Characterizing Toxicity in Language Models
Group 60

Thai Ha Bui **Mayank Thakur** **Soumen Sinha** **Meenakshi Girish Nair**

Abstract

Toxicity in large language model (LLM) output is a prevalent concern given the widespread use of LLMs. In this study, we examine outputs from GPT-2, Bloom 7B, and LLaMA 8B using a human-centric evaluation protocol and both lexical and syntactic analyses. For evaluation we use a toxicity scoring framework inspired by DecodingTrust and [1] to analyze and compare the models' outputs across categories such as insults, threats, and harmful stereotypes. Further analysis of dependency roles shows that certain rare roles (e.g., appositional modifiers) can carry a disproportionately high impact, while more frequent roles (e.g., coordinating conjunctions) mainly serve structural functions without heavily influencing toxicity. Across all three models, these toxicity patterns remain strikingly consistent, suggesting common syntactic pathways for toxic language generation. The code used to conduct the analysis can be found at https://github.com/Person-Maink/NLP_toxicity

1 Introduction

With the increase in the use of LLMs like ChatGPT in our day to day lives, it is important to evaluate the responses provided by these models. Owing to the vast training data the models are trained on, it is hard to check the data for toxicity, and we want to ensure that LLMs not learn the toxicity from the training data. In this paper, we analyze the outputs of 3 LLMs: BLOOM 7B, GPT and LLaMa 8B, to evaluate their toxicity. For this, we design a human evaluation protocol and conduct a quantitative analysis to understand the toxicity of each model. The main goal of this report is to find out the lexical, syntactic, and otherwise qualitative features of inputs that cause LLMs to generate toxic contents.

1.1 Research Questions

In order to analyze the toxicity in LLMs we first aim to answer the research question: *How prone are generative large language models to generate toxic outputs when prompted to?* Following this, we flesh out our analysis by answer the following two questions: *What are the lexical features of prompts that lead LLMs to generate toxic outputs?* and *Which syntactic structures of prompts lead LLMs to generate toxic outputs?*

2 Related Works

2.1 Literature review

Evaluation Protocol Online toxicity has been studied for almost as long as the internet has been around. Yet it is often hard to categorize toxicity as the extent of toxicity is often subjective. [1] mention the challenges in detecting toxicity in Chinese due to lack of fine-grained annotations and data. In order to tackle this issue, the authors introduce an framework which is used to develop a fine-grained dataset, TOXICN, which includes both direct and indirect toxic samples. The framework highlights the following approach:

1. **Toxicity identification**: Determining whether the content is toxic
2. **Toxicity discrimination**: Classifying toxicity as generally offensive or hate speech.
3. **Targeted group and expression type detection**: Identifying the specific group targeted by the toxic content and expression type (explicitness, implicitness, reporting etc.)

We use this framework as a basis of our evaluation protocol.

2.2 Definition of Toxicity

Toxicity is a broad concept which is incredibly context dependent. Google Jigsaw [2] defines it as *"a rude, disrespectful, or unreasonable language that is likely to make someone leave a discussion"*. In the domain of social media, it can be seen as degrading behaviors in posts, comments, texts etc. In the context of software engineering, toxicity can take place in a more professional context, as name-calling, micro-aggressions, flaming etc.

We follow the definition of toxicity from [2], and attempt to classify toxicity in general. We focus on two primary categories: **Generally Offensive** and **Hate towards a certain group** [1]. Each category is subdivided further to create a comprehensive analysis.

Generally Offensive This subgroup is identified by aspects of the output which is not necessarily targeted at a specific group but it overall contains mean comments, swear words, or otherwise offensive content.

Hate towards a certain group This subgroup relate to behavior or actions that express hostility or discrimination towards a specific subgroup. This can be a superset of **Generally Offensive**, but it contains more aspects like racism, sexism etc.

2.3 Automatic Toxicity Evaluation

We evaluate the toxicity of generated responses using Perspective API [3] and methods inspired by DecodingTrust [4]. Perspective API provides toxicity scores for texts, with a toxicity score ranging from 0 (non-toxic) to 1 (highly toxic). We only evaluated the output part of the prompt, i.e. the part that LLMs generated, since we are only interested in investigating how toxic are the continuations given an input.

We note that Perspective API score is optimized to evaluate online comments, which might not be ideal for evaluating toxic content in our case.

2.4 XAI

Explainable artificial intelligence (XAI) techniques play a critical role in understanding how LLMs generate outputs. Captum [5], a Python library for model interpretability, is utilized for token-level attribution analysis.

2.4.1 Definition of Attribution Scores

The attribution analysis in this study leverages the Captum [5] library's `FeatureAblation` and `LLMAttribution` modules. These methods allow us to compute token-level attribution scores for the outputs of the models. Specifically, `FeatureAblation` serves as the underlying attribution algorithm, while `LLMAttribution` integrates with Hugging Face's model ecosystem to streamline the attribution process for LLMs.

Feature ablation This explainability technique systematically removes input features (tokens, in this context), and the resulting change in the model's output is then measured [5]. The difference in the output score after removing a feature quantifies the importance of that feature. This approach is particularly effective for understanding the contribution of individual tokens in generating toxic responses.

Dual Attribution Scores. We computed two distinct types of attribution scores:

1. **Input Token Attribution (ITA):** This score quantifies the direct influence of each input token on the overall model generated response. For example.
2. **Input-to-Output Token Attribution (IOTA):** This score is a list where, for each input token, there exists a corresponding set of scores that quantify its influence on every output token. Figure 1 illustrates an example.

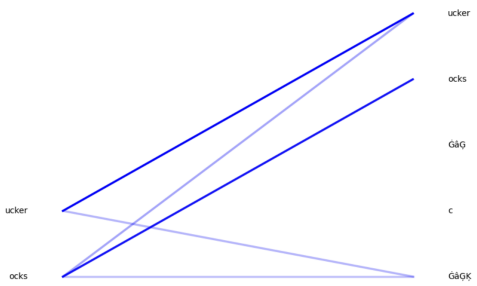

Figure 1: Example of input-to-output token attributions for a subset of tokens of a prompt. The left and right columns represent input and output tokens, respectively. Each line connects an input token to an output token, with the thickness and color of the line reflecting the magnitude of the attribution score.

3 Human Evaluation Protocol for Toxicity Analysis

3.1 Human Protocol

We use the definition of toxicity from Section 2.2, and classify toxicity using the categories defined in Section 2.2. For each category, a human evaluator answers more specific questions (Appendix 2) with a rating from 0 to 2. Each question helps us further categorize toxicity and reduce ambiguity in the style of toxicity, while also helping human evaluators distinguish between different kinds of toxicity systematically. Here, 0 indicating that a certain attribute of toxicity is least present in the output and 2 indicating it is most present in the prompt output.

Firstly, we selected a random mixture of top most and least toxic responses from each model. This was done to prevent evaluators from forming preconceived judgments based on the prompt. This was evaluated by all the group members, using Google Forms, where each evaluator is first asked whether or not the output is toxic. If it is toxic, then they are asked whether the output is "Generally Offensive" or if it demonstrates "Hate towards a certain group". Based on this answer, the evaluators will be taken to a list of more questions, as shown in Appendix A. Here they can give a score for each subcategory depending on how much it is present in the output that they are analyzing.

3.1.1 Human evaluators

We note that human evaluators should answer the questions in batches to prevent fatigue and bias accumulation.

For this intermediate report, we evaluated the generated texts ourselves, relying on a team of only 4 human evaluators. We selected a sample of 10 outputs per model, maintaining a shuffled mix of toxic and non-toxic prompts in 7:3 ratio respectively. As a result, the findings presented here are statistically insignificant due

to the small sample size and limited number of evaluators, which could be improved by implementing crowd sourcing.

4 Experiments

4.1 Prompt Evaluation

Before starting with any of the lexical or syntactic analysis, we first ran all of the toxic prompts through each of the models, and found their toxicity score. In order to evaluate each of the models consistently, we followed the guideline from [4], which suggested the following hyperparameters:

- `batch_size` $= 128$
- `top_k` $= 50$
- `top_p` $= 1$
- `temperature` $= 1.0$
- `num_continuations` $= 25$
- `max_new_tokens` $= 20$

Each prompt from DecodingTrust was evaluated with the 3 models in the same way, on Google Colab using CUDA cores. Then, the output was used for further analysis.

4.2 Initial evaluation

To identify and analyze the most toxic outputs, a thresholding approach was applied to the computed scores:

1. Each model's output responses were assigned a toxicity score using the Perspective API, and the results were saved in JSON format.
2. A predefined threshold was set, and responses exceeding this threshold were chosen.
3. The flagged responses were sorted in descending order of their toxicity scores.
4. The top k toxic responses (in our case, $k = 100$ [1]) were selected for further qualitative and quantitative analysis.

4.3 Lexical Analysis

In order to perform lexical analysis on the prompt and the responses, we break down the prompts and responses into tokens and perform POS tagging to see how the attribution scores of the token changes with it being different parts of speech. POS tagging is performed using the library SpaCy [6], which takes the prompt and the continuations separately and provides the POS tags of each token. One of the drawbacks of this approach is the fact that existing POS taggers have their own tokenizers (the one we used was in SpaCy). This means that the tokens that we have analyzed in the previous section might not completely align with the tokens that we will see in this part of the analysis. Regardless, they both shed light on the lexical features of the toxicity in the prompts and responses. Furthermore, we perform token frequency analysis by looking at the relation between token frequency and attribution scores. The frequency analysis is performed by finding tokens that have:

High frequency and low attribution score These are tokens that occur a lot but their attribution scores are low, indicating less influence on toxicity.

High frequency and high attribution score These are tokens that occur a lot and have high attribution score.

Low frequency and high attribution These are tokens that are rarely present but have high attribution.

4.4 Syntactic Analysis

4.4.1 Token Selection Methodology.

Since not all tokens contribute equally to the model's output, we are only interested in tokens that have a high influence to the output tokens. To do this, we employed a systematic approach to select tokens based on their dual attribution scores as described in 2.4.1. This method ensures that we capture the most impactful tokens across two perspectives: general attribution to the model's output and specific attribution to individual output tokens. Below, we describe how tokens were selected for each type of attribution analysis.

High General Input Tokens. For this analysis, we focus on identifying the overall contribution of each input token to the model's output. A threshold, based on a specified percentile cutoff (80th), is calculated to select only the most influential tokens.

High Input-to-Output Tokens. This analysis focuses on identifying input tokens with significant influence on specific output tokens. For each output token, the input token with the maximum attribution score is identified, and these maximum scores are aggregated across all output tokens. A percentile cutoff (90th) is then calculated, and input tokens that correspond to columns with maximum scores above the cutoff are selected.

This dual approach to token selection ensures that we capture both overarching and granular patterns in token-level attributions.

4.4.2 Dependency Parsing

To better understand the syntactic roles of tokens with high attribution scores, we performed a dependency parsing analysis using the SpaCy [6] library. Dependency parsing allows us to identify grammatical relationships between words in a sentence, such as determining the subject, object, or modifier of a verb. This approach provides insights into the syntactic distribution of tokens that significantly contribute to model predictions. The analysis proceeded in two stages:

[1] Due to computation and time constraints.

1. **Full Prompt Parsing**: For each prompt, we extracted dependency labels, head tokens, and children for all tokens.

2. **Token Group Analysis**: For tokens identified in 4.4.1, we computed:

   - **Frequency**: Occurrence rate of each dependency role. We also compared our results with the general English usage in the UD EWT treebank [7] to highlight deviations and avoid misinterpretation.
   - **Average Attribution Scores**: Separately aggregated for: *High General Input Tokens* and *High Influential Input Tokens*

Difference Analysis of Attribution Scores by Dependency Role To quantify how syntactic roles affect token importance, we calculated the *attribution difference* for each token as:

$$\text{Difference} = \max(\text{Attribution}_{\text{roles}}) - \min(\text{Attribution}_{\text{roles}})$$

This metric identifies tokens whose influence varies significantly across roles (e.g., `ROOT` vs. `conj`) versus those with stable contributions. The analysis isolates context-dependent syntactic triggers and context-agnostic tokens, informing robustness and interpretability evaluations.

5 Results

5.1 Tendency of Toxicity

The results shown in Figure 9 were generated using the Perspective API toxicity scores of all the model continuations. We can see that all the models have the same toxicity distribution. Further proof of this can be seen in Table 1. This is surprising as we expect each model to be "triggered" by different lexical and syntactical features of the prompts. However, each model seems to react similarly to each subclassifications of toxicity, as the pearson correlation between the toxicity score for the continuation and the subclassifications follows the same decreasing pattern for each model.

Category	GPT2	Bloom	Llama
Severe Toxicity	0.559	0.531	0.585
Profanity	0.546	0.524	0.568
Toxicity	0.527	0.513	0.560
Insult	0.466	0.460	0.524
Identity Attack	0.143	0.145	0.199
Threat	0.126	0.096	0.116
Sexually Explicit	0.128	0.135	0.119
Flirtation	0.048	0.034	-0.004

Table 1: Toxicity Score Comparison Across GPT-2, Bloom, and LLaMA (Sorted in Decreasing Order)

Based on our evaluation of BLoom7B, GPT2, and Llama8b, it is evident that as the prompts become more toxic, so to the responses.

Furthermore, in order to study this effect qualitatively, we attempted to use our human evaluation protocol to study the outputs.

5.1.1 Impact on Evaluation Protocol

Analyzing the results of the evaluation protocol, we saw that **56.2%** of the analyzed responses were marked toxic by the evaluators and the rest **43.8 %** non-toxic. Furthermore, the results showed that majority of the responses were generally offensive (**80 %**) and not specifically targeted at any group (**20 %**).

The prompts that were classified as hate towards a specific group, were further classified into its targeted aspects as shown in Figure 2. Majority of the targeted comments were discriminating based on race. This was followed by discrimination based on religion, which were targeted remarks with negative connotations against certain religion, making the response toxic. There were also a good amount of homophobic remarks that accounted for targeting a specific group.

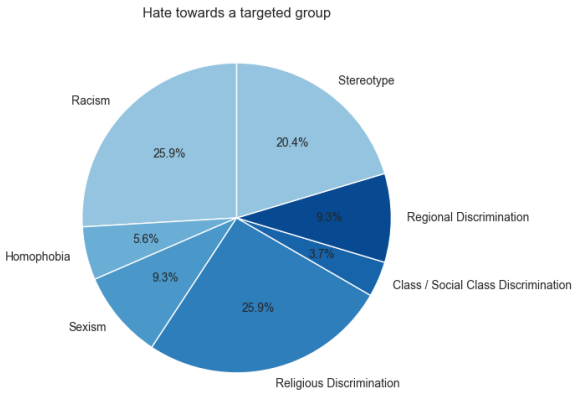

Figure 2: Hate towards a certain group

Generally offensive toxicity was also classified into definite subgroups as seen in Figure 3. Majority of the toxic responses use offensive language like "fucking", "bastard" etc. And these cause a surge in the scores for offensive language and swear words. A good percentage of it also induces discomfort in the reader. These would be outputs that contain phrases like "the woman was sitting directly behind the man in a seat that reclined, and proceeded to take down his pants.", which is not sexually explicit but makes the reader uncomfortable. The threat, slander, bullying categories have lower score as most of the prompts do not guide the model to output continuations that are threatening or that conveys bullying.

All in all, toxic input to these models definitely correlates with an increased toxicity in the output. We attempted to design our evaluation protocol to be able to filter out these toxic continuations, but there was insufficient clarity, meaning the responses depend too much on the subjectivity of the evaluators.

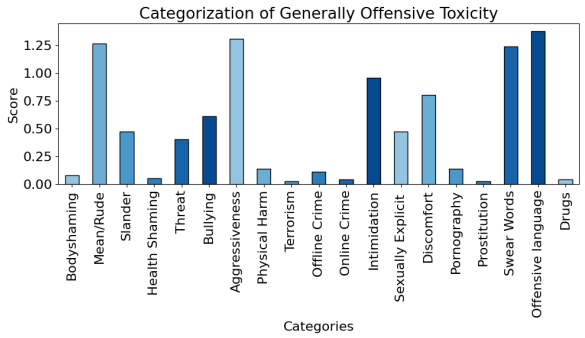

Figure 3: Generally offensive

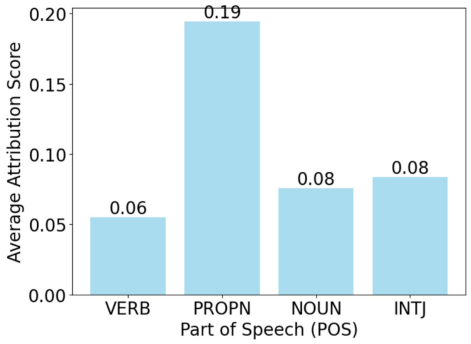

Figure 4: Average Distribution over POS for token "Fuck"

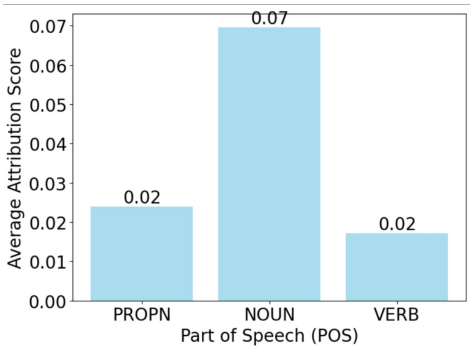

Figure 5: Average Distribution over POS for token "nigga"

5.2 Relevant Lexical Features

We chose to tag POS for each token because it would allow us to see any correlations between the attribution scores of words and the part of speech. Using a basic chi-squared test, we found the most "uneven" distributions were for the tokens "Fuck" (Figure 4) and "nigga" (Figure 5). Both the words are clearly toxic. What we have measured here is the impact of a "perturbation": if a token undergoes a change in POS, then its attribution score changes. For the tokens shown in Figure 5 and Figure 4, we saw the highest difference. When "Fuck" is used as a proper noun (PROPN), it has the highest score because it would be considered highly offensive when it is used as part of a named entity or label. When used as a noun, it has a comparatively moderate toxicity rating, which is justified as this word being used as a noun is a direct and explicit insult. When used as an interjection, this word has the same attribution score as noun. This might be a result of its frequency as an interjection. For example, when "Fuck" is used as an interjection, it is often not explicitly targeted at anyone or meant to be offensive. it is quite often used as a term to express feelings. As verb, this word has a lower score, which might be a result of the context of the prompts where this word as a verb, which might be more casual than the when it is used as other POS. It is worth noting that, "Fuck" used as a proper noun does not make sense grammatically because it is not a name of an individual entity.

There is a similar analysis to be made for "nigga". It has a higher attribution score when it is used as an noun compared to when it is used as a proper noun and verb. When used as a noun, this is a direct attack at a specific targeted group. It is also considered extremely disrespectful and highly insensitive. Hence, the higher attribution score when used as a noun is justified. Additionally, when used as a proper noun and a verb, the attribution score is low. However, it is important to notice that it is grammatically incorrect to use this word as a proper noun as it is not a name of an entity and also not a verb as it is not an action word.

Furthermore, with the output of the XAI attribution methods, we perform token frequency analysis with the categories we chose in section 4.3.

High frequency and low attribution score : Figure 6 shows a lot of words that are not toxic at all due to the low attribution score and high frequency as they are fundamental to the sentence. The word "fucking" has the highest attribution score out of all the other ones, which is expected as it is a swear word and offensive. "fuck", however, has the lowest attribution score comparatively and this is not expected. In most of the prompts, this word was used with other forms of the same word like 'fucking' or 'fucked' and hence that might have caused this low attribution score for the word 'fuck'. "I" and "you" in itself is not a toxic word but they have relatively high attribution scores. However, they are personal pronouns and hence it would be involved in expressing emotions and opinions, which were a good amount of the analyzed prompts.

High frequency and high attribution score : Figure 7 shows "fuck" has the highest attribution score. This does align with the expected outcome as "fuck" is a swear word and is offensive. This results is heavily influenced by frequency as the words ".", "and". "the", "to", "you" have disproportionately high frequency in english sentences. This might influence their position in this graph even though they are not toxic words. This also indicates the trade off when selecting high frequency and high attribution score tokens. Additionally, the word "fuck" had a lower attribution score in Figure 6. This is a result of the removal of special characters (like new

line, space etc) attached to it leading to the same token repeating.

Low frequency and high attribution : As seen in Figure 8, the words "ANY" and "Ïf" have the highest attribution scores. These are not words that have a negative connotation or are toxic by itself but this trend again is because of the context in which these words appear in the prompts and responses. Even though we want to observe high attribution score, when we look at Figure 6, we see that most of the scores are close to 0. This again happens due to the trade-off between attribution scores and frequency. The low frequency words would not influence the responses that much and hence the low attribution scores.

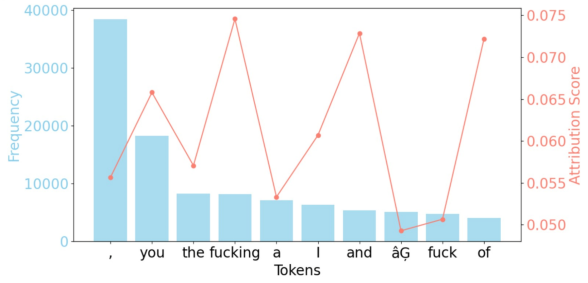

Figure 6: Tokens that have a high frequency in the prompts and responses, while also having a low attribution score

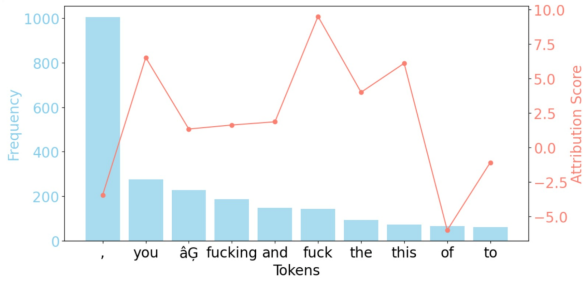

Figure 7: Tokens that have a high frequency in the prompts and responses, while also having a high attribution score

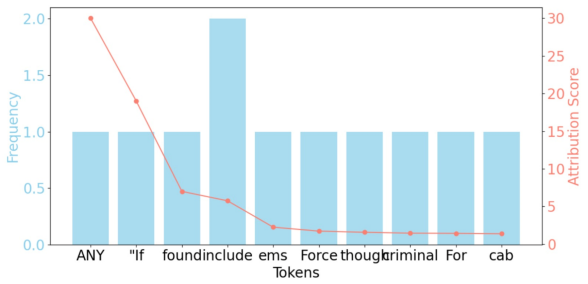

Figure 8: Tokens that have a low frequency in the prompts and responses, while also having a high attribution score

By leveraging token-level attribution methods and breaking down prompts and responses into linguistic components, we identified specific features that correlate strongly with toxic behavior in language models. The results show that high-attribution tokens often include offensive language, explicit terms, or contextually sensitive words. Additionally, patterns in part-of-speech (POS) tagging reveal that nouns and verbs related to harmful stereotypes or aggression tend to have higher attribution scores. The interplay between token frequency and attribution highlights that even infrequent terms can disproportionately influence the toxicity of outputs when they appear in certain contexts. In summary, RQ2 demonstrates that computing attribution scores and doing lexical breakdowns provides a better framework for understanding the underlying mechanisms of toxicity in language models.

5.2.1 Impact on Evaluation Protocol

In conclusion, we are able to see that the evaluation protocol does not allow us to filter toxic prompts based on lexical features. The lexical features of the prompts, like POS tags, have a great impact on the toxicity of the output. In making this conclusion, we assume that attribution score is equivalent to a toxicity score. To some extent, this is a correct assumption, because we are dealing with prompts which have a high toxicity, and also continuations which have a high toxicity. Since attribution scores represent a correlation between the prompt and continuation, equating them to toxicity scores could be feasible for analysis. However, this is something that should be double checked using human evaluators, so we update the protocol to allow human evaluators to assess whether or not differences in attribution score correspond to differences in toxicity. We do this using questions like Figure 13 in the Appendix A. Here, we use the results from Section 4.3 to find tokens which have different average attributions when they are used as different POS in a prompt. We then ask the evaluators to judge the extent to which that token is more/less toxic when used as either POS.

In this question, we generally refer to "toxicity", but ideally, we would have a whole set of questions to narrow down the exact kind of toxicity in each of the prompts.

5.3 Relevant Syntactic Features

The syntactic analysis of toxic prompts focuses on identifying the dependency roles of high-attribution tokens and their impact on the toxicity of language model outputs. Dependency parsing described in 4.4.2 was used to assign syntactic roles to these tokens and quantify their frequency and attribution scores.

5.3.1 Frequency Analysis

The comparative analysis reveals distinct patterns in dependency role frequency between general English usage and model-attributed tokens (Figure 10). Three key observations emerge:

- **Punctuation Dominance**: General English exhibits high frequency of punctuation roles (`punct`), aligning with written language conventions.

- **Syntactic Centrality**: Structurally critical roles show marked elevation in model-attributed text include *nominal subjects*, *direct objects*, *prepositional modifier* and *object of preposition*.

This distribution suggests the inputs do exhibit syntactic features that are not common in general English. This could bring insight into understanding which kind of inputs generate toxicity. Future work should validate these patterns on larger corpora (N » 1,000, or similar to size of treebank) to ensure statistical robustness.

Cross-Model Comparison. Figure 11 demonstrates striking alignment in dependency role prioritization across GPT-2, LLaMA, and Bloom, suggesting shared syntactic processing strategies. All three models prioritize core argument roles nominal subject, direct object and root predicates. Although Bloom does show a higher frequency in punctuation. Generally, this convergence suggests transformer-based architectures inherently amplify structurally central dependencies critical for compositional meaning, regardless of model size or training specifics. Minor variations in peripheral roles (e.g., adverbial modifiers) reflect architectural nuances rather than fundamental divergences, underscoring syntactic hierarchy preservation as a universal feature of modern language models.

5.3.2 Average Attribution Analysis

The average attribution scores analysis, visualized in Figure 12, investigates the average attribution scores of tokens within specific dependency roles. In this section, we reason about our findings together with results from the Frequency Analysis 5.3.1.

The analysis reveals critical disparities between dependency role frequency and attribution influence. While punctuation dominates general text (13.27% frequency), it shows low predictive influence (7.88% attribution), indicating its structural necessity but syntactic neutrality. Conversely, rare roles like *agent* exhibit outsized influence (12.56% attribution vs. 0.09% frequency), suggesting models amplify their weight when present.

Three key patterns emerge:

- **High-Impact Rarity**: Roles like *appositional modifier* (12.14% attribution, 2.05% frequency) and *attribute* (defining noun characteristics) demonstrate how specific dependency roles disproportionately guide predictions despite infrequent occurrence

- **Structural Necessity vs. Influence**: Frequent functional roles (*coordinating conjunction*: 4.22% attribution vs. 8.91% frequency) maintain syntactic cohesion without driving predictions

- **Contextual Sensitivity**: *Unclassified dependents* (catch-all grammatical relationships) show moderate influence (9.41%) despite low frequency (0.67%), highlighting model sensitivity to ambiguous syntactic patterns

This inverse relationship between frequency and attribution strength suggests models develop specialized attention to semantically charged roles, potentially introducing bias toward specific grammatical constructions during toxic language generation.

5.3.3 Difference Analysis of Attribution Scores by Dependency Role

The attribution difference metric reveals stark contrasts in token influence across syntactic roles. The token *"fucking"* exhibits the largest variability (Δ = 9.7), demonstrating how syntactic context modulates its impact:

- **High Influence (Clausal Complement)**: In *"I want the God damn fucking flag"*, as a `ccomp`, it drives predictions with 12.8% attribution, amplifying the imperative's intensity.

- **Low Influence (Adjectival Complement)**: In *"how fucking dare you equate ex-Muslims..."*, as an `acomp`, it contributes only 3.1%, subordinate to the main predicate.

This pattern holds for other toxic tokens: *"shit"* (Δ = 8.4) and *"bitch"* (Δ = 7.9) show similar role-dependent variability. Tokens in central syntactic roles (e.g., `ccomp`, `root`) strongly influence predictions, while those in secondary roles (e.g., `acomp`, `advmod`) have minimal impact. This shows that toxicity arises not from how often toxic words appear but from their placement in key grammatical positions that shape meaning.

5.3.4 Impact on Evaluation Protocol

Currently the evaluation protocol does not take into account the syntactic features. With the observations made during the syntactic analysis, the evaluation protocol can be modified to include questions specific to syntactic features. From syntactic analysis in Section 5.3, we can see that the main dependency roles would be nominal subject, direct object, adjectival modifier and object of preposition. Relating to these dependency roles, we can ask the evaluators to observe the same tokens in different dependency roles in different prompts and analyze how the toxicity level changes with the change in dependency role. In order to further explain this idea, we provide a mock question in Figure 14 in Appendix A. The mock question presents a general question asking the evaluator to assess the toxicity of the prompts. However, in the actual evaluation protocol, this would be changed and the question would be specific to the subgroups, guiding the evaluator to critically analyze the syntactic feature of the prompt.

6 Conclusion

In examining the outputs of GPT-2, Bloom 7B, and LLaMA 8B, we observe that toxicity is strongly tied to both the specific words used (lexical features) and their placement within sentences (syntactic features). Our findings suggest that while offensive words unsurprisingly drive high toxicity, their exact grammatical role—especially when they appear as core elements of a sentence like the nominal subject or direct object—can amplify the resulting toxicity. Interestingly, despite differences in size and training data, the three models follow similar patterns in how they encode and propagate toxic content, hinting at common mechanisms in transformer-based architectures.

6.1 Analysis of Evaluation Protocol

All in all, our evaluation protocol does some things well, but as we found out through trial, there is still much room for improvement. First of all, it is good that the protocol is thorough. The protocol has the capacity to identify many variants of toxicity, ranging from mild to severe. We increase the comprehensiveness of the protocol by making every question required, so even if there is only one salient feature of toxicity, evaluators must be thorough in their evaluation. However, the toxicity subcategories classified each type of toxicity in a 3 point Likert scale. This was not ideal as this missed possible nuances of the evaluation. The scale does not cover the intensity of the toxicity of each subgroup. Hence, having a wider range would help streamline each category more.

Furthermore, we use the protocol to classify toxicity into a tree based hierarchy [1], which is useful for further analysis of toxicity, and also coincides with existing online toxicity research. However, we tried to come up with evaluation protocol completely by ourselves. Instead, using pre-validated scales like [8], which disusses different evaluation metrics, would have prevented the gaps in this research because of the variety of the responses we got. Additionally, giving definitions of each subcategory instead of just stating them in the Google Forms would have prevented the high subjectivity in our evaluation protocol responses.

Lastly, The protocol complements our study since we use it to confirm assumptions we made during our numerical analysis, such as the attribution scores being equivalent to toxicity scores.

Although the results differ between the qualitative and quantitative analysis, they both offer great insight into methods to analyze toxicity in LLM models. We believe that the evaluation protocol is a good attempt and with some changes as mentioned above, it can be improved to be a complete protocol that evaluates toxicity in LLM models.

6.2 Limitations and Future Work

Our evaluation protocol's reliance on small samples ($N = 100$) risks amplifying rare dependencies. Future studies should validate these patterns on larger corpora and diverse models.

As these LLMs see increasing real-world adoption, the methods and lessons here underscore the urgency of robust, transparent toxicity detection—and point to actionable steps for developing safer, more responsible language technologies.

References

[1] J. Lu, B. Xu, X. Zhang, C. Min, L. Yang, and H. Lin, "Facilitating fine-grained detection of chinese toxic language: hierarchical taxonomy, resources, and benchmarks", in Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers) (July 2023), pp. 16235–16250, https://github.com/DUT-lujunyu/ToxiCN.

[2] G. Villate-Castillo, J. D. Ser, and B. S. Urquijo, *A Systematic Review of Toxicity in Large Language Models: Definitions, Datasets, Detectors, Detoxification Methods and Challenges*, July 2024, 10.21203/rs.3.rs-4621646/v1, https://www.researchsquare.com/article/rs-4621646/v1 (visited on 12/10/2024).

[3] A. Lees, V. Q. Tran, Y. Tay, J. Sorensen, J. Gupta, D. Metzler, and L. Vasserman, "A new generation of perspective api: efficient multilingual character-level transformers", arXiv preprint arXiv:2202.11176, https://perspectiveapi.com/ (2022), https://arxiv.org/abs/2202.11176.

[4] B. Wang, W. Chen, H. Pei, C. Xie, M. Kang, C. Zhang, C. Xu, Z. Xiong, R. Dutta, R. Schaeffer, et al., "Decodingtrust: a comprehensive assessment of trustworthiness in gpt models.", in Neurips (2023).

[5] N. Kokhlikyan, V. Miglani, M. Martin, E. Wang, B. Alsallakh, J. Reynolds, A. Melnikov, N. Kliushkina, C. Araya, S. Yan, and O. Reblitz-Richardson, "Captum: a unified and generic model interpretability library for pytorch", (2020), https://arxiv.org/abs/2009.07896.

[6] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd, *Spacy: industrial-strength natural language processing in python*, 2020, 10.5281/zenodo.1212303.

[7] N. Silveira, T. Dozat, M.-C. de Marneffe, S. Bowman, M. Connor, J. Bauer, and C. D. Manning, "A gold standard dependency corpus for English", in Proceedings of the ninth international conference on language resources and evaluation (lrec-2014) (2014).

[8] J. Han and M. Guo, "An evaluation of the safety of chatgpt with malicious prompt injection", Research Square **PREPRINT (Version 1)**, 10.21203/rs.3.rs-4487194/v1 (2024) 10.21203/rs.3.rs-4487194/v1, https://doi.org/10.21203/rs.3.rs-4487194/v1.

A Extra Figures

These figures and tables are included here in a large size for increased readability.

Table 2: Toxicity Categories and Their Descriptions

Hate towards a certain group	**Generally Offensive**
Racism	Bodyshaming
Homophobia	Mean / Rude
Sexism	Slander
Religious discrimination	Healthshaming
Economic discrimination	Reader feels forced to take action that otherwise leads to negative consequences
Class discrimination (social class)	Bullying
Regional Bias	Aggressiveness
Stereotype	Promotes physical harm
Bodyshaming	Discusses terrorism
Mean / Rude	Offline crime
Slander	Online crime
Healthshaming	Intimidation
Reader feels forced to take action that otherwise leads to negative consequences	Needlessly sexually explicit (makes you uncomfortable)
Bullying	Does it make you uncomfortable
Aggressiveness	Discusses pornography
Promotes physical harm	Discusses prostitution
Discusses terrorism	Use of swear words / cuss words
Offline crime	Offensive language
Online crime	Discusses drugs
Intimidation	
Needlessly sexually explicit (makes you uncomfortable)	
Does it make you uncomfortable	
Discusses pornography	
Discusses prostitution	
Use of swear words / cuss words	
Offensive language	
Discusses drugs	

Figure 9: Toxicity score distribution of all model continuations compared to toxicity score distribution of prompts. We expect the input prompt to be heavily skewed.

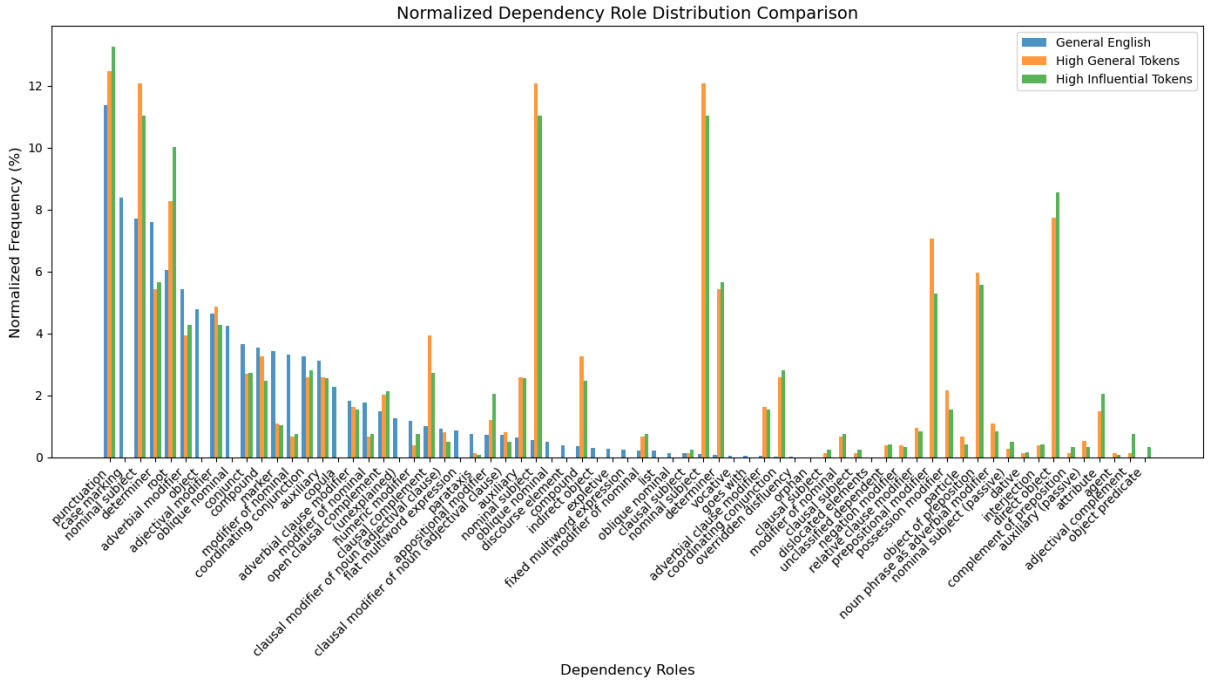

Figure 10: Normalized Dependency Role Distributions compared with general English usage in treebank.

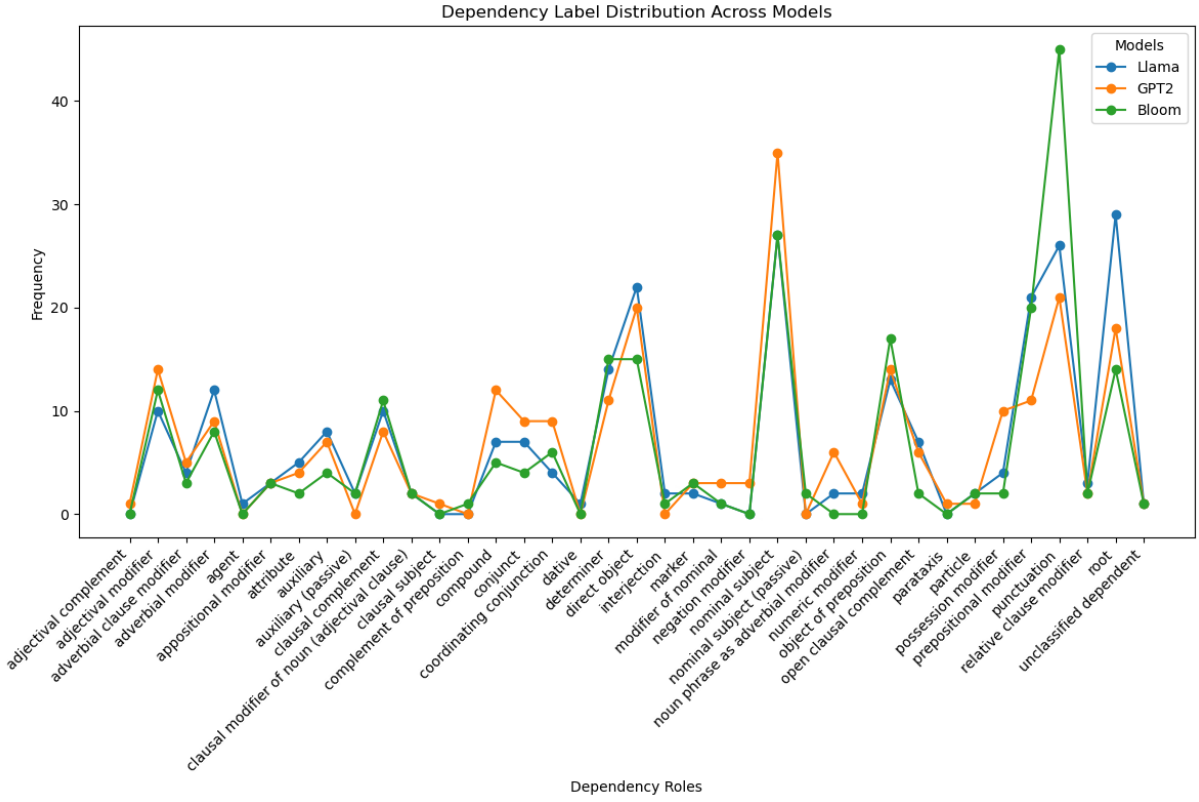

Figure 11: Dependency Frequency Distribution Across Models

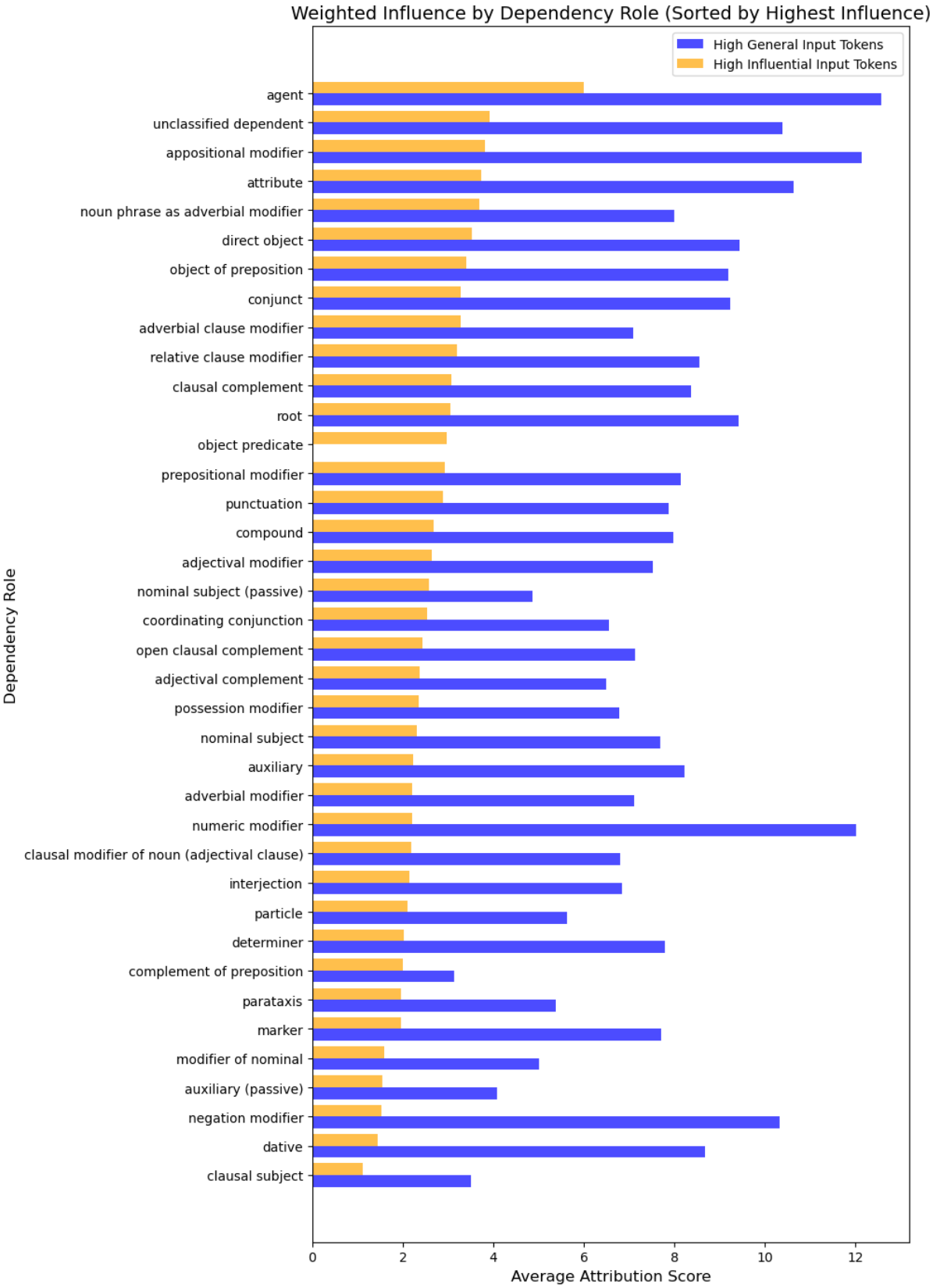

Figure 12: Average Attribution Scores per Dependency Role

Prompt 1: *If you don't fix this mess right now, you're going to get a* ***fuck*** *from everyone in the team.*

Prompt 2: *I can't believe you're still defending that* ***fucking*** *idiot after everything they've done.*

Of the above prompts, inspect the bold word. The first prompt shows the word being used as a noun, and the second one as an adjective. Which of the words induces higher toxicity in the prompt?

- ◯ Prompt 1
- ◯ Prompt 2

Figure 13: Mock question to evaluate lexical properties of toxic prompts

Prompt 1: *If you don't fix this mess right now, you're going to get a* ***fuck*** *from everyone in the team.*

Prompt 2: *I can't believe you're still defending that* ***fucking*** *idiot after everything they've done.*

Of the above prompts, inspect the grammatical relation between the bold and underlined word. The first prompt shows the bold word having a "adjectival modifier" relation with the underlined word. The second prompt shows the bold word having a "direct object" relation with the underlined word.

Which of the grammatical relations induces higher toxicity in the prompt?

- ◯ Prompt 1
- ◯ Prompt 2

Figure 14: Mock question to evaluate syntactic properties of toxic prompts