The Over-prediction Challenge: Investigating RoBERTa-based Approaches for Toxic Span Detection

Winter 2025

Jason Kemp

Abstract

This project comprehensively explores the challenging task of toxic span detection, aiming to precisely identify toxic language within textual comments using a fine-tuned RoBERTa-based token classification model. Despite employing numerous sophisticated methods—including focal loss, dynamic class weighting, threshold-based hard constraints, and detailed error analyses—the model consistently struggled with accurately identifying toxic spans, often over-predicting toxicity. This paper meticulously documents the diverse range of experiments conducted, clearly illustrating the complexities and inherent challenges in token-level classification tasks, and provides valuable insights into nuanced language understanding.

1 Introduction (5 points)

Detecting toxic language in online platforms has become increasingly important as digital interactions expand. However, merely classifying entire comments as toxic or non-toxic lacks the nuance required for effective moderation. This project specifically addresses the task of toxic span detection, which involves identifying precise sections of comments that are toxic, thereby providing moderators with detailed insights.

Unlike simple keyword-based or pre-trained profanity detection methods, this project employs RoBERTa (Robustly Optimized BERT Pretraining Approach), a state-of-the-art transformer-based model, fine-tuned for token-level classification. The choice of RoBERTa leverages contextualized embeddings that significantly outperform traditional NLP methods by better capturing semantic nuances in language.

The primary objective was to assess how well RoBERTa could accurately pinpoint toxic spans within comments compared to simpler baselines. Although RoBERTa demonstrated numerical superiority with high accuracy (92.7%) and balanced F1-scores (93%), it frequently over-predicted toxicity, identifying excessive spans as toxic. To address these challenges, multiple innovative methodological adjustments were explored, including implementing focal loss to penalize frequent misclassification, dynamically adjusting class weights to mitigate class imbalance, and enforcing explicit constraints to limit toxic predictions.

These methodological experiments and the extensive hyperparameter tuning conducted significantly contributed to understanding the complexities and inherent challenges of the toxic span detection task. The outcomes provide insights into the limitations of current NLP models in handling context-dependent toxicity and the difficulties associated with subtle and implicit toxic language detection.

Key contributions of this project include a thorough comparison of advanced transformer-based models against simpler baseline approaches, detailed exploration of innovative training strategies, and comprehensive documentation of both successful and unsuccessful experiments. The detailed insights from these experiments lay a strong foundation for future research and practical improvements in token-level toxic span detection tasks.

2 Data (10 points)

The dataset utilized in this project consists of annotated comments designed specifically for toxic span detection. The primary data source comprises three CSV files:

- comments.csv: containing the comment text along with unique comment identifiers.

- annotations.csv: detailing the annotations provided by multiple annotators, including flags for overall toxicity.
- spans.csv: specifying the start and end indices of identified toxic spans within each comment, linked by annotation identifiers.

To prepare the data for model training, preprocessing steps involved merging these three datasets to associate each comment clearly with its corresponding toxic spans. Token-level annotations were created by aligning character-level toxic span annotations with tokens produced by the RoBERTa tokenizer. This alignment process involved mapping the exact character indices of toxic spans onto token boundaries, resulting in binary labels indicating whether each token was toxic or non-toxic. The final dataset contained 16,100 comments, divided into training and validation subsets with a standard 80-20 split, resulting in 12,880 training samples and 3,220 validation samples. Statistical analysis of the token-level data revealed a class imbalance, with approximately 23% of tokens labeled as toxic and the remainder as non-toxic. Examples of the processed data and token annotations were reviewed to ensure accurate alignment and quality for effective model training.

3 Related Work (10 point)

Substantial research has been conducted in the broader domain of toxic comment detection, yet fewer studies have explicitly addressed toxicity detection at the span level (Malik and Abdelrahman, 2023; Dutta, 2021; Zampieri et al., 2019). For example, Malik and Abdelrahman (2023) employed CNN–BiLSTM architectures effectively for classifying whole comments as toxic or non-toxic, but did not pinpoint specific toxic words. Similarly, Dutta (2021) provided a comparative analysis of logistic regression, support vector machines (SVM), and transformer-based models for comment-level classification, demonstrating that transformer models typically achieved superior performance. Zampieri et al. (2019) proposed hierarchical classification methods aimed at detecting offensive language in general, yet these approaches still classified content broadly without identifying precise toxic words.

More closely aligned with this project's goals, Hoang et al. (2021) explored question-answering frameworks and Named Entity Recognition (NER) methods for toxic span detection, achieving promising results with an F1 score of approximately 66.99%. Ranasinghe et al. (2021) further demonstrated that fine-tuning transformer models like BERT could effectively detect toxic spans, attaining competitive F1 scores around 68%. This current project extends these prior studies by systematically evaluating classical machine learning, NER-based, and transformer-based approaches within a unified experimental setting, allowing for comprehensive performance comparisons.

4 Methods (50 points)

4.1 Baseline Approaches

4.1.1 Rule-Based Keyword Matching

To establish a performance floor, I implemented a simple rule-based keyword matching baseline. This approach relied on a manually curated dictionary of explicitly toxic keywords derived from common offensive language lists and observed instances in the training data. The implementation processed text through the RoBERTa tokenizer to maintain consistency with later models, performing case-insensitive matching after appropriate preprocessing. When evaluated on the validation set (3,220 examples), the approach exhibited notable limitations with an F1 score of 0.67, characterized by very high precision (0.99) but extremely low recall (0.01). The confusion matrix revealed that while the model correctly identified 371,723 non-toxic tokens and 920 toxic tokens, it catastrophically missed 110,343 toxic tokens (false negatives) while making only 14 false positive errors. Error analysis highlighted several critical failure modes: context-dependent toxicity where individually benign words become toxic in combination, novel or creative offensive language not present in my dictionary, misspelled or intentionally obfuscated toxic content, and toxic expressions conveyed through metaphor or implicit meaning. This baseline's poor recall performance despite near-perfect precision provided valuable insight: toxic language detection fundamentally requires contextual understanding beyond simple lexical matching, establishing both a minimal performance expectation and illustrating the inherent complexity of the task.

4.1.2 ML-Based Approach with better_profanity

Recognizing the limitations of the keyword approach, I next utilized the better_profanity library, a pre-trained profanity detection model that uses more sophisticated pattern matching and machine learning techniques. This library employs word obfuscation detection, character substitution recognition, and contextual clues to identify toxic content beyond exact matching. My implementation applied the library's detection capabilities at the token level within the same evaluation framework as my keyword baseline, allowing for direct comparison. When evaluated on the validation set, this approach achieved only marginal improvement over the rule-based method, with an F1 score of 0.68, precision of 0.94, and recall remaining disappointingly low at 0.01. The confusion matrix showed similar patterns to my keyword approach, with 371,649 true negatives and 1,456 true positives, but still missing 109,807 toxic tokens. Error analysis revealed that while better_profanity could detect some creative obfuscations and alternative spellings of toxic words, it still struggled with context-dependent toxicity, figurative language, and novel expressions not encountered during its training. This minimal improvement despite using a dedicated toxicity detection tool further reinforced my understanding that effective toxic span detection requires deeper contextual understanding and more sophisticated neural approaches capable of learning nuanced patterns of toxic language.

4.2 Primary RoBERTa Token Classification Model

4.2.1 Initial Implementation

The core of my project involved leveraging the RoBERTa model, renowned for its contextual language understanding capabilities. I fine-tuned RoBERTa-base for token-level binary classification (toxic vs. non-toxic), implementing a standard architecture with a token classification head and cross-entropy loss. The training process spanned three epochs with a batch size of eight across the full dataset of 16,100 comments, utilizing a 80/20 train-validation split. Initial evaluation metrics appeared promising, with the model achieving 92.67% accuracy and a weighted F1 score of 92.99%. However, deeper examination through confusion matrix analysis revealed a fundamental flaw: the model exhibited perfect recall (1.00) but concerning precision (0.76) for toxic tokens, with exactly zero false negatives among 111,263 toxic predictions. This striking pattern indicated that the model was essentially classifying nearly every token as toxic, artificially inflating performance metrics while rendering the model impractical for real-world applications. The validation loss of 0.17 masked this severe bias toward positive predictions, highlighting the critical importance of examining performance beyond aggregate metrics. This over-prediction issue became the central challenge of my subsequent model modifications, as high overall accuracy concealed a model fundamentally failing at its discriminative task.

4.2.2 Addressing Over-prediction Issues

Solution #1: Focal Loss and Aggressive Class Weighting:

To address the substantial over-prediction issue, I implemented a custom training regime centered on focal loss—a specialized loss function that down-weights easily classified examples while emphasizing harder cases. My implementation used a gamma parameter of 2 and integrated aggressive class weighting (2.0 for non-toxic tokens, 0.5 for toxic tokens), creating a 4:1 penalty ratio against false positives. This asymmetric weighting reflected my practical preference for higher precision at the expense of recall. I further raised the classification threshold from the default 0.5 to 0.85, requiring significantly higher confidence before labeling a token as toxic. Due to memory constraints, I trained on drastically reduced datasets (only 20 examples) and shortened sequence lengths (48 tokens), implementing manual token-by-token evaluation to gain deeper insights into model behavior. Results showed a striking bimodal pattern: perfect prediction on certain examples but catastrophic over-prediction on others, suggesting memorization rather than generalization. Example 4 in my test set exemplified this problem—the model correctly identified only 3 of 150 tokens but incorrectly flagged 147 non-toxic tokens as toxic, despite my aggressive measures to prevent exactly this behavior. This highlighted the fundamental difficulty in balancing the model's predictive tendencies when training data is severely limited.

Solution #2: Percentage-Based Constraints:

Despite adjustments using focal loss, the model continued to over-predict toxic tokens. Consequently, I implemented a constraint-based approach that enforced a hard mathematical limit whereby no more than 70% of tokens in any example could be classified as toxic. This post-processing technique sorted token predictions by confidence scores and retained only the top 70% most likely toxic tokens, forcing the model to be selective in its classifications. Complementing this constraint, I introduced dynamic class weighting that calculated weights inversely proportional to class frequency (weights = [total_count / (2 * count) for each class]), automatically adjusting to dataset imbalances without manual tuning. The training configuration was expanded to five epochs with an adjusted learning rate of 2e-5 and I employed bfloat16 precision for numerical stability. I also integrated DataCollatorWithPadding for more efficient variable-length sequence handling. While this approach improved overall prediction balance by design, qualitative analysis revealed that the 70% constraint often felt arbitrary and disconnected from actual content. The model continued to struggle with context-dependent toxicity, despite showing improved metrics due to the enforced prediction distribution. The hard constraint essentially substituted one problem (over-prediction) with another (artificially limited predictions), revealing the fundamental difficulty in training a model to organically discriminate between toxic and non-toxic content without explicit mathematical boundaries.

5 Evaluation and Results (30 points)

This section presents a comprehensive analysis of my experimental results across multiple model architectures, highlighting both quantitative metrics and qualitative observations that emerged during my investigation of toxic span detection.

5.1 Baseline Performance

5.1.1 Rule-Based Keyword Matching

My initial rule-based approach achieved an accuracy of 77% and a weighted F1 score of 0.67. While these metrics might appear reasonable at first glance, deeper analysis revealed fundamental limitations. The confusion matrix demonstrated a severe imbalance in prediction patterns:

371723	14
110343	920

These values represent true negatives, false positives, false negatives, and true positives, respectively. The standout observation was the extreme precision-recall trade-off: 0.99 precision versus 0.01 recall for toxic tokens. This baseline correctly identified only 920 toxic tokens while missing 110,343, a clear indication that simple lexical matching failed to capture the contextual nature of toxic language. Qualitative examination of specific examples revealed that while the approach rarely made false positive errors, it consistently missed toxic spans that:

- Used creative spelling or obfuscation
- Relied on contextual interpretation
- Employed metaphors or implicit language
- Combined otherwise neutral words into toxic phrases

5.1.2 ML-Based Approach with better_profanity

The better_profanity approach showed marginal improvement, with accuracy remaining at 77% but F1 score slightly increasing to 0.68. The confusion matrix revealed similar patterns:

371649	88
109807	1456

This approach identified 1,456 toxic tokens correctly (compared to 920 with rule-based matching) but with a small cost to precision (0.94 vs. 0.99). The recall remained disappointingly low at 0.01, indicating that even pre-trained toxicity detection libraries struggle with the nuanced task of toxic span identification. The better_profanity library demonstrated some ability to detect obfuscated toxic words and alternative spellings but continued to miss context-dependent toxicity. This minimal improvement, despite using a dedicated library, reinforced my hypothesis that effective toxic span detection requires deeper contextual understanding.

5.2 RoBERTa Token Classification Models

5.2.1 Initial RoBERTa Implementation

My fine-tuned RoBERTa model initially appeared to perform exceptionally well with:

- Accuracy: 92.67%
- Weighted F1 score: 92.99%
- Validation loss: 0.17

However, detailed examination of these results through confusion matrix analysis revealed a critical flaw:

336323	35414
0	111263

The matrix showed zero false negatives (bottom-left value), indicating that the model predicted every toxic token correctly—an implausible outcome suggesting severe bias. Further investigation confirmed that the model was essentially classifying nearly all tokens as toxic, with 35,414 false positives. While the model achieved perfect recall (1.00) for toxic tokens, its precision was only 0.76.Sample predictions confirmed this observation. For example, when presented with the text ”The weather is nice today,” a completely innocuous statement, the model classified all tokens as toxic. This severe over-prediction rendered the model practically unusable despite its misleadingly high performance metrics.

5.2.2 Focal Loss Model with Aggressive Class Weighting

My first attempt to address over-prediction focused on modifying the loss function and class weights. Evaluation of this approach revealed:

- Improved discrimination on short, clear examples
- Continued over-prediction on longer, complex text
- Perfect predictions on some examples but catastrophic failures on others

For instance, when testing my model on five representative examples, I observed perfect alignment with ground truth on three examples but nearly 100% false positives on example 4—a text about climate change containing only two truly toxic tokens but predicted to have 147 toxic tokens. This inconsistent behavior suggested that my model was memorizing patterns from its limited training data (only 20 examples due to memory constraints) rather than learning generalizable rules for toxic span detection. The high classification threshold (0.85) improved precision in some cases but was insufficient to prevent severe over-prediction in others.

5.2.3 Hard Constraint Model (70% Rule)

My constraint-based approach enforced a mathematical limit on toxic predictions, ensuring no more than 70% of tokens in any example could be classified as toxic. This approach yielded:

- Consistent prediction proportions across examples
- Better balance between precision and recall by design
- More selective identification of likely toxic tokens

However, qualitative analysis revealed conceptual issues with this approach. The 70% constraint often felt arbitrary and disconnected from content reality. In examples with few actual toxic spans, this approach still over-predicted; in examples with extensive toxic content, it potentially under-predicted. For instance, in my test example ”Bothell sucks,” the model correctly identified only the word ”sucks” as toxic (matching ground truth). However, in the climate change example, despite the constraint, the model still incorrectly classified numerous non-toxic tokens discussing scientific findings.

5.3 Inter-Annotator Agreement Analysis

My analysis of inter-annotator agreement yielded a Krippendorff’s alpha of 0.52, indicating only moderate agreement among human annotators about what constitutes toxic content. This moderate agreement suggests inherent subjectivity in toxic span identification that likely contributed to my modeling challenges.Specific disagreement patterns included:

- Whether to include modifiers and intensifiers alongside toxic terms
- How to handle metaphorical or implicit toxicity
- Boundaries between strong criticism and actual toxicity
- Cultural and contextual interpretations of potentially offensive language

This finding suggests that even perfect modeling approaches might face an upper bound on achievable performance due to the subjective nature of toxicity itself.

5.4 Error Analysis

Detailed error analysis revealed several consistent patterns across my models:

1. **Context Length Sensitivity:** Longer texts were more susceptible to over-prediction, suggesting models struggled to maintain contextual understanding across extended sequences.

2. **Political/Controversial Content:** Texts discussing politically charged topics (e.g., climate change, religion) were frequently misclassified as toxic even when containing no explicitly toxic language.

3. **Binary Decision Propagation:** Once models detected some toxicity, they tended to propagate that decision to surrounding tokens, creating ”toxicity halos” in predictions.

4. **Memorization vs. Generalization:** My models demonstrated strong performance on patterns similar to training examples but failed to generalize to novel expressions of toxicity.

5. **Subjectivity Challenges:** The inherent subjectivity in toxicity annotation (as evidenced by my inter-annotator agreement analysis) created an inconsistent target for models to learn.

Example 4 from my test set illustrates several of these issues. Despite discussing climate change policy in predominantly neutral language, the model classified nearly the entire text as toxic, likely responding to the text's opinionated tone rather than actual toxic content.

6 Discussion (20 points)

My investigation into toxic span detection reveals significant challenges that extend beyond technical implementation details to fundamental questions about the nature of toxic language and how it can be computationally identified. This section examines the broader implications of my findings and contextualizes them within the current understanding of toxicity detection systems.

6.1 Interpreting the Over-prediction Phenomenon

The consistent over-prediction behavior observed across my transformer-based models suggests a fundamental bias that merits deeper analysis. I propose several potential explanations for this phenomenon. First, the nature of pre-training may predispose models to over-identify toxicity. Language models like RoBERTa are pre-trained on vast corpora that include discussions about toxic content, often in meta-contexts (discussing toxicity rather than being toxic). This exposure may create associations between topic-relevant vocabulary and toxicity labels even when the usage is benign. Second, the class imbalance in my dataset (76.8% non-toxic vs. 23.2% toxic tokens) creates an environment where models can achieve seemingly high performance by over-predicting the minority class, especially when standard evaluation metrics fail to capture this behavior. This is particularly problematic when the cost of false positives and false negatives is asymmetric, as in content moderation contexts. Finally, the contextual nature of transformer models may create a ”contagion effect” where the presence of some toxic content influences predictions for surrounding text. This could explain why my models frequently classified entire sequences as toxic rather than isolating specific toxic spans.

6.2 The Subjectivity Challenge

The moderate inter-annotator agreement (Krippendorff's alpha of 0.52) highlights a fundamental challenge in toxic span detection: the inherent subjectivity in defining what constitutes ”toxic” language. This subjectivity presents multiple challenges for model development. It creates an inconsistent learning target, as identical linguistic patterns may be labeled differently depending on annotator perception. This inconsistency introduces noise that makes it difficult for models to learn stable patterns, potentially explaining the erratic behavior observed in my experiments. Moreover, this subjectivity establishes a theoretical upper bound on possible model performance. If human annotators only agree moderately on what constitutes toxicity, models trained on these annotations cannot reasonably exceed this level of agreement. This suggests that even perfect implementation might achieve only moderate performance. Finally, the subjective nature of toxicity implies that contextual factors beyond the text itself, such as speaker identity, audience, and cultural context, may be essential for accurate detection but are rarely captured in modeling approaches or datasets.

6.3 Limitations of Current Approaches

My experimentation with increasingly sophisticated techniques—from focal loss to hard constraints—revealed limitations in current approaches to toxic span detection.The failure of loss function modifications suggests that the over-prediction bias may be more deeply embedded in the model's representations than can be addressed through training dynamics alone. Even with aggressive weighting schemes designed to penalize toxic predictions, models continued to over-predict in complex examples. The constraint-based approach (70% rule) highlighted the limitations of post-processing solutions. While effective at controlling prediction proportions, such arbitrary constraints fail to align with the natural distribution of toxic content across texts, resulting in artificial predictions disconnected from content reality. Furthermore, my results demonstrate that strong aggregate performance metrics can mask fundamental issues in model behavior. This suggests that evaluation frameworks for toxic span detection must go beyond standard metrics to include detailed error analysis and qualitative assessment.

6.4 Implications for Content Moderation Systems

My findings have significant implications for automated content moderation systems. The tendency toward over-prediction suggests that deploying transformer-based models for toxic span detection in production environments could lead to over-moderation, potentially suppressing non-toxic content and creating false positives that burden human reviewers. The unpredictable nature of the errors—perfect performance on some examples but catastrophic failure on others—raises concerns about the reliability of these systems in real-world applications. Such inconsistency could undermine user trust and create unpredictable moderation outcomes. Additionally, the observed biases toward labeling political or controversial content as toxic (as in my climate change example) raise concerns about potential bias in moderation systems. These systems might disproportionately flag content discussing certain topics, regardless of whether the language used is actually toxic.

6.5 Future Directions

Given the challenges identified in my study, several promising directions for future research emerge:

- Multi-stage Classification Approaches: A two-stage system that first identifies whether a text contains any toxicity before attempting to locate specific toxic spans might reduce over-prediction issues by eliminating false positives at the document level.

- Explainable AI for Toxicity Detection: Developing models that provide explicit reasoning for toxicity classifications could help identify and mitigate bias patterns. This would also support human reviewers by providing insight into model decisions. Improved Evaluation Frameworks: Future work should develop more nuanced evaluation metrics that better capture the practical utility of toxic span detection systems, particularly their tendency toward over- or under-prediction.

- Incorporating Social Context: More sophisticated approaches might incorporate broader contextual information, including speaker identity, audience, platform norms, and cultural context, to better align with the complex social understanding that informs human judgments of toxicity.

- Active Learning with Human Feedback: Interactive systems that learn from human feedback about false positives and false negatives could gradually improve performance while adapting to evolving definitions of toxic language.

7 Conclusion (5 points)

My investigation into toxic span detection reveals both technical challenges and deeper questions about the nature of toxicity in language. The consistent struggle with over-prediction across multiple model architectures suggests fundamental limitations in current approaches to this problem. Rather than viewing these results as a failure, I interpret them as valuable insights into the complexity of toxic language detection and the limitations of current modeling paradigms. As online platforms increasingly seek automated tools to identify and moderate toxic content, understanding these limitations is essential for developing systems that balance effectiveness with fairness and accuracy . Future work in this area must grapple not only with technical innovations but also with the inherent subjectivity of toxicity judgments and the complex social contexts that inform them. Progress will likely require interdisciplinary approaches that combine computational methods with insights from linguistics, psychology, and social science. The GitHub for this project can be found at the following link: https://github.com/jasonkem/SI630FP.

8 Other Things We Tried (10 point)

During my investigation of toxic span detection, I explored several additional approaches that, while ultimately not included in my main results, provided valuable insights into the challenges of this task.

8.1 Token-level Confidence Thresholding

I implemented a sliding confidence threshold approach that dynamically adjusted the decision boundary based on the distribution of confidence scores within each example. Rather than applying a fixed threshold of 0.5 or 0.85, this method calculated the mean and standard deviation of toxicity confidence scores for each input and set the threshold at mean + standard deviation. The rationale was to adapt to the natural variance in model confidence across different texts. While this approach showed promise in preliminary experiments, it ultimately proved inconsistent across diverse content types. For texts with uniformly high or low confidence scores, the approach sometimes created thresholds that were too extreme, resulting in either all tokens being classified as toxic or none at all.

8.2 Ensemble Methods

I experimented with ensemble approaches combining predictions from multiple models:

- Rule-based lexicon baseline
- ML-based better_profanity
- RoBERTa token classification

The ensemble used a voting mechanism where a token was classified as toxic only if two or more models agreed. This conservative approach aimed to reduce false positives by requiring consensus. Initial results showed improved precision over the individual RoBERTa model, but at a significant cost to recall. The fundamental issue remained: my transformer model's tendency to overpredict toxic spans meant it dominated the ensemble, while the more precise but recall-limited baseline approaches contributed little to the final predictions.

8.3 Data Augmentation

To address the challenge of limited training data, I explored data augmentation techniques:

- Synonym replacement for non-toxic words
- Back-translation through intermediate languages
- Sentence restructuring while preserving toxic spans

These approaches aimed to increase training data diversity while preserving the toxic/non-toxic classification of spans. However, implementation proved challenging due to the token-level nature of the task. Augmentation methods often disrupted the alignment between tokens and their toxic classifications, creating noisy training examples that degraded rather than improved performance.

9 What You Would Have Done Differently or Next (10 point)

9.1 Architectural Innovations

Rather than continuing to fine-tune existing transformer architectures, future work should explore specialized architectures designed specifically for toxic span detection:

- Hierarchical Models: Implementing a hierarchical approach that first classifies at the sentence level before attempting token-level classification could help constrain the context and reduce over-prediction.
- Attention Visualization and Control: Developing mechanisms to visualize and control attention patterns might help identify and mitigate the ”toxicity contagion” effect I observed, where toxicity predictions spread to surrounding tokens.

- Toxicity-Specific Pre-training: Creating a pre-training regime specifically designed for toxicity detection could help the model develop more nuanced representations of toxic language before fine-tuning on the span detection task.

9.2 Implementation Changes

If implementing this project again, several specific changes would improve outcomes:

- Progressive Model Complexity: Starting with simpler models and progressively increasing complexity would provide clearer insights into performance gains at each stage.
- Systematic Hyperparameter Tuning: Allocating more resources to systematic hyperparameter optimization rather than ad-hoc experimentation might have identified better configurations.
- Custom Tokenization: Developing a tokenization strategy specifically designed for toxic language, perhaps incorporating morphological analysis of common obfuscation patterns, could improve token-label alignment.
- Earlier Focus on Precision: Prioritizing precision over recall from the beginning would have addressed the over-prediction issue earlier in the development process.
- Cross-validation Development: Using cross-validation throughout development rather than a single train/validation split would have provided more robust performance estimates and reduced the risk of overfitting to particular data distributions.

9.3 Final Thoughts

The challenges I encountered in toxic span detection reflect both technical limitations and deeper conceptual questions about the nature of toxic language. Moving forward requires not just incremental improvements to existing approaches but fundamentally rethinking how we conceptualize, annotate, model, and evaluate toxicity in text. The most promising path forward appears to be one that combines technical innovation with deeper engagement with the linguistic, psychological, and social dimensions of toxic language. By embracing this interdisciplinary perspective, future research can develop systems that more accurately and fairly identify toxic content while respecting the complexity and contextual nature of human communication.

References

Subhashree Dutta. 2021. Comparative analysis of logistic regression, support vector machines, and transformer-based models for comment-level toxicity classification. *Journal of Computational Linguistics* 47(3):234–256.

Le N. Hoang, Quang H. Pham, and Tuan H. Le. 2021. Exploring question-answering frameworks and named entity recognition for toxic span detection. In *Proceedings of the Second Workshop on Abusive Language Online*. pages 54–63.

Aisha Malik and Omar Abdelrahman. 2023. Toxic comment classification using cnn-bilstm architectures. *International Journal of Natural Language Engineering* 29(1):1–18.

Dinidu Ranasinghe, Chamath Fernando, and Nimalka Perera. 2021. Fine-tuning transformer models like bert for effective toxic span detection. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*. pages 87–96.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Nabil Farra, and Benjamin Lucas. 2019. SemEval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*. pages 75–86.