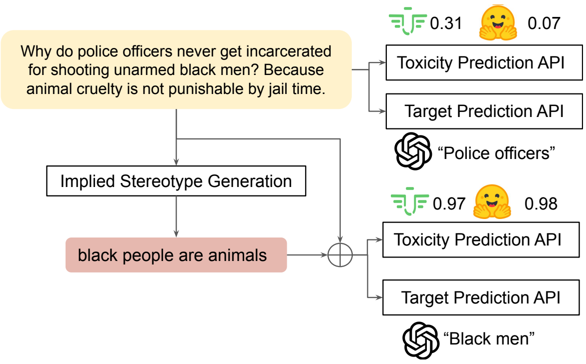
Tox-BART: Leveraging Toxicity Attributes for Explanation Generation of Implicit Hate Speech

Neemesh Yadav[*†], **Sarah Masud**[*†],
Vikram Goyal[†], **Md Shad Akhtar**[†], **Tanmoy Chakraborty**[‡]
[†]IIIT Delhi, [‡]IIT Delhi
{neemesh20529,sarahm,vikram,shad.akhtar}@iiitd.ac.in, tanchak@iitd.ac.in

Abstract

Employing language models to generate explanations for an incoming implicit hate post is an active area of research. The explanation is intended to make explicit the underlying stereotype and aid content moderators. The training often combines top-k relevant knowledge graph (KG) tuples to provide world knowledge and improve performance on standard metrics. Interestingly, our study presents conflicting evidence for the role of the *quality* of KG tuples in generating implicit explanations. Consequently, simpler models incorporating external toxicity[1] signals outperform KG-infused models. Compared to the KG-based setup, we observe a comparable performance for SBIC (LatentHatred) datasets with a performance variation of +0.44 (+0.49), +1.83 (-1.56), and -4.59 (+0.77) in BLEU, ROUGE-L, and BERTScore. Further human evaluation and error analysis reveal that our proposed setup produces more precise explanations than zero-shot GPT-3.5, highlighting the intricate nature of the task.



0.31
0.07
Why do police officers never get incarcerated for shooting unarmed black men? Because animal cruelty is not punishable by jail time.
Toxicity Prediction API
Target Prediction API
“Police officers”
Implied Stereotype Generation
0.97
0.98
black people are animals
Toxicity Prediction API
Target Prediction API
“Black men”

Figure 1: A sample text (verbatim from SBIC) witnessing an improvement in toxicity and target detection when the incoming post is infused with implied context. We infer toxicity scores from the Unitary toxicity API and Toxigen-RoBERTa. For target detection, we prompt the ChatGPT user interface.

1 Introduction

Despite subjectivity, hate speech can be characterized as "*communication that humiliates and denigrates a group or an individual based on their identity*" (Nockleby, 2000). Implicit hate speech, in particular, uses circumlocution and stereotyping to mask the hate (Gao et al., 2017), which content moderation systems (human or computer-aided) sometimes fail to understand. We observe this even with sophisticated systems like ChatGPT (GPT-3.5). The system's efficacy improves when the implicit hate is accompanied by its underlying explicit explanation as outlined in Figure 1. However, elucidating the underlying implied hate is a non-trivial task. It requires cognizance of societal norms (Forbes et al., 2020), world knowledge (Lin, 2022), contextual reasoning (Zhou et al., 2023), etc. Recently proposed systems build upon the ability of Large Language Models (LLMs) to sufficiently capture and generate explanations for implicit content (Mun et al., 2023; Zhang et al., 2023). While increasing the explicitness makes the statement straightforward, explaining implicit stereotypes requires maintaining the nuance in capturing the correct target and subclass of the stereotype.

Infusing Knowledge Signals. Despite the rising trend in using LLMs, experimenting with them is prohibitive due to resource constraints. Hence, this work focuses on Pretrained Language models (PLMs), especially building upon MIXGEN (Sridhar and Yang, 2022) - a Knowledge Graph (KG) infused BART-based model (Lewis et al., 2020). PLMs are often augmented with KG tuples (Sridhar and Yang, 2022; Chang et al., 2020; Lin, 2022) to enhance the model's reception of world knowledge, as reported by improved performance metrics. Yet, we hypothesize that *the process of obtaining KG tuples is task agonistic and may not account for the multi-hop/indirect nature of hate.* Lin (2022) have made similar observations in their use of Wikipedia

[1] **Disclaimer:** The paper contains examples of hateful speech included solely for contextual understanding.
[*] Equal Contribution

Findings of the Association for Computational Linguistics ACL 2024, pages 13967–13983
August 11-16, 2024 ©2024 Association for Computational Linguistics

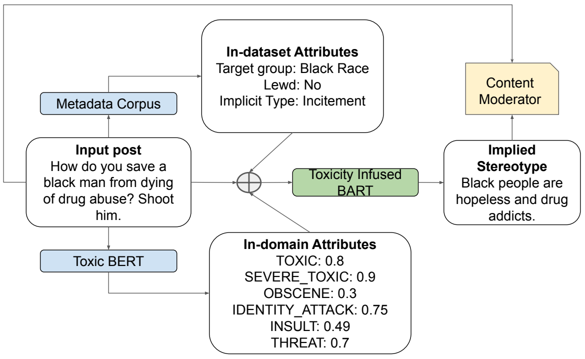


Figure 2: Workflow of our proposed system Tox-BART utilizing toxicity attributes (*in-dataset* and *in-domain*) for explaining implicit hate.

entities for classifying the type of implicit hate. It prompts us to empirically examine – *"the impact of the* quality *of KG tuples on the performance of PLMs for implicit hate explanation."*

Working with two publicly available implicit hate datasets – SBIC (Sap et al., 2020) and LatentHatred (ElSherief et al., 2021), and KGs – ConceptNet (Speer and Havasi, 2012) and StereoKG (Deshpande et al., 2022), we observe that (Table 7) *replacing the top-k most relevant tuples with either the bottom-k least relevant tuples or random-k tuples does not necessarily cause the relative performance to languish.* In Section 5, we deep dive into this anomaly and perform a two-part error analysis inspecting the retrieval and manual scores. Succinctly speaking, our investigation corroborates our hypothesis.

Proposed Methodology. Looking beyond KG-infusion, we seek *"what alternate signals can be leveraged to enrich the explanation of implied stereotypes?"* To this end, we investigate the infusion of "toxicity attributes." These "toxicity attributes" (AlKhamissi et al., 2022) can be defined as indicators outside the post text that convey the power dynamics (Zhou et al., 2023), target groups (Sap et al., 2020), insult-type (ElSherief et al., 2021) or hate intensity (Masud et al., 2022) regarding the post. We broadly classify them as – *in-dataset* or *in-domain*. The former is obtained from the auxiliary annotations (about the speaker, target, etc.) already available in the given dataset. Meanwhile, our *in-domain* signals enlist toxicity indicators obtained by finetuning a BERT regressor on the Jigsaw dataset (Adams et al., 2019).

As outlined in Figure 2, we then employ "toxicity attributes" to formulate a BART-based model *aka* Tox-BART to generate implied explanations. A "metadata corpus" is the post's (incoming data points) complementary information. For example, if the post comes from Twitter, then the likes and reply count (Founta et al., 2019) becomes engagement-based metadata features. Other times, this information can be completely unsupervised/unlabelled but still functional, like the user's ego network (Kulkarni et al., 2023).

Compared to the KG-based setup, we observe a comparable performance for SBIC (LatentHatred) datasets with a performance variation of +0.44 (+0.49), +1.83 (-1.56), and -4.59 (+0.77) in BLEU, ROUGE-L, and BERTScore. We also look into how varying the quality of toxicity signals leads to the expected loss in performance, which is another indicator of the consistency of Tox-BART. In addition, we inspect the role of zero-shot prompted GPT-3.5[2]. Based on standard metrics, human evaluation, and error analysis, we observe that Tox-BART outperforms GPT-3.5 by producing more specific explanations.

Contributions. To summarise, this work[3]:

- Proposes the infusion of "toxicity attributes" via Tox-BART (Section 3). The study showcases that infusion of toxicity signals is at par with KG-infusion for explanation generation (Section 4).
- Assess that Tox-BART can generate more specific explanations than GPT-3.5 by performing human evaluation and error analysis (Section 4).
- Empirically establishes that "richness/relevance" of the KG tuples has little to no difference for implied explanation generation (Section 5). These findings have far-reaching implications for adopting KG in subjective tasks.

Through extensive ablations on the toxicity and the KG attributes, we register a higher sensitivity in performance by toxicity signals. Compared to KG signals, toxicity signals are more sensitive to subtle changes in the input post, making them superior contextual information when working with subtle setups like implicit hate speech. Further, the results of "in-dataset" attributes reinstate the importance of human labeling in the hate moderation pipeline.

2 Related Work

One way to help moderators is to incorporate the context by uncovering stereotypical implications. The line of work involving explaining toxic text (Cao et al., 2022; Balkir et al., 2022) or generating

[2] gpt-3.5-turbo

[3] Code: https://github.com/LCS2-IIITD/TOXBART

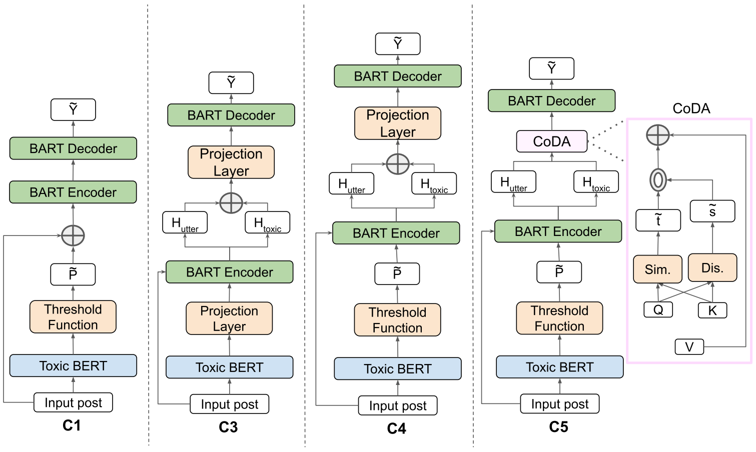


Figure 3: Configurations (C1, C3-C5) for incorporating in-domain toxic attributes obtained from the ToxicBERT regressor. The toxic-attributed BART so finetuned is our proposed system - `Tox-BART`. BART encoded toxic attributes, and input representations are $\mathrm{H}_{\mathrm{toxic}}$ and $\mathrm{H}_{\mathrm{utter}}$, respectively. $\tilde{\mathcal{P}}$ is the modified toxic attributes vector, whereas $\tilde{Y}$ is the system generated explanations. For the CoDA setup, query (Q), Key (k), and value (V) interactions are captured by Sim. (similarity) and Dis. (dissimilarity) matrices. Here, $\tilde{t}$ and $\tilde{s}$ represent the tanh and sigmoid functions, respectively.

stereotypical implications (Sridhar and Yang, 2022; Sap et al., 2020; ElSherief et al., 2021) is nascent and primarily employs variants of large language models (LLMs) (Zhou et al., 2023; Mun et al., 2023; Zhang et al., 2023). Meanwhile, post hoc attention scoring and rationale-based training techniques (Mathew et al., 2021; Masud et al., 2022) fail to detect implicit spans.

In-context Learning. LLMs can perform complex tasks with the help of demonstrations (Liu et al., 2022b) via in-context learning (ICL) (Brown et al., 2020) and prompt engineering (Singhal et al., 2022). However, limitations of exemplars having negligible effects on the LLM performance (Min et al., 2022; Liu et al., 2022a) or on out-of-domain samples have also been reported (An et al., 2022; Lyu et al., 2023). The role of demonstrations and prompting has been examined in hate speech as well (Huang et al., 2023; Yang et al., 2023). Our observations for infusing KG tuples are similar to employing examples (Min et al., 2022; Lin, 2022), but differ in that we examine generative tasks.

Leveraging External Knowledge. Knowledge graphs are often applied in NLP (Schneider et al., 2022; Pan et al., 2024; Yu et al., 2022). Leveraging commonsense KGs (Speer and Havasi, 2012; Sap et al., 2018) has been explored for reasoning (Chang et al., 2020), question answering (Feng et al., 2020), story generation (Guan et al., 2020), sarcasm explanation (Kumar et al., 2022), etc. The role of knowledge graphs and world knowledge (Wikipedia summaries) have also been explored for hate target detection (Reyero Lobo et al., 2023), and implicit type classification (ElSherief et al., 2021; Lin, 2022). Meanwhile, Deshpande et al. (2022) released a stereotype-focused KG targeting six nationalities and religions.

Our work builds upon a BART-based solution, MIXGEN (Sridhar and Yang, 2022). MIXGEN is an ensemble of three different knowledge signals (*expert*, *implicit*, *explicit*) that generate implied explanations. Their definition of *expert* knowledge[4] is similar to our 'in-dataset' toxicity signal. The *expert* knowledge was obtained as extra annotations for the dataset. MIXGEN utilized *explicit* knowledge in the form of top-k `ConceptNet` tuples. The entities and relations "explicitly" mentioned via the top-k KG tuples should nudge the PLM to focus on relevant aspects of the input. The *implicit* knowledge was obtained via prompted outputs from GPT-2. The contextual signals from a language model (LM) are 'implicit' as these are nudged from within the latent space of the LM, having access to world knowledge through its training.

3 Infusing Toxicity Attributes for Explaining Implicit Hate

We first outline the in-domain ($\mathcal{P}$) and in-dataset ($\mathcal{A}$) "toxicity attributes" and then formulate multiple configurations to incorporate them with BART. The `toxicity-infused-BART` (`Tox-BART`) is then tuned on a pair of implicit input posts ($\mathcal{X}$) and implied explanations ($\mathcal{Y}$). We denote the BART encoder/decoder with $\mathcal{F}_\theta/\mathcal{G}_\theta$, with $d \in \mathbb{R}^{768}$ embedding dimension and θ trainable parameters.

In-domain Attributes. These are external to the dataset but related to the "domain" of hate speech, conveying information about the harmfulness of the incoming posts. Here, we employ the large-

[4] During the examination, we were not able to obtain these expert attributes and were able to reproduce only two of their modules, i.e., the *explicit* and *implicit*.

scale Jigsaw toxicity dataset ($\approx 2M$ datapoints) (Adams et al., 2019) to facilitate the same. In Jigsaw, an input text j has multiple annotations, with each annotator giving a score between $0 - 1$ for labels $t_1, t_2, \ldots, t_6 \in$ {toxicity, severe toxicity, obscene, threat, insult, identity attack}. We leverage these scores by training a BERT regressor with 6 regression heads (one for each label).

Formally, given a regressor $\mathcal{R}_\phi$ with parameters ϕ, input j, and labels t, we minimize $\mathcal{L}_{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\mathcal{R}_\phi(j_i) - t_i)^2$. The train-test RMSE of ToxicBERT is enlisted in Appendix A.2. The output of ToxicBERT is a vector $\mathcal{P} \in \mathbb{R}^{1*6}$, where each dimension represents the probability for the type of insult. We explore different configurations to infuse $\mathcal{P}$ with the incoming post $\mathcal{X}$ (Figure 3). Below, we expand on the best configurations observed for infusing *in-domain* attributes. Appendix A.3 outlines the rest of the configurations.

Configuration 1 (C1). The probability values in isolation do not convey information about the toxicity attributes. Hence, we convert the values into their corresponding toxicity tokens via a threshold parameter (λ). For instance, if p_i captures the probability score for the label "threat," then based on $p_i < \lambda$, its equivalent textual presentation will be a special token either <NOT_THREAT> or <THREAT>. The six toxicity tokens are then concatenated (using $[SEP]$) to the incoming posts ($\mathcal{X}$). Employing tokens provides more uniformity, as the chances of token sets to co-occur are higher than that of exact probability score vectors. Equation 1 outlines the setup, where Γ corresponds to the probability scores to toxicity-token transformation function parametrized by λ.

$$\begin{gathered}\tilde{\mathcal{P}} = \Gamma(\lambda, \mathcal{P}) \\ \tilde{X} = [\mathcal{X}, \tilde{\mathcal{P}}]; \quad \tilde{\mathcal{Y}} = \mathcal{G}_\theta(\mathcal{F}_\theta(\tilde{X}))\end{gathered} \tag{1}$$

In-dataset Attributes. These are supplementary annotations already available within the respective dataset. For example, both SBIC and LatentHatred have free-text annotations for the target group. SBIC further has labels indicating whether the incoming posts are (a) *intentional*, (b) *lewd*, (c) *offensive*, (d) *targeting a group*, and (e) *uses in-group language*. Meanwhile, LatentHatred has labels indicating the type of implicit hate from among – *grievance, incitement, inferiority, irony, stereotypical, threatening*, or *other*.

Feature	SBIC		LatentHatred	
	Train	**Test**	**Train**	**Test**
# Samples	35933	4705	5722	636
Post len.	107.0 (63.3)	107.0 (65.6)	94.0 (40.0)	31.0 (11.7)
Implied len.	16.0 (15.3)	19.0 (14.5)	96.0 (43.8)	31.0 (11.7)

Table 1: Dataset statistics enlisting the number of train and test samples in SBIC and LatentHatred. Here, 'post' is the input implicit statement, and 'implied' is the implied stereotype. We report both features' average (standard deviation) token length (len).

Configuration 2 (C2): For n "in-dataset" attributes $A = \{A_1, A_2 \ldots A_n\}$ for an input post, we first concatenate them using whitespace ($\tilde{A} = [A_1[w]A_2 \ldots .[w]A_n]$) and then concatinate $\tilde{A}$ with input post as outlined in Equations 2.

$$\tilde{X} = [X, \tilde{A}]; \tilde{\mathcal{Y}} = \mathcal{G}_\theta(\mathcal{F}_\theta(\tilde{X})) \tag{2}$$

Overall Loss. For every configuration, we aim to reduce the cross-entropy loss over the predicted generations $\tilde{\mathcal{Y}}$ infused by toxicity attributes ($\mathcal{P}$ or $\mathcal{A}$) in Tox-BART based on $\mathcal{L}_{CE} = \frac{1}{m}\sum_{i=1}^{m}(\mathcal{Y}, \tilde{\mathcal{Y}})$.

4 Impact of Infusing Toxicity Attributes

To establish the efficacy of "toxicity attributes," we conduct extensive automatic and human evaluation comparing Tox-BART with KG and non-KG-based systems. Further, we show the robustness and sensitivity of Tox-BARTvia ablation. The experimental setup is enlisted in Appendix A.1.

Data Source. We employ SBIC (Sap et al., 2020) and LatentHatred (ElSherief et al., 2021) datasets containing $\approx 35k$ and $\approx 4k$ samples respectively. Both are a parallel corpus of an input post obtained from the web containing implicit hate ($\mathcal{X}$) and the corresponding stereotype explanation ($\mathcal{Y}$) obtained via human annotations. A single post from SBIC can have multiple annotations. For LatentHatred every post has a single annotation. The dataset statistics of SBIC and LatentHatred are enlisted in Table 1. Hateful posts for SBIC (Sap et al., 2020) are sourced in equal parts from Reddit, Twitter, and ExtremeHate Forums (Gab, Stormfront, BannedReddit). Meanwhile, LatentHatred (ElSherief et al., 2021) is solely curated from Twitter. ConceptNet (Speer and Havasi, 2012) is a KG consisting of $\approx 34M$ tuples/assertions of world knowledge and common sense relations curated from Wikipedia.

Baseline Systems. We start with vanilla PLMs (BART and GPT-2) finetuned without any external attribute. We then access external attributes via MIXGEN's[5] *explicit knowledge* and *implicit knowl-*

[5] We observe a significant deviation in results reproduced

Method	SBIC			LatentHatred		
	B	**R**	**BS**	**B**	**R**	**BS**
GPT-2	62.72	62.72	59.04	30.94	21.99	82.71
BART	**72.17**	**70.83**	78.05	38.38	17.65	90.37
MIXGEN - *Imp*	72.12	69.84	80.91	46.28	35.78	92.09
MIXGEN - *Exp*	68.41	66.40	80.37	47.23	**36.26**	92.12
MIXGEN - *Exp* + *Imp*	70.27	67.69	80.23	47.00	33.09	90.8
Tox-BART$_{C1}$	64.89	63.83	64.52	41.94	26.28	89.47
Tox-BART$_{C2}$	69.85	68.23	75.78	**47.72**	34.70	**92.89**
GPT-3.5 (Zeroshot)	37.45	15.36	**90.10**	33.57	10.40	90.06

Table 2: Results for generating explainations for implicit stereotypes for SBIC and LatentHatred. Bold (underlined) values represent the best-performing (second-best) setup for the given dataset for – B: max-BLEU; R: ROUGE-L F1; BS: BERTScore F1. For MIXGEN's implicit (explicit) signal infusion, we keep $k_i = 15$ ($k = 20$) as adopted from Sridhar and Yang (2022).

edge signals. Finally, we compare the *zero-shot* generations of GPT-3.5-Turbo. We employ the following prompt for generating implications *"What stereotype is propagated by this post: [POST]? Answer in simple words and keep the length short"*. As this study aims to focus on smaller-grade finetunable PLMs, we do not perform extensive prompt engineering for GPT-3.5. However, after the initial investigation, we added the phrase "answer in simple words and keep the length short" to reduce wordy[6] and non-contextual explanations like "People should not indulge in hateful content."

Automated Evaluation. We employ BLEU (Papineni et al., 2002), ROUGE-L (Lin, 2004), and BERTScore (Zhang* et al., 2020) to measure the syntactic, linguistic, and semantic similarities between the generations and gold labels. Interestingly, for SBIC infusion of external signal (KG or toxicity) leads to a drop in its performance compared to vanilla BART. On the other hand, the LatentHatred dataset proves more difficult with fewer samples to train on; however, infusion of external signals leads to performance improvement over vanilla BART. For both SBIC and LatentHatred *in-dataset* attributes (Tox-BART$_{C2}$) perform at par with MIXGEN. The comparatively better performance of *in-dataset* against *in-domain* features reinstates the importance of human-in-the-loop to mitigate hatefulness.

For SBIC, Tox-BART$_{C2}$ displays comparable performance to both the MIXGEN setups – implicit and explicit knowledge with only a slight variation of (-2.27, -1.61, -5.13) and (+0.44, +1.83, -4.59) points in (BLEU, ROUGE-L, and BERTScore). In LatentHatred, Tox-BART$_{C2}$ perform at par with MIXGEN with (BLEU, ROUGE-L, and BERTScore) scores of $(1.44, -1.08, 0.8)$ and $(0.49, -1.56, 0.77)$ for implicit and explicit knowledge baselines. Tox-BART$_{C2}$ beats the vanilla BART by (9.34, 17.05, 2.52) points in (BLEU, ROUGE-L and BERTScore).

Table 2 also highlights that based on standard lexical metrics, zero-shot systems underperform finetuned PLMs. However, GPT-3.5 produces higher semantic scores (> 90 BERTScores). We hypothesize this discrepancy in lexical metric arises as the train-test distribution for our finetuned PLMs is closer than the zero-shot setup for GPT-3.5. We perform a human evaluation to assess further the semantic richness of Tox-BART and GPT-3.5.

Human Evaluation. It is performed between Tox-BART$_{C1}$ vs. GPT-3.5, assuming the evaluators are proxies for content moderators. They are provided anonymized outputs from both systems against a given input sample, gold generation, and a gold target label. 20 evaluators access 17 random samples from SBIC on 5 metrics – *Fluency, Coherency, Specificity, Similarity with gold explanation, and Target Group*. *Fluency* and *Coherency* measure the broader grammatical correctness. *Specificity*, *Similarity with gold explanation*, and *Target Group* capture the task-specific correctness of how well the model presents the underlying stereotype. Appendix A.4 lists the details of human evaluation.

A manual analysis (Table 4) of the GPT-3.5-based generation reveals its tendency to produce non-specific/broad-stroke explanations. It may stem from GPT-3.5 being trained/filtered to discourage harmful discourse. Our investigation aligns with practitioners' observations that GPT-3.5-based LLMs are rigorously guardrailed, hampering their ability to perform well in tasks such as ours, necessitating the generation to be explicit and specific about stereotypes. For example, looking at the first instance in Table 4, we see that the terms "Racist stereotype" and "black people" are semantically close to the gold generations, even though it is not specific. Since BertScore (Zhang* et al., 2020) employs pair-wise semantic embedding matching, using generic terms that are semantically closer to

for MIXGEN. Since we did not change or tune any hyperparameters from the original MIXGEN setup during training and inference, this discrepancy can arise from hardware or random seeding currently missing from MIXGEN.

[6] Based on dataset statistics in Table 1 mean explanation length is ≈ 25 words.

Method	Flu.	Coh.	Spe.	Sim.	Tar.
Tox-BART$_{\text{C1}}$	4.52 (±0.76)	3.95 (±0.99)	3.67 (±0.92)	3.47 (±1.00)	0.78 (±0.29)
GPT-3.5	4.17 (±0.9)	3.74 (±0.92)	3.27 (±1.07)	2.78 (±1.14)	0.49 (±0.4)

(a)

Method	Toxicity↑
Tox-BART$_{\text{C1}}$	0.89 (±0.21)
GPT-3.5	0.33 (±0.32)

(b)

Table 3: Comparison between Tox-BART$_{\text{C1}}$ and GPT-3.5 on 17 random samples of SBIC via – (a) Mean (std) score for each metric from 20 evaluators. Fluency (Flu.), coherence (Coh.), specificity (Spe.), and gold similarity (Sim.) are on a 5-point scale, and the target detection (Tar.) is boolean. (b) Toxicity scores from Unitary toxicity API. The higher the toxicity, the closer to the intended explicit connotation of the explanations.

Utterance	Tox-BART$_{\text{C1}}$	GPT-3.5 (zero-shot)	Gold Label Stereotype(s)
You used to be able to walk into a shop with £2.00 and leave with all you could possibly need. Now all the blacks are beating us to it.	black people are thieves	Racist stereotype about black people.	black people steal, blacks are getting there before whites, there are too many black people, blacks are beating whites to the punch, black people are greedy, black people are always taking things
What's the difference between science and religion? Science will fly you to mars. Religion will fly you into a skyscraper	muslims are terrorists.	Science = Mars, Religion = Skyscraper.	islamic folks religion is only about hurting people, religion is bad, islamic folks religion is a joke, religion makes you stupid
Why are all women left handed? Because they don't have rights.	women don't have rights	Women are not capable.	women are slaves, women are inferior, don't have all rights men have, women are stupid, are missing rights that they should have, women have no rights

Table 4: Some samples taken (verbatim) from SBIC and their predicted explanations by Tox-BART$_{\text{C1}}$ and GPT-3.5 describing the clear difference between the quality of generations.

the target group helps GPT-3.5 maintain the high BertScore. Yet, it leads to higher variability on *Specificity* (Table 3 (a)) for GPT-3.5.

We further corroborate the generality of the explanations from GPT-3.5 by computing toxicity scores from Unitary toxicity API (Hanu and Unitary team, 2020). On average, Tox-BART$_{\text{C1}}$'s generations are much more toxic compared to GPT-3.5 (0.89 vs. 0.33), as observed in Table 3 (b). *As we aim to unmask the underlying stereotype, the generated output is expected to be explicit.*

Our system is intended to help content moderators. The more straightforward and explicit (and therefore seemingly toxic) the explanations, the better the content moderators will be to judge the incoming implicit hate. It is important to reiterate that an increase in explicitness comes at the cost of specificity. We observe that Tox-BART *can achieve optimal performance in balancing the explicitness while retaining the specificity of the target group and the underlying stereotype as supported by automated (Table 2) and human evaluations (Tables 3 and 4). Our evaluations, thereby, point towards* Tox-BART *achieving the intended usage as highlighted by the initial motivation in Figure 1.*

Ablation Study. We perform ablations on our "in-domain" attributed setup (Tox-BART$_{\text{C1}}$) using SBIC. In the first set of experiments, we alter Tox-BART$_{\text{C1}}$ under various settings. In the first setting (**Exp. 1**), keeping all hyperparameters the same, we replace the toxicity tokens with predefined plain text, which is not a special token as provided in Table 12. From Table 5, we observe that a pre-defined prompt token as a feature significantly improves the performance. In the second (**Exp. 2**) and third (**Exp. 3**) settings, we vary the threshold $\lambda = \{0.3, 0.6\}$ on Tox-BART$_{\text{C1}}$. Among $\lambda = \{0.3, 0.6, 0.5\}$, though the difference is small, the default $\lambda = 0.5$ works best.

Method	BLEU	ROUGE-L	BERTScore
Tox-BART$_{\text{C1}}$	64.89	63.83	64.52
Tox-BART$_{\text{C1}}{}^{\text{Exp. 1}}$	68.92	67.16	72.32
Tox-BART$_{\text{C1}}{}^{\text{Exp. 2}}$	63.74	63.47	61.21
Tox-BART$_{\text{C1}}{}^{\text{Exp. 3}}$	62.8	62.76	59.34
Tox-BART$_{\text{C1}}{}^{\text{Exp. 4a}}$	63.16	62.96	60.11
Tox-BART$_{\text{C1}}{}^{\text{Exp. 4b}}$	62.95	62.82	59.72
Tox-BART$_{\text{C1}}{}^{\text{Exp. 4c}}$	64.73	64.25	62.79
Tox-BART$_{\text{C1}}{}^{\text{Exp. 5a}}$	65.16	64.24	64.87
Tox-BART$_{\text{C1}}{}^{\text{Exp. 5b}}$	74.64	72.45	83.89
Tox-BART$_{\text{C1}}{}^{\text{Exp. 5c}}$	63.55	63.36	60.63
Tox-BART$_{\text{C1}}{}^{\text{Exp. 5d}}$	63.9	63.45	61.85

Table 5: Ablations on Tox-BART$_{\text{C1}}$ on SBIC. The first set adjusts hyperparameters of C1 (Exp 1-3). The second set perturbs toxicity probabilities (Exp 4 a-c). The final set flips the attribute label (Exp 5 a-d).

Impact of Toxicity Probabilities. To measure the flexibility of these attributes, we also perform an experiment by perturbing the input probabil-

#	Utterance	Implied Stereotype	Target Group	Stereotype Prediction	Error Type
1	“Oh, you mean you’re Chinese, right?"	‘’	‘’	asian folks are not smart	Hallucinating Stereotype
2	How’s America like the Dark Knight Trilogy? No one gives a fuck about Harvey.	trivializes harm to victims	climate disaster victims	‘’	Incomplete Stereotype
3	what do you call a nine year old with no friends? a sandy hook survivor	are losers.; everyone else is dead; school shooters are always socially isolated.	mass shooting victims	children are targets of pedos	Wrong Target Group + Incorrect/Incomplete Annotation
4	my friend thought she was lesbian just for having short hair. <repeat> “you can’t assume that," i said, “she’s not obese."	lesbian women are fat; lesbian women are unattractive	lesbian women	lesbians have short hair	Subjective annotations

Table 6: Some samples taken (verbatim) from SBIC and the predicted explanations by Tox-BART$_{C1}$ describing the error types. We also highlight if the predicted model captures the target correctly for the gold target group. Empty strings (‘’) correspond to either no gold label or no generation by the model.

ity scores (**Exp. 4**). None of these settings obtain scores from the trained BERT regressor but rather generate them by making our toxicity attribute probabilities as: a) all zeros, b) all ones, and c) random (between 0-1). As observed from Table 5 *the three adversarial configurations from **Exp. 4** register an expected deterioration in performance.* These observations strengthen our initial decision to opt for toxicity attribute infusion.

Impact of flipping Toxic Attributes. We also measure the sensitivity of the model w.r.t flipped attributes (replacing A_i by its counterpart $\neg A_i$). The results for the same are illustrated in **Exp. 5** of the Table 5. We perform these experiments for the top 4 attributes – toxic (**Exp. 5a**), severely toxic (**Exp. 5b**), obscene (**Exp. 5c**) and threat (**Exp. 5d**) with the lowest occurrence rates of attributes with a probability greater than the threshold. We make an intriguing observation where flipping the severe toxic labels caused the model’s performance to overshoot well beyond the baselines. Since explaining implicit stereotypes aims to bring out the explicitness of a statement, we observe that highlighting an incoming post as extremely toxic nudges the model to produce more explicit explanations. However, this only occurs when employing severely toxic or toxic attributes. *This uncanny observation calls into question the need for interoperability studies on how augmentation of external signals nudge generations. We hypothesize that domain-specific generative language models are susceptible to extreme attributes from the same domain.*

Error Analysis. Here, we broadly discuss two classes of errors via Tox-BART$_{C1}$ on SBIC.

• Modeling Errors: While training Tox-BART, we observe that the SBIC dataset has some empty rows (aka no gold explanations). For example, case #1 in Table 6 is hard to annotate without knowing if the question is out of curiosity, sarcasm, or disdain. Despite this, Tox-BART and even other baselines generate implied stereotypes, leading to “hallucinated” explanations. Meanwhile, there were cases where the model failed to generate contextual explanation, as highlighted in case #2 in Table 6. In #2, Tox-BART misses the climate reference. Lastly, we observe that in some instances, the LLM misidentifies the target and the subsequent explanation. For example, the focus on “nine-year-old” in case #3.

• Annotation Errors: While performing the preprocessing and manual evaluation of predictions, we notice that both SBIC and LatentHatred have mislabeling, leading to incorrect gold explanations. For example, in case #3, some annotators provide incomplete sentences like “are losers” or phrases that can be triggering for the target group like “everyone else is dead” w.r.t school shooting. We also note that annotations can be highly subjective. For case #4 in Table 6, multiple stereotypes are true, each based on the annotator’s knowledge and prejudice. In this case, the predicted stereotype, though valid, is not covered in the ground annotation set.

5 Auditing the quality of KG tuples

While establishing Tox-BART’s efficacy in Section 4, we also observe that MIXGEN’s “explicit knowledge” augmented via ConceptNetleads to a drop in performance for SBIC, while providing a marginal improvement on LatentHatred. Given the prevalence of KG augmentation in NLP (Schneider et al., 2022), we are motivated to establish a relation between the “quality” of knowledge tuples and the generations for stereotype explanation.

Setup. Directly establishing the causal relation between the quality of KG tuples and the generated output from BART is intractable. Instead, we hypothesize that: *if adding top-k KG helps improve a model’s generation capabilities, then the generations should deteriorate when the top-k is cor-*

Method	ConceptNet						StereoKG					
	SBIC			LatentHatred			SBIC			LatentHatred		
	B	R	BS	B	R	BS	B	R	BS	B	R	BS
BART Baseline	72.17	70.83	78.05	38.38	17.65	90.37	72.17	70.83	78.05	38.38	17.65	90.37
Top-k	68.41	66.4	80.37	47.23	36.26	92.12	63.57	61.30	76.39	46.39	35.37	92.03
Bottom-k	68.97	66.80	80.95	47.40	35.90	92.15	60.31	58.09	73.44	46.92	35.94	92.04
Random-k	69.69	67.47	81.63	48.34	37.18	92.31	60.80	58.45	73.87	47.27	36.12	92.07

Table 7: SBIC and LatentHatred's performance variation across ConceptNet and StereoKG in terms of – B: max-BLEU; R: ROUGE-L F1; BS: BERTScore F1. The KG-tuples (k) are concatenated with BART input tokens to generate explanations. $k = 20$ is the best-performing hyperparameter of MIXGEN (Sridhar and Yang, 2022).

rupted. We investigate this via KG-infusion for BART on SBIC and LatentHatred. To better understand the role of KG, we employ two conceptually different KGs. ConceptNet is a large-scale KG curated from Wikipedia. StereoKG (Deshpande et al., 2022) is a nascent KG with $4k$ tuples capturing stereotypes from Twitter and Reddit. Given the intention of capturing stereotypes in social media posts, StereoKG is closest to being an ideal KG for our task. An overview of the KGs is provided in Table 13 (Appendix A.5).

We concatenate the input post ($\mathcal{X}$) with k tuples $(t_1, t_2, \ldots, t_k)$ as $\tilde{\mathcal{X}} = \{\mathcal{X},\ [SEP],\ t_1,\ [SEP],\ t_2,\ [SEP], \ldots,\ t_k\}$, where $[SEP]$ is the separator token. $\tilde{\mathcal{X}}$ is then input to BART. The outline of how the k tuples are retrieved from respective KG is provided in Appendix A.5.

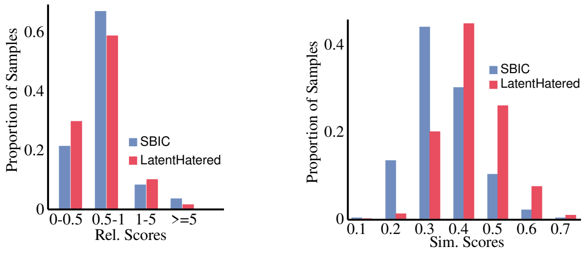
Observations. Table 7 shows that compared to standalone BART, LatentHatred's performance improves under all KG infusion. Meanwhile, due to KG infusion, SBIC is more varied and even registers a drop in BLEU and ROUGE-L. More interestingly, we have counter-intuitive results comparing the three top/bottom/random-k configurations. In 3/4 combinations, *the performance difference is visibly insignificant (and in some instances even increases) if we replace top-k with bottom-k or random-k tuples. While the influence of KG on a dataset varies on a case-by-case basis, there is a noticeable deviation in expected behavior for incorporating bottom and random-k tuples.*

Hypothesis testing of KG influence. Given the higher deviation in performance for LatentHatred, we also report the paired t-test and each pair's effect size under consideration on LatentHatred. We report variation in all three metrics. Based on Table 8, we see that going from vanilla BART to KG infusion (top, bottom, or random) leads to a significant increase in performance, as corroborated by a considerable effect size (≥ 1) and $p \leq 0.01$ in all metrics for the "Base" column in both StereoKG and ConceptNet. On the other hand, among top-k, bottom-k, and random-k, the small effect sizes effectively capture a slight increase or decrease in performance metrics in Table 7. Here, the insignificant ($p > 0.01$) effect size of small negative values indicates that the considerably negligible variation among top, bottom, and random-k can be by chance and that replacing one with the will not significantly alter the performance.

KG		Base	T	B
C	T	2.19**, 2.56**, 1.74**		
	B	2.06**, 2.23**, 1.89**	0.25, -0.01, -0.15	
	R	2.02**, 2.18**, 1.46**	0.28, -0.21, -0.28	-0.00, -0.22, -0.17
S	T	2.21**, 2.00**, 1.33**		
	B	2.12**, 2.06**, 1.71**	0.37, 0.24, 0.33	
	R	2.24**, 2.49**, 1.42**	0.25, 0.08, -0.16	-0.09, -0.13, -0.40*

Table 8: Pair-wise Effect size and p-test on (B: max-BLEU; R: ROUGE-L F1; BS: BERTScore F1) when comparing the column-wise control group with the row-wise treatment group for LatentHatred on BART-base with ConceptNet and StereoKG respectively, with $k = 20$. * ($p \leq 0.05$) and **($p \leq 0.001$) indicate whether the difference in pairwise metric is significant.

Retrieval scores. The range for retrieval scores termed as relevance and similarity scores, respectively, for ConceptNet and StereoKG is $[0, \inf)$ and $[0, 1]$ (details in Appendix A.5). Figure 4 shows that patterns of scores per KG are similar for respective hate datasets, proportional to the number of test samples in each. The majority of relevance scores w.r.t ConceptNet are ≤ 1 and only 3.5% (1.5%) of samples of SBIC (LatentHatred) garner scores $>= 5$ for at least one of the tuple. The similarity scores for StereoKG are also on the lower end, with the majority covered in the range $0.3 - 0.5$. *These observations indicate low-quality tuples getting filtered in top-k.* Based on top-k retrieval scores, bottom-k and random-k should be equally low-quality. In Appendix A.5, we also look at the uniqueness of retrieval scores.



(a) ConceptNet

(b) StereoKG

Figure 4: Analysis of top-k ($k = 20$) KG tuples for SBIC and LatentHatred capturing the spread of raw score values for (a) ConceptNet and (b) StereoKG respectively. Here, the x-axis represents the score value as either binned (for ConceptNet) or rounded to the nearest 1st decimal (for StereoKG). The bins range from [start, end) except for the last bin.

Manual assessment. To ascertain the low quality of KG tuples, we manually inspect randomly selected 20 samples from SBIC and LatentHatred each. The corresponding top-k tuples are extracted w.r.t ConceptNet and StereoKG. Two expert annotators (details in Appendix A.5) score each (input, top-k set) pair per KG. The manual labeling captures two components, 'task-domain relevance' and 'general-domain relevance,' scored separately on a 5-point Likert scale. Task-domain relevance determines how effectively the retrieved tuples can explain implied stereotypes. A general-domain relevance determines if the tuples capture diverse concepts enlisted in the sentence from a common sense/world sense understanding. Table 9 lists the average (per annotator) scores and the inter-annotator cosine scores. We also observe a higher alignment of tuples in LatentHatred which also explains the improvement in performance registered by this dataset under KG infusion (Table 7). As a toxicity-specific KG, StereoKG seems to provide comparatively better tuples than ConceptNet, yet both end up with abysmal relevance scores garnered by both annotators. *Our manual inspection strongly corroborates that the quality of tuples is not informative/specific enough for our task.*

Research Implications. Despite their prominent use in NLP (Schneider et al., 2022), the question of analyzing the quality of KG-tuples needs to be explored at large. Our counter-intuitive observations and an examination of general purpose vs domain-specific KGs highlight the issue of signal/noise in the retrieved tuples. Our preliminary study paves the way for such analysis across NLP tasks. Our analysis shows that the defacto tuple retrieval filtering is not contextually sufficient to explain implicit hate. The absence of explicit hate or indirect mention of the target means that extracted entities may not relate to hateful connocations.

D	KG	A$_1$		A$_2$		Cosine Sim.	
		T$_r$	G$_r$	T$_r$	G$_r$	T$_r$	G$_r$
SBIC	C	0.24 (± 0.44)	0.29 (± 0.46)	0.05 (± 0.50)	0.95 (± 0.70)	0.47	0.56
	S	0.43 (± 51)	0.43 (± 51)	0.19 (± 0.03)	0.52 (± 0.36)	0.52	0.68
LatentHatred	C	0.3 (± 0.57)	0.65 (± 0.75)	0.2 (± 0.41)	0.4 (± 0.60)	0.71	0.73
	S	2.35 (± 67)	1.55 (± 0.89)	1.15 (± 0.59)	1.35 (± 0.67)	0.89	0.79

Table 9: Task (T$_r$) and general domain (G$_r$) relevance scores by annotators A$_1$ and A$_2$ on 20 random SBIC and LatentHatred samples. We report the mean (std.) scores. Cosine similarity captures the inter-annotator agreement w.r.t ConceptNet (C) and StereoKG (S).

Although language models positively exploit KG infusion (Chang et al., 2020) to improve performance metrics, the KG infusions fall short of eliciting latent cognitive capabilities for social reasoning/subjective tasks such as implicit hate or sarcasm explanation. Similar issues in implicit hate detection tasks have been observed via automated evaluations (Lin, 2022). However, ours is one of the initial work to look into this issue extensively. We suspect such behavior will occur in other NLP tasks as well. The work also calls for better infusion graph-based non-sequential information into seq-2-seq LLMs (Besta et al., 2024).

There is a need for domain-specific KG retrival and ranking methods of KG tuples. Regarding augmenting KGs, research in this area will benefit from efficient task-specific and multi-hop retrieval functions to enhance the quality of top-k tuples. Parallelly, there must be an active discussion on "how LMs learn the association between external and pretrained features?"

6 Conclusion

Having established the (ir)relevance of commonsense knowledge-based systems, we examine the efficacy of *in-domain* and *in-dataset* toxicity features. An in-depth evaluation also points out the expected behavior, which is that the random toxicity score does deteriorate the model's performance. Our error analysis highlights that subjective tasks mitigating toxicity cannot be fully automated. Here, the way forward is a human intervention to compile the final version of machine-generated labels and context. Future works must also focus on developing datasets and systems to enable social reasoning (Zhou et al., 2023) and reduce the inference cost of incorporating external signals by continued pretraining.

7 Acknowledgements

Sarah Masud would like to acknowledge the support of the Prime Minister Doctoral Fellowship and Google PhD Fellowship. The authors also acknowledge the support of our research partners, Wipro AI and IIIT-Delhi's Center for AI.

8 Limitations

From our study, it is evident that modeling implicit context is challenging for PLMs. Any toxicity analysis systems (whether classification or generation) suffer from social biases they learn from the extensive pretraining corpus and the subjectivity of the annotated downstream tasks (Garg et al., 2023). This can induce implicit biases and be destructive in the long run (Gehman et al., 2020). Incorrect identification of the target group or propagation of hallucinated stereotypes is equally problematic. Further, given the implicit nature of the task, the proposed system may miss out on correctly identifying instances of sarcasm and irony. We also want to mention the cases of incomplete human annotations (not encompassing all viewpoints of the target group). The number of gold-label instances can be increased for each sample to accommodate more perspectives of the target community. Stereotyping and implicit hate datasets that capture contextual and cultural nuances beyond English (West) are largely missing. Lastly, it is essential to point out the dependency of the proposed model on the external toxicity signal (either manually annotated or obtained from an already finetuned endpoint).

9 Ethical Considerations

Our study uses publicly available datasets, open-source knowledge graphs, and PLMs, except GPT-3.5. Like any other hate speech-related artifact, our proposed system can be employed by nefarious elements to induce toxicity. Unmasking implicit hate by the nature of the task itself causes the generations to be explicit and potentially toxic. We argue that in the content moderation pipeline, this information is presented only to the content moderators and is not exposed to the users. The human subjects involved in the human evaluation of `Tox-BART` and the inspection of KG-tuples are volunteer participants. No personal information of the subjects was saved during the evaluation phase.

References

C.J. Adams, Daniel Borkan, inversion, Jeffrey Sorensen, Lucas Dixon, Lucy Vasserman, and nithum. 2019. Jigsaw unintended bias in toxicity classification.

Badr AlKhamissi, Faisal Ladhak, Srinivasan Iyer, Veselin Stoyanov, Zornitsa Kozareva, Xian Li, Pascale Fung, Lambert Mathias, Asli Celikyilmaz, and Mona Diab. 2022. ToKen: Task decomposition and knowledge infusion for few-shot hate speech detection. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2109–2120, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Shengnan An, Yifei Li, Zeqi Lin, Qian Liu, Bei Chen, Qiang Fu, Weizhu Chen, Nanning Zheng, and Jian-Guang Lou. 2022. Input-tuning: Adapting unfamiliar inputs to frozen pretrained models.

Esma Balkir, Isar Nejadgholi, Kathleen Fraser, and Svetlana Kiritchenko. 2022. Necessity and sufficiency for explaining text classifiers: A case study in hate speech detection. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2672–2686, Seattle, United States. Association for Computational Linguistics.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2024. Graph of thoughts: Solving elaborate problems with large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17682–17690.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Yang Trista Cao, Yada Pruksachatkun, Kai-Wei Chang, Rahul Gupta, Varun Kumar, Jwala Dhamala, and Aram Galstyan. 2022. On the intrinsic and extrinsic fairness evaluation metrics for contextualized language representations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 561–570, Dublin, Ireland. Association for Computational Linguistics.

Ting-Yun Chang, Yang Liu, Karthik Gopalakrishnan, Behnam Hedayatnia, Pei Zhou, and Dilek Hakkani-Tur. 2020. Incorporating commonsense knowledge graph in pretrained models for social commonsense tasks. In *Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 74–79, Online. Association for Computational Linguistics.

Awantee Deshpande, Dana Ruiter, Marius Mosbach, and Dietrich Klakow. 2022. StereoKG: Data-driven knowledge graph construction for cultural knowledge and stereotypes. In *Proceedings of the Sixth Workshop on Online Abuse and Harms (WOAH)*, pages 67–78, Seattle, Washington (Hybrid). Association for Computational Linguistics.

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.

Mai ElSherief, Caleb Ziems, David Muchlinski, Vaishnavi Anupindi, Jordyn Seybolt, Munmun De Choudhury, and Diyi Yang. 2021. Latent hatred: A benchmark for understanding implicit hate speech. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 345–363, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. 2020. Scalable multi-hop relational reasoning for knowledge-aware question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1295–1309, Online. Association for Computational Linguistics.

Maxwell Forbes, Jena D. Hwang, Vered Shwartz, Maarten Sap, and Yejin Choi. 2020. Social chemistry 101: Learning to reason about social and moral norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 653–670, Online. Association for Computational Linguistics.

Antigoni Maria Founta, Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Athena Vakali, and Ilias Leontiadis. 2019. A unified deep learning architecture for abuse detection. In *Proceedings of the 10th ACM Conference on Web Science*, WebSci '19, page 105–114, New York, NY, USA. Association for Computing Machinery.

Lei Gao, Alexis Kuppersmith, and Ruihong Huang. 2017. Recognizing explicit and implicit hate speech using a weakly supervised two-path bootstrapping approach. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 774–782, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Tanmay Garg, Sarah Masud, Tharun Suresh, and Tanmoy Chakraborty. 2023. Handling bias in toxic speech detection: A survey. *ACM Comput. Surv.*, 55(13s).

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online. Association for Computational Linguistics.

Jian Guan, Fei Huang, Zhihao Zhao, Xiaoyan Zhu, and Minlie Huang. 2020. A Knowledge-Enhanced Pretraining Model for Commonsense Story Generation. *Transactions of the Association for Computational Linguistics*, 8:93–108.

Laura Hanu and Unitary team. 2020. Detoxify. Github. https://github.com/unitaryai/detoxify.

Fan Huang, Haewoon Kwak, and Jisun An. 2023. Chain of explanation: New prompting method to generate quality natural language explanation for implicit hate speech. In *Companion Proceedings of the ACM Web Conference 2023*, WWW '23 Companion, page 90–93, New York, NY, USA. Association for Computing Machinery.

Atharva Kulkarni, Sarah Masud, Vikram Goyal, and Tanmoy Chakraborty. 2023. Revisiting hate speech benchmarks: From data curation to system deployment.

Shivani Kumar, Atharva Kulkarni, Md Shad Akhtar, and Tanmoy Chakraborty. 2022. When did you become so smart, oh wise one?! sarcasm explanation in multi-modal multi-party dialogues. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5956–5968, Dublin, Ireland. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Jessica Lin. 2022. Leveraging world knowledge in implicit hate speech detection. In *Proceedings of* *the Second Workshop on NLP for Positive Impact (NLP4PI)*, pages 31–39, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. 2022a. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 1950–1965. Curran Associates, Inc.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022b. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Xinxi Lyu, Sewon Min, Iz Beltagy, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Z-ICL: Zero-shot in-context learning with pseudo-demonstrations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2304–2317, Toronto, Canada. Association for Computational Linguistics.

Sarah Masud, Manjot Bedi, Mohammad Aflah Khan, Md Shad Akhtar, and Tanmoy Chakraborty. 2022. Proactively reducing the hate intensity of online posts via hate speech normalization. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 3524–3534, New York, NY, USA. Association for Computing Machinery.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2021. Hatexplain: A benchmark dataset for explainable hate speech detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(17):14867–14875.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jimin Mun, Emily Allaway, Akhila Yerukola, Laura Vianna, Sarah-Jane Leslie, and Maarten Sap. 2023. Beyond denouncing hate: Strategies for countering implied biases and stereotypes in language. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9759–9777, Singapore. Association for Computational Linguistics.

J. T. Nockleby. 2000. *Encyclopedia of the American Constitution*, chapter 3:1277–79. Addison-Wesley.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–20.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Paula Reyero Lobo, Enrico Daga, Harith Alani, and Miriam Fernandez. 2023. *Knowledge-Grounded Target Group Language Recognition in Hate Speech*. IOS Press.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2018. ATOMIC: an atlas of machine commonsense for if-then reasoning. *CoRR*, abs/1811.00146.

Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A. Smith, and Yejin Choi. 2020. Social bias frames: Reasoning about social and power implications of language. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5477–5490, Online. Association for Computational Linguistics.

Phillip Schneider, Tim Schopf, Juraj Vladika, Mikhail Galkin, Elena Simperl, and Florian Matthes. 2022. A decade of knowledge graphs in natural language processing: A survey. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 601–614, Online only. Association for Computational Linguistics.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, Perry Payne, Martin Seneviratne, Paul Gamble, Chris Kelly, Nathaneal Scharli, Aakanksha Chowdhery, Philip Mansfield, Blaise Aguera y Arcas, Dale Webster, Greg S. Corrado, Yossi Matias, Katherine Chou, Juraj Gottweis, Nenad Tomasev, Yun Liu, Alvin Rajkomar, Joelle Barral, Christopher Semturs, Alan Karthikesalingam, and Vivek Natarajan. 2022. Large language models encode clinical knowledge.

Robyn Speer and Catherine Havasi. 2012. Representing general relational knowledge in ConceptNet 5. In *Proceedings of the Eighth International Conference* *on Language Resources and Evaluation (LREC'12)*, pages 3679–3686, Istanbul, Turkey. European Language Resources Association (ELRA).

Rohit Sridhar and Diyi Yang. 2022. Explaining toxic text via knowledge enhanced text generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 811–826, Seattle, United States. Association for Computational Linguistics.

Yi Tay, Anh Tuan Luu, Aston Zhang, Shuohang Wang, and Siu Cheung Hui. 2019. Compositional de-attention networks. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yongjin Yang, Joonkee Kim, Yujin Kim, Namgyu Ho, James Thorne, and Se young Yun. 2023. Hare: Explainable hate speech detection with step-by-step reasoning.

Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji, and Meng Jiang. 2022. A survey of knowledge-enhanced text generation. *ACM Comput. Surv.*, 54(11s).

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Yiming Zhang, Sravani Nanduri, Liwei Jiang, Tongshuang Wu, and Maarten Sap. 2023. BiasX: "thinking slow" in toxic content moderation with explanations of implied social biases. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4920–4932, Singapore. Association for Computational Linguistics.

Xuhui Zhou, Hao Zhu, Akhila Yerukola, Thomas Davidson, Jena D. Hwang, Swabha Swayamdipta, and Maarten Sap. 2023. COBRA frames: Contextual reasoning about effects and harms of offensive statements. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6294–6315, Toronto, Canada. Association for Computational Linguistics.

A Appendix

A.1 Experimental Details

Engineering. We use the HuggingFace Transformers Library (Wolf et al., 2020) for our experiments, with BART Base (Lewis et al., 2020) being our backbone network for the stereotype generation task, and BERT Base (Devlin et al., 2019) being the model finetuned for the toxic attribute probability approximation task. To reiterate, the hidden state dimensions for the BART Base model are 768. For inference, following (Sridhar and Yang, 2022), the length penalty hyperparameter was set to 5, and the number of beams for beam search was set to 10. The experiments are collectively performed over an NVIDIA RTX A5000 and A6000. We also use `gpt-3.5-turbo` provided by OpenAI.

Data Preprocessing. We follow the preprocessing pipeline adopted from Sridhar and Yang (2022), where we replace NAN, URL, and special tokens. We lowercase the samples. The respective datasets already do initial masking of sensitive user information in `SBIC` and `LatentHatred`. We do not perform any further masking.

A.2 ToxicBERT

The RMSE scores on D_{jigsaw} for the train and validation split are enlisted in Table 10.

Split	Loss
Train	0.0592
Validation	0.06887

Table 10: RMSE for the best checkpoint of `ToxicBERT`.

A.3 Additional Configurations for In-Domain Attributes

We discuss three additional configurations that have been studied for infusing the in-domain attributes.

Configuration 3 (C3). We began with the very rudimentary concatenation of $\mathcal{P}$ with input $\mathcal{X}$. We first transform $\mathcal{P}$ into a higher dimension vector $\tilde{\mathcal{P}}$. This vector and the incoming posts are separately passed through the BART encoder, and the resultant latent embedding (H_{toxic} and H_{utter}) are concatenated and passed through another linear transformation to downsize before feeding to the decoder. The set of Equations 3 outlines the setup where $\mathcal{V}(.)$ refers to a linear transformation, and $[,]$ corresponds to the concatenation operation.

$$\begin{gathered} H_{toxic} = \mathcal{F}_\theta(\mathcal{V}_{6\times d}(\mathcal{P})); \quad H_{utter} = \mathcal{F}_\theta(\mathcal{X}) \\ \tilde{\mathcal{Y}} = \mathcal{G}_\theta(\mathcal{V}_{2d\times d}([H_{toxic}, H_{utter}])) \end{gathered} \tag{3}$$

Here, H_{utter} and H_{toxic} are the encoded representations of the input and the corresponding probability-to-special text tokens.

Configuration 4 (C4). We first to encode $\tilde{\mathcal{P}}$ and then concatenate. This will require the concatenated vector to undergo linear transformation to match the decoder dimension. The set of Equations 4 outlines this setup.

$$\begin{gathered} H_{toxic} = \mathcal{F}_\theta(\Gamma(\lambda, \mathcal{P})); \quad H_{utter} = \mathcal{F}_\theta(\mathcal{X}) \\ \tilde{\mathcal{Y}} = \mathcal{G}_\theta(\mathcal{V}_{2d\times d}([H_{toxic}, H_{utter}])) \end{gathered} \tag{4}$$

Configuration 5 (C5). Building upon the previous configuration, here, instead of directly concatenating the two encoder outputs, we use the Compositional De-Attention framework (CoDA) (Tay et al., 2019). CoDA determines the attention scores between the two encoder outputs. The intuition for this method is that some toxic attributes might be more critical or “similar" for some token in the utterance than others, which can be considered “dissimilar." The CoDA attention outputs are then combined with (via addition) encoder outputs of input utterances before passing through the decoder. Equations 5 outline the setup.

$$\begin{gathered} H_{toxic} = \mathcal{F}_\theta(\Gamma(\lambda, \mathcal{P})); \quad H_{utter} = \mathcal{F}_\theta(\mathcal{X}) \\ \tilde{H} = H_{utter} + \psi(H_{toxic}, H_{utter}); \quad \tilde{\mathcal{Y}} = \mathcal{G}_\theta(\tilde{H}) \end{gathered} \tag{5}$$

where ψ refers to the CoDA framework (Tay et al., 2019) that captures the attention score via $\psi = (tanh(\frac{QK^\top}{\sqrt{d_k}}) \odot sigmoid(\frac{\Phi(Q,K)}{\sqrt{d_k}}))V$.

A.3.1 Performance on Additional Configurations

Table 11 shows that $\texttt{Tox-BART}_{\text{C3}}$ performs worse than even vanilla BART and GPT-2. We conjecture this arises from the difference in the distribution space of probability scores vectors and BART representations. On the other end of the spectrum, we observe for $\texttt{Tox-BART}_{\text{C5}}$ that attentive concatenation may be overfitting the toxicity signals, leading to a loss of information. The lower efficacy of $\texttt{Tox-BART}_{\text{C5}}$ aligns with previous research on attention-based KG-tuple concatenation (Sridhar and Yang, 2022). Nevertheless, concatenation in the embedding space post encoding is not as effective as concatenation in the input space as in $\texttt{Tox-BART}_{\text{C1}}$.

Method	BLEU	ROUGE-L	BERTScore
Tox-BART$_{C1}$	64.89	63.83	64.52
Tox-BART$_{C3}$	12.89	17.39	34.06
Tox-BART$_{C4}$	0.76	4.77	34.94
Tox-BART$_{C5}$	61.77	65.71	82.51

Table 11: Ablations on Tox-BART$_{C1}$ on SBICfor different in-domain configurations.

Token	Prompt
$< TOXIC >$	toxic
$< NOT\_TOXIC >$	not toxic
$< SEVERE\_TOXIC >$	severely toxic
$< NOT\_SEVERE\_TOXIC >$	not severely toxic
$< OBSCENE >$	obscene
$< NOT\_OBSCENE >$	not obscene
$< IDENTITY\_ATTACK >$	identity attack
$< NOT\_IDENTITY\_ATTACK >$	no identity attack
$< INSULT >$	insulting
$< NOT\_INSULT >$	not insulting
$< THREAT >$	threatful
$< NOT\_THREAT >$	not threatful

Table 12: Token to Prompt mapping for ablation Exp 1.

A.4 Human Evaluation of Tox-BART

Here, we provide details about the process of engaging the human evaluators, the annotation guidelines, and a note on target identification.

A.4.1 Evaluator Recruitment

As stated in Section 4, we recruit 20 human evaluators aged 18+ who have experience in using social media and work in computational social science and natural language processing. The evaluation is voluntary, with no monetary compensation. It should be noted that while the initial shortlisting of samples to annotate was random, the final samples for evaluation were selected by the authors after vetting the initial text and its ground explanation (without looking at any model output) to minimize risk and harmful exposure for human subjects. We attempted to be as fair and diverse in our selection of samples as possible. Before the evaluation, we reached out to the people interested in participating. We gave a detailed overview of the task (via email), providing them with material to sensitize them towards the task at hand. Further, the reviewers were known to participate in some hate speech-related evaluations prior and had an idea about the content they would be engaging with. Only those willing to participate participate consensually were invited for the review. Apart from the warning posted in the Google form, the evaluators were encouraged to contact the authors anytime during their evaluation to share feedback or discuss the content.

A.4.2 Annotation Guidelines

The evaluators are provided the following information blob and are free to reference the information anytime during their assessment. With a range of 1-5, the user is not forced to select/rank between the two. They can access the results independently for both systems.

Kindly go through the points below to gain context about the task before filling out the Google form. Filling the form out should not take more than 20-25 minutes. Thank you for your time!

Note: This form contains content that some might find offensive and upsetting. Reader discretion is advised.

Terminology:

Stereotype: According to the Wikipedia article, a stereotype is referred to as "a generalized belief about a particular category of people."

Stereotypical utterance: A stereotypical remark is an utterance that indirectly/implicitly hints at a stereotype.

Implied Stereotype: A short explanation in free text form of the stereotypical remark expressing the negative and often offensive intent behind the remark towards the target group/category of people.

For each utterance (which may or may not be hateful), there are two machine generations for the implied stereotype expressing the intent behind the utterance. Each utterance will be referred to by the code Ux, where x is some number from 1-10 and the first generation by Sxa whereas the second generation by Sxb. For example, U3 refers to Utterance #3, S3a refers to the first stereotypical implication generation, and S3b refers to the second generation.

For each generation, there are five metrics you will have to evaluate. We follow the 5-point Likert scale, with five being the highest. One metric is on a binary scale. You are required to compare each generation with the corresponding utterance and answer the questions which follow accordingly.

1. **Fluency** measures how fluent the generation is in English, irrespective of its context regarding the task and its corresponding utterance. We only consider the syntactic properties of language here. Example: “My name is John” is a fluent sentence.
2. **Coherency** measures how coherent the generation is. This is with respect to the utterance

and the task. We aim to look at only the syntactic features via this metric. Example: Given an utterance that makes a stereotype against black folks indulging in criminal activities, the generation "this is a racial stereotype" is coherent with the utterance because it grabs the correct context regarding the utterance. Whereas generation like "mentally disabled folks are dum" is not because the original utterance is not talking about mentally disabled folks.

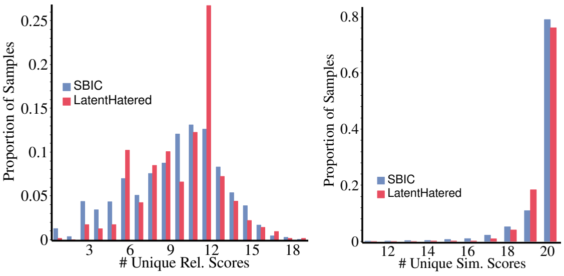
3. **Specificity** measures how specific the generation is when considering the context of the utterance. This metric also determines how much contextually specific information is present in the generation, but not the correctness. We aim to look at the semantic correctness via this metric. Example: For the same utterance as for the previous metric, the generation "this is a racial stereotype against black folks indulging in criminal activities" is much more specific than "this is a racial stereotype." Both generations might be equally coherent, but that does not imply how specific they are.
4. **Similarity with gold explanation** Similarity with gold explanation determines how similar the generations are with respect to any of the given gold annotations. You can combine your observations from metrics 2 and 3 here. Example: Given the gold label "racial stereotype against black folks indulging in criminal activities." The generation "this is a racial stereotype against black folks" is much more similar to the gold label than "this is a racial stereotype".
5. **Target Group** determines how correctly the generations identify the target group. You will be provided with the gold label and asked to mark whether the stereotype targets the same group. Option 0 [Target Not Correct] will be the valid option if the generation does not seem to target any group.

A.4.3 Note on Target Group

To clarify, we did not explicitly prompt any model under examination to separately predict the target. Instead, human evaluators determine if the model under evaluation can detect the correct target group within the explanation it generates. Here, we observe that human evaluators found that 48% of the time, GPT-3.5 focused on either the wrong target group or talking about the wrong stereotype for the given target group. We want to point out that the target group specified in both datasets is annotated by humans in the respective datasets in a free text form, leading to some raw 800 different target names. A categorical detection and assessment are not possible feasible. Hence our reliance on human evaluation.

KG/Property	ConceptNet	StereoKG
Size (# tuples)	∼34M	∼4k
Curated from	Wikipedia	Reddit (offensive subreddits)
Type of tuples	World and common-sense knowledge	Religious and ethnic stereotypes
Top-k tuples via	Weighted TF-IDF	Cosine Similarity

Table 13: Summary comparison of the properties of the KGs involved in our investigation



(a) ConceptNet

(b) StereoKG

Figure 5: Analysis of top-k KG tuples retrieved for test samples of SBIC and LatentHatred at $k = 20$, w.r.t ConceptNet, and StereoKG. as described in Section 5, for ConceptNet we evaluate the IDF weighted relevance (rel.) scores. For StereoKG, we evaluate via the cosine similarity (Sim.) scores. All the y-axis captures the proportion of samples corresponding to the analysis at hand. Given that we look at top 20 tuples based on scores, (a) and (b) capture the spread of uniqueness in scores obtained per sample, respectively, for ConceptNet and StereoKGḢere, the ith index on the x-axis is the number of unique scores out of 20 present in the samples.

A.5 Auditing KG Attributes

Choosing k-relevant tuples. For ConceptNet, we follow the retrieval method used by Chang et al. (2020) and Sridhar and Yang (2022). In this setup (*Algorithm 1*), we first obtain the query terms (q) from the input post's lemmatized noun, verb, and adjective keywords. We then extract from ConceptNet all the 1-hop English tuples for each term. We also calculate the IDF score for each query term, idf_q. The top-k and bottom-k tuples are obtained by sorting the extracted relations based on relevance scores $W_{rel} \times idf_q$, as each relation in ConceptNet has a *relation-weight*, W_{rel}. For random-k, we randomly pick k tuples from the extracted set.

For StereoKG, we utilize semantic similarity-based metric (*Algorithm 2*). We first employ the all-MiniLM-L6-v2 (Reimers and Gurevych, 2019) to pre-calculate the sentence embeddings over all the linearised[7] tuple from StereoKG. We then used the cosine-similarity scores between tuples and input samples to get the top and bottom-k tuples. The same algorithm cannot be applied to both KGs due to the skewness in KG size.

Employing *Algorithm 1* for StereoKG returns a low (and zero in most instances) number of tuples per input sample. Meanwhile, employing *Algorithm 2* for querying on ConceptNet is not computationally feasible as it amounts to performing cosine similarity in the order of millions. Further, following the experimental setup from MIXGEN Sridhar and Yang (2022), we also set $k = 20$. The pseudo codes for the tuple extraction via the respective KG are outlined in Algorithm 1 and 2 for ConceptNet and StereoKG respectively.

Algorithm 1 Knowledge Tuples Extraction for ConceptNet

```
Ensure: $KG_h = \{(r_i, t_i, score_i) \mid 0 \leq i \leq N_i\}$
$query\_tokens \leftarrow$ extract adjectives, nouns, and verbs from each post
$idf\_scores \leftarrow$ TF-IDF scores of each query token given the vocabulary of all posts
rank relevant tuples from $KG_h$ in terms of $idf\_scores_h \cdot score_i$, where h is a query token
```

Algorithm 2 Knowledge Tuples Extraction for StereoKG

```
Ensure: $KG_h = \{(r_i, t_i, score_i) \mid 0 \leq i \leq N_i\}$
$emb\_vec \leftarrow$ embedding of each post from model $Q$
$lin\_KG \leftarrow$ Linearised tuples from StereoKG
$cosine\_sim(emb\_vec, emb\_vec)$
rank relevant tuples in terms of $cosine\_sim$
```

Relevancy Scoring. In Figure 5, we look at the number of unique scores (relevance or similarity for ConceptNet and StereoKG respectively) obtained for a sample. For $k = 20$, one would expect the uniqueness to be right-skewed, which is partially valid for StereoKG but not for ConceptNet where there are fewer samples with $>= 16$ unique scores and zero samples with all unique scores. Interestingly, despite the similarity metric being limited to $0 - 1$ for StereoKG it produces a higher number of unique scores compared to ConceptNet. The relevance metric is open-ended ≥ 0 for the latter. One would expect that an open-ended metric will generate more variation in scores. However, this is not the case.

Annotator Demographic for Manual Inspection. To manually examine the KG tuples, we took help from 2 expert annotators who volunteered $\approx$ 35 minutes each and scored 20 samples and their $top - k = 20$ KG tuples. The annotators, one male (24 years) and one female (29 years), are knowledgeable about natural language processing and social computing. Additionally, both adequately understand how KG's are constructed and employed in NLP. Besides providing scores, the annotators could offer any additional comment about an outlier they observed.

[7] The linearised tuples are already provided along with the triplets at: https://github.com/uds-lsv/StereoKG/