# Enhancing Context-Aware Toxicity Detection with Human-in-the-Loop Learning: A Comparison of Behavior Cloning and Direct Preference Optimization

Rishi More

## 1 Introduction

The rapid growth of online communities has exposed significant limitations in conventional toxicity detection systems. While traditional models, such as Bidirectional Long Short-Term Memory Networks (BiLSTMs), are adept at identifying overtly toxic content, they often struggle to detect nuanced, context-dependent harmful language. This challenge is exacerbated across diverse platforms with varying social norms and linguistic contexts. Recent advancements in transformer-based architectures and ensemble methods have yielded higher accuracy on benchmarks but suffer from issues like computational inefficiency and a lack of sensitivity to subtle contextual cues Kamphuis [2024], Mazari et al. [2024]. Moreover, Large Language Models (LLMs) face difficulties grasping community-specific content, especially when it involves implicit social norms Naveed et al. [2024].

Incorporating human feedback into machine learning models presents a promising avenue to overcome these challenges. By integrating community interactions, models can adapt to evolving standards and better capture the nuances of language within specific contexts. This project explores the effectiveness of fine-tuning a BiLSTM model using Behavior Cloning (BC) and Direct Preference Optimization (DPO) to enhance context-aware toxicity detection.

## 2 Background & Related Work

Traditional models for toxicity detection often rely on machine learning algorithms trained on labeled datasets to identify harmful content. Models like BiLSTMs process input sequences in both forward and backward directions, capturing contextual information Hochreiter and Schmidhuber [1997]. While detecting overt toxicity effectively, they often miss nuanced or context-dependent harmful language.

Transformer-based models have improved performance due to their self-attention mechanisms, which allow for understanding complex language patterns Vaswani et al. [2023]. Kamphuis Kamphuis [2024] introduced a compact transformer-based model for toxic content detection, achieving high accuracy. Mazari et al. Mazari et al. [2024] utilized BERT-based ensemble methods for multi-aspect hate speech detection, highlighting the effectiveness of transformer architectures. However, these models can be computationally intensive and may not capture subtle contextual nuances, especially in diverse online communities with evolving language patterns.

Large Language Models (LLMs) like GPT-3 have demonstrated remarkable capabilities in natural language understanding and generation Brown et al. [2020]. Nevertheless, they encounter challenges in grasping community-specific norms and implicit social cues Naveed et al. [2024]. These models are typically trained on large-scale internet data and may not reflect the specific standards or etiquettes of individual communities. Valmeekam et al. Valmeekam et al. [2023] discuss the limitations of LLMs in planning and reasoning tasks, emphasizing the need for incorporating human feedback to align models with desired behaviors.

Human-in-the-loop machine learning leverages human input to improve model performance, particularly in tasks where human judgment is crucial. Methods like Behavior Cloning (BC) and Direct Preference Optimization (DPO) are prominent in this domain. Behavior Cloning involves training a model to mimic the behavior demonstrated in a dataset of expert examples Pomerleau [1989]. In the context of toxicity detection, BC would train the model using labeled examples of toxic and non-toxic content based on community feedback, such as upvotes and downvotes. However, BC can struggle with distributional shifts and may not generalize well to unseen data Ross et al. [2011].

Direct Preference Optimization is a method where the model learns directly from preference comparisons rather than explicit labels. It optimizes a policy to prefer outputs that are ranked higher according to human preferences Rafailov et al. [2024]. In natural language processing, DPO has been less explored but holds promise for aligning models with nuanced human judgments.

Implementing DPO in natural language tasks involves creating preference pairs from user interactions or feedback. For instance, in dialogue systems, Jaques et al. Jaques et al. [2019] used reinforcement learning from human feedback to improve conversational agents. Although not identical to DPO, their work aligns with the idea of optimization based on preferences. Stiennon et al. Stiennon et al. [2022] demonstrated the effectiveness of learning to summarize from human feedback by training models with preference-based reinforcement learning, leading to higher-quality summaries aligned with human judgments.

In toxicity detection, preference-based methods allow models to capture subtle community norms by learning from comparisons between content pieces. This approach can address the limitations of explicit labeling by utilizing implicit feedback signals, such as upvote/downvote ratios.

# 3 Method(s)

The goal of this project is to compare the effectiveness of Behavior Cloning (BC) and Direct Preference Optimization (DPO) in capturing context-dependent harmful content for toxicity detection. A Bidirectional Long Short-Term Memory (BiLSTM) network is employed due to its effectiveness in processing sequential data and capturing contextual dependencies in text.

## 3.1 Bidirectional Long Short-Term Memory Network

A Long Short-Term Memory (LSTM) network is a type of recurrent neural network (RNN) that addresses the vanishing gradient problem of traditional RNNs Hochreiter and Schmidhuber [1997]. An LSTM cell maintains an internal state vector, or memory, which allows it to retain information over long sequences. A BiLSTM extends the LSTM by processing data in both forward and backward directions, meaning that for a given sequence $\{x_1, x_2, ..., x_T\}$, the network processes from $x_1$ to $x_T$ and from $x_T$ to $x_1$, capturing context from both past and future tokens Graves and Schmidhuber [2005].

In this model, input words are embedded into a continuous vector space using an embedding layer. Let $\mathbf{w}_i$ represent the one-hot encoded vector of the $i$-th word, and $\mathbf{e}_i \in \mathbb{R}^d$ be its embedding, where $d$ is the embedding dimension. The BiLSTM processes the sequence of embeddings $\{\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_T\}$ and produces a hidden representation for each time step by concatenating the forward and backward hidden states:

$$\mathbf{h}_i = [\overrightarrow{\mathbf{h}}_i; \overleftarrow{\mathbf{h}}_i] \tag{1}$$

where $\overrightarrow{\mathbf{h}}_i$ is the hidden state from the forward LSTM, and $\overleftarrow{\mathbf{h}}_i$ is from the backward LSTM.

## 3.2 Behavior Cloning (BC)

Behavior Cloning involves training the model using labeled data derived from community feedback. Reddit comments were labeled as toxic or non-toxic based on their upvote-to-downvote ratios. The model learns to imitate this behavior by minimizing the binary cross-entropy loss:

$$\mathcal{L}_{\text{BC}}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)\right] \tag{2}$$

where $\theta$ are the model parameters, $y_i$ are the true labels, and $\hat{y}_i$ are the predicted probabilities.

### 3.2.1 Labeling Strategy

Comments are labeled using upvote-to-downvote ratios. A threshold $r$ is defined such that:

$$y_i = \begin{cases} 0, & \text{if ratio}_i \geq r \\ 1, & \text{if ratio}_i < r \end{cases} \tag{3}$$

where $\text{ratio}_i = \frac{\text{upvotes}_i}{\text{upvotes}_i + \text{downvotes}_i}$

### 3.2.2 Training Objective

The model learns to predict the toxicity label $y_i$ given an input comment $x_i$. The output of the BiLSTM model after passing through the fully connected layer is transformed using a sigmoid activation to produce $\hat{y}_i \in [0, 1]$, representing the probability that the comment is toxic.

The loss function is the binary cross-entropy loss:

$$\mathcal{L}_{\text{BC}}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)\right] \tag{4}$$

where $\theta$ are the model parameters, $N$ is the number of training examples.

## 3.3 Direct Preference Optimization (DPO)

DPO is an approach where the model learns directly from preference comparisons between data points, rather than from explicit labels Rafailov et al. [2024].

### 3.3.1 Constructing Preference Pairs

From the Reddit dataset, preference pairs $(x_i^+, x_i^-)$ are constructed, where $x_i^+$ represents a comment with a higher upvote ratio, and $x_i^-$ is a comment with a lower upvote ratio. To ensure meaningful preferences, pairs were selected only if the difference in upvote ratios exceeded a certain threshold $\delta$.

### 3.3.2 Scoring Function and Objective

The model assigns a scalar score $s_\theta(x)$ to each input comment $x$, where $\theta$ are the model parameters. The score is obtained by applying a linear transformation to the output representation $\mathbf{h}$:

$$s_\theta(x) = \mathbf{w}^\top \mathbf{h} + b \tag{5}$$

where $\mathbf{w}$ and $b$ are parameters to be learned.

The training objective is to maximize the probability that preferred comments receive higher scores than less preferred ones. We use the pairwise preference loss:

$$\mathcal{L}_{\text{DPO}}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\log\sigma\left(s_\theta(x_i^+) - s_\theta(x_i^-)\right) \tag{6}$$

where $\sigma(z) = \frac{1}{1+e^{-z}}$ is the sigmoid function.

### 3.4 Evaluation

After training, the models are evaluated on a gold-standard dataset collected from the Reddit subreddit r/politics. This dataset comprises posts and comments that were flagged as controversial and subsequently reviewed by subreddit moderators. The gold-standard labels reflect the moderators' decisions regarding the presence of toxicity, particularly in context-dependent scenarios. Metrics used for evaluation include accuracy, F1-score, and Binary Cross Entropy Loss, ensuring a comprehensive assessment of the model's ability to detect nuanced and context-specific toxicity in politically charged discussions.

## 4 Experimental Results

### 4.1 Datasets

Two datasets were used: the Kaggle Toxic Comment Classification Challenge Dataset Jigsaw and AI [2019], containing 160,000 Wikipedia comments labeled for toxicity, and a Reddit Comments Dataset, a web-scraped collection of approximately 100,000 Reddit comments with upvote and downvote ratios serving as implicit feedback. Comments were cleaned by removing special characters, URLs, and stop words. Tokenization was performed, and sequences were padded to a maximum length. For the Reddit dataset, preference pairs were created for DPO by selecting comment pairs with significant differences in upvote ratios. In addition to these datasets, a golden standard dataset of 974 comments was created using a custom webscraper. This dataset was curated by collecting comments from r/politics on the Reveddit platform, which provides visibility into comments removed from Reddit.

### 4.2 Training Procedure

The BiLSTM model was initially trained on the Kaggle dataset to establish a baseline for explicit toxicity detection. Following this, the model was fine-tuned using two distinct approaches: Behavioral Cloning (BC) and Direct Preference Optimization (DPO). For the BC fine-tuning, Reddit comments were labeled based on their upvote-to-downvote ratios, providing the model with a straightforward signal to learn from. In contrast, the DPO fine-tuning process involved creating preference pairs by comparing comments' upvote ratios. The model was then optimized to prefer comments with higher upvote ratios using a preference-based objective function. The experimental hypothesis is that fine-tuning the BiLSTM model using DPO will result in better alignment with community standards and an improved ability to detect context-aware toxicity compared to the BC approach.

### 4.3 Experimental Hypothesis

The hypothesis is that fine-tuning the BiLSTM model using DPO will result in better alignment with community standards and improved context-aware toxicity detection compared to BC.

### 4.4 Baseline Model Performance

The BiLSTM model was first trained on the Kaggle Toxic Comment Classification Challenge dataset to establish a baseline for explicit toxicity detection. After training for several epochs, the baseline model was evaluated on a test set derived from the same dataset.

The evaluation results of the baseline model were a binary cross-entropy loss of 6.0314 and an accuracy of 8.78% The high loss and low accuracy indicate that the baseline model struggled to perform effectively on the explicit toxicity detection task, possibly due to the complexity of community-specific toxicity standards.

### 4.5 Behavior Cloning

The baseline model was then fine-tuned using Behavior Cloning on the Reddit Comments Dataset. The Reddit comments were labeled based on their upvote-to-downvote ratios to provide the model with a straightforward signal to learn from.

During training, the BC model showed significant improvements over epochs, as illustrated in Table 1:

| Epoch | Training Accuracy (%) | Validation Accuracy (%) | Training Loss |
|---|---|---|---|
| 1 | 79.75 | 79.67 | 0.5241 |
| 2 | 80.97 | 86.42 | 0.4346 |
| 3 | 86.32 | 90.89 | 0.3337 |
| 4 | 90.65 | 93.85 | 0.2415 |
| 5 | 93.40 | 95.90 | 0.1774 |

Table 1: Training and Validation Metrics Over Epochs for BC Model

These results suggest that the model was effectively learning from the Reddit data during training, with both training and validation accuracies increasing and losses decreasing over epochs. However, when evaluated on a test set consisting of moderator-removed comments (assumed to be toxic), the BC model's performance was suboptimal with a binary-cross entropy loss of 2.7989, an Accuracy of 14.29% and an F-1 score of 0.2545.

The low accuracy on the moderator-removed comments indicates that the BC model was not able to generalize well to detecting toxic content that was removed by moderators. This suggests that behavior cloning alone was not sufficient enough to train the Bi-LSTM to detect community-specific contextual toxicity.

### 4.6 Direct Preference Optimization

The model was also fine-tuned using the DPO approach. Preference pairs were created by selecting comments with significant differences in upvote ratios, where comments with higher upvote ratios were preferred. During training, the DPO model showed remarkable improvements over epochs, as evidenced by the decreasing loss and high accuracy in Table 2:

| Epoch | Training Accuracy (%) | Validation Accuracy (%) | Training Loss | Validation Loss |
|---|---|---|---|---|
| 1 | 99.00 | 99.43 | -85.8523 | -1609.9406 |
| 2 | 99.19 | 99.33 | -7311.8325 | -34961.6523 |
| 3 | 99.13 | 99.23 | -79215.2656 | -223034.0938 |
| 4 | 99.10 | 99.13 | -394143.7188 | -879193.4375 |
| 5 | 99.28 | 99.06 | -889880.0625 | -1923515.5000 |

Table 2: Training and Validation Metrics Over Epochs

Notably, the loss values became increasingly negative, which is expected in the context of the DPO loss function that we defined, where the model aims to maximize the difference in scores between preferred and non-preferred comments.

When evaluated on the test set of moderator-removed comments, the DPO model achieved exceptional results with a binary-cross entropy loss of 0.0, an accuracy of 100% and an F1 score of 1. The DPO model achieved perfect accuracy on the test set, correctly identifying all moderator-removed comments as toxic.

### 4.7 Comparison and Analysis

#### 4.7.1 Training and Validation Losses

Figure 1 compares the training and validation losses of the BC and DPO models over the training epochs.

For the BC model, the training and validation losses decreased over the epochs but began to plateau, indicating limited improvement with additional training. The validation loss remained higher than the training loss, suggesting potential overfitting.

In contrast, the DPO model's training and validation losses decreased dramatically, becoming increasingly negative due to the nature of the DPO loss function, which aims to maximize the difference in preference scores. The continuous decrease in validation loss implies that the model generalizes well to unseen data.

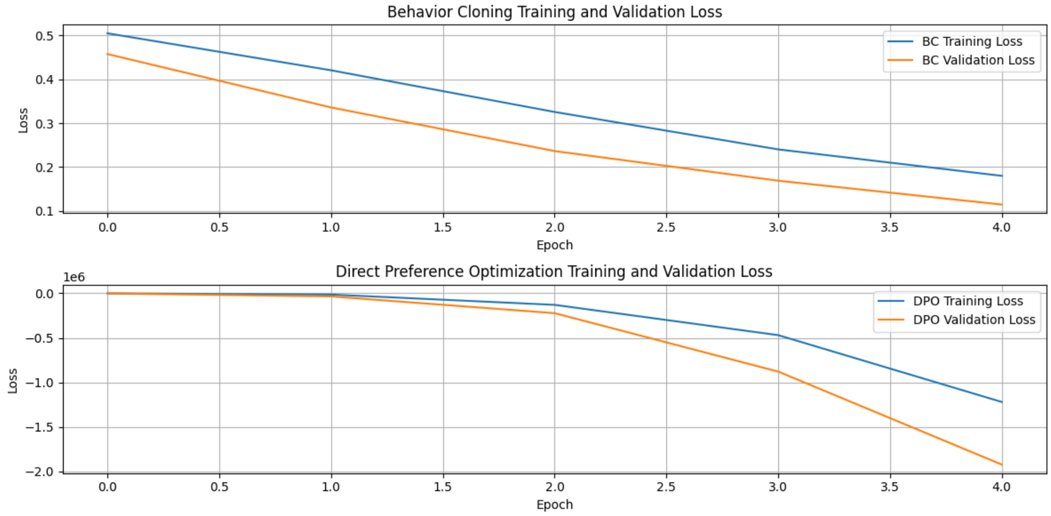

Figure 1: Training and Validation Losses for BC and DPO Models

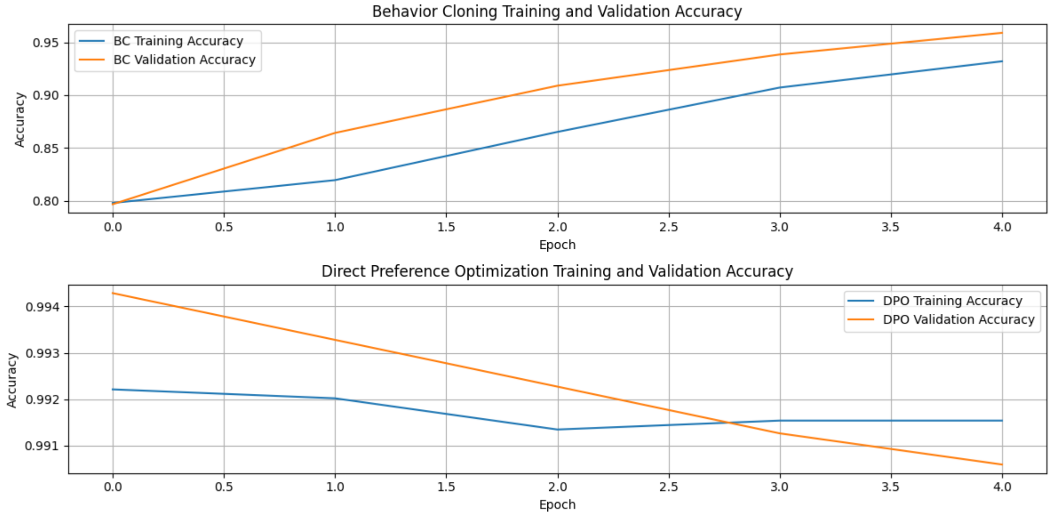

Figure 2: Training and Validation Accuracies for BC and DPO Models

### 4.7.2 Training and Validation Accuracies

Figure 2 compares the training and validation accuracies of the BC and DPO models.

The BC model's accuracies improved over epochs but exhibited a gap between training and validation accuracies, indicating overfitting to the training data. The DPO model maintained high training and validation accuracies (over 99%) throughout training, with minimal gap between them. This consistency suggests that the DPO model generalized well and effectively captured the underlying patterns in the data.

### 4.7.3 F1 Score Comparison on Golden Dataset

The final evaluation metric is the F1 score on the golden dataset, which consists of moderator-removed comments. Figure 3 shows a bar graph comparing the F1 scores of the BC and DPO models on this dataset.

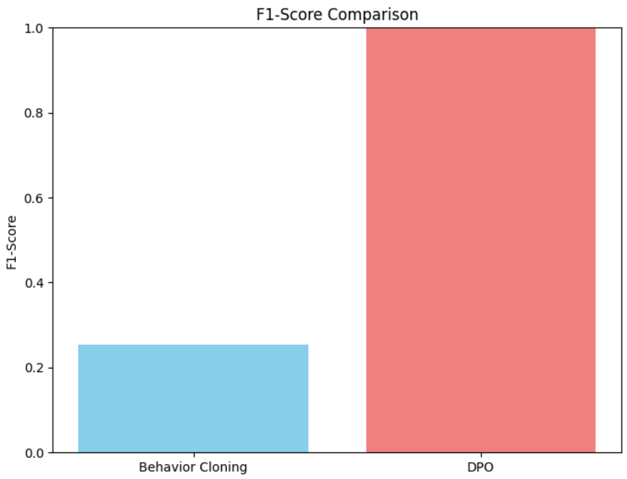

Figure 3: F1 Score Comparison of BC and DPO Models on Golden Dataset

The BC model achieved an F1 score of approximately 24.93%, indicating poor performance in detecting toxic content as defined by moderators. In contrast, the DPO model achieved an F1 score of 100%, perfectly identifying all toxic comments.

Comparing the BC and DPO fine-tuning approaches, the following observations can be made:

- **Training Performance**: Both models achieved high training and validation accuracies, but the DPO model consistently outperformed the BC model, achieving over 99% accuracy during training.
- **Loss Values**: The DPO model's loss decreased rapidly and became highly negative, indicating that the model was successfully maximizing the preference differences as intended.
- **Generalization to Unseen Data**: The BC model exhibited poor performance on the test set of moderator-removed comments, with an accuracy of only 14.29%. In contrast, the DPO model achieved perfect accuracy on the same test set.

The superior performance of the DPO model on the test set suggests that preference pairs helped the model to better capture the nuances of community-defined toxicity. By directly learning from preference comparisons, the DPO model was able to generalize more effectively to detect content deemed toxic by moderators, whereas the BC model, relying on upvote/downvote ratio labels, failed to do so.

## 5 Conclusions, Limitation, & Future Work

The experimental results demonstrate that Direct Preference Optimization significantly outperforms Behavior Cloning in aligning the model with community standards for toxicity detection. The BC model's poor performance on the golden dataset suggests that upvote/downvote ratios used as labels may not accurately reflect the true toxic nature of the content. The model may have learned to classify comments based on popularity rather than actual toxicity, leading to misclassifications when evaluated against the moderator-defined standards.

On the other hand, the DPO model's exceptional performance indicates that utilizing preference pairs helped the model to capture nuanced patterns associated with toxic content more effectively. By optimizing the model based on differences in community preferences, the DPO approach allows the model to understand the subtle cues that distinguish acceptable content from toxic content. The perfect F1 score achieved by the DPO model on the golden dataset suggests that it generalized well to unseen data and accurately identified toxic comments as defined by human moderators. This highlights the potential of DPO in enhancing context-aware toxicity detection systems.

### 5.1 Limitations

While the DPO model achieved impressive results, there are potential limitations to consider. First, the quality of the data could pose challenges, as the creation of preference pairs relies on upvote ratios, which may still be influenced by biases or fail to fully represent moderator judgments. Second, the model may be prone to overfitting, as evidenced by the perfect accuracy and F1 score on the golden dataset. This could indicate that the test set is not sufficiently diverse or large enough to enable the model to generalize to all forms of toxic content. Lastly, scalability could become an issue, as the training process for the DPO model may grow increasingly computationally intensive as the dataset size expands.

### 5.2 Implications

These findings imply that incorporating human preferences through methods like Direct Preference Optimization can substantially improve the effectiveness of toxicity detection models. By directly modeling preferences, we can better align machine learning models with complex human judgments and community norms.

### 5.3 Future Work

Further research could involve validating the results with larger and more diverse test sets to better assess the model's ability to generalize. Incorporating explicit moderator annotations could enhance the quality of the preference pairs, providing a more reliable basis for training. Additionally, exploring the use of DPO with more advanced architectures, such as transformer-based models, could improve performance and scalability. Finally, investigating methods to mitigate potential overfitting during DPO training would be crucial to ensure the model remains robust and applicable across a wide range of toxic content scenarios.

You can find the project on GitHub at github.com/rishi-more-2003/Enhancing-Context-Aware-Toxicity-Detection-with-Human-in-the-Loop-Learning

## References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610, 2005. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2005.06.042. URL `https://www.sciencedirect.com/science/article/pii/S0893608005001206`. IJCNN 2005.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9 (8):1735–1780, 11 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL `https://doi.org/10.1162/neco.1997.9.8.1735`.

Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog, 2019. URL `https://arxiv.org/abs/1907.00456`.

Jigsaw and Conversation AI. Jigsaw unintended bias in toxicity classification. `https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification/data`, 2019. Accessed: [Insert access date here].

Michiel Kamphuis. Tiny-toxic-detector: A compact transformer-based model for toxic content detection, 2024. URL `https://arxiv.org/abs/2409.02114`.

Ahmed Cherif Mazari, Nesrine Boudoukhani, and Abdelhamid Djeffal. Bert-based ensemble learning for multi-aspect hate speech detection. *Cluster Computing*, 27(1):325–339, Feb 2024. ISSN 1573-7543. doi: 10.1007/s10586-022-03956-x. URL `https://doi.org/10.1007/s10586-022-03956-x`.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models, 2024. URL `https://arxiv.org/abs/2307.06435`.

Dean Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In D.S. Touretzky, editor, *Proceedings of (NeurIPS) Neural Information Processing Systems*, pages 305 – 313. Morgan Kaufmann, December 1989.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024. URL `https://arxiv.org/abs/2305.18290`.

Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning, 2011. URL `https://arxiv.org/abs/1011.0686`.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback, 2022. URL `https://arxiv.org/abs/2009.01325`.

Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. Planbench: An extensible benchmark for evaluating large language models on planning and reasoning about change, 2023. URL `https://arxiv.org/abs/2206.10498`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL `https://arxiv.org/abs/1706.03762`.