Fine-Tuning Large Language Models for Detecting Unsafe Speech in Human-AI Interactions with Explainable Bias Analysis

Andrew Tremante
Amherst College
atremante26@amherst.edu

Tyler McCord
Amherst College
tmccord26@amherst.edu

Michael Allers
Amherst College
mallers26@amherst.edu

Abstract

As conversational AI systems are increasingly deployed in real-world settings, failures in content safety can lead to harmful advice, biased language, or policy violations with serious consequences. In this work, we investigate multi-dimensional content safety detection using the DICES-350 dataset, which contains human safety ratings for human–AI conversations. We compare three approaches: a TF–IDF logistic regression baseline, a single-task RoBERTa classifier, and multi-task RoBERTa models with task-specific classification heads. To examine how different architectures make safety decisions, we apply three explainability methods: LIME, SHAP, and Integrated Gradients. We then analyze token-level attribution patterns across models. This study provides a pragmatic comparison of modeling and explainability approaches for understanding and improving safety in conversational AI systems.

1 Introduction

Ensuring content safety has become a central challenge for modern conversational AI systems. As large language models are increasingly deployed in chatbots and virtual assistants, safety failures can lead to the generation of harmful advice, biased language, or violations of platform policies, with significant real-world consequences. Content safety, however, is not a single concept: it spans multiple dimensions, including harmful content detection, bias identification, and policy compliance, each requiring different forms of semantic and contextual reasoning.

Most existing approaches to content safety treat these dimensions independently, training separate classifiers for each safety category. While this enables specialized detection, it ignores the fact that different types of unsafe content often co-occur. Multi-task learning (Caruana, 1997) provides an alternative. By training a single model to jointly predict multiple safety dimensions, the model may learn shared representations that capture common patterns of unsafe behavior while still modeling task-specific distinctions.

Beyond detection accuracy, explainability is a critical requirement for safety systems. Black-box predictions are insufficient in settings where decisions must be transparent, verifiable, and aligned with human intent. Explainability methods help reveal which tokens or phrases drive a model's predictions. Comparing explanations across model architectures further allows us to study how increasing model complexity affects both reasoning behavior and interpretability.

In this work, we study the effectiveness of multi-task learning for content safety detection and analyze how different model architectures make their predictions through explainability. Using the DICES-350 dataset (Aroyo et al., 2023), we train and evaluate three models: (1) a logistic regression baseline with TF-IDF features, (2) a single-task RoBERTa transformer (Liu et al., 2019) trained to predict overall safety, and (3) a multi-task RoBERTa transformer with a shared encoder and task-specific classification heads. We evaluate performance across four safety dimensions: overall safety, harmful content, bias, and policy violations.

2 Related Work

2.1 Content Safety Detection

Existing work on content safety and toxic language detection has explored a wide range of modeling approaches. Elbasani et al. (Elbasani and Kim, 2022) incorporate Abstract Meaning Representation (AMR) into neural models to improve toxic content detection, demonstrating that structured semantic representations can enhance model performance on safety tasks.

More recent work has emphasized the challenges inherent in annotating and evaluating safety-related data. Giorgi et al. (Giorgi, 2025) examine biases in gold-labeled hate speech datasets, showing that both human annotators and LLMs exhibit systematic biases, but often in different forms. This highlights the complexity of establishing ground truth for subjective safety judgments.

2.2 Explainability in Hate Speech Detection

Explainability has become a central concern in machine learning, particularly for neural networks whose decision processes are often opaque. Ibrahim et al. (Ibrahim et al., 2022) study the explainability of machine learning models for hate speech detection, using LIME to compare how different classifiers justify their predictions on Indonesian Twitter data. Their work demonstrates that explanation methods can reveal important differences in how models rely on specific linguistic features.

Amran et al. (Amran et al., 2024) further emphasize the role of explanation methods by applying LIME and other interpretability techniques to existing hate speech classifiers, showing that explainability can improve transparency for content moderation systems. Beyond traditional explainability methods, recent work has explored generative approaches. Piot et al. (Piot and Parapar, 2025) investigate prompt engineering methods like Chain-of-Thought to ask large language models why text is considered hate speech, while Hashir et al. (Hashir et al., 2025) propose frameworks for models to generate explicit natural language explanations alongside predictions. These approaches differ from attribution-based methods like LIME and SHAP but share the goal of improving transparency in safety systems.

3 Dataset

We use the DICES-350 dataset (Aroyo et al., 2023), a collection of 350 adversarial multi-turn conversations designed to evaluate safety in dialog systems. The dataset was created by having human raters interact with a dialog model to generate challenging conversations, which were then rated across multiple safety dimensions by a diverse rater pool.

3.1 Dataset Structure and Preprocessing

Each conversation in DICES-350 consists of multiple turns between a user and an AI assistant, culminating in a final response that is evaluated for safety. The dataset includes both conversation context and final response, which we concatenate as input. A total of 123 raters provided annotations, though we follow the authors' recommendation to filter out 19 low-quality raters, resulting in 104 high-quality raters whose annotations we use.

DICES-350 evaluates conversations across four primary safety dimensions: $Q\_overall$ (overall safety assessment), $Q2\_harmful$ (harmful content including dangerous advice, violence, derogatory language), $Q3\_bias$ (unfair bias targeting beliefs, gender, orientation, or socioeconomic status), and $Q6\_policy$ (policy violations including polarizing topics without appropriate framing).

The original annotations use a three-way classification scheme: "Yes" (unsafe), "No" (safe), and "Unsure". Due to severe class imbalance in the "Unsure" category, we convert to binary classification by mapping Yes $\rightarrow$ unsafe, No $\rightarrow$ safe, and discarding Unsure ratings. This preprocessing removes ambiguous cases and focuses models on clear safety boundaries. After filtering low-quality raters, our final dataset contains 33,528 individual rater-conversation annotations.

3.2 Data Splits and Class Imbalance

We partition data using conversation-level splitting to prevent data leakage. All ratings for a given conversation are assigned to the same split, ensuring models never see test conversations during training. We use an 80/10/10 split ratio: 23,396 training ratings, 6,803 validation ratings, and 3,329 test ratings. This conversation-level approach is crucial because it prevents models from seeing different raters' judgments of the same conversation during training and testing, artificially inflating performance.

The dataset exhibits severe class imbalance across all tasks: $Q\_overall$ (34.5% unsafe), $Q2\_harmful$ (19.3% unsafe), $Q3\_bias$(9.1% unsafe), and $Q6\_policy$(11.3% unsafe). See Appendix A for detailed distributions. This extreme imbalance, particularly for $Q3\_bias$ and $Q6\_policy$, presents a significant challenge for model training and requires careful handling through weighted loss functions (Section 4.3) and undersampling for logistic regression (Section 4.1).

4 Methodology

4.1 Baseline: Logistic Regression

We first introduce a logistic regression model as a non-neural baseline for content classification. This model serves as a reference point for evaluating the benefits of contextual information through transformer-based approaches and provides a baseline for both performance and interpretability.

Model Formulation Given an input text x, we represent it using TF-IDF vectorization, which maps text to a sparse vector where each dimension corresponds to a term weighted by its frequency in the document and inverse frequency across all documents. The logistic regression model then computes the probability that a text is unsafe as:

$$\hat{y} = \sigma(w^\top x + b) \tag{1}$$

where w is a learned weight vector, b is a bias term, and σ is the logistic sigmoid function.

Training and Implementation The model is trained by minimizing binary cross-entropy loss over the training data. We implement this using scikit-learn's `LogisticRegression` with `TfidfVectorizer`, applying balanced class weights to handle imbalance. This approach treats each word as a unigram without surrounding context, providing a lower bound on performance for our task.

4.2 Single-Task Transformer

We introduce a single-task RoBERTa-base (Liu et al., 2019) transformer model for binary classification on the DICES-350 dataset. RoBERTa builds upon BERT (Devlin et al., 2019) with improved pretraining procedures, making it well-suited for fine-tuning on downstream classification tasks.

Architecture The model employs a standard RoBERTa-base encoder followed by a single classification head. Given an input sequence x, the encoder produces contextualized token representations $h = \text{RoBERTa}(x)$, from which we extract the final-layer representation of the special classification token, $h_{[\text{CLS}]}$. This representation is passed through a linear layer to produce two output logits corresponding to the "safe" and "unsafe" classes:

$$z = W h_{[\text{CLS}]} + b, \tag{2}$$

where $W \in \mathbb{R}^{2\times d}$ and $b \in \mathbb{R}^2$ are learned parameters and d denotes the hidden dimension of the RoBERTa encoder. Class probabilities are obtained by applying the softmax function:

$$\hat{y} = \text{softmax}(z), \tag{3}$$

with $\hat{y}_1$ representing the predicted probability that the input is unsafe.

Training Procedure We fine-tune the model using cross-entropy loss, which is standard for binary classification with mutually exclusive labels. Despite the class imbalance of $Q\_overall$, we do not apply explicit class weighting in the single-task setting, instead allowing the model to optimize directly for empirical risk minimization.

Training is conducted using the AdamW optimizer (Loshchilov and Hutter, 2019) with a learning rate of 2×10^{-5}, batch size of 64, and maximum input sequence length of 128 tokens. We apply a linear learning rate schedule, where the learning rate decays linearly from its initial value to zero over the course of training. We train for 3 epochs, selecting the final checkpoint based on validation performance.

4.3 Multi-Task Transformer

We implement a multi-task learning architecture based on RoBERTa-base (Liu et al., 2019), designed to simultaneously predict multiple safety dimensions from the DICES-350 dataset. Our architecture consists of a shared transformer encoder (Vaswani et al., 2017) with task-specific classification heads, enabling the model to learn both shared and task-specific representations for content safety detection.

Architecture The model employs a shared RoBERTa encoder that processes input text into contextualized token representations. For each safety task $t \in \{Q_{\text{overall}}, Q2_{\text{harmful}}, Q3_{\text{bias}}, Q6_{\text{policy}}\}$, we add a task-specific classification head consisting of a linear layer that projects from the encoder's hidden dimension to a single output logit. This architecture allows the encoder to learn generalizable features across all safety dimensions while the task-specific heads capture dimension-specific patterns.

Given an input sequence x, the shared encoder produces contextualized representations $h = \text{RoBERTa}(x)$, from which we use the final-layer representation of the special classification token. For each task t, the classification head computes z_t, the logit (raw prediction score), and $\hat{y}_t$, the predicted probability that the input is unsafe:

$$z_t = W_t h_{[\text{CLS}]} + b_t \tag{4}$$

$$\hat{y}_t = \sigma(z_t) \tag{5}$$

where W_t and b_t are task-specific learned parameters, $h_{[\text{CLS}]}$ is the representation of the [CLS] token, and σ is the sigmoid function.

Training Procedure We train the model using binary cross-entropy loss with logits for each task. For the multi-task transformer, we address the class imbalance directly by applying positive class weighting using the inverse frequency ratio:

$$w_{\text{pos}} = \frac{N_{\text{neg}}}{N_{\text{pos}}} \tag{6}$$

where N_{neg} and N_{pos} are the number of negative and positive samples in the training set for each task respectively. The total loss is the average of weighted losses across all tasks:

$$\mathcal{L} = \frac{1}{|T|} \sum_{t \in T} \text{BCEWithLogits}(z_t, y_t, \text{pos\_w} = w_t) \tag{7}$$

We trained two variants to investigate the impact of task quantity on multi-task learning: a 2-task model ($Q\_overall$, $Q2\_harmful$) and a 4-task model (all safety dimensions). The 2-task variant focuses on the two tasks with relatively higher positive class frequencies, while the 4-task variant includes the severely imbalanced $Q3\_bias$ and $Q6\_policy$ tasks. This comparison allows us to examine whether extreme class imbalance in some tasks negatively impacts multi-task learning through negative transfer.

Training used AdamW optimizer (Loshchilov and Hutter, 2019) with learning rate 2e-5, batch size 16, and maximum sequence length of 256 tokens. We applied linear learning rate warmup for 10% of training steps followed by linear decay. We selected checkpoints based on validation PR-AUC to avoid overfitting: epoch 1 for the 2-task model, epoch 4 for the 4-task model.

5 Explainability Methods

To understand how different model architectures make safety predictions, we apply three complementary explainability methods: LIME, SHAP, and Integrated Gradients. Each method provides different insights into model reasoning, and comparing them across architectures reveals important differences in how models process safety-related content.

5.1 LIME

Local Interpretable Model-agnostic Explanations (LIME) (Ribeiro et al., 2016) is a perturbation-based explanation method that provides local, human-interpretable explanations for individual model predictions. Unlike gradient-based approaches, LIME treats the underlying model as a black box and approximates its behavior in the neighborhood of a specific input using a simple surrogate model.

Methodology Given an input text instance x and a classifier f that outputs class probabilities, LIME constructs a local explanation by generating a set of perturbed samples $\{x'_1, \ldots, x'_N\}$ around x. In the text domain, perturbations are created by randomly removing or masking words from the original input. Each perturbed sample is then passed through the original classifier to obtain predicted probabilities.

LIME fits a sparse linear surrogate model g to approximate the classifier's behavior in the vicinity of x by minimizing:

$$\mathcal{L}(f, g, \pi_x) + \Omega(g), \tag{8}$$

where $\mathcal{L}$ measures the fidelity of the surrogate model to the original classifier weighted by a locality kernel π_x, and $\Omega(g)$ is a regularization term that encourages interpretability through sparsity. The learned weights of the surrogate model indicate the contribution of individual words to the prediction for a specific class.

Implementation We implemented LIME using the `LimeTextExplainer` from the official LIME library (Ribeiro et al., 2016). For each explained instance, we sampled 500 perturbed text variants and trained a local linear surrogate model to approximate the classifier's decision boundary. Explanations were generated for the `unsafe` class by inspecting the top-weighted features (words or phrases) that most strongly increased or decreased the predicted probability of unsafe content.

For transformer-based models, we provided LIME with a custom classifier function that maps a list of input texts to class probabilities. In the single-task RoBERTa model, probabilities were obtained by applying a softmax function to the model's output logits. In the multi-task transformer, probabilities for the target safety dimension were computed using a sigmoid function over the task-specific output logit and converted into a two-class probability vector.

5.2 SHAP

SHapley Additive exPlanations (SHAP) (Lundberg and Lee, 2017) is a framework for model interpretability that explains individual predictions by assigning each feature an attribution value corresponding to its marginal contribution to the model's prediction, averaged across all possible feature groups.

Methodology Given an input instance x and a predictive model f, SHAP computes feature attributions based on Shapley values, a concept derived from cooperative game theory. In the context of a sentence, each word is treated as a "player" in a game where the payout is the model's prediction, and the Shapley value measures how much the feature contributes to the prediction on average across all subsets of features. SHAP decomposes the model output into the sum of feature-level contributions relative to a baseline expectation.

Implementation We implemented SHAP using the `shap` library (Lundberg and Lee, 2017), adapting the explainer to each model. For the logistic regression baseline, we used `LinearExplainer`, which leverages the linear structure of the model to compute exact SHAP values for the TF-IDF features.

For transformer-based models, we used the model-agnostic `shap.Explainer` with a text masker constructed from the corresponding tokenizer. We defined prediction functions that map input text to the probability of unsafe text. In the single-task RoBERTa model, this probability was obtained via a softmax over the output logits, whereas in the multi-task RoBERTa models, task-specific logits were passed through a sigmoid function for the target safety dimension.

5.3 Integrated Gradients

Integrated Gradients (IG) (Sundararajan et al., 2017) is a gradient-based attribution method that assigns importance scores to input features by integrating gradients along a path from a baseline input to the actual input.

Methodology Given an input x and a baseline x' (typically all-zeros or padding tokens), IG computes the attribution for the i-th feature as:

$$\text{IG}_i(x) = (x_i - x'_i) \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha(x - x'))}{\partial x_i} d\alpha \tag{9}$$

where F is the model's prediction function. In practice, this integral is approximated using Riemann summation with m steps.

For transformer models, we apply IG at the token embedding level. The baseline is constructed using padding tokens, and attributions are computed for each token in the input sequence. This provides token-level importance scores that reflect each token's contribution to the model's prediction.

Implementation We implemented IG using the Captum library's (Kokhlikyan et al., 2020) `LayerIntegratedGradients`, applying it to the embedding layer of our transformer models. We used $m = 25$ integration steps, which prior work has shown provides a good balance between computational cost and approximation accuracy (Sundararajan et al., 2017).

6 Experiments

6.1 Experimental Setup

We evaluate all models on the test set using conversation-level splits. We report two primary metrics: F1 score (positive class) to emphasize correct identification of harmful content, and PR-AUC to measure discrimination quality across all thresholds, which is particularly appropriate for imbalanced datasets.

6.2 Research Questions

We structure our experiments around five testable hypotheses:

H1: Transformer Superiority Fine-tuned transformer models (RoBERTa) will outperform traditional logistic regression baselines on multi-dimensional safety detection.

H2: Multi-Task Benefits Multi-task learning with shared representations across safety dimensions will improve performance compared to single-task models.

H3: Class Imbalance Impact Severe class imbalance ($\leq$10% positive examples) will significantly degrade model performance, and including such tasks in multi-task training will cause negative transfer.

H4: High-Confidence Concentration High-confidence predictions (probability > 0.98) will show concentrated attribution patterns, focusing on explicit unsafe keywords.

H5: Borderline Diffuseness Less certain predictions (probability 0.45-0.55) will show diffuse attribution patterns, relying on broader contextual reasoning.

Table 1: Model performance on DICES-350 test set. Best F1 scores per task in **bold**.

Model	Task	F1	PR-AUC	Pos%
LogReg	Q_overall	0.422	0.414	35.1%
Single-Task	Q_overall	0.543	0.660	34.1%
Multi-Task-2	Q_overall	0.469	0.366	32.4%
Multi-Task-4	Q_overall	0.461	0.370	32.4%
Multi-Task-2	Q2_harmful	0.426	0.345	17.7%
Multi-Task-4	Q2_harmful	0.414	0.422	17.7%
Multi-Task-4	Q3_bias	0.322	0.201	9.8%
Multi-Task-4	Q6_policy	0.282	0.163	10.0%

6.3 Results

We present results from three experiments testing our hypotheses, followed by qualitative explainability analysis.

6.3.1 Experiment 1: Model Performance

Table 1 presents F1 and PR-AUC scores for all models across the four safety tasks. See Appendix B for detailed visualizations.

H1: Transformer Superiority The single-task RoBERTa model substantially outperforms the logistic regression baseline on $Q\_overall$, achieving F1=0.543 vs 0.422 (+29% improvement) and PR-AUC=0.660 vs 0.414 (+59% improvement). This confirms that fine-tuned transformers leverage contextual information beyond keyword matching to make more nuanced safety judgments.

H2: Multi-Task Benefits Contrary to our hypothesis, multi-task models underperform the single-task specialist on $Q\_overall$. The 4-head model achieves F1=0.461 and the 2-head model achieves F1=0.469, both significantly below the single-task F1 of 0.543. This suggests negative transfer from auxiliary tasks rather than beneficial shared representations.

H3: Class Imbalance Impact Performance degrades sharply on severely imbalanced tasks. Tasks with $\leq$10% positive examples ($Q3_bias$ at 9.8%, $Q6\_policy$ at 10.0%) show 30-40% lower F1 scores compared to $Q\_overall$. Specifically, $Q3\_bias$ achieves F1=0.322 and $Q6\_policy$ achieves F1=0.282, confirming that extreme imbalance significantly degrades model performance.

6.3.2 Experiment 2: Multi-Task Comparison

To isolate the effect of severely imbalanced auxiliary tasks, we compare the 2-head vs 4-head models on the two shared tasks. See Appendix C for additional visualizations.

Table 2: Multi-task comparison on shared tasks (Q_overall, Q2_harmful)

Model	Task	F1	PR-AUC	Δ F1
2-Head	Q_overall	0.469	0.366	+0.008
4-Head	Q_overall	0.461	0.370	–
2-Head	Q2_harmful	0.426	0.345	+0.011
4-Head	Q2_harmful	0.414	0.422	–

H3: Negative Transfer The 2-head model consistently outperforms the 4-head model on both shared tasks (F1 improvements of +0.008 to +0.011), providing evidence that adding severely imbalanced auxiliary tasks causes negative transfer. The shared encoder must balance learning from $Q\_overall$ with sparse signals from $Q3\_bias$ and $Q6\_policy$, degrading performance on the primary tasks despite weighted loss.

Interestingly, the 4-head model shows better PR-AUC on both tasks, suggesting a trade-off: 2-head provides better threshold-based classification (F1) while 4-head shows improved ranking and calibration (PR-AUC). This indicates that while the 4-head model struggles with binary classification decisions, it may produce improved probability estimates across the full range of predictions.

6.3.3 Experiment 3: Attribution Concentration

We analyze 30 examples per model (15 high-confidence [prob > 0.98], 15 borderline [0.45-0.55]) using Integrated Gradients to test whether prediction confidence correlates with attribution concentration. See Appendix D for supporting visualizations.

Table 3: Attribution concentration metrics by confidence level

Model	Conf.	Top-10 Mass	Entropy	Tokens (Avg)	Prob
Single-Task	High	0.481	3.659	60	0.983
Single-Task	Border	0.180	5.143	254	0.496
Multi-Task	High	0.378	4.054	93	0.993
Multi-Task	Border	0.303	4.518	152	0.501

H4: High-Confidence Concentration Single-task high-confidence predictions show 48% of attribution concentrated in the top-10 tokens vs only 18% for borderline (Δ = +0.301, 167% increase). These predictions focus on explicit unsafe keywords (“bomb”, “kill”, “explosive”) in short texts (average 60 tokens), suggesting the model makes decisive classifications when obvious unsafe content is present.

H5: Borderline Diffuseness Borderline predictions exhibit significantly higher entropy (5.143 vs 3.659, Δ = -1.485, 41% increase in diffuseness), spreading attribution across broader context in much longer texts (average 254 tokens vs 60). This indicates that when models are uncertain, they rely on distributed contextual reasoning across the entire input rather than specific keywords.

6.4 Explainability Analysis

We conduct qualitative analysis using LIME and SHAP to understand how different architectures reason about safety, complementing our quantitative Integrated Gradients analysis.

6.4.1 LIME: Feature Type Analysis

We apply LIME to 5 shared examples across all three model types, analyzing the top 5-8 features contributing to unsafe predictions. We categorize features as *content words* (nouns, verbs, adjectives with semantic meaning) vs *function words* (pronouns, determiners, negations with primarily syntactic roles). See Appendix E for detailed attribution strength distributions.

- **Logistic Regression:** 88% content words (keyword-focused: "abortion", "vote")
- **Single-Task Transformer:** 40% content words (function word-heavy: "not", "I")
- **Multi-Task Transformer:** 60% content words (balanced approach)

Multi-task learning shifts transformer attention patterns toward more content-focused reasoning, bridging the gap between logistic regression's keyword focus and single-task's structural reliance. This suggests multi-task models learn more semantically grounded representations despite lower F1 scores.

6.4.2 SHAP: Method Validation

We test SHAP explainability across all model architectures:

- **Logistic Regression:** SHAP produces clean, interpretable attributions over TF-IDF features (e.g., "america" +0.057).
- **Transformers (Both):** SHAP fails, producing empty token strings and near-zero attributions (0.0001) across all tokens.

This failure is a known limitation of SHAP for deep NLP models operating on high-dimensional embeddings (768-d) with complex attention mechanisms. As a result, we conclude that gradient-based methods (Integrated Gradients) are required for faithful transformer attribution, while SHAP remains appropriate for linear models.

7 Analysis and Discussion

7.1 Limitations of Multi-Task Learning

Our results contradict the assumption that multi-task learning improves performance through shared representations. We identify three factors explaining why multi-task underperforms.

1. Negative Transfer from Imbalance Experiment 2 shows adding $Q3\_bias$ and $Q6\_policy$ (10% positive) degrades performance on $Q\_overall$ and $Q2\_harmful$ by 1-2%. The encoder struggles to balance learning from strong signals with sparse signals. Weighted loss provides some mitigation but cannot fully resolve this tension.

2. Limited Task Similarity While all tasks relate to safety, they may capture sufficiently distinct phenomena that shared representations provide limited benefit. Detecting explicit harm differs fundamentally from identifying bias or policy issues, suggesting multi-task learning works best when tasks share substantial underlying structure.

3. Single-Task Specialization The single-task model optimizes purely for $Q\_overall$, developing focused keyword detection. It dedicates all capacity to learning relevant patterns, while the multi-task model must allocate capacity across multiple objectives, potentially diluting effectiveness on any single task.

7.2 Multi-Task Interpretability Advantage

Despite lower F1 scores, multi-task models show a key advantage: more interpretable, semantically grounded reasoning patterns.

Multi-task models use 60% content words (vs 40% for single-task), focusing more on meaningful semantic features. This makes model decisions more transparent and aligned with human reasoning, as content words are typically the features humans consider when assessing safety.

For production safety systems, the interpretability-performance trade-off may favor multi-task models in scenarios requiring human oversight, auditability, or resistance to adversarial keyword substitutions (Ribeiro et al., 2016). In settings where model decisions must be explainable to content moderators or where adversaries might attempt to evade detection by avoiding specific keywords, the multi-task model's more distributed reasoning may be preferable despite its lower F1 score.

7.3 Class Imbalance Strategies

Class imbalance is a fundamental architectural consideration for multi-task safety systems. For balanced tasks, single-task transformers provide best F1 scores. For imbalanced tasks, both models struggle due to sparse positive signals. For multi-task systems, pair tasks with balance since vastly different imbalance levels cause the shared encoder to struggle balancing optimization objectives.

7.4 Attribution Concentration Insights

The correlation between confidence and attribution concentration (H4/H5) reveals how models allocate resources. High-confidence predictions use fast, keyword-based decisions ("bomb", "kill") with high certainty, efficient but potentially vulnerable to adversarial attacks. Borderline predictions distribute attention broadly, integrating information across long contexts for nuanced decisions approaching human-level uncertainty. Text length correlates with complexity: high-confidence examples average 60 tokens (simple, explicit) while borderline average 254 tokens (complex, contextual), suggesting longer texts require more complex reasoning.

8 Conclusion

In this work, we evaluated three approaches to content safety detection on DICES-350: TF-IDF logistic regression, single-task RoBERTa, and multi-task RoBERTa. Fine-tuned transformers substantially outperformed the baseline, but multi-task learning did not improve F1 and exhibited negative transfer when highly imbalanced tasks (bias and policy) were included. The single-task transformer achieved strongest performance.

Using LIME, SHAP, and Integrated Gradients, we found logistic regression relies on explicit content words, single-task emphasizes syntactic cues, and multi-task balances both. IG showed high-confidence predictions concentrate on few tokens while borderline cases use diffuse, context-dependent reasoning. SHAP worked for linear models but failed for transformers, highlighting the need for gradient-based methods for deep architectures.

Our work reveals a key trade-off: single-task achieves higher F1 through focused keyword detection, while multi-task offers more interpretable reasoning at lower performance. The choice depends on deployment requirements: maximizing accuracy versus ensuring interpretability and robustness. These findings suggest the *best* approach depends on operational context and constraints.

Limitations Our work has several constraints. DICES-350's small scale (350 conversations, 33,528 ratings) limits the generalizability of our findings to larger, more diverse datasets. We used equal task weighting in multi-task training; adaptive weighting schemes or task curriculum learning might mitigate negative transfer. We tested only shared-encoder multi-task architectures; task-specific intermediate layers or mixture-of-experts approaches might better handle class imbalance. Our Integrated Gradients analysis covered 30 examples per model; larger-scale attribution analysis would strengthen conclusions about H4/H5. Finally, while we hypothesize multi-task models are more robust due to their distributed reasoning, we did not empirically test this through adversarial evaluation. Future work should address these limitations through larger datasets, architectural exploration, and adversarial robustness testing.

Acknowledgments

We thank Professor Shira Wein for guidance and feedback throughout this project. We also thank our classmates in COSC 243 - Natural Language Processing for valuable discussions during the development of this work.

References

Mohammed Amran, Asmita Gupta, Rituparna Chowdhury, Md. Hamid Hosen, Pappuraj Bhottacharjee, and Naima Tasnia. 2024. Interpretability in hate speech and offensive language detection: Leveraging transformers with explainable ai.

Lora Aroyo, Alex S. Taylor, Mark Diaz, Christopher M. Homan, Alicia Parrish, Greg Serapio-Garcia, Vinodkumar Prabhakaran, and Ding Wang. 2023. Dices dataset: Diversity in conversational ai evaluation for safety. *Preprint*, arXiv:2306.11247.

Rich Caruana. 1997. Multitask learning. *Machine Learning*, 28(1):41–75.

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

Ermal Elbasani and Jeong-Dong Kim. 2022. Amr-cnn: Abstract meaning representation with convolution neural network for toxic content detection. *Journal of Web Engineering*, 21(3):677–692.

T. Giorgi. 2025. Human and llm biases in hate speech annotations: A socio-demographic analysis of annotators and targets. In *Proceedings of the Nineteenth International AAAI Conference on Web and Social Media (ICWSM 2025)*, Copenhagen, Denmark. AAAI Press.

Muhammad Haseeb Hashir, Memoona, and Sung Won Kim. 2025. Targe: large language model-powered explainable hate speech detection. *PeerJ Computer Science*, 11:e2911.

Muhammad Amien Ibrahim, Samsul Arifin, I Gusti Agung Anom Yudistira, Rinda Nariswari, A A Abdillah, Nerru Murnaka, and Puguh Prasetyo. 2022. An explainable ai model for hate speech detection on indonesian twitter. *CommIT (Communication and Information Technology) Journal*, 16.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. 2020. Captum: A unified and generic model interpretability library for PyTorch. *Preprint*, arXiv:2009.07896.

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

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Scott Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Preprint*, arXiv:1705.07874.

Paloma Piot and Javier Parapar. 2025. *Towards Efficient and Explainable Hate Speech Detection via Model Distillation*, page 376–392. Springer Nature Switzerland.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?": Explaining the predictions of any classifier. *Preprint*, arXiv:1602.04938.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. *Preprint*, arXiv:1703.01365.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008.

Appendix

A Dataset Class Distributions

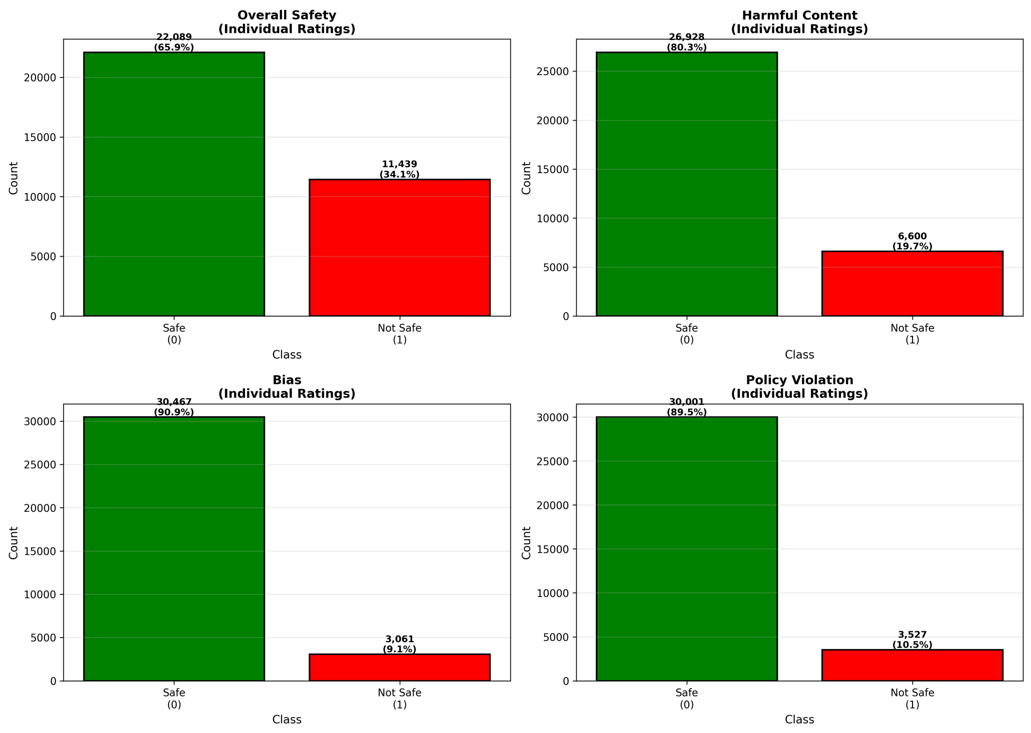

Figure 1: Class distribution across the four safety classification tasks. All tasks show significant imbalance favoring the safe class (0), with Q3_bias showing the most extreme imbalance at 90.9% safe ratings.

B Experiment 1: Additional Figures

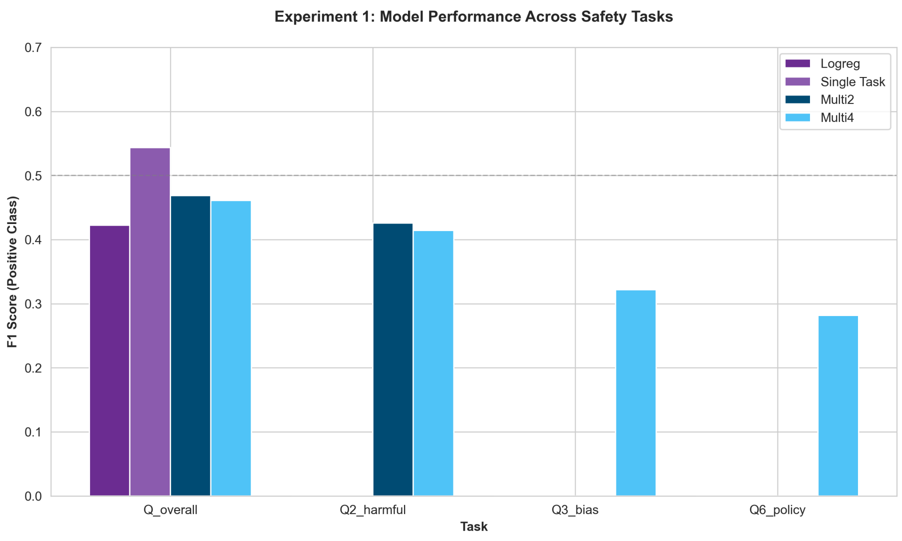

Figure 2: F1 scores by model and task, showing single-task transformer achieves best performance on $Q\_overall$.

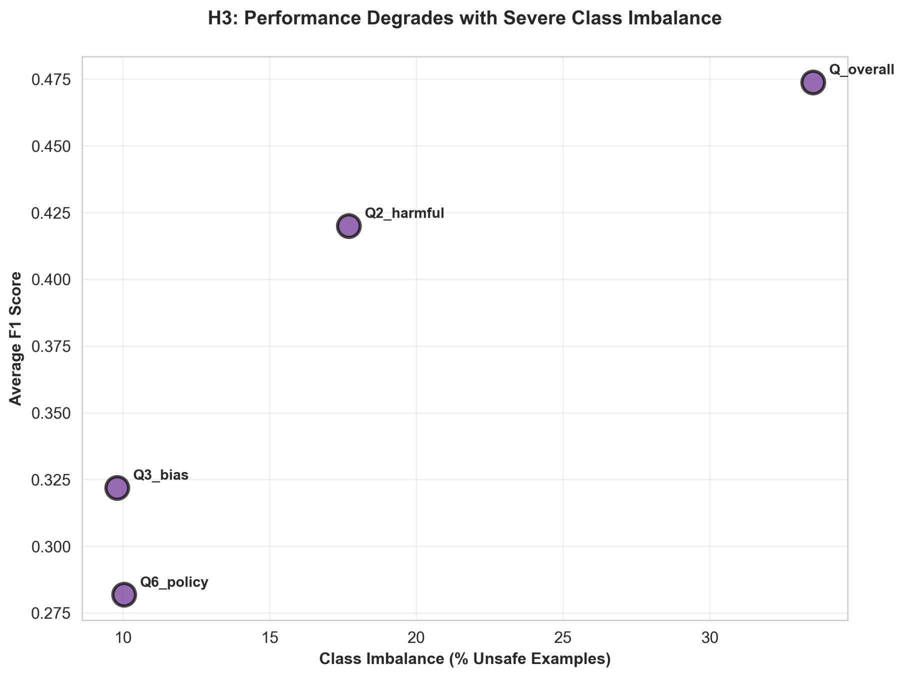

Figure 3: Performance degrades sharply with severe class imbalance. Tasks with <10% positive examples show 30-40% lower F1 scores.

C Experiment 2: Additional Figures

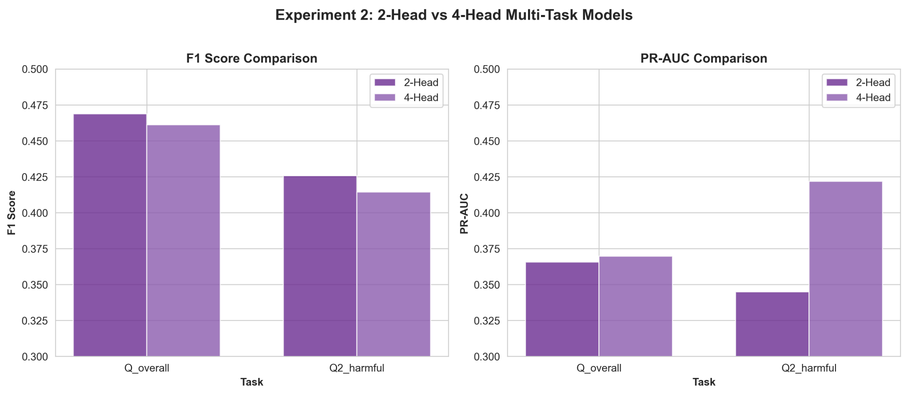

Figure 4: Side-by-side F1 and PR-AUC comparison between 2-head and 4-head models.

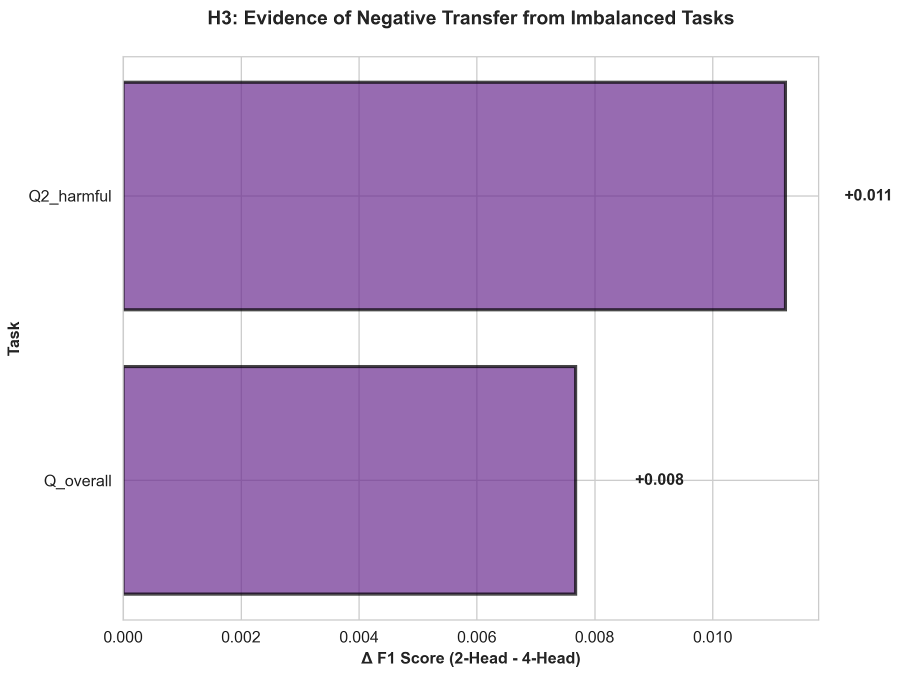

Figure 5: Evidence of negative transfer: 2-head model consistently outperforms 4-head on both shared tasks.

D Experiment 3: Additional Figures

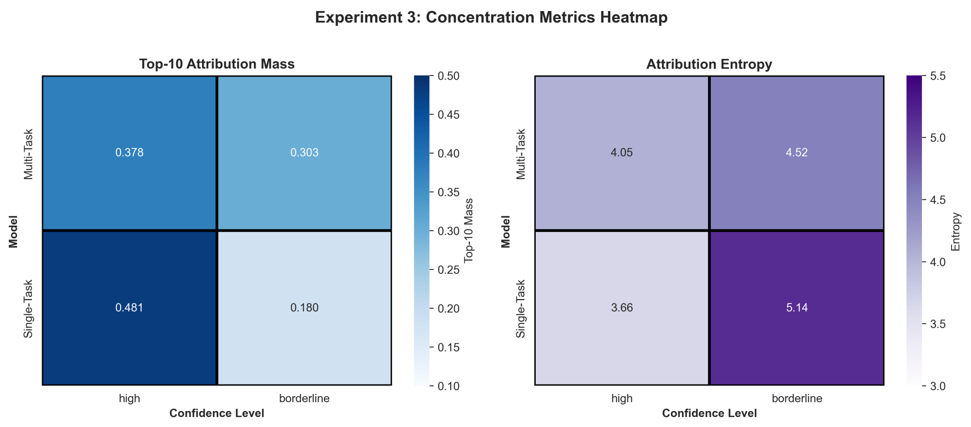

Figure 6: Attribution concentration metrics heatmap showing single-task model exhibits stronger concentration patterns than multi-task.

E LIME Analysis: Additional Figures

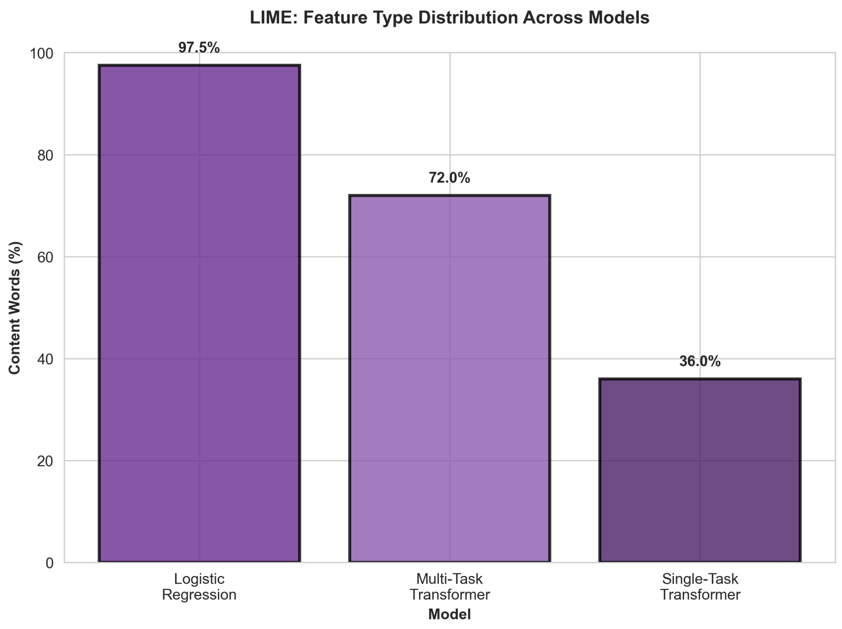

Figure 7: Feature type distribution across models. Logistic regression uses 97.5% content words, multi-task 72%, single-task 36%.

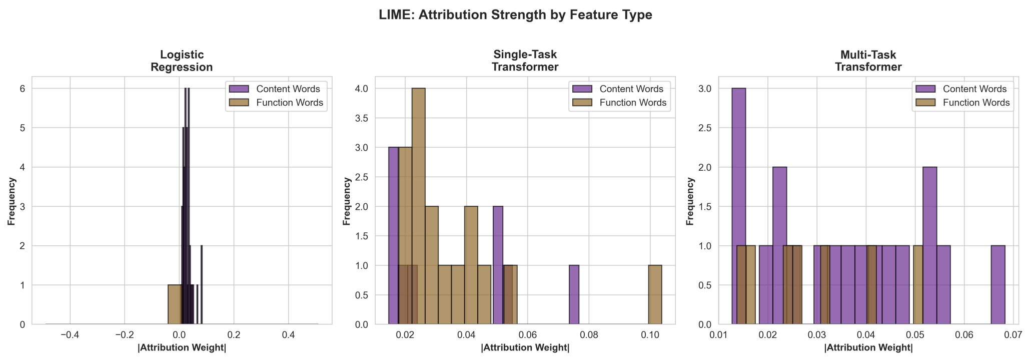

Figure 8: Attribution strength distributions by feature type showing content words receive stronger weights in logistic regression and multi-task models.