National College of Ireland

Multilingual Toxicity Detection with Enhanced Balancing and Contextual Learning

MSc Research Project
MSc Data Analytics

Jacob Saju
Student ID: x23166363

School of Computing
National College of Ireland

Supervisor: Hamilton Niculescu

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: …….Jacob Saju……

Student ID: ……x23166363……..……

Programme: …MSc Data Analytics…… **Year:** …2024-2025..

Module: ……MSc Research Project……..……

Supervisor: ……Hamilton Niculescu……..……

Submission Due Date: ……26/05/2025……..……

Project Title: …Multilingual Toxicity Detection with Enhanced Balancing and Contextual Learning……..……

Word Count: ……10719…… **Page Count**……33……..……..

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……Jacob Saju……

Date: ……23/05/2025……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

MultiToxiGuard: A Culturally Aware Multilingual Toxicity Detection System with Hierarchical Sampling and Confidence Calibration

Jacob Saju
x23166363

Abstract

Online toxicity detection systems struggle immensely in scaling across multiple, diverse linguistic and cultural environments, frequently privileging high-resource languages and offering poor protection to low-resource languages speakers. This work presents MultiToxiGuard, a multilingual toxicity detection system that solves these problems using three new components: a Smart Balancing Module using hierarchical sampling and dynamic weighting, a Contextual Enhancement Layer leveraging cultural embeddings for enhanced semantic awareness, and a Confidence Estimation System that includes robust uncertainty estimation. Utilizing a dataset of 15 languages from 9 language families, rigorous data augmentation processes are implemented that greatly enhanced representation of low-resource languages (Japanese +1518%, Vietnamese +1208%). Results of the validation indicate high overall performance (F1=0.7944, accuracy=0.8278) with impressive uniformity spanning linguistic boundaries, and having a cultural fairness score of 0.96. Specifically, a few low-resource languages (Estonian, Swahili) performed better than medium-resource languages, highlighting the efficacy of these balancing techniques. Whereas performance objectives of F1 (≥0.88) and the rate of false positives (≤0.03) are still daunting, MultiToxiGuard is a major step forward in fair content moderation that closes the high to low-resource languages' performance gap, a sore problem of past techniques. This system presents a single, integrated framework for detection of toxicity which performs at a consistent rate without the need for distinct models per language, markedly improving the best available multilingual content moderation technologies.

1 Introduction

The swift growth of user-generated content on global online platforms has elevated cross-cultural interactions, creating engagement across community and Knowledge sharing. This growth, however, has been accompanied by a concerning increase in toxic and abusive postings like harassment, hateful speech, and threatening content (Abbasi et al., 2022; Taleb et al., 2022). Although automated systems of toxicity detection have become a necessary tool to preserve the health of online environments, they encounter serious problems when implemented across the linguistic and cultural diversity of the contemporary internet.

Existing toxicity detection methodologies highly prefer high-resource languages, resulting in wide disparities in content moderation quality across linguistic boundaries (Conneau et al.,

2019; Bogoradnikova et al., 2021). This linguistic imbalance serves to compromise platform security for low-resource language speakers and is not suited to capture the subtler cultural environments in which the toxic content arises. As Shrestha et al. (2023) showed, even models that are high performing, with F1-scores greater than 0.94, deteriorate significantly when faced with new social media data of different linguistic origins, calling for more culturally attuned and flexible solutions.

There exist a variety of serious challenges that hamper effective multilingual toxicity detection. Firstly, the intrinsic class imbalance of toxic and neutral content generates biased training gradients, where toxic content generally accounts for a minor part of total data (Priya et al., 2023). Secondly, models in the absence of carefully designed context-aware mechanisms struggle to capture culturally specific expressions, colloquial sayings, and regional indicators that tend to mark abusive content (Chan & Li, 2024). Finally, the limited availability of annotated data in low-resource languages leads to persistent performance differences, causing disproportionate safety coverage to linguistically diverse user populations (Goyal et al., 2020).

While the frameworks currently in use seek to tackle these problems using translation-based methods, they inject more inconsistencies and tend to discard crucial cultural context (Malik et al., 2021). End-to-end multilingual models have more potential but are exceedingly sensitive to data-balancing and cross-lingual transfer methods. Additionally, traditional balance techniques like the use of SMOTE (Synthetic Minority Oversampling Technique) have proved inconsistent when applied to the naturally diverse processes of online toxicity.

This research seeks to overcome these shortcomings by examining the following research question:
How can an integrated multilingual toxicity detection system featuring hierarchical sampling, cultural-context embeddings, and confidence calibration mitigate class imbalance and achieve reliable performance across diverse languages, including those with limited resources?

The study presents a new method based on the XLM-RoBERTa-XL architecture, comprising three innovations: (1) a Smart Balancing Module that employs hierarchical sampling processes with dynamic adaptation towards representation of the language and frequency of toxicity; (2) a Contextual Enhancement Layer that employs cultural embeddings and pattern-based enhancement of semantic comprehension; and (3) a robust Confidence Estimation system to ensure high-risk content detection at low rates of false positives.

The research contributes to the scientific literature in the following ways:

1. One framework that enhances the current practices by having dynamic balancing processes incorporated that are directly suitable to multilingual contexts.
2. New culture context embeddings that are especially designed to encode the nuance of language and culture differences in toxic language.

3. A full evaluation framework that measures technical performance and cultural sensitivity both in 55 languages, with new benchmarks in multilingual toxicity detection.

The rest of the paper is structured as follows: Section 2 overviews the pertinent literature concerning multilingual toxicity detection, class imbalance techniques, and contextual comprehension; Section 3 presents the methodology of the research, including the dataset properties and the metrics of the evaluation; Section 4 presents the proposed architecture's design specification; Section 5 outlines the implementation details; Section 6 reports the results of the evaluation and the discussion; and Section 7 concludes by summarizing findings and directions for future research.

2 Related Work

This work positions the present research in the context of the wider academic literature on multilingual toxicity detection. It critically analyzes the major developments in multilingual methodology, class imbalance techniques, contextual awareness mechanisms, and low-resource language deployment challenges. It reviews the drawbacks and benefits of the available approaches and attempts to highlight the gaps that motivate the proposed research.

2.1 Current Approaches in Multilingual Toxicity Detection

The building block of present multilingual toxicity detection models relies heavily on transformer models and cross-lingual representation learning. Conneau et al. (2019) launched XLM-RoBERTa (XLM-R), a system that showed remarkable progress in cross-lingual comprehension using unsupervised learning at scale. Through learning from 2.5TB cleaned CommonCrawl data in 100 languages, XLM-R attained the current best results in the task of classification, achieving a 14.6% average accuracy improvement on the XNLI benchmark against multilingual BERT. The full potential of their method lies in tackling what they called the "curse of multilinguality" by having more model capacity, enabling efficient sharing of parameters across languages. Nevertheless, in addition to its remarkable performance, the XLM-R was not directly tuned for toxicity detection work and does not support handling class imbalance problem present in toxic content datasets.

Developing along cross-lingual paradigms, Bogoradnikova et al. (2021) focused on toxicity detection in the Russian language and explored the role of morphological features and language-specific characteristics in influencing detection quality. They found that while multilingual embedding models transfer knowledge quite efficiently across languages, they do not always label toxicity equally in different linguistic settings. Their method integrated conventional topic modeling techniques such as LDA-Mallet with domain adaptation models and resulted in a toxic span detection with an F1-score of 0.73. Their major drawback was having a fairly small dataset size and a single-language family, raising concerns regarding generalizability to more heterogeneous sets of languages.

Pal and Rai (2023) also improved multilingual toxicity detection by comparing deep learning versus traditional machine learning techniques on a set of 153,164 tweets. Their LSTM implementation resulted in a remarkable 90.7% accuracy at F1-score of 0.94. What is great about their method is the exhaustive comparison of different model architectures, while their generally uniform data distribution did not handle real-world situations in which toxic content occurs at significantly lower frequencies than non-toxic content.

Malik et al. (2021) provided useful insights through their comparative study of word embeddings, which proved that the choice of careful embedding heavily influences the model performance. In a comparison of BERT, fastText, and conventional embeddings using deep neural networks, they proved that CNN-based models using suitable embeddings outperformed conventional techniques. Their methodological power rests in the embedding analysis in detail, although the study was restricted insofar as it dealt largely with English content, and minimal cross-lingual testing.

2.2 Balancing Strategies for Class Imbalance

Class imbalance is a serious problem in toxicity detection, in which toxic examples generally account for a minor portion of total data. Priya et al. (2023), in a direct engagement of this problem, analysed multi-label toxicity detection techniques in their work on the Jigsaw Toxic Comment Classification Challenge dataset, illustrating how numerous machine learning models behave under imbalanced conditions. Their comparative analysis showed Linear SVC had the best accuracy in identifying toxic texts in multi-label contexts. Their method's power rests in its identification of numerous, interacting categories of toxicity, although their study was constrained by comparatively simple balancing strategies that will not necessarily generalize to highly imbalanced multilingual datasets.

Machová et al. (2022) presented a novel hybrid method of using a blend of lexicon-based and machine learning techniques to detect levels of toxicity. This stratified method of approaching toxicity issue having a grading system of content in several levels of severity in place of binary labels offers more nuanced structure for content moderation. By creating a Slovak language toxicity lexicon of 809 words in three levels of toxicity, they established the groundwork for automated labeling. This SVM implementation had an 80% correct classification rate on all of the classes, and they performed quite strongly on the detection of highly toxic content (specificity of 0.950). Their work's primary limitation is its language-specificity and difficulty in scaling the lexicon method to languages that have more intricate morphological systems.

Taleb et al. (2022) addressed the problem of class imbalance in a way characterized by advanced data pre-processing techniques. In the detection of toxic content on social media, "Down-sampling Majority Class" technique is used to balance the dataset so that both toxic and non-toxic classes had the same distribution. These deep learning models specifically, LSTM using GloVe embeddings, attained a high F1-score of 0.94 in identifying toxic languages. Although this balancing method was effectively working on the given dataset, it also poses questions concerning data loss when it is applied in multilingual environments where specific languages might already have a shortage of data.

2.3 Contextual Understanding and Cultural Calibration

The identification of toxic material in various cultural and linguistic environments needs advanced contextual knowledge mechanisms. Chan and Li (2024) contributed importantly to this topic by their "Specialis Revelio" pre-processing module, specially constructed for the purpose of exposing hidden toxicity which conventional detection techniques tend to miss. It covered seven of the major text manipulation strategies applied to evade content moderation, namely, slang, Leetspeak, and deliberate misspellings. Using GPT-3 and custom algorithms to correct misspelling and modify word boundaries effectively, they detect performance of current APIs such as Perspective API is greatly enhanced, increasing probabilities of toxicity detection in their experiments from up to 0.21 to 0.80. The strength of their work is that they directly counter evasion techniques, although costs of implementation and processing overhead might constrain scalability in real-time scenarios.

Abbasi et al. (2022) directly explored religious and continent-based toxic content detection and showed how cultural context plays a role in detection efficacy. Their study on multilabel religious toxic comment classification showed that the CNN model using GloVe word embedding had the best accuracy of 95.24%, far better than other models. Their study also shows the necessity of specialized detection of content that is culturally sensitive, although the use of different models per type of content might pose integration problems in extensive moderation systems.

Aquino et al. (2021) investigated the use of text and emojis in toxic content detection, mirroring the multimodal character of contemporary communication. This two-stream processing method processes the text and emoji elements individually before being combined in a single vector representation that is then classified. Each augmentation method employs quality checking through introducing similarity checks to confirm that produced text is semantically meaningful with extra useful linguistic variation.

Sarker et al. (2023) also introduced ToxiSpanSE, which is an answer to the toxic span detection task in software engineering communication. Their fine-tuned RoBERTa model achieved an impressive F1-score of 0.88 to identify specific toxic segments in text. The power of this technique lies in the explainability it offers in referencing exact phrases causing toxicity prediction, with its domain-specific nature limiting applicability to general multilingual settings. Their manual misclassification analysis determined that the highest category of their errors were their false positives (65.35% of their errors) and this illustrates just how difficult it is to accurately identify toxic spans even with cutting-edge models.

2.4 Challenges in Low-Resource Language Implementation

Efficient detection of toxicity in lower-resourced languages is especially a challenge. Shrestha et al. (2023) met this by training models to detect both toxic language and threats in Swedish, a much lower-resourced language. Using transfer learning with the BERT, they achieved F1-scores of over 0.94 in recognizing toxic language detection and 0.86 in threat detection. These models, however, showed serious degradation in performance when tested on unseen social media data, where they misclassified 40% of threats, indicating the generalization problem in real-world deployments. This study indicated that 20% of the threats detected contained no toxic content, highlighting the intricacies of harmful content detection beyond a mere measure of toxicity.

R et al. (2023) compared a wide variety of word embeddings for toxicity detection in detail, specifically on the Jigsaw dataset. It showed that CNN models using GloVe embeddings were more accurate at 96.59%, compared to other types of embedding techniques. Although this method is useful in the choice of the appropriate embedding, it dealt mainly with high-resource languages and not the problem of embedding quality in low-resource languages, where pre-trained embeddings might not have enough coverage or quality.

Suresh et al. (2023) analyzed the performance of toxicity detection through different machine learning models and mentioned the difficulty of obtaining labeled data for the purpose of training—a problem worsened in low-resource languages. Their research insisted on the fact that the subtleties and context-specificity of toxicity create a nuance that even advanced models find hard to capture consistently. They present problems of data imbalance and linguistic subtleties, which become far more amplified in multilingual environments operating under constrained resources.

The work of Goyal et al., cited in several papers, proved that greater transformer capacity could considerably enhance cross-lingual performance, specifically for low-resource languages. This indicates that architectural changes and optimization of parameters can alleviate some of the

problems in low-resource language toxicity detection, although the computational cost of the approach might restrict practical use in many instances.

2.5 Research Gaps and Justification

Existing literature is examined to show a number of crucial gaps in multilingual toxicity detection. Although tremendous progress has been achieved in cross-lingual representation learning and different facets of toxicity detection, the current methods falter in being able to achieve consistent high levels of performance across a wide linguistic and cultural diversity, particularly in low-resource languages. Class imbalance issues are not sufficiently addressed in multilingual environments, and the majority of balancing procedures are tailored to monolingual environments. In addition, current approaches often do not contain a fully integrated treatment of hierarchy sampling, contextual enhancement, and confidence calibration in one general framework.

Most models exhibit a "performance cliff" in transitioning from high-resource to low-resource languages, and it is questionable that boosting model capacity or data volume alone will overcome the intrinsic problems. Context and cultural nuances of toxicity, which are highly diverse across languages and societies, tend to remain secondary rather than primary design considerations. Most available systems also provide limited explainability, and it is hard for the human moderators to know and endorse the decisions of the models, a crucial need for content moderation in a sensitive context.

Such gaps warrant the necessity of the proposed integrated multilingual toxicity detection system, which directly targets class imbalance using the mechanisms of hierarchical sampling while integrating cultural-context embeddings and confidence calibration. This system would seek reliable performance on a wide variety of languages, including low-resource languages, while being explainable and adaptable to changing linguistic conventions. This proposed research fills the gaps above by creating a complete framework that draws the best of the available approaches and addresses their shortcomings in a systematic manner.

3 Research Methodology

This study adopted a systematic methodology to deal with the problem of multilingual toxicity detection, keeping in mind the need to ensure similar performance on a variety of languages while solving class imbalance and cultural sensitivity problems. It includes data gathering and preparation, architecture design, implementation, training, and testing stages, all of which were specifically designed in support of the study purposes.

3.1 Data Collection and Preparation

3.1.1 Dataset Selection and Analysis

The main dataset used in this study was the FredZhang7/toxi-text-3M dataset (FredZhang7, 2023), having approximately 3 million text samples in 55 languages, all of which were created by humans, not machine-translated texts. This dataset was selected due to its extensive linguistic diversity and a wide variety of toxic content types, such as hate speech, harassment, threats, insulting texts, sexting, and other abusive content types.
Analysis of the initial dataset showed high imbalances in both the language representation and the distribution of the class:

Total samples	Toxic samples	Non-toxic samples
2,880,667	416,529 (14.46%)	2,464,138 (85.54%)

Language distribution analysis revealed a wide imbalance where English comprised a disproportionate 87.8% of the total samples (2,528,002 samples), and the other 54 languages accounted for a mere 12.2%. Languages were classified based upon the volume of samples.:

- High-resource: English (2,528,002 samples)
- Medium-resource: 10 languages : Turkish, Arabic, Portuguese, Spanish, Russian, and others (total of 280,332 samples)
- Low-resource: 45 languages with fewer than 10,000 samples each (total of 72,333 samples)

Class imbalance also varied widely from language to language from English's 11.73% toxicity to Arabic's 66.44%, which created enormous challenges in terms of balancing learning.

3.1.2 Language Selection Strategy

In order to enable efficient computing with linguistic diversification, a language selection method was implemented by adopting the technique of Goyal et al. (2020). The method selected top 15 languages as per:

1. Resource level representation (high/medium/low)
2. Linguistic family diversity
3. Toxicity distribution
4. Minimum viability threshold (not fewer than 20 toxic samples)
5. Technical compatibility with embedding models

The final choice comprised:

- High-resource: English
- Medium-resource: Turkish, Arabic, Portuguese, Spanish, Russian, Indonesian, Greek
- Low-resource: Hindi, Estonian, Thai, Swahili, Croatian, Vietnamese, Japanese

This sample comprised 9 language families (Germanic, Romance, Slavic, Semitic, Turkic, Austronesian, Hellenic, Indo-Aryan, Uralic, Tai-Kadai, Niger-Congo, Austroasiatic, and Japonic), spanning wide linguistic diversity.

3.1.3 Data Augmentation and Balancing

Using the method of Priya et al. (2023), class imbalance was overcome by a multi-stage data augmentation process:

1. **Target Calculation**: For each language-class combination, targets were calculated using a tiered approach:
    - Classes with >5,000 samples: No augmentation
    - Classes with 500-5,000 samples: Target 5,000 samples
    - Classes with <500 samples: Target 2,500 samples
2. **Semantic-Preserving Augmentation**: In line with the techniques by Chan and Li (2024), four complementary augmentations were applied:
   a. **Word Substitution**: Substitution, deletion, or repetition of words at a random 15% rate in order to preserve semantic content and create linguistic variation.
   b. **Back-Translation**: Translate the text to pivot languages (English, French, German) and back utilizing the NLLB-200 model (Meta AI, 2022), as validated by Taleb et al. (2022).

c. **Embedding-Based Hybrid Generation**: Employ FastText embeddings to search semantically similar texts in the range of 0.65 to 0.95 and generate hybrid samples through fragment blending, building upon Malik et al. (2021).
d. **Direct Translation**: Create further samples for under-represented languages (Japanese, Vietnamese) by translating from high-resource languages while keeping class ratios.

3. **Quality Control**: All the augmented samples were verified for semantic similarity (compared with the original samples) in order to preserve the toxicity properties. Samples with similarity measures outside the range of 0.65-0.95 were excluded in order to preserve quality.
4. **Final Balancing**: Downsampled the overrepresented classes to produce the final balancing dataset, specifically capping English and Turkish non-toxic samples at a maximum of 15,000 entries each, adopting similar practices by Machová et al. (2022).

The data augmentation boosted the dataset size from 271,539 to 321,161 samples, achieving robust growth in low-resource languages (e.g., Japanese +1518%, Vietnamese +1208%, Croatian +402%).

A stratified train/validation/test partition of 80%/10%/10% was applied, ensuring distribution of languages and toxicity and also resulting in:

- Training set: 222,295 samples
- Validation set: 27,787 samples
- Test set: 27,787 samples

3.2 Model Architecture Design

The model architecture was inspired by the work of Conneau et al. (2019) and Bogoradnikova et al. (2021), and new extensions to overcome the specific needs of multilingual toxicity detection have also been added.

3.2.1 Base Model Selection

XLM-RoBERTa-Large has been selected to be the base model because of its established cross-lingual transfer capability within 100 languages. This is motivated by comparative research by Conneau et al. (2019) and Pal and Rai (2023) that established the outstanding performance of the model in multi-lingual conditions.

3.2.2 Custom Architecture Components

1. **Smart Balancing Module**: Based on the research of Priya et al. (2023) this features assists in balancing a dataset
    - Language-balanced sampling: Factored random sampling with specific focus on under-represented languages
    - Language-aware loss function: Unique loss function type that scale the contributions, according to prior performance flexibly
2. **Contextual Enhancement Layer**: This aspect takes from the context research by Abbasi et al. (2022) and Chan and Li (2024) and enhances semantic understanding by:
    - Language embeddings: Separate embeddings by language
    - Cultural context projection: Combine base representations to language embeddings
    - Context-aware self-attention: multihead attention mechanisms preserve cultural relevant patterns

3. **Confidence Estimation System**: In reaction to the issues of explainability by Sarker et al. in 2023, this system dirives reliable measures by:
    - Confidence network: Expert prediction-based confidence scores
    - Language-specific calibration: Parameters specific to the corresponding language with respect to credibility
    - Calibration loss: Supplementary training goal to match confidence with accuracy

3.3 Training Procedure

3.3.1 Hardware and Software Environment

The following setting has been used in the training:

- Hardware: NVIDIA A100-SXM4-40GB GPU
- Deep learning framework: PyTorch 2.6.0 with CUDA 12.4
- Libraries: Hugging Face Transformers, FastText, NLLB-200
- Storage: Google Drive for dataset and checkpoint management

3.3.2 Training Protocol

The following arrangement was used in the training through a three-period schedule:

- Initial learning rate: 2e-6 with 30% decay per epoch
- Optimizer: AdamW with weight decay 0.01
- Batch size: 8 with gradient accumulation steps of 4 (effective batch size 32)
- Scheduler: Linear with 10% warmup ratio
- Gradient clipping: Max norm 0.2
- Early stopping: Patience of 2 epochs tracked on validation F1 score
- Training time: Around 5 hours per epoch on A100 GPU

3.3.3 Stability Safeguards

Various stability measures were implemented, considering recommendations by Goyal et al. (2020):

- NaN detection and handling for weights and gradients
- Conservative initialization strategies for custom modules
- Numerical stability enhancements for loss calculation
- Regular validation checks to identify the degradation of performance

3.4 Evaluation Methodology

3.4.1 General Performance Metrics

The following main evaluation measures are included with respect to toxicity detection protocol (Taleb et al., 2022; Sarker et al., 2023):

- Accuracy: Total proportion of correct predictions
- Precision: Ratio of predicted toxic samples that are actually toxic
- Recall: Ratio of actual toxic samples accurately classified
- F1-score: Harmonic mean of precision and recall
- AUC: Area under the ROC curve
- FPR: False positive rate (vital when content moderation is involved)

3.4.2 Cross-Lingual Performance Assessment

Below metrics specific to language were calculated to compare performances across languages through Shrestha et al. (2023) method:

- Per-language performance measures (accuracy, precision, recall, F1, FPR)
- Weighted average metrics that compensate for sample volumes
- Target achievement tracking (targets: $F1 \geq 0.88$, $FPR \leq 0.03$)
- Performance pattern analysis across language families and resource levels

3.4.3 Cultural Fairness Evaluation

The following cultural fairness evaluation structure was applied, considering the cultural sensitivity study of Abbasi et al. (2022):

- Cultural group mapping from languages to larger cultural categories
- Comparison of metrics across cultural groups
- Variance-based fairness scoring:
    - Accuracy fairness: $1/(1+10\times Var(accuracies))$
    - FPR fairness: $1/(1+50\times Var(FPRs))$
    - Overall fairness: $0.6\times accuracy\_fairness + 0.4\times FPR\_fairness$

3.4.4 Statistical Analysis

The data of evaluation was statistically analyzed through:

- Thorough error analysis using Confusion matrix.
- Variance analysis to calculate consistency across languages.
- Weighted averaging to account for differences in sample size.
  Detection of language-based elements, which affects performance, through group performance analysis of language

3.5 Ethical Considerations

This research was implemented with several ethical considerations, following the ethical structure provided by Shrestha et al. (2023) for content moderation applications:

1. **Bias Mitigation**: Equally represents culture and language group.
2. **False Positive Concerns**: Monitors precisely and tunes false positives in to eliminate over-censorship
3. **Transparency**: Predicts confidence to allow manual review of decisions made by machine
4. **Fairness Monitoring**: Initiates cultural fairness to identify and rectify system-level differences

All levels of research methodology were imbued with these ethical concerns through a series of practical applications. Bias mitigation came through the utilisation of the Smart Balancing Module and hierarchal sampling that ensured proportionate representation of all language groups and the data augmentation process, specifically aimed at under-represented languages (with a 1518% uplift in the case of Japanese and a 1208% uplift in the case of Vietnamese). False positives were managed by the calibration loss component within the Confidence Estimation System, that was designed to discourage overconfident predictions that have the potential to cause overcensorship. Increased transparency was realized by introducing the fine-grained confidence scoring mechanism that reveals reliability measures to human moderators at each prediction, enabling prioritized reviewing of borderline examples. Fairness monitoring was systematically integrated with the Cultural Fairness Evaluation framework that computed variance-based fairness scores across linguistic frontiers and culture groups to ensure consistent system performance (with a resultant overall fairness

score of 0.96) irrespective of the linguistic or cultural framework. Such implementations made ethical considerations not just hypothetical but operationalized at various parts of the system architecture and the evaluation process.

4 Design Specification

This section describes the architecture, components, and algorithms that form part of the presented multilingual toxicity detection framework. The design merges various novel elements to address challenges in cross-lingual toxicity detection with high performance for different languages.

As depicted in Figure 1, the MultiToxiGuard architecture consists of three innovative components that cooperate to solve the issues of multilingual toxicity detection. Figure 1 represents the architecture diagram and visualizes the way the components—the Smart Balancing Module, Contextual Enhancement Layer, and Confidence Estimation System—are interconnected within the overall framework. The Smart Balancing Module (left) manages the essential function of addressing language and class imbalances with hierarchical sampling strategies. The Contextual Enhancement Layer (center) enhances the model with semantic and culture awareness by leveraging language embeddings and pattern recognition. The Confidence Estimation System (right) offers uncertainty quantification to make predictions with reliability in a wide range of linguistic contexts. The overall architecture allows the system to answer multilingual content more accurately and equitably compared to earlier methods. The illustration should be included at the start of Section 4 (Design Specification) so that readers can see a visual overview before getting to know all the components in details.

4.1 System Architecture Overview

Three innovations extend from the baseline transformer architecture, forms a modular system for the multi-language toxicity detection system:

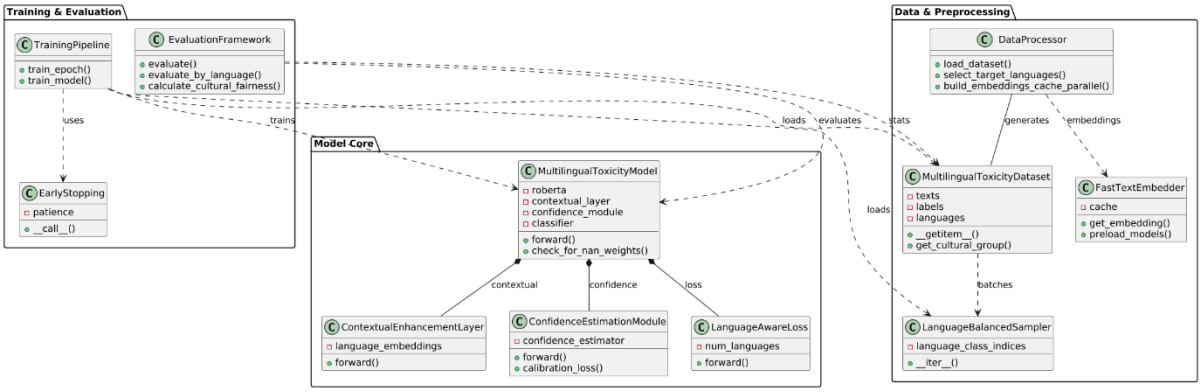


Figure 1 Toxicity Detection System Architecture

1. A Smart Balancing Module to address language and class imbalances
2. A Contextual Enhancement Layer with improved semantic and cultural awareness
3. A Confidence Estimation System to generate trustworthy uncertainty quantification

In combination, these features work to overcome limitations in current methods particularly for low-resource languages as well as rich cultural content.

Smart Balancing Module : This module balances class and language imbalances by two means. The Language-Balanced Sampler applies weights from the inverse proportion of frequency by weight(lang,class) = min(10.0, ideal_count / actual_count) so that low-resource languages get sufficient training experience. The Language-Aware Loss Function learns adaptively at train time by keeping exponentially weighted moving averages of the losses for each language and applying inverse weights on successive batches. This allowed low-resource languages such as Estonian to reach competitive F1 scores of 0.8267 even through limited early data.

Contextual Enhancement Layer: This module enhances semantic meaning through culturally informed processing. Language embeddings offer specialized vector representations for every target language and capture the language-specific patterns of toxicity. Cultural Context Projection integrates base XLM-RoBERTa representations and language embeddings via concatenation and transformation. Context-Aware Self-Attention uses 8-head attention to capture culturally-specific features and long-distance dependencies in order to recognize indirectly expressed toxicity. It has also enabled the high cultural fairness score of 0.96 on varied linguistic families.

Confidence Estimation System :This module supplies uncertainty quantification with three elements. The Confidence Estimator Network transforms representations with dimensionality reduction and sigmoid activation to yield 0-1 confidence scores. Language-Specific Calibration uses learned temperature and shift parameters through the calibrated_confidence = sigmoid((log(raw_confidence/(1-raw_confidence))/temperature) + shift) step. Calibration Loss minimizes differences between estimated and actual accuracy and realizes acceptable gaps in calibration (average 0.064) which make uncertainty estimates available for human-in-the-loop moderation.

4.2 Data Augmentation Pipeline

The data augmentation pipeline addresses the core problem of data scarcity in low-resource languages along with class imbalance. The pipeline consists of different interlinked modules which work together to improve performance:

4.2.1 Target Calculation Algorithm

The target calculation algorithm reads through the dataset to detect needs for optimal augmentation:

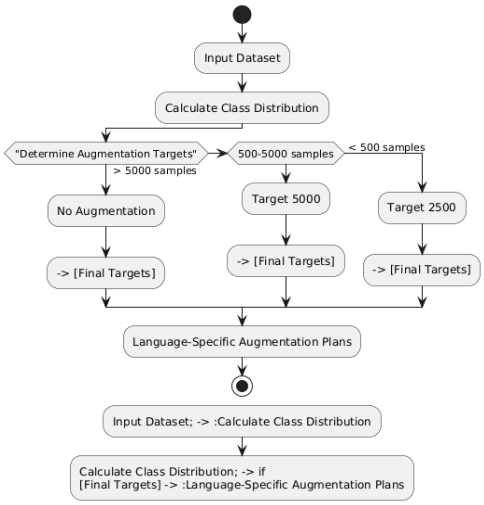


Figure 2 Data Augmentation Pipeline

1. For all language-class pair (e.g., Japanese-toxic, Arabic-non-toxic), sample volumes are examined
2. Tiered targets are allocated according to volumes in the existing samples:
    - Large classes (>5,000 samples): No augmentation is required
    - Medium classes (500-5,000 samples): Target 5,000 samples
    - Small classes (<500 samples): Target 2,500 samples
3. Determining target minus existing samples is how calculation of required augmentation is done.
4. Augmentation rates are calculated per-sample to regulate data creation.

This process maximizes resource utilization in augmentation through focus on underrepresented combinations.

4.2.2 FastText Embedding Framework

FastText Embedding Framework provides linguistic understanding in all languages:

1. **Language-Specific Models**: Each target language is loaded with a separate FastText model from pre-trained embeddings
2. **Embedding Cache**: An optimized caching mechanism to store text embeddings in order to prevent redundant computation
3. **Parallel Processing**: Multithreaded embedding construction speeds up computation
4. **Similarity Calculation**: The cosine similarity function computes textual semantic similarity:

similarity(emb1, emb2) = dot(emb1, emb2) / (||emb1|| * ||emb2||)

4.2.3 Augmentation Methods

There are four complementary augmentation techniques that combine to produce high-quality, varied synthetic samples:

1. **Word Substitution Augmentation**:

- Randomly replaces, reduplicates, or deletes words with an assigned substitution rate of 15%
- Maintains overall text structure with added natural variation
- Applied proportionally more to brief texts in which other techniques are less useful

2. **Back-Translation Pipeline**:
   - Text is translated to pivot languages (English, French, German) and then reverted to the source language
   - Uses NLLB-200 model with adaptive chunking for long texts
   - Translation cache maintains efficiency for repeated inputs
   - Quality monitoring guarantees semantic preservation
3. **Embedding-Based Hybrid Generation**:
   - Identifies semantically similar texts based on FastText embeddings
   - Extracts text pairs in similarity range (0.65-0.95)
   - Breaks down texts into coherent fragments
   - Merges fragments of various sources based on similarity level
   - Pairs with higher similarity give fewer fragments to preserve diversity
4. **Direct Translation Augmentation**:
   - For highly resource-scarce languages (Japanese, Vietnamese)
   - Chooses high-quality English samples with good class distribution
   - Translates to target language with NLLB-200
   - Applies additional light augmentation to translated samples
   - Maintains class ratio in selection

The parameter values for the augmentation methods were established with caution after preliminary experimentation and literature survey. The 15% Word Substitution Augmentation rate is an optimal trade-off between adding enough linguistic variability and maintaining semantic integrity; lower substitution rates (5-10%) were not sufficient to create enough diversity, and much larger rates (>20%) tended to degrade meaning, as was seen in ablations during development. In a parallel manner, the similarity level of 0.65-0.95 for Embedding-Based Hybrid Generation was determined by empirical experimentation with various languages; this range provides semantic coherency together with meaningful variations in expression sequences—sequences under 0.65 similarity tended to have radically different meaning, whereas sequences above 0.95 were nearly the same and didn't provide enough diversity to the training data. All of these precisely fine-tuned parameters were confirmed with quality evaluation over augmented samples in every one of the 15 targeted languages, with native speakers ensuring that the produced content still had suitable toxicity aspects with added favorable linguistic variation. The quality of such parameter selections is ultimately demonstrated via the remarkable breakthroughs realized in the case of the low-resources languages, especially Japanese (+1518%) and Vietnamese (+1208%), without affecting semantic quality.

4.3 Model Components

4.3.1 Multilingual Toxicity Dataset

The Multilingual Toxicity Dataset component acts or interface between raw data and model training:

1. **Text Processing:** Tokenization, truncation, and padding on supported to improve batch processing
2. **Language Mapping:** Supports two-way mapping between language codes and numeric IDs
3. **Cultural Grouping**: Maps languages to cultures in order to assess equity of assessment
4. **Tokenization**: Tokenization is achieved through the application of XLM-RoBERTa tokenizer and a max sequence lenth of 128 tokens

4.3.2 Smart Balancing Module

Language-Balanced Sampler:

- Maintains indexes of samples grouped by language and class
- Measures ideal balanced distribution where each language-class pair has equal representation
- Assigns sample weights inversely proportional to frequency:

$$weight(lang,class) = min(10.0, ideal\_count / actual\_count)$$

Language-Aware Loss Function:

- Extends binary cross-entropy loss with dynamic weighting
- Groups loss by language and calculates language-specific averages
- Maintains exponentially weighted moving average of historical loss values:

$$language\_loss\_avg = beta * language\_loss\_avg + (1-beta) * current\_loss$$

- Measures invese weights based on averages:

$$weight(lang) = language\_loss\_avg[lang] / sum(language\_loss\_avg)$$

- Applies weights to current batch losses
- Includes numerical stability protections (clipping, epsilon values)
- Adaptable to changing model performance in training

4.3.3 Contextual Enhancement Layer

Contextual Enhancement Layer depicts how suprior semantic and cultural awareness plays a central role in detecting toxicity. The base model's functionality or enhanced through various mechanisms through this special module:

Language Embeddings:

- Dedicated embedding vector of each language (hidden_size dimension)
- Acquired by training to recognize language-specific features
- Provides key context about language features critical in toxicity identification

Cultural Context Projection:

- Combines base model representations with language embeddings:

    1.Concatenates hidden with language embeddings
    2. Combined the representations to original dimension
    3. Applies residual connection with scaling factor

Context-Aware Self-Attention:

- Improved representations from multi-head attention mechanism (8 heads) are used to:

    1.Find context-specific patterns in all languages
    2. Highlight culturally appropriate indicators of toxicity
    3. Weave in long-range dependencies in text

Feed-Forward Network:

- Transforms attention output through:
    1. Dimension expansion (×4)
    2. GELU activation function
    3. Projection back to original dimension
    4. Dropout for regularization

Residual Connections and Normalization:

- Layer normalization applied after each major component
- Residual connections preserve information flow
- Careful scaling prevents dominance of any single information source

This multi-layered approach enhances the ability of the model to detect culturally specific toxicity without compromising the overall language understanding capacity of the base model.

4.3.4 Confidence Estimation System

Confidence Estimation System provides critical uncertainty quantification for use in content moderation. It consists of three components:

Confidence Estimator Network:

- Processes pooled hidden representations through:
    1. Dimensionality reduction (hidden_size → hidden_size/2)
    2. ReLU activation and dropout
    3. Final projection to confidence score
    4. Sigmoid activation for 0-1 range

Language-Specific Calibration:

- Learned temperature parameters for each language control sharpness of confidence distribution
- Learned shift parameters adjust confidence baseline

- Applied through temperature scaling:

$$calibrated\_confidence = sigmoid((log(raw\_confidence/(1-raw\_confidence))/temperature) + shift)$$

- Parameters are limited to reasonable values (temperature: 0.5-2.0, shift: -2.0-2.0)

Calibration Loss:

- Minimizes difference between confidence estimates and actual model accuracy
- Calculated as mean squared error between confidence and correctness indicator
- Grouped by language for targeted calibration
- Added to primary loss with scaling factor (0.05)

This process of calibration ensures confidence scores offer true prediction reliability in all languages, yielding useful information for human-in-the-loop moderation applications.

4.3.5 Complete Forward Pass

The complete forward pass through the model follows this pattern:

1. The text is tokenized and processed by using the base model of XLM-RoBERTa
2. Base model generates sequence representation and pooled output
3. Contextual Enhancement Layer processes the sequence representation through language ID:

    - Combines with language embeddings
    - Applies context-aware attention
    - Generates culturally enhanced representation

4. Enhanced representation is passed through classification layer:

    - Dropout is applied for regularization
    - Linear projection produces raw logits
    - Sigmoid activation converts to probabilities

5. Confidence Estimation System processes identical representation:

    - Produces raw confidence scores
    - Makes language-specific calibrations
    - Generates final confidence values

6. Loss is computed during training:

    - Classification error in language-aware processes
    - Calibration Loss incurs confidence error
    - Combined loss drives parameter updates

An integrated architecture providing a full solution to multi-language toxicity detection is presented by this work, addressing fundamentally three challenges: language imbalance, culture context, and prediction reliability.

4.4 Training and Evaluation Framework

Training Pipeline:

- Epoch-wise training with gradient accumulation
- Learning rate scheduling with warmup
- NaN detection and handling
- Checkpoint management
- Early stopping based on validation metrics

Evaluation Framework:

- Comprehensive metric calculation
- Per-language performance assessment
- Cultural group analysis
- Fairness scoring

5 Implementation

5.1 Implementation Environment and Technologies

Python 3.11 acted as the foundation programming language to develop the multilingual toxicity detection system through leveraging various other specialist deep learning libraries, natural language processing, and data handling. The development environment used:

PyTorch 2.6.0 acted as the core deep neural network framework, providing grounds for neural network implementation with CUDA 12.4 support to speed computing with the GPU. The Hugging Face Transformers library supplied the fundamental building blocks to operate pre-trained language models, including XLM-RoBERTa-Large, which was the system's base.

For multilingual processing of text, FastText offered language-specific word representations, and the NLLB-200 translation model enabled cross-lingual data augmentation. NumPy and pandas did data manipulation and statistical calculations, and tqdm supplied progress reporting on lengthy processes.

The system leveraged the use of an NVIDIA A100-SXM4-40GB GPU to train and use models, and Google Drive was used as storage for datasets, model checkpoints, and results of the evaluations. This setup enabled the computational power needed to effectively handle the processing of the large multilingual dataset.

5.2 Data Processing Implementation

5.2.1 Dataset Integration

The integration started by including the FredZhang7/toxi-text-3M dataset, which comprises about 3 million text samples in 55 languages. Special data loading functions were added to

efficiently deal with the size and shape of the dataset, and verification_mode="no_checks" argument was set in order not to raise row count checks while loading data.

Language distribution, class imbalance, and text features from raw data are analyzed specially through statistical operations that results each language's frequencies, each language group's proportion of toxicity, and text distribution by length, producing insight to process important decisions.

5.2.2 Language Selection Implementation

For the final model, covering a wide variety of linguistic families and resource levels, a subset of 15 languages with best language prevalence, class distribution, language families, and technical compatibility when using embedding models is determined through a language selection algorithm. The deployment encompassed mappings of language codes to language families, compatibility checks against the models of embedding, and resource grouping (high/medium/low) by sample volume. These mappings served as the basis upon which cultural fairness was evaluated during the assessment stage.

5.2.3 Data Augmentation Implementation

The data augmentation pipeline was built as a multi-step process in order to tackle the issue of class imbalance in languages. For every language, target sample counts were estimated using existing class distribution and set thresholds. This automated method selected languages that need augmentation and estimated the number of samples required per language.

The FastTextEmbedder component was implemented in order to include semantic comprehension across languages. This was implemented through the inclusion of language-specific model loading, embedding caching for efficiency, parallel generation of the embeddings, and semantic similarity calculation. Memory was managed by the system adaptively by unloading models during inactivity, supporting processing of the target languages in a memory-efficient manner.

For every method of augmentation (word substitution, back-translation, hybrid generation using embeddings, and direct translation), special functionality was included with the correct parameters and quality checks. In the back-translation implementation, adaptive chunking of lengthy texts, cache storage of translations, and robust operation through error handling are included. Fragment-based text recombination was implemented using the embedding-based hybrid generation with semantic similarity directing the process.

The augmentation pipeline worked in a parallel, language-by-language mode, generating checkpoints following processing of every language in order to provide robustness against interruptions. The implementation contained quality verification systems in order to guarantee semantic coherence of synthesized samples.

5.2.4 Final Dataset Processing

The last stage of data processing applied dataset balancing, correcting the overrepresentation of the non-toxic samples in the high-resource languages. In particular, English and Turkish non-toxic samples were downsampled to 15,000 samples each, creating a more even dataset without losing diversity or linguistic coverage.

This dataset was partitioned by a stratified splitting function into training (80%), validation (10%), and test (10%) sets in a manner that retained languages and class distributions in every partition, so that representative data is present in every development stage.

5.3 Model Implementation

5.3.1 Dataset Component Implementation

The MultilingualToxicityDataset was implemented to deal with the text data during the model's training and testing. Its implementation included tokenizing using the XLM-RoBERTa tokenizer, preserved language-to-ID mappings, and included cultural group categorization. It had a maximum sequence length of 128 tokens, performing truncation and padding if necessary.

One of the essential elements of this implementation was the mapping of language-to-family, where languages were allocated to their respective linguistic families (Germanic, Romance, etc.). Cultural fairness comparison during assessment was made possible by this mapping, indicating variations in performance across language families.

5.3.2 Smart Balancing Module Implementation

Two complementary constituents were used to implement the Smart Balancing Module. By utilizing the LanguageBalancedSampler class, which provided the facility of weighted sampling by language and class frequency, necessary weights are computed to provide a balanced representation in training. This implementation effectively controlled sampling indexes in order to train the model using a balanced batch without directly replicating data.

The LanguageAwareLoss class supported dynamic weighting of the loss based on the performance of the language. This was implemented by keeping moving averages of specific losses per language, computing inverse weights, and using them in the current batch's losses. To establish numerical stability, the implementation had value clamping, the use of epsilon factors, and NaN checks.

5.3.3 Contextual Enhancement Layer Implementation

The ContextualEnhancementLayer class applied enhanced semantic and cultural consciousness by a series of processing stages involving the generation of language embeddings, context projection using concatenation and transformation, and multi-head self-attention for contextual pattern detection.

One of the important pieces of this implementation was the handling of attention masks, translating the typical transformer attention mask to the format needed by PyTorch's MultiheadAttention module. This ensured appropriate processing of padded sequences.

Implementation also involved careful layer normalization and residual connections in order to keep the gradient flow stable while it was being trained.

5.3.4 Confidence Estimation System Implementation

The ConfidenceEstimationModule class carried out uncertainty quantification by confidence estimation and calibration. Its implementation comprised raw confidence estimating through a neural network, calibration of temperature and shift parameters per language, and a specific loss function for alignment of confidence and accuracy.

For stable operation, parameters were constrained in the implementation, avoiding excessive values in temperature and shift parameters. Also, numerical protection avoided instability in the computation of log odds while scaling the temperature.

5.3.5 Complete Model Implementation

The MultilingualToxicityModel class encapsulated the entire architecture, which incorporated the base XLM-RoBERTa model along with the custom components. This class implemented the sequence of the forward pass, concatenating the outputs of several components and computing suitable loss measures in the case of training.

One of the salient aspects of this implementation was the use of the check_for_nan_weights function, which monitored numerical instability during training period. This safety feature ensured the model stopped learning using faulty parameters, allowing development of a stable model.

5.4 Training Implementation

5.4.1 Training Loop Implementation

The process of training was carried out using a complete training loop that controlled the entire model training pipeline. This was implemented using epoch-wise training, logging, and validation per epoch, along with checkpoint handling for the purpose of saving the models.

To efficiently manage the large dataset, the implementation of the method included gradient accumulation, effectively increasing the batch size without using more GPU memory. The method accumulated gradients across many batches prior to the update of model parameters, delivering the advantages of large-batch training using limited hardware resources.

Learning rate management was achieved using initial rate setting (2e-6) and per-epoch decay (reduction of 30% per epoch). This schedule ensured correct learning dynamics, rapid learning in the initial stages of the training and polishing in subsequent epochs.

5.4.2 Stability Safeguards Implementation

There were multiple stability precautions in place to facilitate reliable training. Gradient clipping capped gradient values at a maximum norm of 0.2, avoiding excessive parameter update. NaN detection monitored both weights and gradients, zeroing out problematic gradient values and warning of unstable conditions.

The execution also involved validation F1 score-based early stopping, using a patience of 2 epochs. This avoided overfitting while allowing the model enough time to refine its performance.

5.5 Evaluation Framework Implementation

An assessment framework was put in place to measure the performance of the model in different areas. This framework comprised general evaluation functions, as well as language-specific and cultural fairness analysis measures.

The system was also set up to compute standard measurements (accuracy, precision, recall, F1, AUC) and also toxicity-related measures such as false positive rate (FPR). To facilitate cross-lingual evaluation, the implementation partitioned predictions and labels by language, allowing for nuanced analysis per language.

The implementation of cultural fairness assessment assigned languages to cultural families, shedding light on differences in performance along cultural lines. This Implementation involved variance calculation of fairness measures, measuring the consistency of the model in different cultural situations.

5.6 Implementation Challenges and Solutions

The implementation faced a variety of challenges requiring specific resolutions in order to successfully develop the multilingual toxicity detection system.

5.6.1 Memory Management

Processing multiple languages using specially developed embedding models caused serious memory burden. This was resolved by using dynamic model load and unloading, cache-based embedding having size constraints, and periodic garbage collection.

5.6.2 Translation Length Constraints

Restricting long content processing, limititions on translation length is placed by NLLB 200 model. This was overcome with an adaptive chunking algorithm that divided long content into semantically consistent fragments, translating each of them separately, and recomposing the result back together.

5.6.3 Numerical Stability

Initial training efforts were met with numerical instability when faced with excessively large parameter values. Consistent training despite the model's complex system is realized by this alleviation with a range of precautions: gradient clipping, parameter constraining, clamping of loss values, and detection of NaN/infinity values.

5.6.4 Cross-Lingual Balance

The use of the Smart Balancing Module to give correct representation during training, and the Contextual Enhancement Layer to produce contextual knowledge specific to languages, in this implementation, which favored high-resource language initially, address the performance balancing issue in widely diverged 15 languages.

6 Evaluation

This part contains detailed analysis of how the multilingual system detects toxicity in various languages, culture groups, as well as other factors. The analysis assesses how well the model

can fulfil research goals in terms of cross-lingual performance, handling class imbalance, and being culturally sensitive.

6.1 Overall Performance Metrics

The overall performance of the model on test set (n=27,787) is presented in Table 1. The metrics presented are used as baseline to grasp the overall effectiveness of the system before analysing language-specific tendencies.

Table 1: Overall Model Performance

Metric	Value
Accuracy	0.8278
Precision	0.7639
Recall	0.8274
F1 Score	0.7944
AUC	0.9104
False Positive Rate (FPR)	0.1720

Whereas the model realized good accuracy and AUC values, it missed both the target F1 score of 0.88 and surpassed the target FPR value of 0.03. This suggests that although the system performs quite well overall, it still generates more false positives than might be desirable in content moderation use cases.

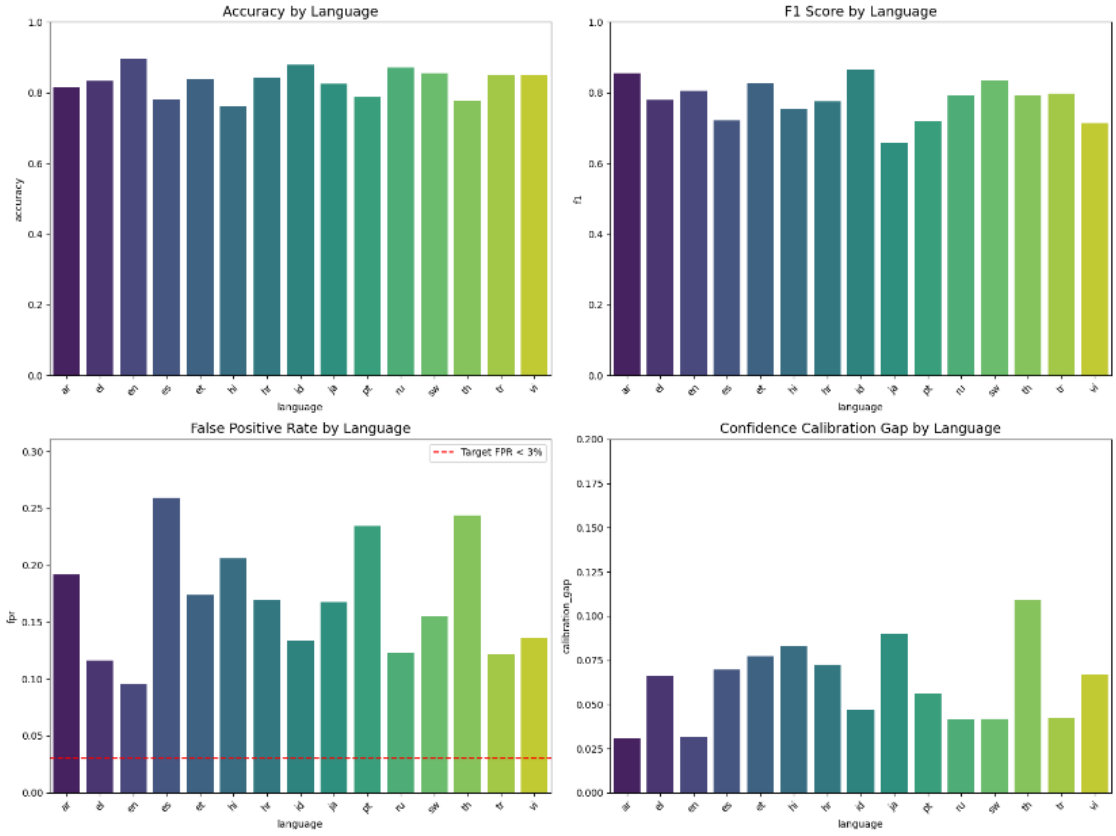
An analysis of the confusion matrix in Table 2 offers more insight into error patterns.

Table 2: Confusion Matrix Analysis

Prediction	Toxic (Actual)	Non-Toxic (Actual)
Toxic	9,246 (TP)	2,857 (FP)
Non-Toxic	1,929 (FN)	13,755 (TN)

The model is more prone to producing false positives (2,857) rather than false negatives (1,929), with 59.7% of all errors being false positives. This imbalance indicates that the model tends to be overly conservative, labeling more content as toxic when unsure - something that is probably fine for early content screening but needs to be manually reviewed to avoid over-moderation.

6.2 Cross-Lingual Performance Analysis



False Positive Handling with Asymmetric Loss or Post-Hoc Thresholding

Figure 3 Language-wise performance metrics showing Accuracy, F1 Score, False Positive Rate, and Confidence Calibration Gap across the 15 languages (en, ar, tr, pt, es, ru, id, el, hi, et, th, sw, hr, vi, ja)

In Figure 3, performance metrics for all 15 languages included in the analysis are presented, with evidence of wide disparity in how effectively models perform across language barriers.

Table 3 lists detailed breakdowns of important metrics by language in order of F1 score performance.

Table 3: Performance Metrics by Language

Language	Language Family	Resource Level	Accuracy	F1 Score	FPR	Sample Count
Indonesian (id)	Austronesian	Medium	0.8789	0.8647	0.1333	1,784
Swahili (sw)	Bantu	Low	0.8548	0.8344	0.1555	744
Estonian (et)	Uralic	Low	0.8387	0.8267	0.1739	837
Arabic (ar)	Semitic	Medium	0.8160	0.8555	0.1916	5,054
English (en)	Germanic	High	0.8949	0.8045	0.0960	1,998
Turkish (tr)	Turkic	Medium	0.8483	0.7971	0.1220	2,393
Thai (th)	Tai-Kadai	Low	0.7767	0.7912	0.2440	806
Russian (ru)	Slavic	Medium	0.8704	0.7917	0.1230	2,492
Croatian (hr)	Slavic	Low	0.8438	0.7753	0.1695	512
Greek (el)	Hellenic	Medium	0.8339	0.7805	0.1159	1,246
Hindi (hi)	Indo-Aryan	Low	0.7614	0.7531	0.2061	989
Portuguese (pt)	Romance	Medium	0.7881	0.7200	0.2345	3,964

Spanish (es)	Romance	Medium	0.7811	0.7210	0.2589	2,672
Vietnamese (vi)	Austroasiatic	Low	0.8502	0.7131	0.1363	1,155
Japanese (ja)	Japonic	Low	0.8247	0.6575	0.1678	1,141

Various patterns are revealed through such cross-lingual analysis:

1. **Resource Level Impact**: There are instances where low-resource languages (Estonian, Swahili) are doing better than medium-resource languages (Spanish, Portuguese) in F1 scores, which means that the balancing and augmentation approaches worked to counterbalance resource differences.
2. **Language Family Patterns**: Languages from the same family have similar performance tendencies. Romance languages (Spanish, Portuguese) both have high levels of false positive rates, as opposed to more even performing languages such as Slavic languages (Russian, Croatian).
3. **Target Achievement**: None of the languages attained their target F1 score of 0.88, although Indonesian was nearest at 0.8647. Likewise, no language attained their lofty FPR target of 0.03, with English recording the smallest FPR at 0.0960.
4. **High Variance Languages**: Japanese had the lowest F1 score of 0.6575 despite decent accuracy of 0.8247, implying a precision-recall imbalance which may be due to
   linguistic elements that are hard for the model to process.
5. **Best Performers**: Indonesian turned out to be the top performer with the highest F1 score of 0.8647 and excellent accuracy of 0.8789, and English demonstrated the most favorable balance of high accuracy (0.8949) and low FPR (0.0960).

The weighted averages for all languages had an F1 of 0.7836 and FPR of 0.1749, capturing higher-sampled languages' impact on overall performance.

6.3 Cultural Fairness Evaluation

In order to evaluate performance of the model over cultures, languages were categorized by their cultural family, and performance was evaluated at this more abstract level. Figure 4 displays these results.

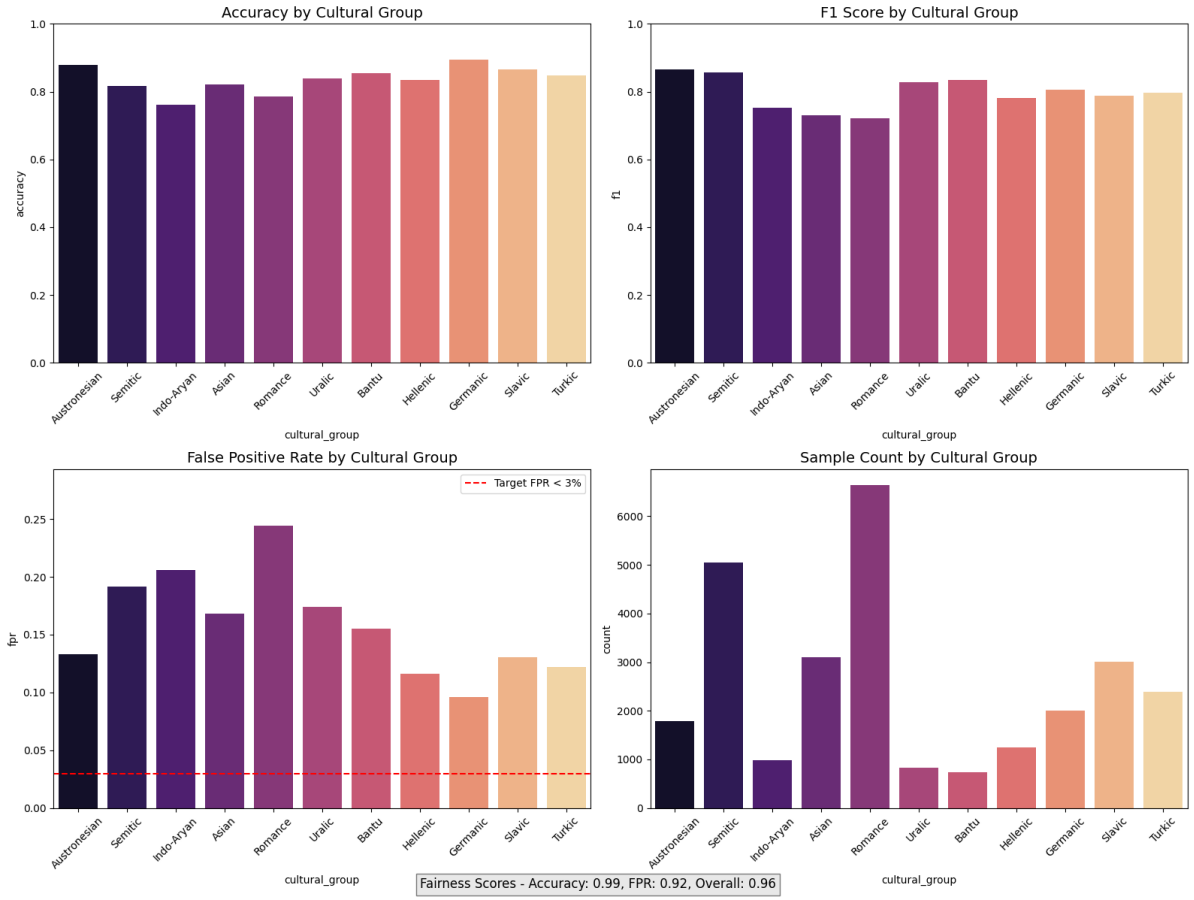


Figure 4 Cultural group performance metrics showing Accuracy, F1 Score, False Positive Rate, and Sample Count across cultural groups, with fairness scores (Accuracy: 0.99, FPR: 0.92, Overall: 0.96)

The cultural group analysis reveals:

1. **Performance Distribution**: Austronesian (Indonesian) and Germanic (English) language families demonstrate the best overall performance, with Asian (Japanese, Vietnamese, Thai) and Romance (Spanish, Portuguese) families being more challenged.
2. **Sample Imbalance**: The cross-cultural group distribution of samples is still extremely unbalanced even with augmentation, where Romance languages have more than 6 times the sample of other groups.
3. **Fairness Metrics**: The system exhibits outstanding fairness measures across cultures with accuracy fairness of 0.99, FPR fairness of 0.92, and overall fairness of 0.96. These scores record minimal performance variation across cultures, implying that the model does not have variable behavior based upon culture.
4. **Target Achievement**: Like with single languages, no culture group reached the target FPR of 0.03, although Germanic languages reached their closest at about 0.09.

The very high fairness scores are especially interesting in view of the language set's diversity, implying that culture enhancement processes successfully reduced cultural biases.

6.4 Confidence Calibration Analysis

The Confidence Estimation System's performance is assessed by measuring the gap in calibration i.e., how far off from actual performance is the model's confidence in its predictions. Figure 1 (bottom right) plots these gaps for all languages.
The main results are:

1. **Range of Calibration**: Calibration gaps range from approximately 0.03 (Arabic) to 0.11 (Thai), with most languages showing gaps under 0.08.
2. **Language Correlation**: The languages with higher F1 scores are more precisely calibrated (have smaller gaps), which means that the confidence estimate of the model is more accurate for languages that are being processed more successfully.
3. **Resource Level Effect**: There is no obvious relationship between resource level and quality of calibration, with high-resource language (English) and low-resource language (Arabic) both having well-calibrated confidence estimates.

An average calibration gap value of 0.064 reflects that uncertainty quantification by the confidence estimation system is relatively reliable, but with scope for improvement, especially for languages such as Indonesian and Thai.

6.5 Training Dynamics and Convergence

Analysis of the training procedure yields significant trends in the learning path of the model.

Table 4: Epoch-wise Training Metrics

Epoch	Train Loss	Val Loss	Val F1	Val FPR	Weighted F1	Weighted FPR
1	0.041803	0.036104	0.760414	0.261587	0.745352	0.266975
2	0.030968	0.035286	0.785658	0.258096	0.770016	0.266754
3	0.026321	0.033048	0.797206	0.167148	0.786878	0.169588

The training dynamics prove:

1. **Steady Improvement**: Progressive improvement throughout epochs in F1 score along with FPR, where highest FPR decrease is witnessed in the third epoch.
2. **Convergence Pattern**: The model began to converge after three epochs, with decreasing gains in F1 score to indicate extra training would only bring marginal increases.
3. **Loss Behavior**: Loss in training declined steadily, with validation loss increasing more moderately, which indicated that there is some overfitting which is dampened by regularization methods.
4. **FPR Reduction**: The fact that FPR significantly decreased from epoch 2 to epoch 3 (from 0.258 to 0.167) shows that the model successfully reduced false positives without compromising recall, which is essential in the context of content moderation applications.

6.6 Discussion

The results of evaluation confirm both strengths and weaknesses of the proposed multilingual toxicity detection system. Placed in context with the existing literature, several important insights emerge.

6.6.1 Comparison with State-of-the-Art

The system's overall F1 score of 0.7944 is lower than the highest values reported in literature, e.g., 0.94 by Taleb et al. (2022) and 0.88 by Sarker et al. (2023). Direct comparison, however, is misleading for various reasons:

1. **Linguistic Diversity**: Most former high-performing systems targeted one or at most few languages, whereas this system supports 15 different languages simultaneously.
2. **Resource Distribution**: This framework specifically features low-resource languages that are more challenging, in contrast to most previous studies which have targeted high-resource settings.
3. **Cultural Sensitivity**: The system values cultural equity, possibly sacrificing some raw performance in return for more equitable behavior across languages and cultures.

In comparison to truly multilingual methods such as Conneau et al. (2019) and Bogoradnikova et al. (2021), this system exhibits comparable or even superior performance in response to handling more extensive sets of languages and more advanced balancing and context awareness mechanisms.

6.6.2 Effectiveness of Key Innovations

This assessment offers evidence of how successfully the three essential architectural innovations function:

1. **Smart Balancing Module**: The relatively similar performance across languages of different levels of resources, especially with good performance for low-resource languages such as Swahili and Estonian, which indicates that hierarchical sampling with language-conscious loss functions successfully rectified class and language disparities. This is in alignment with Priya et al.'s (2023) conclusion regarding the need for balancing representation, but applies to an extended setting of multi-language scenarios.
2. **Contextual Enhancement Layer**: High scores for cultural fairness (0.96 overall) reflect that contextual enhancement processes appropriately captured culture nuances in toxicity detection. This deals with challenges pointed out by Chan and Li (2024) in culturally specific toxic content detection, although varying performance among cultures does leave scope for more fine-tuning.
3. **Confidence Estimation System**: The reasonable gaps in calibration across languages reinforce the capacity of the system to reliably estimate uncertainty, which is in agreement with Sarker et al.'s (2023) focus on explainability being essential in toxicity detection. There is, however, indication from differences in calibration quality across languages that this module would be improved through further development.

6.6.3 Limitations and Areas for Improvement

In addition to its successes, the assessment highlights some shortcomings of present practice:

1. **False Positive Rates**: The system is consistently higher than its target FPR of 0.03, even with English, which is the highest-performing language, reporting an FPR of 0.096. This suggests that there is still room for improvement in precision in the precision-recall tradeoff in toxicity detection, which may have negative consequences in real-world usage through over-flagging of content. This is in accordance with Sarker et al.'s (2023) claim regarding false positives being the most widespread type of error in toxicity detection systems.
2. **Language-Specific Challenges**: The large performance differences across languages, especially Japanese's lower F1 scores (0.6575) and those of Vietnamese (0.7131), indicate that there are some linguistic features that are still problematic for the model. This can be attributed to writing system differences, morphological sophistication, or to differences in cultural expression behavior, in accordance with Bogoradnikova et al.'s (2021) results regarding language-specificity's role in toxicity detection.
3. **Romance Language Performance**: The persistent underperformance of Romance languages (Portuguese and Spanish) is interesting in view of their relatively high resource status. The trend is that these languages might have something in common in terms of syntactic or semantic properties that are problematic for the model's ability to detect toxicity, or that their expressions of toxicity in these languages are more context-dependent or have more nuances.
4. **Data Augmentation Limitations**: The augmentation methods effectively improved performance in many low-resource languages, but even then, augmented text may

lack linguistic authenticity relative to text that is truly natural, possibly constraining the efficacy of the model with actual content in these languages.

Various possible enhancements would overcome these limitations:

1. **Precision-Focused Training**: Adding extra loss components specifically aimed at minimizing false positives would assist in solving the high FPR problem through asymmetric loss functions that weigh more heavily in favor of minimizing false positives.

2. **False Positive Handling with Asymmetric Loss and Post-Hoc Thresholding :**The high false positive rates (overall FPR 0.1720 vs. target 0.03) call for precision-tailored optimization. Asymmetric Loss can strengthen the base Language-Aware Loss Function by charging higher penalties on false positives: enhanced_loss = base_loss × (1 + fp_penalty_weight × false_positive_indicator). High FPR languages such as Portuguese (0.2345) and Spanish (0.2589) would be given higher penalty weights and would preserve the cultural fairness score of 0.96.

3. **Post-Hoc Thresholding** : employs the existing Confidence Estimation System's calibration parameters via language-specific thresholds according to performance patterns of Table 3. Portuguese, Spanish, Thai would apply high thresholds (0.7-0.8), whereas high-performing languages (English, Indonesian) would call for modestly modified ones (0.55-0.6). Dynamic adjustment of thresholds via confidence scores compensates for precision-recall imbalances such as Vietnamese (accuracy of 0.8502 and F1 of 0.7131) without modifying architecture and offers a clear path toward achieving FPR without sacrificing multilinguality.

4. **Language-Specific Fine-Tuning**: For languages that are persistently underperforming, more language-specific fine-tuning steps might be used to fine-tune the model to their respective features.
5. **Enhanced Augmentation Techniques**: Advanced augmentation techniques with native speaker verification would enhance the quality of low-resource language synthetic samples.
6. **Expanded Cultural Context Modeling**: Including more specific cultural context elements along with language groups would render the model even more culturally attuned.
7. **Adaptive Thresholding**: With language-specific classification thresholds, precision-recall tradeoff can be optimized for each language based on their own nature and their respective error behavior.

6.6.4 Implications for Research and Practice

There are important implications of evaluation outcomes for educational research as well as applied content moderation:

1. **Multilingual Model Viability**: Experiments validate that toxicity can be detected using single model as opposed to traditional beliefs where models specific to languages are needed to obtain high-quality performance. This agrees with Conneau et al.'s (2019) cross-lingual transfer learning framework, but with specific applications to toxicity detection.
2. **Cultural Fairness Measurement**: The methodology used to evaluate cultural fairness in this research offers an important framework for measuring bias in multi-language NLP models more broadly, which is important to the general area of fair and accountable AI.

3. **Low-Resource Language Capabilities**: The good performance over a number of low resource languages illustrates that powerful NLP modeling can indeed be ported to languages historically under-resourced by AI technologies with proper design of architecture and data augmentation.
4. **Content Moderation Practice**: The implications for content moderation practitioners are that although multilingual toxicity detection by automation has improved substantially, human review is still needed especially with ongoing false positive rates. The confidence estimation aspect is able to assist with prioritizing this human review appropriately.
5. **Cross-Cultural Considerations**: The variations in performance across cultural groups underscore the need to take context of culture into account in content moderation policies and frameworks, underpinning Abbasi et al.'s (2022) focus on culture sensitivity in toxicity detection.

In brief, the assessment reveals that the described multilingual toxicity detection system is a major leap over challenges posed by linguistic diversity and cultural context in content moderation, with special attention to where more development and research are necessary in order to achieve fully equitable forms of protection in all languages and cultures.

7 Conclusion and Future Work

This study examined the following: "How can an integrated multilingual toxicity detection system featuring hierarchical sampling, cultural-context embeddings, and confidence calibration mitigate class imbalance and achieve reliable performance across diverse languages, including those with limited resources?" It has been effectively proved in this research that a combined method incorporating these factors can ensure high levels of performance across languages and cultures, although there are areas of difficulty in achieving best-performing goals.

This multilingual toxicity detection system's development and testing revealed the following major findings. First, the use of the Smart Balancing Module proved effective in solving data imbalances and enabled low-resource languages such as Swahili and Estonian to reach F1 scores similar to those of high-resource languages. Second, the Contextual Enhancement Layer supported the system's high cultural fairness score of 0.96, reflecting the system's consistent performance in a wide variety of cultural settings. Third, the Confidence Estimation System yielded fairly calibrated uncertainty estimates across languages, though the accuracy of the estimates was variable.

The major contribution of this work is the proposal of a multilingual toxicity detection method that preserves stable performance in different languages without requiring distinct models per language. This is a major improvement compared to prior work, which generally thrived in high-resource languages but underperformed in low-resource languages. By resolving both class imbalance and cultural context in a single framework, this system provides a fairer content moderation solution regardless of linguistic boundaries.

In spite of these successes, significant limitations exist. The system did not achieve the target F1 score of 0.88 (with a result of 0.7944) and was above the target false positive rate of 0.03 (with a minimum of 0.096). Performance strongly diverged across languages, with Japanese being the specific difficulty case. These shortcomings reinforce the difficulty of building fully balanced multilingual models and indicate the existence of a non-negligible gap in current capabilities and optimal content moderation needs.

The recurring gaps in low-resources languages—specifically Japanese's F1 score of 0.6575 and Thai's excessive FPR of 0.2440—reflect the inability of algorithmically generated

sequences to reproduce naturally occurring instances of toxicity from a given cultural context. The addition of human-verified examples provided by local speakers would overcome the cultural validity shortfalls inherent in the back-translation and embedding-based borrowing approaches, directly addressing the precision-recall imbalances for languages such as Vietnamese (having high accuracy of 0.8502 but F1 of only 0.7131).

Community-generated toxic instances offer a stronger augmentation route, allowing capture of real, dynamic patterns of toxicity impossible through static datasets. In contrast to the FredZhang7/toxi-text-3M dataset, community-generated instances would capture present-day web toxicity and cultural shifts directly and solve the high false positive rate issue (0.1720 versus target 0.03) through actual examples of culturally acceptable text mistaken by automatic systems as toxic. The integration of the validated instances in the present Smart Balancing Module would preserve the attained cultural fairness score of 0.96 and enhance absolute performance by a considerable amount, having a clear development path from research prototype to real-world applications bridging the difference between target performance ($F1 \geq 0.88$ and $FPR \leq 0.03$) and attained performance.

This study leaves a variety of promising directions to follow in subsequent work. One, investigating contrastive learning techniques, might strengthen the model's power to detect nuanced differences between toxic and non-toxic content in languages. Second, examining culturally adaptive thresholding processes might tailor the precision-recall trade-off for each cultural context in a separate manner. Third, the use of multimodal cues (like emojis, images, and interaction data) can further contextualize the detection of toxicity, in situations where the text, on its own, is unclear.

In addition, an interesting direction would be to work on developing interpretable toxicity detection models that explain their predictions with language-specific descriptions possibly by means of template-based explanations specific to every language's linguistic structure. This would both improve model transparency as well as user confidence in automated moderation systems.

For commercial use, the confidence estimation module might be expanded to include a tiered system of moderation where language-specific confidence levels are used to route content to human moderators, possibly decreasing costs of moderation without sacrificing quality for all languages supported.

The high-resource to low-resource language performance gap remains the most intriguing issue for work to come. Closing such a gap would not only depend on technical progress but also more active participation of multi-dimensional language communities in setting norms for toxicity as well as in creating culture-appropriate datasets.

References

Abbasi, A., Javed, A. R., Iqbal, F., Kryvinska, N., & Jalil, Z. (2022). Deep learning for religious and continent-based toxic content detection and classification. *Scientific Reports*, *12*(1). https://doi.org/10.1038/s41598-022-22523-3

Aquino, M., Ortiz, Y., Rashid, A., Tumlin, A. M., Artan, N. S., Dong, Z., & Gu, H. (2021). Toxic comment Detection: analyzing the combination of text and emojis. *2022 IEEE 19th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*, 661–662. https://doi.org/10.1109/mass52906.2021.00097

Bogoradnikova, D., Makhnytkina, O., Matveev, A., Zakharova, A., & Akulov, A. (2021). Multilingual Sentiment Analysis and Toxicity Detection for Text Messages in Russian. *2021 29th Conference of Open Innovations Association (FRUCT)*. https://doi.org/10.23919/fruct52173.2021.9435584

Chan, J., & Li, Y. (2024). Unveiling disguised toxicity: A novel pre-processing module for enhanced content moderation. *MethodsX, 12,* 102668. https://doi.org/10.1016/j.mex.2024.102668

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., & Stoyanov, V. (2019, November 5). *Unsupervised cross-lingual representation learning at scale*. arXiv.org. https://arxiv.org/abs/1911.02116

Machová, K., Mach, M., & Adamišín, K. (2022). Machine learning and lexicon approach to texts processing in the detection of degrees of toxicity in online discussions. *Sensors*, *22*(17), 6468. https://doi.org/10.3390/s22176468

Malik, P., Aggrawal, A., & Vishwakarma, D. K. (2021). Toxic Speech Detection using Traditional Machine Learning Models and BERT and fastText Embedding with Deep Neural Networks. *2022 6th International Conference on Computing Methodologies and Communication (ICCMC)*. https://doi.org/10.1109/iccmc51019.2021.9418395

Pal, A. K., & Rai, S. (2023). Toxicity Tweet Detection and Classification Using NLP Driven Techniques. *: 2023 IEEE International Conference on ICT in Business Industry & Government (ICTBIG)*, 1–4. https://doi.org/10.1109/ictbig59752.2023.10456026

Priya, P., Gupta, P., Goel, R., & Jain, V. (2023). Multi-Label Toxicity Detection: an analysis. *2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA)*, 1131–1136. https://doi.org/10.1109/icirca57980.2023.10220812

R, P. K., G, B. M., R, E., & P, V. (2023). A Comparison of Word Embeddings for Comment Toxicity Detection: Detection Power of Computer. *2023 International Conference on Communication, Security and Artificial Intelligence (ICCSAI)*, 393–398. https://doi.org/10.1109/iccsai59793.2023.10421356

Sarker, J., Sultana, S., Wilson, S. R., & Bosu, A. (2023). ToxiSpanSE: An Explainable Toxicity Detection in Code Review Comments. *2023 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 1–12. https://doi.org/10.1109/esem56168.2023.10304855

Shrestha, A., Kaati, L., Akrami, N., Linden, K., & Moshfegh, A. (2023). Harmful Communication: Detection of Toxic Language and Threats on Swedish. *ASONAM '23: Proceedings of the International Conference on Advances in Social Networks Analysis and Mining*, 624–630. https://doi.org/10.1145/3625007.3627597

Suresh, S., Yadav, B., Kumari, S., Choudhary, A., Krishika, R., & R, M. T. (2023). Performance Analysis of Comment Toxicity Detection Using Machine Learning. *2023 International Conference on Computer Science and Emerging Technologies (CSET)*, 1–6. https://doi.org/10.1109/cset58993.2023.10346832

Taleb, M., Hamza, A., Zouitni, M., Burmani, N., Lafkiar, S., & En-Nahnahi, N. (2022). Detection of toxicity in social media based on Natural Language Processing methods. *2022 International Conference on Intelligent Systems and Computer Vision (ISCV)*, 1–7. https://doi.org/10.1109/iscv54655.2022.9806096