Approximated Variational Bayesian Inverse Reinforcement Learning for Large Language Model Alignment

Yuang Cai, Yuyu Yuan, Jinsheng Shi, Qinhong Lin

Beijing University of Posts and Telecommunications
{cyang,yuanyuyu,jinsheng,linqinhong}@bupt.edu.cn

Abstract

The alignment of large language models (LLMs) is crucial for generating helpful and harmless content. Existing approaches leverage preference-based human feedback data to learn the reward function and align the LLM with the feedback data. However, these approaches focus on modeling the reward difference between the chosen and rejected demonstrations, rather than directly modeling the true reward from each demonstration. Moreover, these approaches assume that the reward is only obtained at the end of the sentence, which overlooks the modeling of intermediate rewards. These issues lead to insufficient use of training signals in the feedback data, limiting the representation and generalization ability of the reward and potentially resulting in reward hacking. In this paper, we formulate LLM alignment as a Bayesian Inverse Reinforcement Learning (BIRL) problem and propose a novel training objective, Approximated Variational Alignment (AVA), to perform LLM alignment through Approximated Variational Reward Imitation Learning (AVRIL). The BIRL formulation facilitates intermediate reward modeling and direct reward modeling on each single demonstration, which enhances the utilization of training signals in the feedback data. Experiments show that AVA outperforms existing LLM alignment approaches in reward modeling, RL finetuning, and direct optimization.

Introduction

Large language models (LLMs) trained on massive corpus encode a large amount of knowledge and demonstrate powerful linguistic and reasoning capabilities in various domains (OpenAI 2022; Achiam et al. 2023). However, due to the inevitable harmful and useless information in the training data, LLMs can potentially generate content inconsistent with human values or requirements (Holtzman et al. 2019; Zhang et al. 2019; Weidinger et al. 2021). LLM alignment is a prevalent and effective approach for LLMs to generate harmless and helpful content. The alignment task typically relies on human feedback data in the form of preferences, where each preference data consists of a chosen sentence and a rejected sentence, labeled by human annotators (Zopf 2018; Tay et al. 2020). Reinforcement Learning from Human Feedback (RLHF) and Direct Preference Optimization (DPO) are two common approaches to align LLMs with human feedback data (Shen et al. 2023). RLHF first performs reward modeling to learn a reward function from the feedback data and then fine-tunes the LLM policy to maximize the expected reward achieved by its generated content using Reinforcement Learning (RL) (Ouyang et al. 2022; Bai et al. 2022; Touvron et al. 2023; Yang et al. 2023). DPO formulates the reward modeling objective as a ranking objective based on the LLM policy, which facilitates the joint performance of reward modeling and LLM policy finetuning through a unified training objective (Yuan et al. 2023; Rafailov et al. 2024; Song et al. 2024).

The Inverse Reinforcement Learning (IRL) problem generally involves learning a reward model from observed demonstration data produced by a Markov Decision Process (MDP) (Ng, Russell et al. 2000). Conversely, the Natural Language Generation (NLG) process can be viewed as an MDP where the generated sentences are considered demonstration data (Ranzato et al. 2016). Therefore, the alignment task performed by RLHF and DPO can be seen as addressing an IRL problem that infers the implicit reward function hidden in the preference-based human feedback data and learns the LLM policy either separately or jointly. However, existing RLHF and DPO alignment approaches only model the reward difference between the chosen and rejected demonstrations without explicitly modeling the true reward of every single sentence. This limitation means that the demonstration data is not fully utilized, which restricts the representation ability of the reward model and can lead to reward hacking (Skalse et al. 2022; Gao, Schulman, and Hilton 2023; Coste et al. 2023; Zhang et al. 2024). Additionally, current approaches generally model the end-to-end sentence-level reward without considering the reward of intermediate states. Model generalization may be limited when confronted with data that have similar intermediate state distributions but different complete sentence distributions. It is more intuitive to model intermediate rewards since humans can not only provide overall feedback on the entire text but also explain which parts of the text influenced their feedback.

In this paper, we propose a novel LLM alignment training objective, **Approximated Variational Alignment (AVA)**, based on Bayesian Inverse Reinforcement Learning (BIRL) (Ramachandran and Amir 2007). Specifically, we formulate the reward distribution as a posterior distribution condi-

tioned on the demonstration data and perform Approximated Variational Reward Imitation Learning (AVRIL) (Chan and van der Schaar 2021) to jointly approximate the reward distribution (i.e., the reward model) and the demonstration likelihood (i.e., the policy). Unlike most previous LLM alignment approaches, which only model the reward difference between chosen and rejected demonstrations, AVA directly models the reward of every single demonstration through the AVRIL training objective, thereby making better use of the training signals from feedback data. Additionally, we do not adhere to the assumption that the reward is only obtained at the end of the sentence. Instead, we leverage the AVRIL training objective to model the intermediate reward conditioned on the intermediate demonstration data. To demonstrate flexibility, we use the AVA training objective on data in different formats through different pipelines.

Our work makes the following main contributions:

- We present a novel insight into LLM alignment by formulating the alignment task as a BIRL problem, which enhances the utilization of training signals and improves the representation and generalization ability of the LLM.
- We demonstrate the flexibility of AVA by employing it for both reward modeling and direct optimization on either preference data or demonstration data.
- We empirically show that AVA surpasses Bradley-Terry and Preference Transformer in reward modeling and downstream RL fine-tuning, and outperforms DPO and AfD in direct optimization, which indicates a reduction in the reward hacking issue and an improvement in representation and generalization ability.

Related Work

LLM Alignment The Bradley-Terry model (Bradley and Terry 1952) formulates preference likelihood using the reward model and is widely adopted by RLHF alignment approaches for reward modeling. After reward modeling, the LLM is fine-tuned to maximize the expected reward achieved by LLM-generated content through downstream RL training (Ouyang et al. 2022; Bai et al. 2022; Touvron et al. 2023; Yang et al. 2023). A more concise approach for preference alignment is Direct Preference Optimization (DPO) (Rafailov et al. 2024), which denotes preference as the relative log-likelihood difference between the chosen sentence and the rejected sentence. DPO unifies reward modeling and LLM fine-tuning into a single process, facilitating LLM alignment with a simple classification loss. In addition to aligning LLMs with pairwise human preference data, some recent works also align LLMs with non-pairwise demonstration data. Sun and van der Schaar (2024) propose Alignment from Demonstrations (AfD), which leverages high-quality demonstration data to overcome challenges such as noisy labels and privacy concerns in preference datasets.

Intermediate Reward Modeling The above alignment approaches only model the end-to-end reward of a complete sentence, without considering the reward of intermediate states. This lack of intermediate reward modeling stems from the assumption that the reward is only achieved when the sentence is fully generated, regarding the Natural Language Generation (NLG) process as an MDP (Ranzato et al. 2016). To address this issue, we refer to related work on preference modeling in classic RL problems without the aforementioned assumption. Notably, the Preference Transformer (Kim et al. 2023) uses the attention weights computed by the Transformer architecture (Vaswani et al. 2017) to estimate the weighted non-Markovian reward of each intermediate state of the trajectory. The reward of a complete trajectory is then the weighted sum of all intermediate rewards. The preference between the chosen and rejected trajectories is formulated by their rewards and optimized through a contrastive training objective, similar to Bradley-Terry.

Preliminaries

MDP Formulation of NLG

At time step t, the state is the previously generated tokens denoted as $\mathbf{y}_{1:t} = (y_1, y_2, \cdots, y_t)$, the action is the currently generated token y_{t+1}. Note that in auto-regressive decoding, the output tokens are time-shifted. The action space is the vocabulary $\mathcal{V}$ containing all possible tokens. In the text generation setting, the state transition is deterministic, so we do not consider the transition probability function. The reward of taking action y_{t+1} under state $\mathbf{y}_{1:t}$ is denoted as $R(\mathbf{y}_{1:t}, y_{t+1}) = R(\mathbf{y}_{1:t+1})$, i.e., the reward can be the function of either the current state and the current action or merely the function of the next state due to the deterministic state transition. It is worth noting that for simplicity of denotation, we do not separately denote the prompt text and the response text but denote them as a whole sentence $\mathbf{y}$. The separation of prompt and response is trivial during implementation. The policy can be denoted as $\pi_w(y_{t+1}|\mathbf{y}_{1:t})$, which is also the distribution of the language model parameterized by w. For simplicity, we sometimes denote the policy as $\pi_w(\mathbf{y}) = \prod_{t=1}^{|\mathbf{y}|-1} \pi_w(y_{t+1}|\mathbf{y}_{1:t})$, where $|\mathbf{y}|$ is the length of sequence $|\mathbf{y}|$. Note that the accumulated product starts from $\pi_w(y_2|\mathbf{y}_{1:1})$ instead of $\pi_w(y_1)$ since we assume that all sequences start with a special token denoting the start of the sequence.

Bayesian Inverse Reinforcement Learning

Inverse Reinforcement Learning (IRL) is the problem of extracting a reward function of a Markov Decision Process (MDP) given observed optimal behavior (Ng, Russell et al. 2000). Bayesian Inverse Reinforcement Learning (BIRL) regards the reward function R as the hidden variable affecting and motivating the behavioral data $\mathcal{T}$. The objective of BIRL is to learn the posterior distribution $p(R|\mathcal{T})$. Approximate Variational Reward Imitation Learning (AVRIL) (Chan and van der Schaar 2021) adopts variational inference to approximate the posterior distribution. Specifically, AVRIL employs a parameterized distribution q_ϕ and minimizes the Kullback-Leibler (KL) divergence between q_ϕ and the posterior distribution $p(R|\mathcal{T})$, as shown in Eq. 1. This KL divergence is hard to compute since the posterior distribution is intractable. A common solution is to maximize the Evidence Lower Bound (ELBO), as shown in Eq. 2, where the second

term is to minimize the KL divergence between q_ϕ and the tractable prior distribution.

$$\min_{\phi} D_{\mathrm{KL}}[q_\phi(R)||p(R|\mathcal{T})] \tag{1}$$

$$\max_{\phi} \mathbb{E}_{R\sim q_\phi(\cdot)}[\log p(\mathcal{T}|R)] - D_{\mathrm{KL}}[q_\phi(R)||p(R)] \tag{2}$$

The first term of Eq. 2 is to maximize the log-likelihood of the observed optimal behaviors given any reward sampled from q_ϕ. AVRIL denotes the action distribution as a Boltzmann policy, as shown in Eq. 3, where $Q_R^{\pi_\mathcal{T}}$ is the state-action value function following policy $\pi_\mathcal{T}$ under reward function R. Intuitively, we can approximate the state-action value using a Deep Q Network (DQN) (Mnih et al. 2013) Q_θ parameterized by θ. An important problem is that in the RL setting, the reward function is fixed when optimizing Q_θ. However, in the AVRIL setting, the reward function is also being optimized during the optimization of Q_θ. The reward function and the state-action value function should satisfy $R(s,a) = \mathbb{E}_{s'\sim P(\cdot|s,a),a'\sim\pi(\cdot|s')}[Q_R^\pi(s,a) - \gamma Q_R^\pi(s',a')],\ \forall\ s\in\mathcal{S}, a\in\mathcal{A}$, i.e., the reward should equal the expectation of the TD error. By adding a penalty term forcing the TD error to follow the reward distribution, the final objective to be maximized is shown in Eq. 4, where $q_\phi(R|s,a)$ denotes the distribution of reward values given the state s and the action a. In this way, the behavior is indirectly conditioned on the reward, which is consistent with the likelihood $p(\mathcal{T}|R)$ in the ELBO (Eq. 2). Here, $B(a|s;Q_\theta)$ is the Boltzmann policy upon the state-action value function Q_θ parameterized by θ. The third term in the square brackets is to restrict the TD error to satisfy the constraint $R(s,a) = \mathbb{E}_{s',a'}[Q_R^\pi(s,a) - \gamma Q_R^\pi(s',a')],\ \forall\ s\in\mathcal{S}, a\in\mathcal{A}$. $q_\phi(R|s,a)$ denotes the distribution of reward values given the state s and the action a.

$$B(a|s;Q_R^{\pi_\mathcal{T}}) = \frac{\exp(\beta Q_R^{\pi_\mathcal{T}}(s,a))}{\sum_{a'\in\mathcal{A}}\exp(\beta Q_R^{\pi_\mathcal{T}}(s,a'))} \tag{3}$$

$$\max_{\phi,\theta} \sum_{(s,a,s',a')\in\mathcal{T}} \begin{bmatrix} \log B(a|s;Q_\theta) - D_{\mathrm{KL}}\left[q_\phi(\cdot|s,a)||p(\cdot)\right] \\ +\lambda \log q_\phi\big(Q_\theta(s,a) - \gamma Q_\theta(s',a')|s,a\big) \end{bmatrix} \tag{4}$$

Approximated Variational Alignment

In this section, we formulate the LLM alignment tasks as the BIRL problems and perform alignment with the Approximated Variational Alignment (AVA) training objectives. The AVA training objectives involve AVA from Demonstration (AVA-d) and AVA from Preference (AVA-p), both of which are BIRL training objectives based on the Approximated Variational Reward Imitation Learning (AVRIL) training objective (Chan and van der Schaar 2021). AVA-d is the implementation of the AVRIL training objective under the NLG setting, which learns on non-pairwise demonstration datasets. AVA-p is a contrastive variant of AVA-d, which learns on pairwise preference datasets.

Alignment from Demonstration

We first consider the problem of aligning an LLM policy with the demonstration data $\mathcal{D}$, where each sentence $\mathbf{y}\in\mathcal{D}$ is the ground-truth sentence. The alignment objective is to encourage the LLM policy to generate sentences like the demonstration data. Instead of building a direct training objective (e.g., supervised fine-tuning) to optimize the LLM policy, we focus on performing BIRL to learn a reward function from the demonstration data $\mathcal{D}$, i.e., to learn the posterior $p(R|\mathcal{D})$ with a parameterized distribution $q_\phi(R)$.

As illustrated in the preliminaries, the optimization of q_ϕ can be achieved by maximizing the AVRIL training objective (Eq. 4), where each element $(s,a,s',a')\in\mathcal{T}$ is a state-action quadruplet consisting of the current state s, the current action a, the next state s' and the next action a'. As for the Natural Language Generation (NLG) setting, at each time step t, the current state is the current sub-sentence $\mathbf{y}_{1:t}$, the current action is the token-to-be-generated y_{t+1}, the next state is $\mathbf{y}_{1:t+1}$, the concatenation of $\mathbf{y}_{1:t}$ and y_{t+1}, and the next action is y_{t+2}. By substituting the state-action quadruplet in Eq. 4 with the new quadruplet $(\mathbf{y}_{1:t}, y_{t+1}, \mathbf{y}_{1:t+1}, y_{t+2})$ and rewrite the summation in timestep-wise form, we can obtain the AVRIL training objective applicable to the NLG setting, as shown in Eq. 5. We refer to this training objective as Approximated Variational Alignment from Demonstration (AVA-d), which is a variant of the AVRIL training objective in the NLG setting.

$$\mathcal{F}_d(\mathcal{D}) = \sum_{\mathbf{y}\in\mathcal{D}}\sum_{t=1}^{|\mathbf{y}|-2} \begin{bmatrix} \log B(y_{t+1}|\mathbf{y}_{1:t};Q_\theta) - d_t(\phi) \\ +\lambda \log q_\phi\big(\delta_t(\theta)|\mathbf{y}_{1:t+1}\big) \end{bmatrix} \tag{5}$$

$$d_t(\phi) = D_{\mathrm{KL}}\left[q_\phi(\cdot|\mathbf{y}_{1:t+1})||p(\cdot)\right] \tag{6}$$

$$\delta_t(\theta) = Q_\theta(\mathbf{y}_{1:t}, y_{t+1}) - \gamma Q_\theta(\mathbf{y}_{1:t+1}, y_{t+2}) \tag{7}$$

Here, $q_\phi(R|\mathbf{y}_{1:t+1})$ is the reward distribution of the subsequence $\mathbf{y}_{1:t+1}$. The Boltzmann policy $B(y_{t+1}|\mathbf{y}_{1:t};Q_\theta)$ built upon the Q-value model Q_θ acts as the LLM policy for text generation. By maximizing $\mathcal{F}_d$, the Q-value model (i.e., the LLM policy) Q_θ as well as the reward distribution q_ϕ will be jointly optimized to be aligned with the demonstration dataset $\mathcal{D}$.

Similar to the original AVRIL objective, the AVA-d objective consists of three sub-objectives: the log-likelihood maximization, the KL divergence minimization, and the TD-error constraint. The first objective trains the LLM policy to maximize the likelihood of the demonstration data, which is identical to supervised fine-tuning. The second objective is to ensure the reward distribution satisfies the prior distribution assumption. The third objective, TD-error constraint, distinguishes AVA-d from conventional supervised fine-tuning. With the constraint, the update of the Q-value model will not only increase the Q-value of the ground-truth token in demonstration data but also make the TD error of the Q-values 7 close to the reward obtained after generating the current token, which ensures the consistency between the reward and the policy.

TQR Architecture

The original AVRIL adopts the architecture with a reward encoder and a Q-value decoder. To compute the AVA-d training objective and leverage the pre-trained weights of the

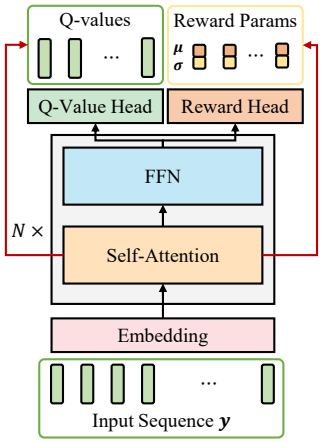

Figure 1: Overview of the TQR architecture.

backbone transformer model, we add a reward head and a Q-value head at the top of the Transformer decoder, as shown in Fig. 1. We refer to this architecture as **T**ransformer with **Q**-value and **R**eward Heads (TQR). The Q-value head takes the hidden states of the last decoder layer as input and outputs the Q-value of each action (i.e., token), as shown in Eq. 8. The reward is assumed to follow Gaussian distribution, and the reward head takes in the same hidden states and outputs the mean and standard deviation of the reward of each state, as shown In Eq. 9. Here, $\mathbf{h}_t$ is the hidden state vector of time step t, $Q_\theta(\mathbf{y}_{1:t}, \cdot) \in \mathbb{R}^{|\mathcal{V}|}$ is a vector whose i-th element equals $Q_\theta(\mathbf{y}_{1:t}, v^{(i)})$, where $v^{(i)}$ is the i-th token in the vocabulary, and $\mu_t, \sigma_t \in \mathbb{R}$ are mean and standard deviation of reward $R(\mathbf{y}_{1:t+1})$ at time step t. Now we can compute the training objective in Eq. 5 based on the above outputs of the Q-value head and the reward head.

$$Q_\theta(\mathbf{y}_{1:t}, \cdot) = \text{QHead}(\mathbf{h}_t; \theta), \forall\, t \in \{1, \cdots, |\mathbf{y}|\} \quad (8)$$

$$[\mu_t; \sigma_t] = \text{RHead}(\mathbf{h}_t; \phi), \forall\, t \in \{1, \cdots, |\mathbf{y}|\} \quad (9)$$

$$R(\mathbf{y}_{1:t+1}) \sim q_\phi(R|\mathbf{y}_{1:t+1}) = \mathcal{N}(R; \mu_t, \sigma_t) \quad (10)$$

Inspired by preference transformer (Kim et al. 2023), we further compute a reward weight for each time step of reward based on attention weights, as shown in Eq. 11, where $\mathbf{q}_i$ is the i-th row of the query matrix of the attention mechanism, $\mathbf{k}_{t'}$ is the t'-th row of the key matrix. We then apply reward weights to the outputs of the Q-value head (Eq. 8) and reward head (Eq. 9). Specifically, we simply multiply the output of the t-th position of the heads by the reward weight w_t, as shown by the red arrows in Fig. 1.

$$w_t = \frac{1}{|\mathbf{y}|} \sum_{i=1}^{|\mathbf{y}|} \text{softmax}\left(\{\mathbf{q}_i \cdot \mathbf{k}_{t'}\}_{t'=1}^{|\mathbf{y}|}\right)_t \quad (11)$$

Besides using a randomly initialized Q-value head, we can also construct a pre-trained Q-value model from the pre-trained LLM policy. The Boltzmann policy formulates the action probability as the softmax function of Q-values. Inversely, we can also formulate the Q-value as the log-softmax function of action probabilities, as shown in Eq. 12, where α is the temperature hyperparameter, π_w is the LLM policy parameterized by w. Note that the log-softmax operation is a non-strict inversion of the softmax operation, which means we can tune α to find the best way to map token-level probabilities to token-level Q-values.

$$Q_w(\mathbf{y}_{1:t}, y_{t+1}) = \log \frac{\exp(\alpha \pi_w(y_{t+1}|\mathbf{y}_{1:t}))}{\sum_{y' \in \mathcal{V}} \exp(\alpha \pi_w(y'|\mathbf{y}_{1:t}))} \quad (12)$$

By substituting with the above Q-value model, the AVA-d training objective can be denoted as Eq. 13, which facilitates us to initialize the Q-value model from a pre-trained LLM policy and adopt the AVA-d objective to fine-tune the LLM policy. The TD error can be denoted as Eq. 14.

$$\mathcal{F}_d(\mathcal{D}) = \sum_{\mathbf{y} \in \mathcal{D}} \sum_{t=1}^{|\mathbf{y}|-2} \begin{bmatrix} \beta \log \text{softmax}(\alpha \pi_w(y_{t+1}|\mathbf{y}_{1:t})) \\ -d_t(\phi) + \lambda \log q_\phi(\delta_t(w)|\mathbf{y}_{1:t+1}) \end{bmatrix} \quad (13)$$

$$\delta_t(w) = \log \frac{\text{softmax}(\alpha \pi_w(y_{t+1}|\mathbf{y}_{1:t}))}{\text{softmax}(\alpha \pi_w(y_{t+2}|\mathbf{y}_{1:t+1}))^\gamma} \quad (14)$$

Alignment from Preference

We then consider the problem of aligning an LLM policy π_w with preference data $\mathcal{P}$, where each data item $(\mathbf{y}^+, \mathbf{y}^-) \in \mathcal{P}$ consists of the chosen sentence $\mathbf{y}^+$ and the rejected sentence $\mathbf{y}^-$. We denote the set of all chosen sentences as $\mathcal{P}^+ = \{\mathbf{y}^+ | (\mathbf{y}^+, \mathbf{y}^-) \in \mathcal{P}\}$ and the set of all rejected sentences as $\mathcal{P}^- = \{\mathbf{y}^- | (\mathbf{y}^+, \mathbf{y}^-) \in \mathcal{P}\}$. The alignment objective is to encourage the LLM policy to generate sentences like the chosen demonstrations $\mathcal{P}^+$ while discouraging the LLM policy from generating sentences like the rejected demonstrations $\mathcal{P}^-$.

Similar to the derivation of the AVA-d training objective, we first focus on performing BIRL to learn a reward function from the preference data $\mathcal{P}$. We need to consider not only the chosen sentences as positive demonstrations but also the rejected sentences as negative demonstrations. We consider two posterior distributions, which are the reward conditioned on the chosen demonstrations $p(R|\mathcal{P}^+)$ and the reward conditioned on demonstrations that differ from rejected demonstrations $p(R|\overline{\mathcal{P}^-})$. Here, $\overline{\mathcal{P}^-}$ denotes demonstrations that differ from $\mathcal{P}^-$. Therefore, we define the training objective as Eq. 15, where the first term drives the reward distribution q_ϕ close to rewards that motivate the positive behaviors $\mathcal{P}^+$, while the second term drives q_ϕ close to rewards that motivate behaviors that differ from the negative demonstrations. We refer to the training objective as Contrastive Bayesian Inverse Reinforcement Learning (CBIRL).

$$\min_\phi D_{\text{KL}}[q_\phi(R)||p(R|\mathcal{P}^+)] + D_{\text{KL}}[q_\phi(R)||p(R|\overline{\mathcal{P}^-})] \quad (15)$$

Unsurprisingly, the minimization of these two KL divergences is infeasible. We derive the equivalent ELBO objective, as shown in Eq. 16. The derivation is shown in the Technical Appendix.

$$\max_\phi \begin{bmatrix} \mathbb{E}_{R \sim q_\phi(\cdot)} \left[\log p(\mathcal{P}^+|R) + \log[1 - p(\mathcal{P}^-|R)]\right] \\ -D_{\text{KL}}[q_\phi(R)||p(R)] \end{bmatrix} \quad (16)$$

Towards implementation, we need to further derive the ELBO objective as an approximated variational objective. Note that the main difference between the ELBO of CBIRL and the ELBO of conventional BIRL is the second optimization term in Eq. 16, which minimizes the log-likelihood of the negative demonstrations $\mathcal{P}^-$. Therefore, the approximated variational objective also contains the minimization of the negative demonstrations, as shown in Eq. 17. We refer to this training objective as the Approximated Variational Alignment from Preference (AVA-p). By maximizing $\mathcal{F}_p(\mathcal{P})$, on one hand, the LLM policy π_w will be encouraged to generate sentences like $\mathcal{P}^+$ and discouraged to generated sentences like $\mathcal{P}^-$; on the other hand, the policy and the reward will stay consistent under the TD-error constraint.

$$\mathcal{F}_p(\mathcal{P}) = \sum_{\mathbf{y}^{+/-}\in\mathcal{P}} \sum_t \begin{bmatrix} \beta \log \text{softmax}(\alpha\pi_w(y^+_{t+1}|\mathbf{y}^+_{1:t})) \\ -\beta \log \text{softmax}(\alpha\pi_w(y^-_{t+1}|\mathbf{y}^-_{1:t})) \\ -d_t(\phi) + \lambda \log q_\phi\big(\delta_t(w)|\mathbf{y}_{1:t+1}\big) \end{bmatrix} \tag{17}$$

To ensure the reward difference between the chosen and rejected demonstrations, we adopt a more intuitive auxiliary training objective, the Contrastive Expected Return (CER) training objective, as shown in Eq. 18, which encourages the reward of the positive demonstrations to be higher than the reward of the negative demonstrations. Note that we only consider the reward of the last timestep in the CER objective. Although we model the intermediate rewards, we still assume that the reward of the last timestep is decisive for the overall expected return, since empirical practice and research (Geva et al. 2023; Hanna, Liu, and Variengien 2024) show that the last position of the Transformer gathers most of the knowledge.

$$\mathcal{F}_c(\mathcal{P}) = \sum_{\mathbf{y}^{+/-}\in\mathcal{P}} \sigma\left[\mathbb{E}_{q_\phi(R|\mathbf{y}^+)}[R] - \mathbb{E}_{q_\phi(R|\mathbf{y}^-)}[R]\right] \tag{18}$$

AVA Pipelines

The AVA training objectives facilitate the joint optimization of the reward function and the policy. Therefore, AVA can be leveraged for both reward modeling and direct optimization, which are two common pipelines in LLM alignment. Both pipelines have their advantages and disadvantages. The reward modeling pipeline can produce a lightweight and reusable reward function for downstream RL fine-tuning while it suffers from the high RL training cost. The direct optimization pipeline is more efficient than reward modeling with RL during training but cannot produce a lightweight reward function for other uses and may suffer from overfitting. The AVA pipeline is shown in Alg. 1. For direct optimization, the initial policy is the initial LLM policy $\pi_{w^{(1)}}$.

AVA for Reward Modeling For reward modeling, the initial policy $\pi_{\theta^{(1)}}$ in Alg. 1 is the initial implicit policy $\pi_{\psi^{(1)}}$. In the TQR architecture, the reward function shares the same backbone model with the policy. Our purpose of reward modeling is to obtain an accurate and lightweight reward model. Therefore, we initialize the TQR architecture with a lightweight backbone model. In other words, we initialize the policy with a lightweight pre-trained language model $\pi_{\psi^{(1)}}$ instead of a large language model. Meanwhile, the reward distribution is also initialized and denoted by $q_{\phi^{(1)}}$ After the initialization, we leverage either AVA-d or AVA-p training objectives to optimize the reward distribution according to the type of the dataset $\mathcal{D}$. Note that the AVA training objectives require us to jointly train the reward function with the policy, although finally we only need the reward function. After reward modeling, we can leverage RL algorithms to fine-tune the LLM policy π_w to maximize the expected reward produced by the trained reward distribution q_ϕ, as shown in Eq. 19.

$$J(w) = \mathbb{E}_{\mathbf{y}\sim\pi_w(\cdot)}\left[\sum_{t=1}^{|\mathbf{y}|-1} \mathbb{E}_{R\sim q_\phi(\cdot|\mathbf{y}_{1:t+1})}[R]\right] \tag{19}$$

AVA for Direct Optimization For direct optimization, the initial policy in Alg. 1 is the initial LLM policy $\pi_{w^{(1)}}$. In other words, we directly initialize the policy with the pre-trained LLM $\pi_{w^{(1)}}$ and leverage the AVA training objectives to jointly optimize the policy and the reward distribution $q_{\phi^{(1)}}$. After training, the LLM policy and the reward distribution are both aligned with the demonstration or preference dataset $\mathcal{D}$.

Algorithm 1: The AVA pipeline.

Data: Dataset $\mathcal{D}$, initial policy $\pi_{\theta^{(1)}}$, initial reward distribution $q_{\phi^{(1)}}$, training epochs T
Result: The trained reward distribution $q_{\phi^{(T)}}$

for $i \in \{1, \cdots, T\}$ **do**
  if $\mathcal{D}$ *is demonstration dataset* **then**
    $\phi^{(i+1)} \leftarrow \phi^{(i)} + \nabla_{\phi^{(i)}}\mathcal{F}_d(\mathcal{D})$;
    $\theta^{(i+1)} \leftarrow \theta^{(i)} + \nabla_{\theta^{(i)}}\mathcal{F}_d(\mathcal{D})$;
  else
    $\phi^{(i+1)} \leftarrow \phi^{(i)} + \nabla_{\phi^{(i)}}\mathcal{F}_p(\mathcal{D}) + \nabla_{\phi^{(i)}}\mathcal{F}_c(\mathcal{D})$;
    $\theta^{(i+1)} \leftarrow \theta^{(i)} + \nabla_{\theta^{(i)}}\mathcal{F}_p(\mathcal{D}) + \nabla_{\theta^{(i)}}\mathcal{F}_c(\mathcal{D})$;
  end
end
return $q_{\phi^{(T)}}, \pi_{\theta^{(T)}}$

Experiment

Experiment Setup

Datasets For preference datasets, we consider Anthropic-Harmless, Anthropic-Helpful, and OpenAI-Summary and perform reward modeling, RL fine-tuning, and direct optimization on these datasets. For demonstration datasets, we consider Alpaca-GPT-4 and Math-GPT-4o and only perform direct optimization on these datasets.

Metrics For reward modeling, we evaluate the accuracy at which the reward of the chosen sentence is greater than that of the rejected sentence, as well as the win rates of the Best-of-N sampling (Stiennon et al. 2020; Nakano et al. 2021) results. The detailed calculation of win rate is in the Technical Appendix. For RL fine-tuning, we evaluate the win rates

of the LLMs fine-tuned with different reward models (i.e., AVA-p/d and baselines). For direct optimization, we evaluate the win rates of LLMs fine-tuned with AVA-p/d against LLMs fine-tuned with baseline approaches.

Pre-trained Models For reward modeling, we initialize the implicit policy with GPT-2 (117M) and BART-base (140M) to see the reward modeling performance with different initializations. For RL fine-tuning and direct optimization, we initialize the LLM policy with Llama-2-7b-chat-hf. The reward models adopted in RL fine-tuning only involve those initialized with GPT-2.

Baselines For reward modeling, we adopt Bradley-Terry (Bradley and Terry 1952) and Preference Transformer (Pref-Trans) (Kim et al. 2023) as baselines. For direct optimization from preference, we adopt DPO (Rafailov et al. 2024) as the baseline. For direct optimization from demonstration, we adopt AfD (Sun and van der Schaar 2024) as the baseline. Since AfD constructs preference data from demonstration data and relies on preference-based training objectives, we combine AfD with different preference-based training objectives. Specifically, for reward modeling, we construct AfD w/ Bradley-Terry, Afd w/ Pref-Trans, and AfD w/ AVA-p. For direct optimization, we construct AfD w/ DPO. For win rate evaluations of aligned LLMs, we also adopt supervised fine-tuning (SFT) as the baseline.

Ablation Variants We construct the following variants of AVA-p and AVA-d training objectives for ablation studies:

- **AVA-p/d w/o rwt**: AVA-p/d without reward weighting, which removes the computation of reward weights and the weighted rewards from the TQR architecture.
- **AVA-p w/o neg**: AVA-p without the negative demonstration, which removes the minimization of the likelihood of the negative demonstrations. Note that the objective does not completely degenerate into the AVA-d training objective since we still keep the CER auxiliary objective.
- **AVA-p w/o irl**: AVA-p without inverse reinforcement learning, which removes the TD-error constraint and the reward prior assumption and only keeps the likelihood optimization, which can be regarded as contrastive supervised fine-tuning.
- **AVA-p w/o cer**: AVA-p without CER auxiliary objective.
- **AVA-p/d w/o ptq**: AVA-p/d without pre-trained Q-value head, which does not reuse the LM head of the pre-trained policy as the Q-value head but initializes the Q-value head from scratch.

Moreover, the prior reward distribution is assumed to be the standard Gaussian distribution. For detailed experiment setup, please refer to our code and the Experiment Details section of the Technical Appendix.

Reward Modeling

Table 1 reports the reward accuracy of baseline and AVA training objectives. The results show that AVA-p surpasses Bradley-Terry and Pref-Trans in reward accuracy on all reported reward modeling tasks with different initial models and datasets. The ablation results further reveal that AVA-p achieves the highest reward accuracy on the greatest number of tasks compared to ablated training objectives, which suggests that removing any module from AVA-p diminishes the reward accuracy on most tasks. Furthermore, we consider the chosen half of the preference data as demonstration data and train the reward model on it using the AVA-d training objective. Surprisingly, AVA-d achieves the best performance on 2 out of 6 tasks, despite learning solely from the chosen demonstrations. We also evaluate the RewardBench score (Lambert et al. 2024) of the ensemble model over all 6 tasks for Bradley-Terry, Pref-Trans, and AVA-p, which achieve scores of 55.91, 57.05, and 59.84, respectively.

	Harmless		**Helpful**		**Summary**	
	gpt2	bart	gpt2	bart	gpt2	bart
Baselines						
Bradley-Terry	70.02	68.96	69.39	67.56	59.27	59.27
Pref-Trans	70.26	71.32	71.37	72.37	59.31	56.91
Ours						
AVA-p	70.27	**72.30**	**72.37**	**74.84**	61.79	**64.31**
AVA-p w/o rwt	70.06	70.73	69.81	69.32	60.55	58.89
AVA-p w/o neg	**70.54**	70.36	69.75	69.15	**62.06**	58.65
AVA-p w/o irl	69.48	67.46	68.87	65.38	58.96	58.46
AVA-p w/o cer	70.06	70.73	69.81	69.32	60.55	58.58
AVA-p w/o ptq	68.67	68.69	68.51	67.60	61.25	57.76
AVA-d	**70.54**	70.36	69.75	69.15	**62.06**	59.00

Table 1: Reward accuracy of AVA and baseline objectives.

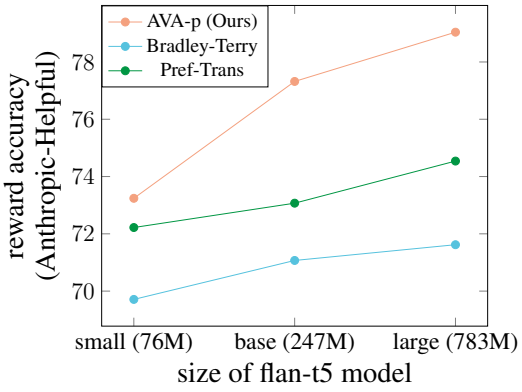

Figure 2: Scalability of AVA-p and baseline objectives.

We demonstrate the scalability of AVA-p for reward modeling by performing the Helpful task using google flan-t5 models (Chung et al. 2024) with different sizes. Specifically, we perform reward modeling based on flan-t5-small, flan-t5-base, and flan-t5-large models using baseline and AVA-p objectives. The scalability result in Fig. 2 shows that AVA-p scales better with the increase of model size compared to baseline objectives.

To further evaluate reward modeling performance, we employ Best-of-N (BoN) sampling. We evaluate the win rates of BoN w/ AVA-p against BoN w/ Bradley-Terry and BoN w/ Pref-Trans, where "BoN w/ xxx" means that the reward

Task	Opponent	Win↑	Tie	Lose↓
Harmless	Stochastic	43.0	17.4	39.6
	BoN w/ Bradley-Terry	28.8	42.6	28.6
	BoN w/ Pref-Trans	35.6	31.9	32.5
Helpful	Stochastic	26.1	50.4	23.5
	BoN w/ Bradley-Terry	13.2	76.1	10.7
	BoN w/ Pref-Trans	19.3	62.3	18.4
Summary	Stochastic	60.2	0.8	39.0
	BoN w/ Bradley-Terry	34.6	34.5	30.9
	BoN w/ Pref-Trans	43.6	25.8	30.6

Table 2: Win rates of BoN with AVA-p reward model.

Task	Opponent	Win↑	Tie	Lose↓
Harmless	SFT	42.5	23.4	34.1
	PPO w/ Bradley-Terry	9.2	81.7	9.1
	PPO w/ Pref-Trans	9.0	83.2	7.8
Helpful	SFT	23.3	58.8	18.0
	PPO w/ Bradley-Terry	1.8	97.2	1.0
	PPO w/ Pref-Trans	2.6	95.8	1.6
Summary	SFT	73.8	1.4	24.7
	PPO w/ Bradley-Terry	18.5	66.3	15.2
	PPO w/ Pref-Trans	33.9	34.6	31.5

Table 3: Win rates of PPO with AVA-p reward model.

model used for BoN is trained with the "xxx" training objective. Additionally, we evaluate the win rate of BoN w/ AVA-p against the stochastic sampling results without BoN. Table 2 reports the win rates of the reward model trained with AVA-p against reward models trained with baseline objectives in BoN sampling. The results further demonstrate that AVA-p surpasses Bradley-Terry and Pref-Trans in reward modeling.

RL Fine-tuning

We adopt the PPO algorithm (Schulman et al. 2017) to fine-tune LLMs to maximize the reward produced by different reward models. We evaluate the win rates of PPO w/ AVA-p against PPO w/ Bradley-Terry and PPO w/ Pref-Trans, where "PPO w/ xxx" means that the reward model used for PPO fine-tuning is trained with the "xxx" training objective. We also evaluate the win rate of PPO w/ AVA-p against supervised fine-tuning (SFT), where the LLM is fine-tuned on the chosen half of the preference data with supervised learning. The results in Table 3 show that AVA-p outperforms the baseline reward modeling objectives on all reported tasks in downstream RL fine-tuning of the LLM.

Direct Optimization

From Preference We adopt AVA-p and DPO (Rafailov et al. 2024) to directly optimize the LLM from preference data and evaluate the win rates of AVA-p against DPO and SFT. The results in Table 4 show that AVA-p outperforms DPO in direct optimization from preference data.

Task	Opponent	Win↑	Tie	Lose↓
Harmless	SFT	37.1	28.9	34.0
	DPO	13.7	73.8	12.5
Helpful	SFT	22.5	59.6	17.9
	DPO	14.4	72.4	13.2
Summary	SFT	59.0	7.3	33.7
	DPO	44.9	11.0	44.1

Table 4: Win rates of direct optimization with AVA-p.

Task	Opponent	Win↑	Tie	Lose↓
Alpaca	SFT	58.1	7.2	34.7
	DPO w/ AfD	57.2	6.9	35.9
	AVA-p w/ AfD	56.5	7.1	36.4
Math	SFT	47.0	9.7	43.3
	DPO w/ AfD	44.3	11.4	44.3
	AVA-p w/ AfD	45.4	11.4	43.1

Table 5: Win rates of direct optimization with AVA-d.

From Demonstration We adopt AVA-d and AfD (Sun and van der Schaar 2024) to directly optimize the LLM from demonstration data. We evaluate the win rates of AVA-d against SFT, DPO w/ AfD, and AVA-p w/ AfD, where "xxx w/ AfD" means applying the "xxx" training objective on AfD-format data. The results in Table 5 show that AVA-d outperforms the AfD approaches in direct optimization from demonstration data. Moreover, AVA-d is more training-efficient since AfD requires supervised fine-tuning and sampling from LLM policies.

Limitation The direct optimization pipeline achieves little improvement over the DPO baseline in most tasks. Future work should be performed to address the limitations of the direct optimization pipeline.

Conclusion

We present AVA, a flexible novel LLM alignment objective with enhanced capabilities. The flexibility of AVA is evident in two aspects. Firstly, AVA can utilize either preference data or demonstration data for alignment purposes. Secondly, AVA can be integrated into the reward modeling and RL fine-tuning pipeline or used to directly optimize the LLM. The representation and generalization capabilities of AVA are also evident in two aspects. Theoretically, AVA formulates reward modeling as a BIRL problem, facilitating both intermediate reward modeling and direct reward modeling on demonstration. Experimentally, AVA achieves superior reward accuracy in reward modeling tasks and higher win rates in RL fine-tuning and direct optimization of LLMs, which demonstrates the alleviation of the reward hacking issue and improved alignment performance.

Acknowledgements

This work is supported by Super Computing Platform of Beijing University of Posts and Telecommunications.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Bai, Y.; Jones, A.; Ndousse, K.; Askell, A.; Chen, A.; DasSarma, N.; Drain, D.; Fort, S.; Ganguli, D.; Henighan, T.; et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Bradley, R. A.; and Terry, M. E. 1952. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4): 324–345.

Chan, A. J.; and van der Schaar, M. 2021. Scalable Bayesian Inverse Reinforcement Learning. In *International Conference on Learning Representations*.

Chung, H. W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, Y.; Wang, X.; Dehghani, M.; Brahma, S.; et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70): 1–53.

Coste, T.; Anwar, U.; Kirk, R.; and Krueger, D. 2023. Reward model ensembles help mitigate overoptimization. *arXiv preprint arXiv:2310.02743*.

Gao, L.; Schulman, J.; and Hilton, J. 2023. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, 10835–10866. PMLR.

Geva, M.; Bastings, J.; Filippova, K.; and Globerson, A. 2023. Dissecting recall of factual associations in auto-regressive language models. *arXiv preprint arXiv:2304.14767*.

Hanna, M.; Liu, O.; and Variengien, A. 2024. How does GPT-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. *Advances in Neural Information Processing Systems*, 36.

Holtzman, A.; Buys, J.; Du, L.; Forbes, M.; and Choi, Y. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Kim, C.; Park, J.; Shin, J.; Lee, H.; Abbeel, P.; and Lee, K. 2023. Preference transformer: Modeling human preferences using transformers for RL. *arXiv preprint arXiv:2303.00957*.

Lambert, N.; Pyatkin, V.; Morrison, J.; Miranda, L.; Lin, B. Y.; Chandu, K.; Dziri, N.; Kumar, S.; Zick, T.; Choi, Y.; et al. 2024. Rewardbench: Evaluating reward models for language modeling. *arXiv preprint arXiv:2403.13787*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Nakano, R.; Hilton, J.; Balaji, S.; Wu, J.; Ouyang, L.; Kim, C.; Hesse, C.; Jain, S.; Kosaraju, V.; Saunders, W.; et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

Ng, A. Y.; Russell, S.; et al. 2000. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, 2.

OpenAI. 2022. Introducing ChatGPT. https://openai.com/index/chatgpt/.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Ramachandran, D.; and Amir, E. 2007. Bayesian Inverse Reinforcement Learning. In *IJCAI*, volume 7, 2586–2591.

Ranzato, M.; Chopra, S.; Auli, M.; and Zaremba, W. 2016. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shen, T.; Jin, R.; Huang, Y.; Liu, C.; Dong, W.; Guo, Z.; Wu, X.; Liu, Y.; and Xiong, D. 2023. Large language model alignment: A survey. *arXiv preprint arXiv:2309.15025*.

Skalse, J.; Howe, N.; Krasheninnikov, D.; and Krueger, D. 2022. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35: 9460–9471.

Song, F.; Yu, B.; Li, M.; Yu, H.; Huang, F.; Li, Y.; and Wang, H. 2024. Preference ranking optimization for human alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 18990–18998.

Stiennon, N.; Ouyang, L.; Wu, J.; Ziegler, D.; Lowe, R.; Voss, C.; Radford, A.; Amodei, D.; and Christiano, P. F. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33: 3008–3021.

Sun, H.; and van der Schaar, M. 2024. Inverse-RLignment: Inverse Reinforcement Learning from Demonstrations for LLM Alignment. *arXiv preprint arXiv:2405.15624*.

Tay, Y.; Ong, D.; Fu, J.; Chan, A.; Chen, N.; Luu, A. T.; and Pal, C. 2020. Would you rather? a new benchmark for learning machine alignment with cultural values and social preferences. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 5369–5373.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Weidinger, L.; Mellor, J.; Rauh, M.; Griffin, C.; Uesato, J.; Huang, P.-S.; Cheng, M.; Glaese, M.; Balle, B.; Kasirzadeh, A.; et al. 2021. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*.

Yang, A.; Xiao, B.; Wang, B.; Zhang, B.; Bian, C.; Yin, C.; Lv, C.; Pan, D.; Wang, D.; Yan, D.; et al. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

Yuan, Z.; Yuan, H.; Tan, C.; Wang, W.; Huang, S.; and Huang, F. 2023. Rrhf: Rank responses to align language models with human feedback without tears. *arXiv preprint arXiv:2304.05302*.

Zhang, X.; Ton, J.-F.; Shen, W.; Wang, H.; and Liu, Y. 2024. Overcoming reward overoptimization via adversarial policy optimization with lightweight uncertainty estimation. *arXiv preprint arXiv:2403.05171*.

Zhang, Y.; Sun, S.; Galley, M.; Chen, Y.-C.; Brockett, C.; Gao, X.; Gao, J.; Liu, J.; and Dolan, B. 2019. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*.

Zopf, M. 2018. Estimating summary quality with pairwise preferences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 1687–1696.