Mimicking How Humans Interpret Out-of-Context Sentences Through Controlled Toxicity Decoding

Maria Mihaela Trusca* and **Liesbeth Allein***
Department of Computer Science
KU Leuven
firstnames.lastname@kuleuven.be

Abstract

Interpretations of a single sentence can vary, particularly when its context is lost. This paper aims to simulate how readers perceive content with varying toxicity levels by generating diverse interpretations of out-of-context sentences. By modeling toxicity, we can anticipate misunderstandings and reveal hidden toxic meanings. Our proposed decoding strategy explicitly controls toxicity in the set of generated interpretations by (i) aligning interpretation toxicity with the input, (ii) relaxing toxicity constraints for more toxic input sentences, and (iii) promoting diversity in toxicity levels within the set of generated interpretations. Experimental results show that our method improves alignment with human-written interpretations in both syntax and semantics while reducing model prediction uncertainty.

1 Introduction

Misunderstandings online can often be traced back to misalignment between the meanings of text intended by the author and those inferred by the readers. This is even further amplified when text is taken out of context – which is commonplace on social media – resulting in frustration and heated discussion. In this paper, we aim to mimic how readers may interpret out-of-context sentences. We do this by modeling and generating for each sentence a diverse set of interpretations (Allein et al., 2025). Toxicity is taken as the control factor during generation as we want to simulate human interpretation behavior of sentences that exhibit varying degrees of surface-level toxicity. Generating diverse interpretations can help anticipate misunderstandings, explain reactions from readers, and recover underlying toxicity, which is especially beneficial for capturing implicit hostility and harm online (ElSherief et al., 2021; Hartvigsen et al., 2022).

This paper introduces *a novel decoding strategy for interpretation generation that explicitly controls the toxicity level of generated interpretations.* Our decoding strategy enforces three key objectives that are inspired by toxicity patterns observed in human-written sentence interpretations: *Align the toxicity level* of generated interpretations with that of the input sentence (Objective 1); *Progressively relax toxicity constraints* on the interpretations for increasing toxicity in the sentence (Objective 2); *Promote diversity in the toxicity levels* across the generated interpretations (Objective 3). These objectives are enforced iteratively during the decoding process, enabling fine-grained control over toxicity while maintaining coherence and diversity in generated text. Controlling generation in the decoding phase is particularly desirable as it bypasses the need for alterations to model architectures, allowing a plug-and-play integration with existing language models.

Our results demonstrate the soundness and effectiveness of our decoding strategy. Controlling the decoding of interpretations using all three objectives consistently leads to generated interpretations that better align with human-written interpretations in terms of syntax and semantics, compared to when generation is not controlled. Our strategy also lowers uncertainty for the base models when predicting the interpretations.

2 Related Work

Text generation can be controlled using a range of control factors, including text attributes (e.g., sentiment, style) (Hu et al., 2017; Dathathri et al., 2020), syntactic structures (Li et al., 2022), speaker or reader characteristics (Dinan et al., 2020; Majumder et al., 2020), and structured data (e.g., tables, knowledge graphs) (Zhang et al., 2023). A popular approach to condition text generation is in-context learning, where these control factors are integrated into the input (Yang et al., 2023). An-

*Equal contribution.

Proceedings of the 5th Workshop on Trustworthy NLP (TrustNLP 2025), pages 291–297

other method is to control generation during the decoding phase, e.g., by manipulating the output token distributions (Pascual et al., 2021; Yang and Klein, 2021; Kim et al., 2023).

This paper controls the toxicity of generated interpretations based on the surface-level toxicity of the original sentence during decoding. While much of the existing work on controlling toxicity in text generation focuses on reducing toxicity (Gehman et al., 2020; Liu et al., 2021; Prabhumoye et al., 2023; Wingate et al., 2022), our work builds on the idea that the toxicity of the original sentence is perceived differently among readers. We aim to capture this variability by constraining generation following three objectives.

3 Methodology

3.1 Preliminaries

Language models generate text sequences y of length T by decoding the probability of the sequence y calculated using the chain rule: $p(y) = \prod_{t=1}^{T} p(y_t|y_{<t})$, where $y_{<t} = \{y_1, ..., y_{t-1}\}$. The probabilities $p(y_t|y_{<t})$ are obtained by projecting the logits computed by the language model into the space of the model's vocabulary $\mathcal{V}$ typically using a softmax transformation. By applying the logarithmic differentiation over the chain rule, the *softmax* scores are given by $score(y_t|y_{<t}) = \log p(y_t|y_{<t})$. Once the scores are computed, a decoding algorithm such as nucleus sampling or beam search is applied to autoregressively generate y.

In our work, we aim to control the toxicity of the interpretations generated by a language model for an input sentence in a plug-and-play manner. We do this by calibrating the *softmax* scores for toxicity control before applying the decoding algorithm. To ensure the correct summation of all probabilities in the $\mathcal{V}$ space to 1, we apply another *softmax* transformation over the calibrated scores.

3.2 Toxicity control

We define a set of objectives for our decoding strategy that should closely align the generated interpretations with the toxicity behavior observed in the input sentence and human interpretations. The implementation of these objectives is summarized in Algorithm 1.

Objective 1: Match toxicity level of the interpretations to the sentence The toxicity of the generated interpretations should match the toxicity level of the input sentence, as maintaining consistency in toxicity prevents the interpretations from unintentionally intensifying or minimizing the original tone. Adopting this hypothesis, we ensure that the generated interpretation preserves the meaning of the input sentence in terms of toxicity. Since the text generation process is sequential, it is necessary to calibrate the toxicity level of the generated text after each time step t.

Algorithm 1 The implementation of Objectives 1-3

Input $s, tox(s), tox(y_t) \in \mathbb{R}^V, y = \{\}$
Output y

```
if Objective 3 and (∃)y' then
    if tox(y') < tox(s) then
        tox(s) = tox(s) + (tox(s) − tox(y'))
    else if tox(y'_T) > tox(s) then
        tox(s) = tox(s) − (tox(y') − tox(s))
    end if
end if
while t ≤ T do
    Compute score(y_t|y_<t)
    if Objective 1 then
        if Objective 2 then
            λ = 1/(tox(s) ∗ 100)
        else if not Objective 2 thenλ = 1
        end if
        if tox(y_<t) < tox(s) then
            score(y_t|y_<t) = score(y_t|y_<t)+
            λ ∗ tox(y_t)
        end if
        if tox(y_<t) > tox(s) then
            score(y_t|y_<t) = score(y_t|y_<t)−
            λ ∗ tox(y_t)
        end if
    end if
    y*_t = argmax(score(y_t|y_<t))
    y_<t = y_<t + y*_t
end while
if t=T then
    y = y_<t
end if
```

Knowing that the $tox(*)$ function indicates the toxicity level (the codomain of the function is $[0, 1]$) and given the *softmax* scores $score(y_t \mid y_{<t})$ computed by the language model for the t-th generated token y_t based on the already generated sequence of $t-1$ tokens $y_{<t}$, we calibrate the scores as follows:

$$score(y_t|y_{<t}) = score(y_t|y_{<t}) + \lambda * tox(y_t), \quad \text{if } tox(y_{<t}) < tox(s)$$
$$score(y_t|y_{<t}) = score(y_t|y_{<t}) - \lambda * tox(y_t), \quad \text{if } tox(y_{<t}) > tox(s) \quad (1)$$

where s is the input sentence, λ adjusts the toxicity control, and $tox(y_t) \in \mathbb{R}^{\mathcal{V}}$ indicates the toxicity level of y_t in $\mathcal{V}$ used by the language

Toxicity Interval of the Input Sentence	Toxicity Average Standard Deviation of the Interpretations
(0.0 - 0.2)	0.05
(0.2 - 0.4)	0.10
(0.4 - 0.6)	0.13
(0.6 - 0.8)	0.20
(0.8 - 1.0)	0.23

Table 1: Comparison between the toxicity intervals of the input sentences and the average standard deviations of the toxicity scores of all interpretations per input sentences. The average is computed at the interval level.

model. All toxicity scores are computed using the well-established BERT-HateXplain model (Mathew et al., 2021).

By implementing **Objective 1** using Eq. 1, we correct the toxicity of the generated interpretation after every time step t to ensure that the toxicity of the final interpretation converges to that of the input sentence.

Objective 2: Gradually relax control as sentence toxicity rises Empirically, we observe that input sentences with higher toxicity scores are more likely to have human interpretations with a broader toxicity range than less toxic input sentences. As shown in Table 1, the standard deviation of the toxicity scores observed in the human interpretations of an input sentence is higher for more toxic input sentences than for less toxic ones. Based on this observation, we gradually loosen the toxicity control over the generated interpretations as the toxicity of the input sentence increases. To implement this, we set the weight λ in Eq. 1 as $1/(tox(s) \cdot 100)$. If **Objective 2** is not implemented, λ is set to 1.

Objective 3: Promote diversity by alternating toxicity While the generated interpretations should preserve the meaning of the input sentence, we also want to capture the range of possible interpretations. To encourage diversity in the set of generated interpretations, we set a heuristic rule that the current generated interpretation should be higher in toxicity than the input sentence when the previous interpretation was lower in toxicity, and vice versa. To implement this, we update the toxicity score of the input sentence, $tox(s)$, after every generated interpretation as follows:

$$\begin{aligned} tox(s) &= tox(s) + (tox(s) - tox(y^{'})), \\ &\quad \text{if } tox(y^{'}) < tox(s) \\ tox(s) &= tox(s) - (tox(y^{'}) - tox(s)), \\ &\quad \text{if } tox(y^{'}) > tox(s) \end{aligned} \quad (2)$$

where $y^{'}$ is the previously generated interpretation.

Note that our decoding strategy defines the toxicity of interpretations as a function of the input sentence toxicity, meaning that we can always substitute the toxicity score of the input sentence with an arbitrary value. This feature is particularly important for content moderation by producing interpretations that deliver the meaning of the input sentence in a non-toxic manner.

4 Experimental Setup

Dataset We rely on the OrigamIM dataset[1] (Allein and Moens, 2024) to evaluate our decoding strategy. OrigamIM is the first dataset that specifically supports the interpretation modeling task (Allein et al., 2025) and includes 9,851 human-written interpretations of 2,018 sentences from Reddit posts. To accommodate the language models for this task, we fine-tune and validate them on the OrigamIM training and validation sets. The test set is used to evaluate our decoding strategy.

Models To evaluate our method for toxicity control, we use three open-source language models: BART (139M parameters) (Lewis et al., 2020), T5 (223M parameters) (Raffel et al., 2020), and LLAMA 7b (6.74B parameters) (Touvron et al., 2023). We test various combinations of our proposed objectives and compare it against the base models without explicit control.

Implementation details We fine-tune the language models on an NVIDIA GeForce RTX GPU with 24GB of GPU RAM during 8 epochs. We set the learning rate to 0.0001 and the batch size to 4 for T5 and BART and to 1 for LLAMA. We use nucleus sampling (Holtzman et al., 2020) with $p = 0.9$ during inference. Compared with the commonly used beam search, nucleus sampling is more effective and can better prevent text degeneration (Holtzman et al., 2020). The matching between the generated interpretations and the human interpretations is done using the Hungarian algorithm. Our code is available here: `https://github.com/mtrusca/ToxicityControl`.

Metrics We use METEOR (Banerjee and Lavie, 2005) to measure the syntactic similarity between the human interpretations and the generated ones. We measure semantic similarity using COMET

[1] `https://github.com/laallein/origamIM`.

Method	$METEOR(\uparrow)$	$COMET(\uparrow)$	$Perplexity(\downarrow)$	$Correlation(\uparrow)$
$BART$	29.22 ± 0.21	82.36 ± 0.31	1.27 ± 0.2	0.43 ± 0.56
$BART + Obj_1$	**29.82 ± 0.12**	83.74 ± 0.21	1.27 ± 0.1	0.41 ± 0.49
$BART + Obj_{1,2}$	29.48 ± 0.23	83.11 ± 0.3	**1.26 ± 0.1**	0.45 ± 0.23
$BART + Obj_{1,3}$	29.01 ± 0.22	84.16 ± 0.36	**1.26 ± 0.2**	0.42 ± 0.31
$BART + Obj_{1,2,3}$	29.79 ± 0.12	**85.81 ± 0.37**	1.27 ± 0.1	**0.46 ± 0.34**
$LLAMA$	27.13 ± 0.44	86.16 ± 0.26	13.19 ± 0.3	0.41 ± 0.32
$LLAMA + Obj_1$	27.73 ± 0.38	83.78 ± 0.26	13.19 ± 0.4	0.42 ± 0.41
$LLAMA + Obj_{1,2}$	**27.97 ± 0.11**	84.47 ± 0.29	13.33 ± 0.2	0.43 ± 0.64
$LLAMA + Obj_{1,3}$	27.14 ± 0.07	90.02 ± 0.4	**13.11 ± 0.1**	0.4 ± 0.35
$LLAMA + Obj_{1,2,3}$	27.84 ± 0.22	**91.07 ± 0.15**	**13.11 ± 0.4**	**0.43 ± 0.42**
$T5$	27.44 ± 0.31	79.61 ± 0.33	**1.43 ± 0.3**	0.38 ± 0.46
$T5 + Obj_1$	27.61 ± 0.1	79.07 ± 0.28	**1.43 ± 0.2**	0.41 ± 0.35
$T5 + Obj_{1,2}$	28.19 ± 0.18	81.39 ± 0.46	1.44 ± 0.2	0.42 ± 0.51
$T5 + Obj_{1,3}$	27.52 ± 0.39	81.98 ± 0.37	1.44 ± 0.3	0.42 ± 0.24
$T5 + Obj_{1,2,3}$	**28.25 ± 0.12**	**82.9 ± 0.27**	**1.43 ± 0.2**	**0.44 ± 0.36**

Table 2: Quantitative evaluation of our decoding strategy for controlling toxicity in text generation (mean and standard deviation; three runs).

Method	$METEOR(\uparrow)$	$COMET(\uparrow)$	$Perplexity(\downarrow)$	$Correlation(\uparrow)$
$LLAMA + Obj_{1,3}(\lambda = .25)$	27.44 ± 0.09	88.93 ± 0.38	13.11 ± 0.1	0.41 ± 0.36
$LLAMA + Obj_{1,3}(\lambda = .50)$	27.54 ± 0.28	89.93 ± 0.22	13.12 ± 0.2	0.4 ± 0.28
$LLAMA + Obj_{1,3}(\lambda = .75)$	27.36 ± 0.38	90.44 ± 0.22	13.11 ± 0.1	0.41 ± 0.21
$LLAMA + Obj_{1,3}(\lambda = 1)$	27.14 ± 0.07	90.02 ± 0.4	13.12 ± 0.1	0.4 ± 0.35
$LLAMA + Obj_{1,2,3}$	**27.84 ± 0.22**	**91.07 ± 0.15**	**13.11 ± 0.4**	**0.43 ± 0.42**

Table 3: The effect of λ on the decoding mechanism for toxicity control. While the first four models use a fixed λ, model $LLAMA + Obj_{1,2,3}$ implements a decreasing λ as the toxicity of the input sentence increases.

(Rei et al., 2020). COMET is suitable for interpretation modeling because it was trained to recognize human preferences between correct and incorrect translations, which can be applied to the "translations" of meaning in interpretations. Additionally, COMET considers both the similarity between the generated interpretation and the human interpretation, as well as between the generated interpretation and the input sentence. The third metric we report is perplexity, which shows the level of uncertainty the models have in predicting the generated interpretations. The final metric is the Spearman correlation computed between the toxicity scores of the generated interpretations and the scores of the human interpretations.

5 Results

Quantitative analysis Table 2 presents the quantitative results of integrating our method into the text decoding of T5, LLAMA, and BART models. Syntactically, we notice that controlling toxicity in text generation consistently enhances the capacity of the models to generate interpretations similar to the input sentence. Analyzing METEOR scores, we observe that the implementation of the first objective has the strongest capacity to increase syntactic similarity, while the implementation of the other two objectives further enhances this similarity, as observed in the cases of LLAMA and T5. Regarding semantic similarity, the meaning of the input sentence is better preserved when toxicity is directly adjusted during decoding. When toxicity is controlled using all three objectives, COMET scores show a substantial increase compared to the results of the base models, with improvements of 4.10% for BART, 5.54% for LLAMA, and 4.04% for T5.

Regarding perplexity, implementing our decoding strategy generally results in lower model uncertainty when generating the interpretations. Correlation scores further confirm that the toxicity-controlled interpretations better capture the toxicity behavior observed in human interpretations than when toxicity is not controlled. Lastly, the results show overall improvement in the interpretation generation performance when all three objectives are enforced.

To demonstrate that a variable λ value (as required by **Objective 2**) is more advantageous than a fixed value, we evaluate our decoding

Sentence	*Over the past few years across Western nations, there have been several attempts by 'Progressives' to gain power.*	Tox.: 0.04
Ground-Truth Interpretations	1. Politics that have stagnated are due to be contested. 2. Over the past few years across Western nations, there have been several endeavors by 'Progressives' to gain puissance.	Tox.: 0.03 Tox.: 0.04
Generated Interpretations	1. Western countries are not better than the Russians when it comes to politics. 2. In recent years across Western countries, many left-leaning politicians have been trying to gain power	Tox.: 0.03 Tox.: 0.08
Sentence	*Zoomers are inexperienced, idealistic do-gooders who are woke, and Boomers are world destroyers who consume everything without thinking, and will be the death of us all.*	Tox: 0.47
Ground-Truth Interpretations	1. The young generation is too idealistic and not realistic while the older generation consumes resources mindlessly. 2. Zoomers are callow, idealistic do-gooders who are woke, and Boomers are world destroyers who consume everything without mentally conceiving, and will be the demise of us all.	Tox: 0.18 Tox: 0.45
Generated Interpretations	1. Millennials (zoomers) are naive idealists who are attuned to the environment and boomers are extinctionists who consume everything without thought, and will ultimately kill us. 2. Zoomers and Boomers' lifestyles are completely different.	Tox: 0.53 Tox: 0.19
Generated Interpretations ($\boldsymbol{tox(s) = 0.2}$)	1. Zoomers and Boomers have different ideals on how to deal with the world. 2. The writer seems to be pointing to a kind of inter-generational difference that motivates and polarizes extreme political movements.	Tox: 0.17 Tox: 0.21

Table 4: Examples from the OrigamIM test dataset that present toxicity behavior in $LLAMA + Obj_{1,2,3}$.

strategy using different fixed λ values ($\lambda = 0.25, 0.50, 0.75, 1$). As shown in Table 3, a variable λ results in better manipulation of the toxicity level in the generated text and achieves higher semantic and syntactic similarity to the human interpretations, compared to when λ is fixed.

Qualitative analysis Table 4 presents several interpretations generated by LLAMA using our decoding strategy. When the toxicity score of the input sentence is low, the generated interpretations are also non-toxic. However, this does not prevent LLAMA from being creative and discussing Russian politics in the context of Western political systems. Conversely, when the input sentences have a high level of toxicity, the generated interpretations either reflect the toxicity or produce milder interpretations. Note that we can moderate the toxicity of an input sentence by replacing its toxicity score $tox(s)$ with a lower value that allows generation of non-toxic interpretations (last line in Table 4).

6 Conclusion

In this work, we proposed a modular decoding algorithm with three objectives designed to explicitly guide the generation of interpretations of out-of-context sentences. We showed that specifically constraining text decoding on toxicity brings generated interpretations closer to those written by humans. However, human interpretation is driven by many factors beyond toxicity like cultural background and personal experiences. We therefore strongly encourage future research to also consider these contextual factors when modeling the diverse ways in which a sentence's meaning is perceived.

Limitations

Due to the external classifier used to detect toxicity, the ability to control the toxicity of our decoding strategy is strongly correlated with the data used to train the classifier. As a result, our strategy depends on the quality of the classifier's training data.

Ethical Considerations

Our decoding method intentionally amplifies toxicity in certain generated interpretations to better replicate human interpretations of out-of-context sentences with varying levels of toxicity. While promoting toxicity in text generation may seem controversial, it is not inherently negative in all contexts. Minimizing or even entirely removing toxicity is crucial for applications like customer service, education, or mental health support – where safety and ethics are non-negotiable. However, some systems actually benefit from the ability to produce texts with varying degrees of toxicity. For example, explicitly highlighting toxicity in generated text can help improve content filtering systems

and facilitate better detection of harmful language. As such, we believe that developing methods for the controlled and adaptable regulation of toxic language is valuable. Nevertheless, it is important to exercise caution in designing and implementing these methods to ensure they are used responsibly and ethically.

Acknowledgements

This work has been funded by the Research Foundation - Flanders (FWO) under grant G0L0822N through the CHIST-ERA iTRUST project.

References

Liesbeth Allein and Marie-Francine Moens. 2024. OrigamIM: A dataset of ambiguous sentence interpretations for social grounding and implicit language understanding. In *Proceedings of the 3rd Workshop on Perspectivist Approaches to NLP (NLPerspectives) @ LREC-COLING 2024*, pages 116–122, Torino, Italia. ELRA and ICCL.

Liesbeth Allein, Maria Mihaela Trusca, and Marie-Francine Moens. 2025. Interpretation modeling: Social grounding of sentences by reasoning over their implicit moral judgments. *Artificial Intelligence*, 338:104234.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization@ACL 2005, Ann Arbor, Michigan, USA, June 29, 2005*, pages 65–72. Association for Computational Linguistics.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*.

Emily Dinan, Angela Fan, Adina Williams, Jack Urbanek, Douwe Kiela, and Jason Weston. 2020. Queens are powerful too: Mitigating gender bias in dialogue generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8173–8188, Online. 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.

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.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. ToxiGen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3309–3326, Dublin, Ireland. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *International Conference on Machine Learning*, pages 1587–1596. PMLR.

Minbeom Kim, Hwanhee Lee, Kang Min Yoo, Joonsuk Park, Hwaran Lee, and Kyomin Jung. 2023. Critic-guided decoding for controlled text generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4598–4612, Toronto, Canada. 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, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. 2022. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, 35:4328–4343.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. DExperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6691–6706, Online. Association for Computational Linguistics.

Bodhisattwa Prasad Majumder, Harsh Jhamtani, Taylor Berg-Kirkpatrick, and Julian McAuley. 2020.

Like hiking? you probably enjoy nature: Persona-grounded dialog with commonsense expansions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9194–9206, Online. Association for Computational Linguistics.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2021. Hatexplain: A benchmark dataset for explainable hate speech detection. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14867–14875. AAAI Press.

Damian Pascual, Beni Egressy, Clara Meister, Ryan Cotterell, and Roger Wattenhofer. 2021. A plug-and-play method for controlled text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3973–3997, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shrimai Prabhumoye, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2023. Adding instructions during pretraining: Effective way of controlling toxicity in language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2636–2651, Dubrovnik, Croatia. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:140:1–140:67.

Ricardo Rei, Craig Stewart, Ana C. Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 2685–2702. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

David Wingate, Mohammad Shoeybi, and Taylor Sorensen. 2022. Prompt compression and contrastive conditioning for controllability and toxicity reduction in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5621–5634, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kevin Yang and Dan Klein. 2021. FUDGE: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3511–3535, Online. Association for Computational Linguistics.

Kexin Yang, Dayiheng Liu, Wenqiang Lei, Baosong Yang, Mingfeng Xue, Boxing Chen, and Jun Xie. 2023. Tailor: A soft-prompt-based approach to attribute-based controlled text generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 410–427, Toronto, Canada. Association for Computational Linguistics.

Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. 2023. A survey of controllable text generation using transformer-based pre-trained language models. *ACM Computing Surveys*, 56(3):1–37.