Detecting Harmful Instructions via Interpretable Feature Analysis: A Hybrid Approach

Serafima Patrikejeva
BA Computational Linguistics
sepat101@hhu.de

Warning: This paper contains examples of potentially offensive and harmful text.

Abstract

The use of large language model (LLM) assistants has expanded rapidly in recent years. Despite technological advances, ensuring their safety remains a major challenge, as these systems are exposed to vast amounts of unfiltered input. User prompts may elicit responses that are immoral, harmful, or covertly illegal. In this paper, we investigate such instances using targeted, manually annotated datasets of safe and unsafe prompt–response pairs. We further explore linguistically motivated hypotheses for distinguishing harmful advice, applying common natural language processing toolkits to analyze grammatical, lexical, and semantic patterns.

1 Introduction

Over the past five years, the use of chatbot services has expanded rapidly. Users now rely on AI assistance across a wide range of domains in their daily lives. While this trend demonstrates the growing integration of large language models (LLMs) into society, it also raises concerns about safety and reliability. To mitigate the risks of spreading misinformation or exposing users to harmful advice, LLMs require effective self-regulation mechanisms. A variety of strategies have been proposed, including rule-based filtering, reinforcement learning from human feedback, and prompt engineering. However, many existing approaches depend heavily on common-sense reasoning, which in turn requires external knowledge not always available to the models.

The goal of this project is to investigate how natural language processing (NLP) techniques can be applied to detect harmful content generated by LLMs without relying on external common-sense knowledge. To this end, we formulate a set of linguistically motivated hypotheses. The overall challenge is framed as a classification problem, with a focus on distinguishing safe and unsafe responses; the methodological approach is outlined in Section 3. Section 4 highlights the results of our findings.

2 Data and Resources

In this project, two datasets are considered: BeaverTails and Safe-RLHF-QA, both released by the PKU-Alignment team. While they are conceptually similar, each dataset exhibits distinct characteristics.

2.1 BeaverTails

The BeaverTails dataset comprises approximately 16,000 unique prompts, many of which are associated with multiple responses, yielding a total of around 330,000 prompt–response pairs. Each record contains a prompt, a response, and a boolean variable `is_safe` that indicates whether the pair involves potentially unsafe topics or harmful instructions. Furthermore, the dataset includes detailed annotations with up to 14 harm categories, allowing a single prompt–response pair to belong to multiple categories simultaneously.

Due to the significant overlap between the training and test splits in terms of prompts, the training split of BeaverTails was adopted as the primary dataset. Given the limited number of unique entries overall, the dataset is used for testing purposes only.

2.2 Safe_RLHF

The PKU-SafeRLHF-QA dataset extends BeaverTails and provides a large-scale benchmark for studying safety in instruction-following models. In total, it contains 265K question–answer pairs to corresponding 34,5K unique prompts. Each data point is annotated with three complementary labels:

a binary safety indicator (is_safe), one or more harm categories that specify the type of risk, and a severity level ranging from 0 to 3. The severity scale captures the degree of potential harm, from minimal impact (0) to severe consequences with broad societal implications (3).

To avoid duplication, overlapping entries originating from BeaverTails were removed, resulting in the exclusion of approximately 20% of the initial dataset.

2.3 Preparation steps

Manual inspection of both datasets revealed several recurring types of noisy instances, including responses that contained continuation instructions (presumably associated with other prompt–response pairs), emoji symbols, "end of message" markers, and personal information such as URLs, email addresses, and phone numbers. These artifacts were identified using regular expressions. Sensitive elements such as website links and personal data were subsequently anonymized through masking ([LINK], [PHONE], [EMAIL]).

Additionally, responses with extreme lengths were removed: those shorter than 70 characters or exceeding 2000 characters. This filtering step resulted in an average of 480 characters per response, with only about 1,000 entries being discarded.

To ensure consistency across datasets, the original harm categories were consolidated into seven broader domains:

- Violence and Threats,
- Crime and Illegal Activities,
- Hate Speech and Discrimination,
- Sexual Content,
- Privacy Violations,
- Misinformation and Manipulation, and
- Public Safety and Health.

All categories from both datasets were mapped to these domains, which were then incorporated into the preprocessing pipeline for each entry. The full mapping table can be found in the preprocessing.py module, among all other preparation steps.

After applying the preprocessing pipeline, the resulting datasets were stored in a HuggingFace repository for faster and more convenient access.

The BeaverTails dataset comprises 91,811 entries corresponding to 75,985 unique prompts, with 40,318 safe and 51,493 unsafe responses (approximately 44 % safe vs. 56 % unsafe). The average response length is around 373 tokens.

The Safe-RLHF dataset contains 203,405 entries across 168,835 unique prompts, with 107,124 safe and 96,281 unsafe responses (about 53 % safe vs. 47 % unsafe). Here, the responses are on average longer, at about 587 tokens.

3 Method

To develop a model for harm prediction, we restrict our focus to the linguistic properties of the statements. Instead of relying on contextual or commonsense knowledge, we formulate a set of linguistic hypotheses that serve as the initial basis for distinguishing between safe and harmful responses.

3.1 Proposed hypotheses

We hypothesize that harmful advice can be identified through a range of linguistic signals.

- From a lexical perspective, such responses are expected to frequent references to entities in sensitive domains (e.g., money, weapons, chemical substances) and the presence of obscene vocabulary.
- From a morphosyntactic perspective, harmful advice may involve a higher frequency of modal verbs, nominalizations, negation markers, and imperative constructions.
- Finally, we assume that the distribution of part-of-speech categories differs systematically between safe and unsafe statements.

To test the hypotheses, we implemented a pipeline that extracts the relevant features from each response in the corpora and quantifies potential harmfulness markers. For each marker, raw counts were transformed into ratio values, calculated separately for safe and unsafe responses across the corpus.

The majority of subtasks were addressed using the spaCy library. Certain phenomena required additional heuristics. For instance, identifying imperatives in English is particularly challenging from a purely morphological perspective. We therefore applied a simplified rule-based approximation, assuming that a sentence can be considered imperative if:

- it lacks a subject directly connected to the root, and

- its root dependency is a predicate of the VB form.

Obscene lexicon was detected with the `better_profanity` Python package.

For technical reasons related to computational constraints, each hypothesis was implemented as a separate function that can be executed independently. After feature extraction, results were aggregated at the sentence level with a median score for each class. In addition, the proportion of non-zero values was calculated to capture the distribution of responses containing the respective feature.

The resulting statistical observations are discussed in Section 4.

3.2 Word embeddings

For lexical representation, we employ Word2Vec embeddings trained directly on the combined training corpora of BeaverTails and Safe-RLHF. Each token is mapped into a 300-dimensional vector space, and sentence-level representations are constructed by aggregating token vectors through four complementary statistics: mean, standard deviation, minimum, and maximum. The resulting four vectors (each of length 300) are concatenated end-to-end, forming a single 1200-dimensional embedding for each text. This embedding is then further concatenated with the engineered linguistic feature vector of length 44, producing the complete input representation for the downstream classifier.

3.3 Model

For the classification task, we adopt LightGBM (Light Gradient Boosting Machine), a high-performance gradient boosting library developed by LightGBM is specifically designed for efficiency, scalability, and handling high-dimensional structured data. We selected LightGBM as the final classifier because it slightly outperformed baseline models in an accuracy scoring.

We first benchmarked a set of standard classifiers on our feature representations to establish baselines. Among the tested models, LinearSVC achieved the highest accuracy (0.70) with balanced precision (0.69) and recall (0.69), yielding an F1 score of 0.67. The MLP classifier performed similarly, with slightly lower accuracy (0.68) but higher precision (0.74), again converging to an F1 of 0.67. Logistic Regression matched this level of performance (F1 = 0.67) with accuracy 0.67, while SGDClassifier underperformed slightly, reaching F1 = 0.66 due to lower precision (0.61). A concise summary of the experimental results from all prior trials is presented in Table 1.

Model	**Acc.**	**Prcs.**	**Rcl.**	**F1**
MLP	0.68	0.74	0.68	0.67
LinearSVC	0.70	0.69	0.69	0.67
LogisticRegression	0.67	0.72	0.67	0.67
SGDClassifier	0.67	0.61	0.67	0.66

Table 1: Baseline classifier results on our feature representations.

Overall, the results indicate that linear models provide consistent, but limited, performance in this setting (F1 around 0.66–0.67). While simpler linear classifiers are efficient and interpretable, they struggled to fully capture the complex, non-linear interactions between the engineered features and the high-dimensional Word2Vec embeddings. LightGBM, by contrast, leverages gradient-boosted decision trees, which can naturally model such interactions while still being fast and memory-efficient.

Training is done with a binary objective and both `binary_logloss` and AUC as validation metrics. We train on the train split and validate on a test set, then score the validation probabilities, selecting the best iteration reported by LightGBM. To convert probabilities to labels, we sweep the decision threshold on a fixed grid (0.10. . . 0.90) using our `best_f1_threshold` routine and pick the threshold that maximizes F1; we also report AUC and accuracy at this threshold. The final artifact consists of the trained LightGBM booster and the selected decision threshold.

4 Results and Discussion

4.1 Features

Beyond the aggregate metrics, the summaries show some results for the previously defined hypotheses:

POS. Unsafe texts are slightly more action-oriented: they use a higher share of verbs and are more likely to contain numbers and proper names. Safe texts show a modest tilt toward connective/scaffolding tokens (e.g., subordinators) and a slightly higher proportion of auxiliaries. Overall effects are small but consistent, so POS works best as supporting context.

Named entities (NER). This is the clearest separator. Unsafe responses mention numbers and step markers (CARDINAL/ORDINAL) and refer to people or groups (PERSON/NORP) more often, fitting procedural or targeted instructions. Safe responses more often include dates, organizations, and legal references (DATE/ORG/LAW), reflecting policy-style framing and attribution.

Modality. Safe answers rely more on auxiliaries, which aligns with cautious, qualified phrasing in refusals or guidance. Classic modals appear a bit more frequently in unsafe texts by presence, but the magnitude is small. Net effect: useful but weaker than NER/negation.

Negation. One of the strongest "safe" markers. Safe responses contain more explicit negation ("cannot," "do not," "not allowed"), which captures refusal and risk-mitigation language. This signal is robust across both mean and presence metrics.

Imperative. Imperatives lean slightly toward the safe class when they occur as warnings or recommendations ("Do not. . . ", "Please consult. . . "). The gap is small, so imperatives help as a complementary cue rather than a primary discriminator.

Profanity. Very weak discriminative power. Presence is a touch higher in unsafe texts, but the effect size is tiny; on its own it won't move performance unless heavily weighted or combined with stronger signals.

Ratios. Generic ratios like the noun-to-verb ratio are effectively neutral. They add little beyond noise and can be dropped or kept as control variables without affecting the main picture.

Overall, the engineered features offer interpretable, concentrated signals—strongest for NER (numbers/people vs. dates/orgs/law) and negation, with POS and modality acting as supporting context. This matches the model's qualitative behavior (good at catching instruction-like unsafe content, conservative in refusal-style safe content) while also explaining why the net performance lift from features is modest: most individual effects are small, and the useful signal lives in a relatively narrow subset of indicators.

4.2 Classifier

The classifier was trained on 203,405 training instances and evaluated on a validation set of 91,811 instances. The underlying word representations were obtained using a Word2Vec model (vocabulary size 29,384, embedding dimension 300), aggregated into document vectors of size 1200. The final results are shown in Table 2. Quantitatively, the best-performing iteration achieved an AUC of 0.8182, F1-score of 0.7288 at threshold 0.690, and an overall accuracy of 0.7183. The class-wise evaluation shows a trade-off between precision and recall:

Model	Acc.	Prcs.	Rcl.	F1
LightGBM	0.72	0.75	0.72	0.72

Table 2: LightGBM classifier results.

- For the safe class (label 0), the model reached precision 0.849, recall 0.606, and F1-score 0.707 over 51,493 examples.
- For the unsafe class (label 1), the results were precision 0.631, recall 0.862, and F1-score 0.729 over 40,318 examples.

This indicates that the model is more recall-oriented for unsafe cases, capturing the majority of potentially harmful responses at the cost of producing more false alarms. The macro-average F1 (0.718) and weighted-average F1 (0.717) suggest balanced but slightly skewed performance due to class distribution.

Qualitatively, some borderline cases remain difficult for the classifier, especially where prompts contain ambiguous or indirect formulations of unsafe intent. In particular, neutral-looking requests that conceal harmful goals tend to be misclassified as safe, while unusually long but benign answers sometimes get flagged as unsafe. These cases highlight the challenges of distinguishing subtle pragmatic cues in natural language.

5 Challenges and Open Issues

Several challenges and limitations emerged throughout the project:

Data limitations. Although the dataset was relatively large, model performance indicates that additional high-quality annotated data could improve robustness. Some classes, especially those involving subtle unsafe intent, remain underrepresented. More balanced coverage across categories would likely lead to stronger generalization.

Feature engineering trade-offs. The engineered count- and ratio-based features (POS distributions, modality usage, profanity indicators, etc.) provided useful signals but only marginally improved performance when combined with embeddings. The main challenge here is that while these features capture surface-level linguistic patterns, they often lack the semantic depth required to distinguish borderline safe/unsafe cases. In hindsight, the feature extraction logic could be refined further to better align with the safety categories.

Model setup. The current pipeline relies on Word2Vec embeddings aggregated at the document level. While this representation is simple and computationally efficient, it loses context and word order information. More advanced architectures (e.g., transformer-based encoders or contextual embeddings) could likely yield stronger performance. Within the chosen setup, further hyperparameter tuning (embedding dimension, window size, LightGBM parameters, threshold selection) remains an open opportunity.

Abandoned approaches. Some potential directions (e.g., incorporating semantic similarity measures, experimenting with class-specific thresholds, or enriching the feature space with discourse-level markers) were considered but not fully pursued due to time constraints and complexity.

Overall, while the current setup demonstrates solid baseline performance, it also highlights the inherent difficulty of safety classification in natural language: subtle pragmatic cues and low-resource harm categories require richer representations and larger, more balanced datasets to achieve reliable detection.

6 Summary and Conclusion

The project set out to investigate whether a machine learning pipeline can reliably classify prompt–response pairs into safe vs. unsafe categories and, if unsafe, capture relevant linguistic patterns through engineered features. The experiments demonstrated that such an approach is indeed feasible: the final classifier reached an AUC of 0.82 and an F1-score of 0.73, showing that the combination of Word2Vec embeddings and engineered linguistic features yields a predictive potential.

With regard to the research questions, the findings suggest that it is possible to separate safe and unsafe responses with reasonable reliability using additional linguistic data; however, the answers are not fully reliable yet, as borderline and ambiguous cases remain challenging, and the engineered features only add a modest performance boost.

In retrospect, the chosen approach can be rated as a good baseline strategy: it provided interpretable results, highlighted strengths (high recall on unsafe content) and weaknesses (precision trade-offs, difficulty with subtle unsafe intent), and established a quantitative benchmark for future work. Overall, the work delivered useful first answers to the research question and outlined a clear path for further improvement.

References

Matthew Caron, Frederik Bäumer, and Oliver Müller. 2022. Towards automated moderation: Enabling toxic language detection with transfer learning and attention-based models. In *Proceedings of the 55th Hawaii International Conference on System Sciences*.

Ine Gevers, Ilia Markov, and Walter Daelemans. 2022. Linguistic analysis of toxic language on social media. In *Computational Linguistics in the Netherlands Journal 12 (2022), pp. 33-48*.

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. 2024. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems, vol. 36*.

Alex Mei, Anisha Kabir, Sharon Levy, et al. 2022. Mitigating covertly unsafe text within natural language systems. In *Findings of the Association for Computational Linguistics: EMNLP 2022, pp. 2914–2926*.

Nedjma Ousidhoum, Xinran Zhao, Tianqing Fang, Yangqiu Song, and Dit-Yan Yeung. 2021. Probing toxic content in large pre-trained language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pp. 4262–4274*.

Maribeth Rauh, John Mellor, Jonathan Uesato, Po-Sen Huang, Johannes Welbl, Laura Weidinger, et al. 2022. Characteristics of harmful text: Towards rigorous benchmarking of language models. *36th Conference on Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks*.