Training Value-Aligned Reinforcement Learning Agents Using a Normative Prior

Md Sultan al Nahian ©, Spencer Frazier, Mark Riedl, and Brent Harrison ©

Abstract—Value alignment is a property of intelligent agents wherein they solely pursue non-harmful behaviors or human-beneficial goals. We introduce an approach to value-aligned reinforcement learning (RL), in which we train an agent with two reward signals: a standard task performance reward plus a normative behavior reward. The normative behavior reward is derived from a value-aligned prior model that we train using naturally occurring stories. These stories encode societal norms and can be used to classify text as normative or nonnormative. We show how variations on a policy shaping technique can balance these two sources of reward and produce policies that are both effective and perceived as more normative. We test our value-alignment technique on three interactive text-based worlds; each world is designed specifically to challenge agents with a task as well as provide opportunities to deviate from the task to engage in normative and/or altruistic behavior.

Impact Statement—Value-aligned autonomous systems exhibit values that align with those of humans. This should enable them to safely interact with people, thus increasing their general applicability. Creating value-aligned systems is hard because it can be difficult to specify societal values in a form suitable for training agents. In this work, we show that children’s stories can provide a robust value signal that can be used to train reinforcement learning (RL) agents. We show that transformers can achieve up to 90% accuracy in classifying normative and nonnormative behavior using children’s stories and that this signal can be used to train RL agents in a variety of learning environments. The work presented here is one of the first methods for performing practical value alignment. We believe this will lower the barrier for creating these types of agents, thus making it more likely for these techniques to be adopted by the general public.

Index Terms—Autonomous agents, natural language processing, reinforcement learning (RL).

I. INTRODUCTION

VALUE alignment is a desirable property of an intelligent agent, indicating that it can only pursue goals and behaviors that are beneficial to humans [1], [2], [3], [4]. Russell [5] and others have argued that value alignment is one of the most important tasks facing artificial intelligence (AI) researchers today. Value-aligned intelligent systems, however, are hard to build. As argued by Soares and Fallenstein [1], it is nontrivial to directly specify values; there are infinitely many undesirable outcomes in an open world.

Machine learning-based approaches to value alignment have largely relied on learning from observations, demonstrations, preferences, or other forms of imitation learning [6], [7], [8], [9], [10]. Values can thus be cast as preferences over action sequences, and preference learning can be formulated as reward learning or imitation learning [11]. There are a number of challenges faced by value alignment via imitation learning that are of relevance to this work.

1) Demonstrations do not always generalize concepts beyond the scope of the observation.
2) It can be time-consuming to provide sufficient demonstrations, and if the agent is learning online, it can be performing harmful actions until learning is complete.
3) Lastly, it can be difficult for humans to provide high-quality demonstrations that exemplify certain values, especially those related to negation or not doing something.

In situations where imitation learning is difficult to achieve, such as those above, we propose that a strong prior belief over the quality of certain actions or events can complement imitation learning-based approaches. A strong prior for value-aligned actions may replace the need for imitation learning or, more likely, make it easier for an imitation learner to align itself with values.

How can we acquire this strong prior? One solution is to learn this prior through stories [12]. Stories contain examples of normative and nonnormative behavior [13]. We define normativity as behavior that conforms to expected societal norms and contracts, whereas nonnormativity aligns with values that deviate from these expected norms.

In this work, explore how a strong prior can be best learned from naturally occurring story corpora. First, one must be able to reason about the context of individual sentences. To address this, first, we turn to language modeling techniques that can extract contextual semantics from sentences. Second, there is presently a lack of readily available, labeled datasets with
Goofus identified a children’s instructional comic strip named Gallant (G&G) (see Fig. 1). Using this prior, we can provide guidance to reinforcement learning (RL) agents without needing to demonstrate normative behavior. The normative reward is thus an intrinsic behavioral signal while the environmental reward is an extrinsic behavioral signal. In this work, we explore many different ways of combining these intrinsic and environmental reward signals and find that policy shaping [14], [15] is more effective in balancing normative behavior and environmental task behavior compared to other techniques such as summing reward signals. Policy shaping trains an RL agent on a regular environmental reward but uses a secondary criterion to re-rank action choices at every step to bias the agent away from certain courses of action. We update policy shaping for deep RL agents in which a noisy normative action classifier provides the shaping signal. To evaluate different RL techniques, we created a suite of three virtual simulation environments, each of which emulates a situation where an agent must make tradeoffs between environmental reward and intrinsic normative reward.

To summarize, our contributions are as follows.

1) We introduced an alternative approach for AI value alignment, proposing the construction of a prior knowledge model of human values by utilizing children’s stories.

2) We created a text corpus consisting of textual descriptions of social behaviors to enable the construction of a prior knowledge model.

3) We also created two additional text corpora to assess the transfer ability of the prior knowledge model.

4) We proposed RL techniques aimed at training value-aligned agents by incorporating the prior knowledge model.

5) We finally developed test environments to evaluate the effectiveness of value-aligned agents.

II. RELATED WORK

Past approaches to value alignment include learning from expert demonstrations [16], [17], preference learning [18], [19], imitation [8], and inverse RL [20]. Cooperative inverse RL [21], for example, works to derive the reward function exhibited by a human for some task. These methods are costly in terms of the amount of human input required to train the model. These approaches assume that values are latent within people but can be teased out in the form of a reward from which an agent can learn. As with any problem with a sparse or expensive to acquire signal, there is a need for a strong prior to assure transferability [22].

Learning from stories [12], [23] is similar to learning from demonstration, except the demonstrations are replaced by natural language stories; an RL agent extracts a reward signal from the stories to perform more human-like action sequences. It was shown that agents could learn to avoid nonnormative behavior whenever possible. However, the stories used were crowdsourced instead of using a naturally occurring corpus and thus still expensive. Our work differs by focusing on value alignment as learning a prior and using this prior to learn a value-aligned policy instead of directly learning a value-aligned policy. Our work complements LfS and other approaches involving learning from demonstration or imitation learning by providing a means of a priori biasing the agent toward certain actions.

The most similar work is that by Ziegler et al. [24] in which the transformer-based language model, GPT-2, is fine-tuned to learn preferences for generating sentences. While sentiment is not the same as values, it shows that language models can be trained from human preference data. Delphi [25] is one of the recent efforts to embed human values information in transformer-based language models. The dataset they have created to train the model was crowdsourced from online platforms which consist of many inappropriate and biased examples. Therefore, Delphi often makes improper and biased moral judgments. In contrast, we create our dataset from children’s comic books that are meant to teach social norms and values. Thus, it was carefully curated to avoid biased and inappropriate social behavior. In our work, we also introduce techniques that shape the behavior of RL agents toward value-aligned agents using value-aligned language models as a prior knowledge model.

III. BACKGROUND

In our proposed method for training value-aligned agents, we utilize RL. RL is a subfield of machine learning that aims to find optimal solutions for sequential decision-making problems through iterative learning from experience [26]. Specifically, RL is intended to be used on Markov decision processes (MDPs). An MDP is formally defined by a tuple \((S, A, P, R, \lambda)\), where \(S\) represents the set of states in the environment, \(A\) denotes the set of available actions, \(P\) is the state transition probability function, \(R\) is the reward function, and \(\lambda(0 < \lambda < 1)\) represents the discount factor that determines the importance of future rewards. An RL algorithm aims to find an optimal behavior policy \(\pi\) that maximizes the expected return

\[
q_\pi(s, a) = E_\pi \left[ \sum_{k=0}^{\infty} \lambda^k R_{t+k+1} | S_t = s, A_t = a \right].
\]
Advantage actor-critic (A2C) architectures for RL have been found to be effective for playing text-based games [27]. Actor-critic is an RL method that makes use of two neural networks: an actor network that learns a probability distribution over the actions to take in a given state and a critic network that learns the value function for an action in a given state. At each timestep, \( s_t \) represents the state as an input to the actor network \( \pi_\theta(s_t, a) \) and the critic network \( \hat{q}_w(s_t, a) \) where \( a \) represents the action taken by the agent. \( \theta \) and \( w \) are weights of the actor and critic networks, respectively. The actor network’s policy update is

\[
\Delta \theta = \alpha \nabla \theta (log \pi_\theta(s, a)) \hat{q}_w(s, a)
\]

(2)

where \( \hat{q}_w(s, a) \) is a \( q \)-based approximation function of the action’s value. The critic’s update function is given by

\[
\Delta w = \beta (R(s, a) + \gamma \hat{q}_w(s_{t+1}, a_{t+1}) - \hat{q}_w(s_t, a_t))
\]

\[
\times \nabla_w \hat{q}_w(s_t, a_t).
\]

(3)

\( \alpha \) and \( \beta \) represent different learning rates for each model.

For A2C, which we use in this work, this value function is replaced with an advantage function, which compensates for the high degree of variability in value-based RL methods.

IV. METHODS

We aim to introduce a method to train an RL agent to distinguish between socially normative and nonnormative actions and accomplish its goal by performing normative actions while also optimizing for reward. We propose that a prior model that has knowledge of socially normative/nonnormative actions can be used to aid in the RL process. Our proposed method of training a value-aligned agent consists of two steps. First, we build a normative prior model to characterize an action as normative or nonnormative. Then, we use techniques to integrate the normative prior model into the RL process. In the following section, we provide further details on these general steps.

A. Prior Model for Value-Aligned Agent

The first step of creating a value-aligned agent is to construct a model that will serve as a prior belief over the normativity and nonnormativity of various actions as described by text. We build this model using a dataset of normative behavioral natural language examples where we seek to show that the trained model can: 1) identify socially normative behavior and 2) transfer that knowledge to previously unseen examples of behavior. In doing so, we train multiple state-of-the-art language models with a training corpus manually created from the children’s comic strip, G&G, and evaluate how well they transfer to other domains.

1) Datasets: We describe the G&G training corpus, a source of textual descriptions of everyday life situations and ground-truth labels of normative and nonnormative behavior. To show how well models trained on G&G transfer to other tasks, we collect two other datasets that describe various scenarios, which are labeled via crowdsourcing.

a) G&G: For this work, we curated the G&G corpus, composed of excerpts taken from the popular children’s comic strip of the same name. G&G (Fig. 1) is a children’s comic strip that features two main characters, Goofus and Gallant, who are depicted in common everyday scenarios that young children might find themselves in. These comics are meant to illustrate the proper way to navigate a situation and the improper way to navigate the situation based on which character is performing the action. Gallant is meant to act “properly” or in a socially acceptable way. In contrast, Goofus is meant to navigate the situation “improperly” or in a way that violates social conventions or norms.

To better ensure that our machine learning models learn relevant social norms, we have curated a corpus of G&G comics that consist only of recent comics from 1995 to 2017. Since we only use text to train our model, we extract only the text from each comic panel. We then remove explicit references to G&G by replacing their names with pronouns like “he,” “she,” or “they.” This provides us with 1387 sentences. For all of the experiments with this corpus, we use a training set consisting of 50% of the corpus and a test set of the remaining 50% of the corpus.

b) Plotto dataset: Plotto is a book written to help provide inspiration and guidance to potential writers by providing a large library of thousands of predetermined narrative events, called plot points, commonly found in fiction. Within each plot point, there are one or more character slots with one character always being the primary actor/actress. The corpus was extracted from the book with the aid of open-source software described in [28].

In Plotto, there are 1462 plot points provided. This book was originally published in 1928 and contains several plot events which are overtly racist or misogynistic. For our experiments, we removed these plot events, which reduced the total number of plot points available from 1462 to 900.

To test transfer on this dataset, we require normative/nonnormative labels for each plot event. We crowdsourced labels via TurkPrime [29], a service that manages Amazon Mechanical Turk tasks with US-based workers. We designed a survey in which participants are asked to label each phrase extracted from Plotto plot points as normative or nonnormative. Plot points receiving more than one dissenting classification were discarded, and the remaining ones were given a label based on consensus. After this process, the corpus contained 555 phrases subsequently used in our transfer experiments.

c) Science fiction summaries dataset: To further test the transfer capabilities of our trained machine learning models, we used a second, open-source dataset composed of plot summaries taken from fan wikis for popular science fiction shows and movies [30]. In this corpus, we make the assumption that each sentence encodes at least one plot event in the overall story. To curate this corpus, we first manually extracted sentences containing character-driven events. During this process, we identified that some sentences actually encode multiple events and contain both normative and nonnormative behaviors. In these cases, we manually divided the sentence into multiple separate events. After this manual extraction, this corpus contained 800
story events. As with the G&G dataset, we replace character names with pronouns. To label the plot events, we followed a procedure similar to that used to tag the *Plotto* dataset. After this process, our science fiction corpus contained 445 annotated sentences with consensus.

2) Models: Using the text of the G&G corpus, we have trained binary classifiers that can classify events in a story as normative or nonnormative. The classifiers take a single sentence as input, and the output is whether the sentence contains normative behavior or a nonnormative behavior. We used four different machine learning techniques to build the classifiers: 1) bidirectional long short-term memory (BiLSTM), 2) deep pyramid CNN (DPCNN), 3) BERT, and 4) XLNet.

The BiLSTM [31] works as follows. An input sentence is encoded using a bidirectional multilayer long short-term memory (LSTM) cell having two layers with a size of 512. Pretrained GloVe word embeddings are used to embed the input sentence as input, and the output is whether the sentence contains normative behavior or a nonnormative behavior. We used four different machine learning techniques to build the classifiers: 1) bidirectional long short-term memory (BiLSTM), 2) deep pyramid CNN (DPCNN), 3) BERT, and 4) XLNet.

The BiLSTM [31] works as follows. An input sentence is encoded using a bidirectional multilayer long short-term memory (LSTM) cell having two layers with a size of 512. Pretrained GloVe word embeddings are used to embed the input sentence before passing it through the LSTM layer. The hidden state of the LSTM layer is passed through a fully connected (FC) layer followed by a classification layer to make the label prediction. The dimension of the FC layer is $4H \times 512$, and the classification layer is $512 \times K$, where $H$ is the hidden state size of the LSTM cell which is 512 and $K$ is the number of classes.

Using sentiment as a classification signal is a common strategy for performing binary classification on text corpora. DPCNNs [32] were originally designed for sentiment classification and achieved state-of-the-art sentiment classification results, so we explore how they perform in identifying normative behavior. A simple network architecture achieves the best accuracy with 15 weight layers. We re-trained a DPCNN on the G&G dataset. No pretrained word embeddings were used as the network applies text region embeddings enhanced by unsupervised encodings [33].

Bidirectional encoder representations from transformers (BERT) [34] is a transformer that makes use of an attention mechanism to learn contextual relations between words (or sub-words) in a text. It achieves strong results on many tasks through its bidirectionality, enabled by token masking. We utilize BERT’s binary classification mode. The [CLS] token is omnipresent within the BERT model but only active for classification. The final hidden state of the [CLS] token is taken as the pooled representation of the input text. This is fed to the classification layer which has a dimension of $H \times K$, where $K$ is the number of classes and $H$ is the size of the hidden state. Class probabilities are computed via softmax.

XLNet [35] is a generalized autoregressive pretrained model based on the state-of-the-art autoregressive language model TransformerXL [36], which removes MASK tokens while incorporating permutation language modeling to capture the bidirectional context. We utilize XLNet for classification by following the same procedure used for BERT.

B. Training Value-Aligned Agent Using the Prior Model

Training RL agents with environmental reward alone may result in behavior that humans would consider nonnormative if the greatest expected environmental reward is achieved by performing behaviors that deviate from expected norms. However, if an agent is capable of generating an intrinsic normative reward, then it may learn to make tradeoffs that incorporate normative behaviors. We propose a set of techniques to incorporate the normative prior model introduced in the previous step to help guide RL.

1) Environment Preliminaries: For each state in a TextWorld environment, an RL agent receives an observation consisting of: a) a description text of the current state, b) items in inventory, c) facts about the state of objects in the environment (e.g., “A drawer is open”), and d) previous reactive text (e.g., “You can’t go west”) if any. TextWorld additionally provides a set of admissible actions, which are actions that can be executed in the current state. We allow our agents to access the list of admissible actions and choose from them instead of having to generate a command word token by word token. After an action is taken in timestep $t$, the agent increments to timestep $t+1$ and TextWorld provides an environmental reward $R_{t+1}^\text{env}$, which may be zero.

We augment the standard TextWorld environment to use action elaborations. Each admissible action that TextWorld provides to the agent is accompanied by a longer descriptive text. The descriptive text of the taken admissible action is selected randomly and uniformly from among three crowdsourced elaboration texts at each step. This elaboration text serves two purposes. First, the GG normative model operates on natural language text sequences, so action elaborations give the model more information to use in its classification. Second, since it is crowdsourced, it is authored by neutral sources, which mitigates the possibility of experimental bias.

2) Agent Implementations: Our proposed technique to train a value-aligned agent is based on the A2C architecture. In our experiments, the actor network outputs a distribution over admissible actions of a given state of the environment. An action is sampled from this distribution and passed to the environment for execution. The agent then receives environment reward $R_{t+1}^\text{env}$. Typically, the only reward an A2C agent receives is the environment reward. In this work, we augment this environmental reward with a signal generated from the normative prior model.

The normative prior model, GG, receives the natural language elaboration of the chosen action and outputs a distribution of unnormalized log probabilities from the final dense layer of the network. Specifically, the normative prior produces two logits $L_{\text{norm}}$ and $L_{\text{norm}}$ for the belief that the input is normative and the belief that the input is nonnormative, respectively. Note that the GG model is only fine-tuned on the G&G dataset and all experiments are effectively zero-shot transfer to the three TextWorld environments.

In order to understand how best to make use of the normative prior model, we propose multiple approaches for how to incorporate the outputs of the normative prior to update the agent’s policy. We will discuss each of these in detail in the following sections.

a) GG-pos: This agent is an A2C agent that adds the normative prior’s normative class logit values $L_{\text{norm}}$ to the environmental reward received for the action chosen by A2C, specifically

$$R_{t+1} = R_{t+1}^\text{env} + L_{\text{norm}}. \quad (4)$$
The agent simply receives more reward when the action is judged to be normative; it is the simplest means of combining two rewards. One thing to note about this approach is that it fundamentally changes the optimal policy associated with the learning environment since it modifies the reward that an agent receives. Thus, the agent is now optimizing for both environmental reward and reward generated by the normative prior. It is difficult, however, to determine exactly how this will impact behavior as it depends on the logit values obtained from the normative prior. This will, however, cause the agent to prefer policies comprised of normative behaviors unless the environmental reward received is much higher than the normative logit values.

b) GG-mix: This agent is an A2C agent that applies the combined logits from GG, unnormalized log probabilities for the normative and nonnormative classes, to the environmental reward. If the normative prior is equally certain about the normativity of the input, they cancel each other out, specifically

\[ R_{t+1} = P_{t+1}^{\text{GG}} + (L_{\text{norm}} - L_{\text{nonnorm}}). \] (5)

Note that GG-mix incorporates information about the probability that an action is nonnormative. Thus, if \( L_{\text{norm}} \) is greater than \( L_{\text{nonnorm}} \), then a negative value will be added to the environmental reward to calculate the overall reward associated with an action.

Like GG-pos, this strategy alters the optimal policy associated with an MDP. Also, it is difficult to determine how this will affect agent behavior because it depends on the scale of probabilities being produced by the normative prior model. The intuition behind this method, however, is to discourage the agent from choosing actions that have a high probability of being nonnormative, even if the environmental reward associated with that action is high.

c) GG-shaped: This is a variant of the base A2C architecture implementing policy shaping [14], [15], [37]. Policy shaping agents produce a probability distribution over actions, which is then adjusted by an externally produced source of value information, biasing the agent toward certain actions or states during exploration. Policy shaping was originally introduced to incorporate human action preferences into tabular RL with finite state and action spaces. Lin et al. [38] show that policy shaping can be applied to deep q-learning to incorporate human preferences into the learning process. In this work, we make use of policy shaping with the following two modifications: 1) we use an intrinsic source of value information derived from an action classifier instead of human feedback and 2) we apply value-aligned policy shaping to the A2C RL architecture rather than q-learning.

We sample the distribution of unnormalized log probabilities (logits) over potential actions from the final dense layer of the actor critic network: \([L_{a_1}, ..., L_{a_n}]\). For each admissible action, \(a_i\) is altered by GG’s assessment of the action elaboration

\[ L'_{a_i} = L_{a_i} \times (L_{\text{norm}} - L_{\text{nonnorm}}). \] (6)

\((L_{\text{norm}} - L_{\text{nonnorm}})\) is the policy shaping component that modifies the action probabilities of the A2C network and provides a new distribution. The agent then samples the action from this “reranked” distribution. Since this is more complicated than the standard A2C architecture, we show our new architecture in Fig. 2.

It’s important to note that policy shaping does not fundamentally change the optimal policy associated with the MDP, like GG-pos and GG-mix do. This is because policy shaping only biases exploration. We can say, however, that this process will bias initial exploration toward normative actions. Since we multiply the action probabilities with the normativity score, we expect that actions considered to be normative and have high probabilities according to the actor model to receive highly positive values when combined. Similarly, if an action is considered nonnormative, it will receive a negative value from the normativity model, which will cause the overall evaluation of the action to be low even if the actor model rates the action highly. As the training progresses, this process of reranking admissible actions based on the normative score encourages the model to assign lower probabilities to nonnormative actions and higher probabilities to neutral or normative actions.

V. EXPERIMENTS

To validate our approach, we perform two sets of experiments: 1) experiments on the quality of the normative prior models and 2) experiments on our approaches for training value-aligned agents. For the normative prior model, we investigate how well it can classify normative/nonnormative actions in text corpora and transfer to other domains. Then, we investigate how the normative prior model can aid in the training process of RL agents.

A. Experimental Setup

To evaluate the quality of our normative prior model, we first train several baseline classifiers. The Bi-LSTM and DPCNN are trained on the G\&G training set. We produced several versions of BERT and XLNet models: BERT-Base and XLNet-Base receive no training on G\&G, while BERT-GG and XLNet-GG are fine-tuned on the G\&G training set. All models are tested on a held-out testing set. For the transferability experiment of
B. Evaluation Metrics

Metrics used to evaluate the normative prior models include accuracy, precision (TP/TP + FP), recall (TP/TP + FN), $F_1$-score, and classification quality as determined by the Matthews correlation coefficient (MCC).

To evaluate value-aligned RL agents, we need a way to characterize and assess the differences in behavior. Unlike most RL research, we cannot compare the optimality of the agents as measured by the environmental reward received. Each agent is operating under a slightly different way of computing rewards—for example, GG-pos will always receive more reward per step than GG-mix or GG-shaped. All agents may be highly optimal for their reward functions but behave very differently. To characterize and assess differences in execution behavior, we label a subset of admissible actions as “normative” or “task-oriented” and measure the normalized ratio of normative actions to task-oriented actions the agent takes: $n_{\text{norm}}/(n_{\text{norm}} + n_{\text{task}})$. Task-oriented labels are derived from the minimum set of admissible actions required to complete quests in the world. In two of our TextWorld environments, superhero and playground, these are all actions along the shortest path to completion of the main quest. In the final environment, clerkworld, this set includes moving, taking, and stamping—also the actions required for the shortest main quest completion. Normative actions are the difference between the set of all admissible actions and the task-oriented set, excluding actions that lead to the failure of the main quest. The agents never have access to these ground-truth labels.

C. Experiments: Normative Prior Model

We test our hypothesis that stories contain a great deal of knowledge about sociocultural norms that can be generalized to different situations. To show this, we conduct two experiments. The first experiment seeks to determine the most effective machine learning technique for producing a classification model for descriptions of normative and nonnormative events. This is done by training the machine learning models (introduced in Section IV-A2) on the G&G training corpus and then measuring classification accuracy on the G&G testing set. In the second experiment, we explore how the trained models from the first experiment can transfer to other, unrelated story domains with various amounts of fine-tuning. For this experiment, we use the models trained on the G&G corpus to classify events in the Plotto dataset and the science fiction summary datasets.

1) Experiment 1: G&G Classification: The Bi-LSTM network was trained for 80 epochs, and the DPCNN was trained for 20 epochs. Both used Adam optimizer and a learning rate of 0.001. Fine-tuning for the BERT-GG and XLNet-GG models was done using the following parameters: maximum sequence length of 128 characters, one gradient accumulation step, and the learning rate is $3e^{-5}$. Model performance peaked at six epochs.

Additionally, we conducted a human participant study to determine human accuracy on the task of classifying G&G events as normative or nonnormative. The study used the same protocol that was used to label the Plotto and Sci-Fi corpora. $N = 20$ participants tagged sentences from G&G, and we compared their tags to the ground truth from the original cartoons.

Experiment results for case study 1 are given in Table I. First, it shows that humans have strong agreement with the G&G ground truth labels. Among the non-transformer models, DPCNN better classifies normative and nonnormative behavior from the G&G dataset. This is likely because the CNN can identify the global sentence structure better than a simple bidirectional LSTM cell. While the BERT-Base and XLNet-Base models struggle to classify events from the G&G corpus (achieving accuracies of 61.4% and 60.6%, respectively), fine-tuning drastically improves each model’s performance. BERT-GG obtains the best results in each of our metrics, obtaining a 21.33% accuracy improvement over the DPCNN.

2) Experiment 2: Transfer: In this experiment, we investigate how well machine learning models trained to identify normative and nonnormative behavior in the G&G corpus can transfer to other story domains. Specifically, we explore how well these models can classify events from the Plotto and science fiction summary corpora. We evaluate how well these models perform on fine-tuned and zero-shot transfer learning. Fine-tuned transfer learning means using a model trained for one task on a different, but related, task utilizing some additional training for fine-tuning. Zero-shot transfer, however, involves using the previously trained model on the new task with no additional training.

a) G&G to Plotto transfer: Table II shows the results of transfer learning for the Plotto dataset. Zero-shot transfer results are achieved by testing BERT-GG and XLNet-GG on the Plotto dataset; these models were trained on G&G but have never seen Plotto plot events. BERT-GG outperforms all the other models in zero-shot transfer in terms of accuracy and MCC. These results demonstrate that the knowledge of normative and nonnormative behavior gathered from the G&G stories facilitates a strong prior over normative/nonnormative behavior without overfitting to G&G scenarios.
Table II: Results for Plotto Transfer Experiments

| Model          | Test acc | F1-score | Precision | Recall | MCC  |
|----------------|----------|----------|-----------|--------|------|
| BERT-Base      | 0.529    | 0.402    | 0.297     | 0.619  | 0.103|
| XLNet-Base     | 0.46     | 0.436    | 0.297     | 0.817  | 0.148|
| BERT-GG        | 0.741    | 0.514    | 0.494     | 0.535  | 0.338|
| XLNet-GG       | 0.543    | 0.506    | 0.349     | 0.915  | 0.307|
| BERT-Plotto    | 0.838    | 0.634    | 0.75      | 0.549  | 0.544|
| XLNet-Plotto   | 0.838    | 0.651    | 0.724     | 0.592  | 0.552|

Note: The BERT-Plotto and XLNet-Plotto models were first trained on G&G and then additionally trained on the plotto corpus. The bold values represent the results of the best performing models which is discussed in Section V.

Table III: Results for Science Fiction Summary Transfer Experiments

| Model          | Test acc | F1-score | Precision | Recall | MCC  |
|----------------|----------|----------|-----------|--------|------|
| BERT-Base      | 0.43     | 0.38     | 0.6       | 0.279  | −0.037|
| XLNet-Base     | 0.538    | 0.658    | 0.55      | 0.066  | 0.081|
| BERT-GG        | 0.65     | 0.655    | 0.86      | 0.529  | 0.381|
| XLNet-GG       | 0.731    | 0.784    | 0.79      | 0.779  | 0.427|
| BERT-Sci-Fi    | 0.874    | 0.895    | 0.94      | 0.85   | 0.747|
| XLNet-Sci-Fi   | 0.839    | 0.87     | 0.882     | 0.857  | 0.658|

Note: The BERT-Sci-Fi and XLNet-Sci-Fi models were first trained on G&G and then additionally trained on the Sci-Fi corpus. The bold values represent the results of the best performing models which is discussed in Section V.

To further investigate the transferability of the models, we fine-tuned all the G&G models (BERT-GG and XLNet-GG) on Plotto stories. When fine-tuning each model, we use the same parameter settings except in experiment 1 except for the number of training epochs. We fine-tuned the BERT-Plotto and XLNet-Plotto models for three epochs.

Results from the experiment show that fine-tuning these models on the Plotto dataset significantly increases model performance, especially that of the transformer models.

b) G&G to Sci-Fi transfer: Events in G&G stories are from our daily life, whereas Sci-Fi plots are fictional, consisting of strange objects and events. We use the science fiction plot summary dataset to show the capability these models have for transfer learning in another narrative context. The results of this second experiment are shown in Table III. As before, we find that transformer-based models perform well on zero-shot transfer, though in this case, they perform worse than they did with the Plotto task. As with the Plotto task, we also fine-tuned our models on the Sci-Fi training data using the same training protocol. We see a dramatic increase in performance when given access to even a small amount of task-specific normative labels for fine-tuning.

3) Discussion: Our results demonstrate that transformer-based models trained on the G&G corpus are highly accurate in classifying previously unseen descriptions of normative behavior taken from that comic strip. A more notable observation is that the best models, the transformer models, can attain high accuracy when classifying event descriptions from unrelated corpora. This is significant in that it means the model can transfer to other tasks without requiring any normative/nonnormative labels of situations from the new tasks. When a small number of labels from the transfer tasks are available, the classification accuracy increases to nearly the same level as when the model is used to classify the G&G corpus.

D. Experiments: Value-Aligned Agent

To evaluate the value-aligned RL agents, we have created three text-based environments depicting different situations in which normative behavior can be investigated. Each environment has different qualities that enable us to investigate how incorporating normative information as intrinsic reward can change agents behavior based on the environment.

We conduct three experiments on these text environments. The first experiment examines how agents that incorporate intrinsic normative rewards in different ways fare against a baseline A2C when it comes to environmental reward. The second experiment quantifies behavioral differences when it comes to using normative and task-oriented actions. The third experiment looks at the effect of natural language phrase choices on the behavior of agents.

1) Test Environments: We have created three new environments to evaluate normative interactions with social entities while simultaneously trying to perform a task with an environmental reward. That is, there is a task that must be performed, but there are preferred and non-preferred ways of accomplishing the task that align with notions of normativity and nonnormativity for a particular society.

Each environment is designed such that, in the absence of an intrinsic normative reward signal, agents will learn a policy that, when executed, will likely appear to be nonnormative. Each environment pits the environmental reward against intrinsic normative reward in a different way. The agent may need to avoid nonnormative behaviors that are not part of solution trajectories, avoid nonnormative behaviors that comprise a less costly solution, or be given opportunities to take altruistic behaviors that are not strictly necessary and potentially in conflict with environmental rewards.

Each environment that we investigate in this article was constructed in the TextWorld [39] framework. We use this because it affords the ability to construct scenarios with social entities and more complex action spaces than the grid worlds more conventionally used for AI safety experiments [40]. These environments, thus, challenge the agent to reconcile task-oriented behavior and normative behavior. Consistent with text-based games, each scenario is composed of multiple rooms (discrete locations), entities, and task-oriented rewards. We have simplified each environment so that agents do not need to learn to read the descriptions and can instead learn to recognize states by their unique location names, observable entities, and observable items. The admissible commands in each location are also given.

One of the difficulties of working with a text-based environment, especially with respect to normativity, is that the way an action or its description is phrased can have an effect on whether it is deemed normative or nonnormative. To control for experimental biases that may exist in author-created action descriptions, we crowdsource action elaborations. These would then be used to describe the actions that an agent can take in...
each environment and would also be evaluated for normativity when choosing which action to perform.

To construct this dataset of action elaborations, we recruit 50 participants on Amazon Mechanical Turk. Each participant is provided an environment description, an action that is being performed, and the role of the character performing said action. They are then prompted to write a phrase describing how someone in their role would perform that action given the environmental context. Each action in each environment received three elaborations after pruning incomplete submissions, chosen at random from the generated set.

In the following subsections, we will describe each of the environments examined in this article in greater detail. We will focus on the states, actions, and rewards of each environment; we will also explain the purpose that each environment serves with respect to examining normative alignment.

a) Playground world: The first environment we explore is called playground world. The playground world environment is meant to simulate a situation that might occur when a child is playing on a playground. This is designed to be a simple proof that an agent can be trained to avoid nonnormative behaviors since the scenario can be successfully completed by performing only actions that have neutral normativity. In addition, this environment is meant to be the most aligned with the Goofus and Gallant normative prior model since it presents a social situation that closely resembles events that may occur in those comics. This allows us to investigate how a normative shaping approach performs when knowledge transfer due to an unfamiliar context is not an issue.

The playground world depicts a sports team dressing room before a game begins. The agent’s role is as a member of a team whose goal is to get prepared before the game. The environment is composed of three rooms: a locker room, a meeting room, and the playground itself. To complete the scenario, the agent must collect sporting gear for the game, wear the gear, go to the meeting room and join the team meeting, then go to the playground after the meeting is finished. The agent receives a large reward for joining the meeting and a small reward for completing each of the other parts of the process and finishing the full scenario.

These actions make up the task-oriented actions in playground world. Besides these task-oriented actions, the player can take several other optional actions that are ostensibly normative (e.g., “help your teammates,” “give attention in the meeting”) or nonnormative (e.g., “interrupt your teammate”). However, no actions are explicitly labeled as such.

An RL agent should learn that it can complete the scenario and maximize the expected environmental reward by never conducting nonnormative actions.

b) Superhero world: The second environment we explore is called superhero world. The superhero world environment depicts a fictional scenario in which the agent plays the role of a superhero who must stop a bank robbery. This environment is meant to provide a challenge to a normative aligned agent for several reasons. First, given that we are utilizing the G&G prior on action normativity, some amount of transfer is required. Second, this scenario presents the agent with more significant moral quandaries that have to be navigated when compared with the playground world. The scenario can be accomplished by completing two sub-tasks: 1) collecting information about the bank robbery and 2) using the information to stop the robbery. After completing each sub-quest (task), the agent gets a small environmental reward and a final, equal-sized small reward after finishing the game successfully. Going to the wrong bank or allowing the robber to escape are the failure states of the game.

In order to test how the normative prior could potentially alter the agent’s moral decision-making, each of these sub-tasks can be completed in two different ways: one that consists of nonnormative actions such as “beat the informant” and “shoot the robber” and another that consists of normative actions such as “do a favor to the informant,” “ask the informant about the robbery,” and “convince robber to surrender.” To make the choice to behave normative versus nonnormative more difficult, the nonnormative path was designed to be shorter than the normative path. This is significant because a RL agent that only responds to environmental reward is likely to learn a policy that utilizes the shorter, nonnormative path.

c) Clerk world: Clerk world is designed to investigate a scenario where tradeoffs exist between task efficiency and socially conscious actions that ignore or hinder task performance. In addition, this is another scenario where knowledge transfer will be necessary to effectively utilize the normative prior as this is a situation unexplored by the G&G prior model.

The clerk world scenario simulates a small Post Office. The agent plays the role of a worker in the office tasked with finding forms and stamping them. There are a number of customers and one coworker. Ten forms are scattered around the environment, and the agent must move around to find them. Not all forms are required to complete the scenario objective or subgoals, only a preset few are main task objectives. The agent receives a small reward for each form stamped, and a final, larger reward is given upon scenario completion. Actions that advance the scenario include locomotion, picking up forms, and applying the “stamp” action to forms in inventory.

Non-player character objects (coworker, customer) can be the targets of two other actions: “aid” and “ask.” To emulate a time tradeoff, when the agent chooses to aid or ask non-player characters, a subgoal involving a random form fails, lowering an agent’s environmental reward. The agent may still stamp that form but will not receive a reward for doing so. This scenario differs from the first two in that it requires the agent to make a tradeoff between stamping as many forms as possible and taking actions such as “aid” or “ask” which might be informally referred to as altruistic. An agent that is only responding to environmental reward can complete the scenario without “aid” or “ask” actions.

We noted in the experiments for superhero world and playground world that the behaviors of the GG-Pos and GG-mixed agents were very similar. Due to this similarity, we chose to only investigate the GG-Pos, the GG-shaped, and the A2C agents in the clerkworld experiments.

2) Experiment 1: Environmental Reward: In this experiment, we seek to understand the effect of the normative prior on acquired environmental reward. We should expect an agent that
ignores the intrinsic normative reward to achieve a greater total environmental reward over time. For each environment, we train our three agents that use the intrinsic normative reward and an A2C agent that only uses environment reward.

We train each agent for 1000 episodes in the clerk world environment, averaging over five training iterations. The playground and superhero world are trained for 4000 and 2500 episodes, respectively, as they take more time to converge. Performance in playground and superhero world is also averaged over five training iterations. At every step, the agent chooses an action and then randomly chooses one of three crowdsourced action elaborations. We measure the amount of environmental reward over time. In Figs. 3–5, we illustrated this measurement by plotting the ratio between the agent’s score and the observed score. The agent’s score represents the score achieved by the agent in a given episode, while the observed score corresponds to the highest achievable score in that episode.

As depicted in Figs. 3 and 4, in playground world and superhero world, all normative agents, as well as the baseline A2C agent, converge to policies that achieve maximum reward. Clerk world is a more challenging environment. For all clerk world runs (Fig. 5), the baseline A2C achieves the highest environmental reward score. The \textit{GG-shaped} agent achieves \( \sim 40\% \) of the maximum observed environmental score; in clerk world, opportunities for environmental reward are lost with each altruistic action.

Normative and altruistic actions in clerk world and playground world environments require the agent to perform actions that do not progress the scenario. Therefore, it is necessary, especially in clerk world, where opportunities for reward are lost with each altruistic action, to give up some environmental reward in order to act in ways that will be perceived as normative. The significance of this experiment shows that a policy shaping approach sacrifices more environmental score in order to take more normative actions than other means of using the normative reward. This confirms our hypothesis, and experiment 2 (next section) shows how different techniques qualitatively make the tradeoff between normative and nonnormative behaviors.

3) Experiment 2: Behavioral Analysis: Here, we analyze the behavioral differences between agent techniques. We use the ratio of task-specific to normative actions to visualize qualitative differences between agents. As with experiment 1, we train each agent for 1000 episodes in clerk world, 2500 episodes in superhero, and 4000 episodes in the playground environment, averaging over five training iterations.

In playground world (Fig. 6), the \textit{GG-pos} and \textit{GG-shaped} agents learn policies that execute normative actions \( \sim 40\% \) of the time. In contrast, the baseline A2C agent learns that normative actions are unnecessary.

In superhero world, we must use a slightly different formulation of our metric. Since in this environment, the agent
NAHIAN et al.: TRAINING VALUE-ALIGNED REINFORCEMENT LEARNING AGENTS USING A NORMATIVE PRIOR 3359

Fig. 7. Ratio of normative actions taken for all agent types in superhero world, smoothed with a 20-episode sliding window. In this environment, GG-mix and GG-pos outperform GG-shaped in total normative actions taken.

Fig. 8. Normalized ratio of normative actions taken for all agent types in clerk world, at that episode, smoothed with a 20-episode sliding window. This indicates that the decrease in environmental reward later in training is not attributed to an increase in normative actions.

can complete the scenario using normative or nonnormative actions, Fig. 7 shows the normalized ratio of normative to nonnormative actions. The GG-pos and GG-mix agents learn to almost exclusively follow the trajectories of “normative” actions. The baseline A2C agent discovers that the trajectories featuring “nonnormative” actions are shorter and learns a policy that favors them. The GG-shaped agent favors the normative trajectories (> 0.5) but not consistently. We observe that the GG model misclassifies some of the elaborations for “normative” actions in superhero world as “nonnormative” (see next section), which confuses the agent because some actions are sometimes re-ranked high and sometimes re-ranked low depending on which elaboration gets used.

In clerk world (Fig. 8), the baseline A2C agent learns not to use altruistic actions, which not only do not progress the scenario but also reduce the maximum reward achievable. The GG-pos and GG-mix agents also learn policies that use almost no altruistic actions. This is likely because the intrinsic normative reward added to the environmental loss does not make up for lost reward due to altruistic actions. The GG-shaped agent learns a policy that uses significantly more altruistic actions than any of the other alternatives. As seen from Experiment 1, this is done at the expense of environmental reward because this scenario penalizes the environmental reward for every altruistic action taken. The extent to which the GG-shaped agent attempts to use normative actions can be modulated by scaling the output of the GG model, however.

4) Experiment 3: Action Elaboration Phrasing: In experiment 2, we see how elaboration phrasing has an effect on the agent. In this experiment, we assess how the crowdsourced action elaborations affect agent behavior. In the test environments section, we discuss how each admissible action has three action elaborations. Because the GG model can be sensitive to certain phrasings of the same action, we seek to understand how different natural language phrasings for action elaborations alter agent behavior when all else is kept constant. For each of the three sets of paraphrases, we test with the GG-mix agent in each environment.

Fig. 9 shows the ratio of normative actions to task actions (e.g., a score of 0.9 means 90% normative actions and 10% task-oriented actions) in the superhero world. For two of the three crowdsourced phrase sets, we see that the GG-mix agent learns a policy that strongly prefers actions that we labeled as normative. For one phrase set (phrase set 1), some action elaborations are classified with the opposite of the ground-truth label. As a consequence, the agent’s resultant policy selects a mix of normative and nonnormative actions.

These results tell us two things. First, our ground-truth labels for our metrics are in agreement with crowd workers when considering a majority of elaborations. Second, the way in which commands are elaborated into natural language for normative classification can have an effect on agent behavior.

5) Discussion: Our experiments show that the three proposed techniques for incorporating intrinsic normative reward into a deep RL agent achieve desired behavioral change, increasing the use of actions perceived to be normative. Experiments in the superhero environment show that even though the nonnormative path is shorter, hence more efficient, agents learn the policy that prefers taking the normative path to reach the goal in the presence of a normative prior model. Even if the normative actions do not contribute to accomplishing goals, agents still may take some of these actions without sacrificing its objectives, as seen in the playground environment experiments. The clerk world experiments show that the policy shaping agent, GG-shaped, is more robust to complicated tradeoffs. The GG-shaped receives a lower task reward but is: a) robustly 2–6x more normative throughout its training iterations and b) can be
useful in situations where normative behavior during training is beneficial (e.g., apprenticeship learning).

The results also show that how actions are described can have a significant effect on the behavior of the agents. The normative prior can be sensitive to particular wordings. This is an artifact of our use of crowdsourcing to avoid experimenter bias but serves to remind us that normativity is subjective and that things that are normative can be described in ways that present as nonnormative, or vice versa.

In general, we see that GG-pos and GG-mix do not lose as much environmental reward as GG-shaped and are able to find “normative” solutions in the playground and superhero scenarios. However, GG-pos and GG-mix are unable to handle the complexities of the clerk world where normative rewards can only be achieved at the expense of environmental reward. GG-shaped is able to balance these rewards and, when the GG model is not misled by action elaborations, performs equally or better in other domains. Finally, we show how this prior can be used to create safe agents by providing a practical pathway for creating value-aligned agents. By harnessing the rich sociocultural knowledge contained in stories, we are able to train agents that can better understand the normativity of the actions available to them and incorporate this information into its own decision-making process, thus, constructing an agent that is more likely to be able to coexist with humans.

VI. LIMITATIONS & FUTURE WORKS

In this research, we have introduced machine learning approaches that leverage children’s stories to learn a strong prior over behaviors and demonstrated that stories offer a potentially rich source of information on human values. However, our current approach is limited to a binary view of values, opting to describe behavior as either being normative, aligning with expected social norms, or nonnormative, deviating from expected social norms. It does not provide the specific social norms or principles that determine why the behavior falls into one category or the other, which would be beneficial for making more informed decisions, both for humans and value-aligned agents. Therefore, there is a scope to further explore this area to identify specific principles embedded within text-based descriptions and examples of normative behavior, which will enable a more nuanced understanding of descriptions of human behavior in the context of normative values.

One potential limitation of our proposed value-aligned technique is its difficulty in finding the optimal policy in an environment where every possible path to the goal is nonnormative, and the primary objective is to achieve the goal, even if it means compromising the agent’s normative nature. To address this specific scenario, the agent should have the ability to balance its task performance and normativity based on its objective, which requires further research investigation.

VII. CONCLUSION

In this article, we show children’s stories can be used as the source of a normative prior over the actions that an agent takes. This can be used to train value-aligned RL agents. We show that this knowledge can be modeled using modern language understanding technology such as transformers and that the knowledge extracted from the stories is highly transferable to other domains. Finally, we show how this prior can be used to train value-aligned agents that can balance environmental reward with intrinsic reward as defined by this normative prior.

This work represents a significant step forward in the quest to create safe agents by providing a practical pathway for creating value-aligned agents. By harnessing the rich sociocultural knowledge contained in stories, we are able to train agents that can better understand the normativity of the actions available to them and incorporate this information into its own decision-making process, thus, constructing an agent that is more likely to be able to coexist with humans.

REFERENCES

[1] N. Soares and B. Fallenstein, “Aligning superintelligence with human interests: A technical research agenda,” Machine Intelligence Research Institute, Berkeley, CA, USA, Tech. Rep., 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:14393270
[2] S. Russell, D. Dewey, and M. Tegmark, “Research priorities for robust and beneficial artificial intelligence,” AI Mag., vol. 36, no. 4, pp. 105–114, 2015.
[3] T. Arnold, D. Kasenberg, and M. Scheutz, “Value alignment or misalignment—What will keep systems accountable?” in Proc. AAAI Workshop, AI, Ethics, Soc., 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:1725783
[4] J. H. Moor, “The nature, importance, and difficulty of machine ethics,” in IEEE Intell. Syst., vol. 21, no. 4, pp. 18–21, Jul./Aug. 2006.
[5] S. J. Russell, Human Compatible: Artificial Intelligence and the Problem of Control. Baltimore, MD, USA: Penguin, 2019.
[6] B. Stadie, P. Abbeel, and I. Sutskever, “Third-person imitation learning,” 2017, arXiv:1703.01703.
[7] M. Wulfmeier, “Efficient supervision for robot learning via imitation, simulation, and adaptation,” KI - Künstliche Intell., vol. 33, pp. 401–405, Dec. 2019.
[8] J. Ho and S. Ermon, “Generative adversarial imitation learning,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2016, pp. 4565–4573.
[9] J. Ho, J. Gupta, and S. Ermon, “Model-free imitation learning with policy optimization,” in Proc. Int. Conf. Mach. Learn., 2016, pp. 2760–2769.
[10] P. Abbeel and A. Y. Ng, “Apprenticeship learning via inverse reinforcement learning,” in Proc. 21st Int. Conf. Mach. Learn., 2004, p. 1.
[11] S. Russell, Human Compatible: Artificial Intelligence and the Problem of Control. Baltimore, MD, USA: Penguin, 2019.
[12] B. Harrison and M. O. Riedl, “Learning from stories: Using crowdsourced narratives to train virtual agents,” in Proc. Int. Conf. Mach. Learn. (ICML), 2015, pp. 3366–3372.
[13] S. Schaal, “Learning from demonstration,” in Proc. Adv. Neural Inf. Process. Syst., 1997, pp. 1040–1046.
[14] M. K. Ho, M. Littman, J. MacGlashan, F. Cushman, and J. L. Austerweil, “Showing versus doing: Teaching by demonstration,” in Proc. Adv. Neural Inf. Process. Syst., 2016, pp. 3027–3035.
[15] R. Akroun, M. Schoenauer, and M. Sebag, “APRIL: Active preference learning-based reinforcement learning,” in Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases, Cham, Switzerland: Springer-Verlag, 2012, pp. 116–131.
[16] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, “Deep reinforcement learning from human preferences,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 4299–4307.
[17] A. Y. Ng et al., “Algorithms for inverse reinforcement learning,” in Proc. 17th Int. Conf. Mach. Learn. (ICML), Jun. 2000, pp. 663–670.
[18] D. Hadfield-Menell, S. J. Russell, P. Abbeel, and A. Dragan, “Cooperative inverse reinforcement learning,” in Proc. 30th Int. Conf. Neural Inf. Process. Syst., 2016, pp. 3916–3924.
[22] B. Zoph, D. Yuret, J. May, and K. Knight, “Transfer learning for low-resource neural machine translation,” 2016, arXiv:1604.02201.
[23] M. O. Riedl and B. Harrison, “Using stories to teach human values to artificial agents,” in Proc. Workshops 30th AAAI Conf. Artif. Intell., 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:289511
[24] D. M. Ziegler et al., “Fine-tuning language models from human preferences,” 2019, arXiv:1909.08593.
[25] L. Jiang et al., “Delphi: Towards machine ethics and norms,” 2021, arXiv:2110.07574v1.
[26] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction. Cambridge, MA, USA: A Bradford Book, 2018.
[27] P. Ammanabrolu and M. Hausknecht, “Graph constrained reinforcement learning for natural language action spaces,” in Proc. Int. Conf. Learn. Representations, 2020. [Online]. Available: https://openreview.net/forum?id=B1x6w0EtwH
[28] M. Eger and K. W. Mathewson, “dAIrector: Automatic story beat generation through knowledge synthesis,” 2018, arXiv:1811.03423.
[29] L. Litman, J. Robinson, and T. Abberbock, “TurkPrime.com: A versatile crowdsourcing data acquisition platform for the behavioral sciences,” Behav. Res. Methods, vol. 49, no. 2, pp. 433–442, 2017.
[30] P. Ammanabrolu et al., “Story realization: Expanding plot events into sentences,” 2019, arXiv:1909.03480.
[31] Z. Huang, W. Xu, and K. Yu, “Bidirectional LSTM-CRF models for sequence tagging,” 2015, arXiv:1508.01991.
[32] R. Johnson and T. Zhang, “Deep pyramid convolutional neural networks for text categorization,” in Proc. 35th Annu. Meeting Assoc. Comput. Linguistics, Vancouver, BC, Canada, Association for Computational Linguistics, Jul. 2017, pp. 562–570.
[33] R. Johnson and T. Zhang, “Semi-supervised convolutional neural networks for text categorization via region embedding,” in Proc. Adv. Neural Inf. Process. Syst., vol. 28, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., Cambridge, MA, USA: MIT Press, 2015, pp. 919–927.
[34] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2018, arXiv:1810.04805.
[35] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le, “XLNet: Generalized autoregressive pretraining for language understanding,” 2019, arXiv:1906.08237.
[36] Z. Dai, Z. Yang, Y. Yang, J. G. Carbonell, Q. V. Le, and R. Salakhutdinov, “Transformer-XL: Attentive language models beyond a fixed-length context,” 2019, arXiv:1901.02860.
[37] T. K. Faulkner, E. S. Short, and A. L. Thomaz, “Policy shaping with supervisory attention driven exploration,” in Proc. IEEERSJ Int. Conf. Intell. Robots Syst. (IROS), Piscataway, NJ, USA: IEEE Press, 2018, pp. 842–847.
[38] Z. Lin, B. Harrison, A. Keech, and M. O. Riedl, “Explore, exploit or listen: Combining human feedback and policy model to speed up deep reinforcement learning in 3D worlds,” 2017, arXiv:1709.03969.
[39] M.-A. Coté et al., “TextWorld: A learning environment for text-based games,” in Proc. Workshop Comput. Games, Cham, Switzerland: Springer-Verlag, 2018, pp. 41–75.
[40] J. Leike et al., “AI safety gridworlds,” 2017, arXiv:1711.09883.