dušek and Filip Jurcicek. 2016. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 45–51.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2020. Evaluating the state-of-the-art of end-to-end natural language generation: The E2E NLG Challenge. *Computer Speech & Language*, 59:123–156.

Bradley Efron and Robert J Tibshirani. 1994. *An introduction to the bootstrap*. CRC press.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The WebNLG challenge: Generating text from rdf data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133.

Sebastian Gehrmann, Falcon Dai, Henry Elder, and Alexander M Rush. 2018. End-to-end content and plan selection for data-to-text generation. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 46–56.

Heng Gong. 2018. Technical report for e2e nlg challenge. *E2E NLG Challenge System Descriptions*.

Alex Graves, Marc G Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. 2017. Automated curriculum learning for neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1311–1320. JMLR. org.

Jiatao Gu, Kyunghyun Cho, and Victor OK Li. 2017. Trainable greedy decoding for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1968–1978.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 140–149.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. 2018. Dynamic multi-level multi-task learning for sentence simplification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 462–476.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. 2019a. AutoSeM: Automatic task selection and mixing in multi-task learning. 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 3520–3531.

Zhijiang Guo, Yan Zhang, Zhiyang Teng, and Wei Lu. 2019b. Densely connected graph convolutional networks for graph-to-sequence learning. *Transactions of the Association for Computational Linguistics*, 7:297–312.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. 1996. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285.

Yanghoon Kim, Hwanhee Lee, Joongbo Shin, and Kyomin Jung. 2019. Improving neural question generation using answer separation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6602–6609.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Vishwajeet Kumar, Ganesh Ramakrishnan, and Yuan-Fang Li. 2019. Putting the horse before the cart: A generator-evaluator framework for question generation from text. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 812–821.

Branislav Kveton, Csaba Szepesvari, Sharan Vaswani, Zheng Wen, Tor Lattimore, and Mohammad Ghavamzadeh. 2019. Garbage in, reward out: Bootstrapping exploration in multi-armed bandits. In *International Conference on Machine Learning*, pages 3601–3610.

Gerasimos Lampouras and Andreas Vlachos. 2016. Imitation learning for language generation from unaligned data. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1101–1112.

Tor Lattimore and Csaba Szepesvári. 2019. *Bandit Algorithms*. Cambridge University Press (preprint).

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213.

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*, pages 1192–1202.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. 2018. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185):1–52.

Percy Liang, Michael I Jordan, and Dan Klein. 2009. Learning semantic correspondences with less supervision. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 91–99. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Proceedings of Workshop on Text Summarization Branches Out, Conference Workshop of ACL*.

Bang Liu, Mingjun Zhao, Di Niu, Kunfeng Lai, Yancheng He, Haojie Wei, and Yu Xu. 2019a. Learning to generate questions by learningwhat not to generate. In *The World Wide Web Conference*, pages 1106–1118.

Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. 2017. Improved image captioning via policy gradient optimization of spider. In *Proceedings of the IEEE international conference on computer vision*, pages 873–881.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

William G Macready and David H Wolpert. 1998. Bandit problems and the exploration/exploitation tradeoff. *IEEE Transactions on evolutionary computation*, 2(1):2–22.

Diego Marcheggiani and Laura Perez-Beltrachini. 2018. Deep graph convolutional encoders for structured data to text generation. In *INLG*.

Hongyuan Mei, TTI UChicago, Mohit Bansal, and Matthew R Walter. 2016. What to talk about and how? selective generation using lstms with coarse-to-fine alignment. In *Proceedings of NAACL-HLT*, pages 720–730.

Andreas Merentitis, Kashif Rasul, Roland Vollgraf, Abdul-Saboor Sheikh, and Urs Bergmann. 2018. A bandit framework for optimal selection of reinforcement learning agents. In *NeurIPS 2018 Workshop on Deep Reinforcement Learning*.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2018. Regularizing and optimizing LSTM language models. In *International Conference on Learning Representations*.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290.

Preksha Nema and Mitesh M Khapra. 2018. Towards a better metric for evaluating question generation systems. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3950–3959.

Graham Neubig and Taro Watanabe. 2016. Optimization for statistical machine translation: A survey. *Computational Linguistics*, 42(1):1–54.

Eric W Noreen. 1989. *Computer-intensive methods for testing hypotheses*. Wiley New York.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Ramakanth Pasunuru and Mohit Bansal. 2017a. Multi-task video captioning with video and entailment generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1273–1283.

Ramakanth Pasunuru and Mohit Bansal. 2017b. Reinforced video captioning with entailment rewards. In *EMNLP*.

Ramakanth Pasunuru and Mohit Bansal. 2018. Multi-reward reinforced summarization with saliency and entailment. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, volume 2, pages 646–653.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *ICLR*.

Carl Edward Rasmussen and Christopher K. I. Williams. 2005. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

Ehud Reiter. 1995. NLG vs. templates. In *Proc of the Fifth European Workshop on Natural-Language Generation*.

Ehud Reiter. 2007. An architecture for data-to-text systems. In *Proceedings of the Eleventh European Workshop on Natural Language Generation*, pages 97–104. Association for Computational Linguistics.

Ehud Reiter. 2017. You need to understand your corpora-the weathergov example. *Blogpost-https://ehudreiter. com/2017/05/09/weathergov*.

Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–87.

Ehud Reiter, Somayajulu Sripada, Jim Hunter, Jin Yu, and Ian Davy. 2005. Choosing words in computer-generated weather forecasts. *Artificial Intelligence*, 167(1-2):137–169.

Zhou Ren, Xiaoyu Wang, Ning Zhang, Xutao Lv, and Li-Jia Li. 2017. Deep reinforcement learning-based image captioning with embedding reward. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 290–298.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1179–1195. IEEE.

Herbert Robbins. 1952. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *CoRR*.

Baskaran Sankaran, Anoop Sarkar, and Kevin Duh. 2013. Multi-metric optimization using ensemble tuning. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 947–957.

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

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, pages 3776–3784.

Amr Sharaf and Hal Daumé III. 2019. Meta-learning for contextual bandit exploration. *arXiv preprint arXiv:1901.08159*.

Sahil Sharma and Balaraman Ravindran. 2017. Online multi-task learning using active sampling. In *ICLR 2017 Workshop*.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*.

Linfeng Song, Zhiguo Wang, and Wael Hamza. 2018a. A unified query-based generative model for question generation and question answering. In *NAACL*.

Linfeng Song, Zhiguo Wang, Wael Hamza, Yue Zhang, and Daniel Gildea. 2018b. Leveraging context information for natural question generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 569–574.

Xingwu Sun, Jing Liu, Yajuan Lyu, Wei He, Yanjun Ma, and Shi Wang. 2018. Answer-focused and position-aware neural question generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3930–3939.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2015. Sequence to sequence-video to text. In *Proceedings of the IEEE international conference on computer vision*, pages 4534–4542.

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

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. In *Proceedings of ICML Deep Learning Workshop*.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015b. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 189–198.

Xin Wang, Wenhu Chen, Jiawei Wu, Yuan-Fang Wang, and William Yang Wang. 2018. Video captioning via hierarchical reinforcement learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4213–4222.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, pages 2048–2057.

Xingdi Yuan, Tong Wang, Caglar Gulcehre, Alessandro Sordoni, Philip Bachman, Saizheng Zhang, Sandeep Subramanian, and Adam Trischler. 2017. Machine comprehension by text-to-text neural question generation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 15–25.

Wojciech Zaremba and Ilya Sutskever. 2015. Reinforcement learning neural turing machines-revised. *arXiv preprint arXiv:1505.00521*.

Shiyue Zhang and Mohit Bansal. 2019. Addressing semantic drift in question generation for semi-supervised question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2495–2509.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 584–594.

Chao Zhao, Marilyn Walker, and Snigdha Chaturvedi. 2020. Bridging the structural gap between encoding and decoding for data-to-text generation. In *ACL*.

Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. 2018. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3901–3910.

Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. 2018. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8739–8748.

Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. 2017. Neural question generation from text: A preliminary study. In *National CCF Conference on Natural Language Processing and Chinese Computing*, pages 662–671. Springer.

A Exp3 Bandit Algorithm

A stochastic bandit is completely determined by the distribution of rewards of respective actions. However, it will be hard to argue that rewards are truly randomly generated, and even if they are randomly generated, the rewards could be correlated over time (e.g., the validation performance at the next step will be correlated with validation performance at this time step). Taking all these factors into account makes the algorithm overly complicated, and thus an alternative is to assume nothing about the underlying mechanism that generates the rewards while still trying to achieve the lowest possible regret. This is called the adversarial bandit problem, where the goal is to design an algorithm that keeps the regret small regardless of what rewards are assigned to actions.

Exponential-weight algorithm for Exploration and Exploitation, or Exp3 (Auer et al., 2002b), was created to handle the non-stochastic adversarial bandit problem. We use this algorithm in our DORB framework. Exp3 works by maintaining a set of weights for each candidate action, and the weights are used to decide randomly which action to take next. The empirical observation is fed back to the bandit to either increase or decrease the relevant weights. The algorithm also has a hyperparameter $\gamma \in [0, 1]$ that decides the probability to take action uniformly at random. Specifically, at round t, the bandit picks action (arm) i among K arms based on the arm selection probability which is defined as follows:

$$p_t(i) = (1-\gamma)\frac{w_{t,i}}{\sum_{j=1}^{K} w_{t,j}} + \frac{\gamma}{K} \tag{5}$$

where the weights $w_{t,i}$ are updated based on the observed bandit reward r_t^B:

$$\hat{r}_{t,j}^B = \begin{cases} r_t^B / p_t(i) & \text{if } j = i \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

$$w_{t+1,i} = w_{t,i} \exp(\gamma \hat{r}_{t,i}^B / K) \tag{7}$$

B Training Details

All the hyperparameters are tuned on the validation set for both question generation and data-to-text tasks. We use TITAN X and GeForce GTX 1080

GPUs for all our experiments, where all our RL models roughly take 1 day to train on a single GPU.

For the question generation task, we use two layers for both bi-directional encoder and uni-directional decoder. We set the hidden size of LSTM-RNN to 600 and use BERT-based contextual embeddings as input instead of word embeddings. The number of parameters in our model is 33.3 million. We use a batch size of 32, encoder maximum length of 512 and decoder maximum length of 50, and maximum gradient clipping of 5. We use Adam optimizer (Kingma and Ba, 2015) with a learning rate of 1e-3 and 1e-6 for cross-entropy model and RL models, respectively. We use a dropout of 0.3 for the cross-entropy model and no dropout for RL models. For multi-reward bandit models, we set the bandit coefficient (γ) to 0.1, and each round of the bandit consists of optimization of 100 mini-batches of training data. For HM-Bandit, we set the controller round size to 300 mini-batches. We consider the following short hyperparameters ranges and manually tune on: learning rate in the range [1e-5, 1e-7]; bandit coefficient in the range [0.01, 0.5]; bandit round - {10, 100}; and controller round size - {30, 300}.

For WebNLG data-to-text task, we first serialize and reorder the RDF data as an intermediate planning setup, and then feed the plan into an encoder-attention-decoder style architecture with copy mechanism, to generate the text describing the RDF data. We use same hyperparameters as discussed in Zhao et al. (2020) for the cross-entropy model, e.g., we use Adam with a batch size of 64, initial learning rate of 0.001, and a dropout of 0.3. All RL models are initialized with the best cross-entropy model checkpoint, and use Adam with a learning rate of 1e-6. We do not use dropout for RL models. The number of parameters in our model is 5.9 million. For multi-reward bandit models, we set the bandit coefficient (γ) to 0.15, and each round of the bandit consists of optimization of 10 mini-batches of training data. For HM-Bandit, we set the controller round size to 30 mini-batches. We consider the following short hyperparameters ranges and manually tune on: learning rate in the range [1e-5, 1e-7]; bandit coefficient in the range [0.01, 0.5]; bandit round - {10, 100}; and controller round size - {30, 300}.