National College of Ireland

Configuration Manual

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.....

Lecturer: .....Hamilton Niculescu.....

Submission Due Date: .....24/04/2025.....

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

Word Count: .....1842..... **Page Count:** .....16.....

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: .....23/04/2025.....

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):	

Configuration Manual

Jacob Saju
x23166363

1 Introduction

1.1 Project Overview

The initial setup guide describes the deployment and configuration of an XLM-RoBERTa system that processes text of various languages and classifies the text to be toxic or not with specific support to process language and culture variations of a maximum of 15 languages.

1.2 Model Details

XLM-RoBERTa Model Specifications:

- **Base Architecture**: XLM-RoBERTa Large
- **Parameters**: 560 million
- **Pre-trained on**: 2.5TB of CommonCrawl data in 100 languages
- **Fine-tuned on**: Customized multilingual toxicity dataset

Custom Enhancement Components:

- **Contextual Enhancement Layer**: Adapts to language-specific features
- **Confidence Estimation Module**: Provides calibrated confidence scores
- **Language-Aware Loss**: Balances performance across languages
- **Cultural Grouping System**: Aggregates languages by cultural/linguistic families

2 System Requirements

2.1 Hardware Requirements

- **GPU**: NVIDIA GPU with at least 8GB VRAM (A100 40GB recommended)
- **RAM**: 16GB minimum, 32GB recommended
- **Storage**: 10GB for model and 5GB for dataset
- **CPU**: 4+ cores recommended for preprocessing

2.2 Software Requirements

- **Python**: 3.8+
- **CUDA**: 11.0+
- **Core Libraries**:

- PyTorch 1.12+
- Transformers 4.28+
- Pandas
- Numpy
- Scikit-learn
- Matplotlib
- Seaborn
- Tqdm

3 Setup Instructions

3.1 Google Colab Setup

1. **Initial Setup**
   - Open web browser and navigate to https://colab.research.google.com
   - Sign in with your Google account
   - Click 'New Notebook' or File → New Notebook
2. **GPU Configuration**
   - Click 'Runtime' in the top menu
   - Select 'Change runtime type'
   - In the popup window:
     - Set 'Runtime type' to 'Python 3'
     - Set 'Hardware accelerator' to 'GPU'
     - Set 'GPU type' to any available GPU (A100 recommended)
     - Click 'Save'
   - Verify GPU access:

!nvidia-smi

3. **Google Drive Mounting**

- Import and mount drive:

python
from google.colab import drive
drive.mount('/content/drive')

- *Verify mounting successful:*

python
!ls "/content/drive/MyDrive"

3.2 Directory Structure Setup

1. **Create Main Project Directory**
   1. /content/drive/MyDrive/Thesis_XLNEt
   2. checkpoints :store the final model check point hereL
   3. final_evaluation → store final evaluation results

2. **Final Directory Structure Should Look Like:**

```
Thesis_XLNEt/
├── checkpoints/
│   └── model_epoch_3.pt
├── selected_language_samples_final_balanced.csv
└── final_evaluation/
    ├── overall_metrics.csv
    ├── class_metrics.csv
    ├── language_metrics.csv
    ├── confusion_matrix.png
    ├── language_metrics_plot.png
└── cultural_metrics_plot.png
```

3.3 Package Installation

!pip install -q transformers pandas numpy scikit-learn matplotlib seaborn tqdm

4 Dataset Preparation

4.1 Dataset Organization

- **Dataset Source**: Multilingual toxicity dataset with 15 supported languages
- **Total Samples**: 27,787 text samples
- **Class Distribution**:
  - Toxic: 11,175 samples (40.2%)
  - Non-toxic: 16,612 samples (59.8%)
- **Language Distribution**:
  - Arabic (ar): 5,054 samples
  - Portuguese (pt): 3,964 samples
  - Spanish (es): 2,672 samples
  - Russian (ru): 2,492 samples
  - Turkish (tr): 2,393 samples
  - English (en): 1,998 samples
  - Indonesian (id): 1,784 samples
  - Greek (el): 1,246 samples
  - Vietnamese (vi): 1,155 samples
  - Japanese (ja): 1,141 samples
  - Hindi (hi): 989 samples
  - Estonian (et): 837 samples
  - Thai (th): 806 samples
  - Swahili (sw): 744 samples
  - Croatian (hr): 512 samples

4.2 Cultural/Linguistic Grouping

Languages are grouped into the following cultural families:

- **Romance**: Spanish (es), Portuguese (pt)
- **Germanic**: English (en)
- **Slavic**: Russian (ru), Croatian (hr)
- **Asian**: Japanese (ja), Vietnamese (vi), Thai (th)
- **Semitic**: Arabic (ar)
- **Indo-Aryan**: Hindi (hi)
- **Turkic**: Turkish (tr)
- **Austronesian**: Indonesian (id)
- **Hellenic**: Greek (el)
- **Uralic**: Estonian (et)
- **Bantu**: Swahili (sw)

4.3 Data Preprocessing

1. **Text Preprocessing**:
    - Tokenization using XLM-RoBERTa tokenizer
    - Maximum sequence length: 128 tokens
    - Padding to max length
    - Attention masks for padded tokens
2. **Dataset Splitting**:
    - Training set: 80% of data
    - Validation set: 10% of data
    - Test set: 10% of data
    - Stratified by language and toxicity label

5 Model Configuration

5.1 Base Model Setup

pythonfrom transformers import XLMRobertaTokenizer, XLMRobertaModel

 # Initialize base model
tokenizer = XLMRobertaTokenizer.from_pretrained("xlm-roberta-large")
base_model = XLMRobertaModel.from_pretrained("xlm-roberta-large")

5.2 Custom Model Components

Contextual Enhancement Layer:
pythonclass ContextualEnhancementLayer(nn.Module):
def __init__(self, hidden_size, num_languages):
super()._init_()
self.hidden_size = hidden_size
self.num_languages = num_languages

 # Language embeddings
self.language_embeddings = nn.Embedding(
self.num_languages,
self.hidden_size
)

 # Context projection
self.context_projection = nn.Linear(
*self.hidden_size * 2,*
self.hidden_size
)

 # Self-attention and other components
 # ...

Confidence Estimation Module:
pythonclass ConfidenceEstimationModule(nn.Module):
def __init__(self, hidden_size, num_languages):
super().__init__()
self.hidden_size = hidden_size
self.num_languages = num_languages

 # Confidence network
self.confidence_estimator = nn.Sequential(
nn.Linear(hidden_size, hidden_size // 2),
nn.ReLU(),
nn.Dropout(0.1),
nn.Linear(hidden_size // 2, 1),
nn.Sigmoid()
)

 # Calibration parameters
*self.calibration_temp = nn.Parameter(torch.ones(num_languages) * 1.0)*
self.calibration_shift = nn.Parameter(torch.zeros(num_languages))
 # ...

Language-Aware Loss:
pythonclass LanguageAwareLoss(nn.Module):
def __init__(self, num_languages, beta=0.9999):
super().__init__()
self.num_languages = num_languages
self.beta = beta
self.register_buffer('language_loss_avg', torch.ones(num_languages))
 # ...

5.3 Training Parameters

- Epochs: 3
- Batch Size: 8 (with gradient accumulation steps = 4)
- Learning Rate: 2e-6
- Optimizer: AdamW with weight decay = 0.01
- Loss Function: Language-aware binary cross-entropy with confidence calibration
- Evaluation Metrics: F1 score, False Positive Rate (FPR)

6 Training Pipeline

6.1 Dataset and Dataloader Setup

```
python# Create dataset classes
train_dataset = MultilingualToxicityDataset(train_df, tokenizer, max_length=128)
val_dataset = MultilingualToxicityDataset(val_df, tokenizer, max_length=128)
test_dataset = MultilingualToxicityDataset(test_df, tokenizer, max_length=128)

# Create dataloaders with balanced sampling
train_loader = DataLoader(
    train_dataset,
    batch_size=8,
    sampler=LanguageBalancedSampler(train_dataset),
    num_workers=2
)

val_loader = DataLoader(
    val_dataset,
    batch_size=16,
    shuffle=False,
    num_workers=2
)

test_loader = DataLoader(
    test_dataset,
    batch_size=16,
    shuffle=False,
    num_workers=2
)
```

6.2 Training Process

```
python# Training loop with early stopping
best_f1 = 0
best_model_path = None
best_epoch = 0
gradient_accumulation_steps = 4

for epoch in range(num_epochs):
    # Train for one epoch
    train_loss = train_epoch(model, train_loader, optimizer, scheduler, device,
                             gradient_accumulation_steps=gradient_accumulation_steps)

    # Evaluate on validation set
    val_metrics, val_preds, val_probs, val_labels, val_lang_ids, val_confidences =
evaluate(model, val_loader, device)

    # Calculate detailed metrics
    class_metrics = calculate_class_metrics(val_preds, val_labels)
```

```
# Calculate per-language metrics
lang_metrics = evaluate_by_language(model, val_loader, device, val_dataset)

# Record best model
if val_metrics['f1'] > best_f1:
    best_f1 = val_metrics['f1']
    best_model_path = f"{CHECKPOINT_DIR}/model_epoch_{epoch+1}.pt"
    best_epoch = epoch + 1

    # Save best model
    torch.save({
        'epoch': epoch + 1,
        'model_state_dict': model.state_dict(),
        'optimizer_state_dict': optimizer.state_dict(),
        'scheduler_state_dict': scheduler.state_dict(),
        'val_metrics': val_metrics,
        'lang_metrics': lang_metrics,
        'class_metrics': class_metrics,
        'lang_to_id': train_dataset.lang_to_id
    }, best_model_path)
```

6.3 Checkpointing

- Checkpoint Frequency: After each epoch
- Best Model Selection: Based on F1 score on validation set
- Checkpoint Directory: /content/drive/MyDrive/Thesis_XLNEt/checkpoints/
- Checkpoint Contents:
    - Model state dict
    - Optimizer state dict
    - Scheduler state dict
    - Validation metrics
    - Language-specific metrics
    - Class metrics
    - Language-to-ID mapping

7 Inference Setup

7.1 Google Colab Environment Configuration

7.1.1 Connecting to Google Colab with T4 GPU

To set up the inference environment in Google Colab:

1. **Access Google Colab**:
    - Navigate to Google Colab in your web browser
    - Create a new notebook or open an existing one
2. **Configure T4 GPU**:
    - Click on "Runtime" in the top menu
    - Select "Change runtime type"
    - From the dropdown menu, choose "T4 GPU" under hardware accelerator
    - Click "Save" to apply the configuration
3. **Verify GPU Allocation**:

- Run a system check to confirm T4 GPU allocation
- Verify CUDA and PyTorch are correctly detecting the GPU
- Check available GPU memory (typically 16GB for T4)

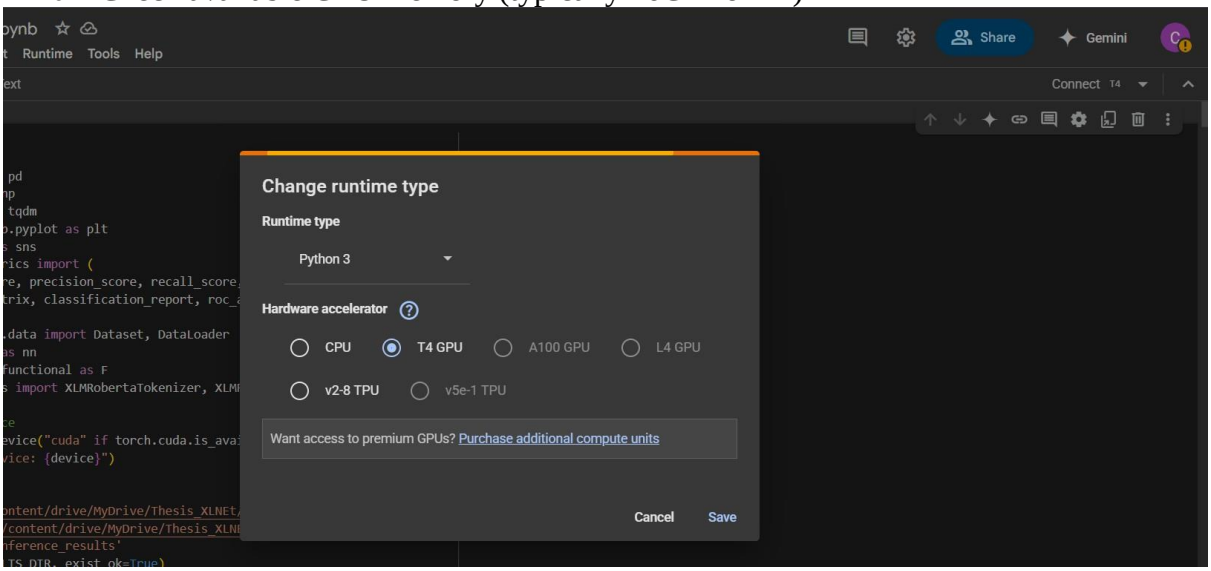


Figure 1 : Google Colab runtime configuration dialog showing the selection of T4 GPU hardware accelerator. This step is crucial for ensuring sufficient computational resources for the multilingual toxicity detection model, which requires GPU accelerat

7.1.2 Mount Google Drive

To access model files and datasets:

1. **Connect to Google Drive**:
   - Use the Google Colab Drive integration
   - Authenticate by following the provided link
   - Enter the authorization code when prompted
2. **Verify Drive Mounting**:
   - Confirm the drive is properly mounted by listing directories
   - Check access to the Thesis_XLNEt folder containing model checkpoints
   - Ensure read/write permissions are working correctly
3. **Environment Path Configuration**:
   - Set path variables to relevant directories
   - Configure checkpoint path to model_epoch_3.pt
   - Set dataset path to selected_language_samples_final_balanced.csv
   - Create output directory for inference results

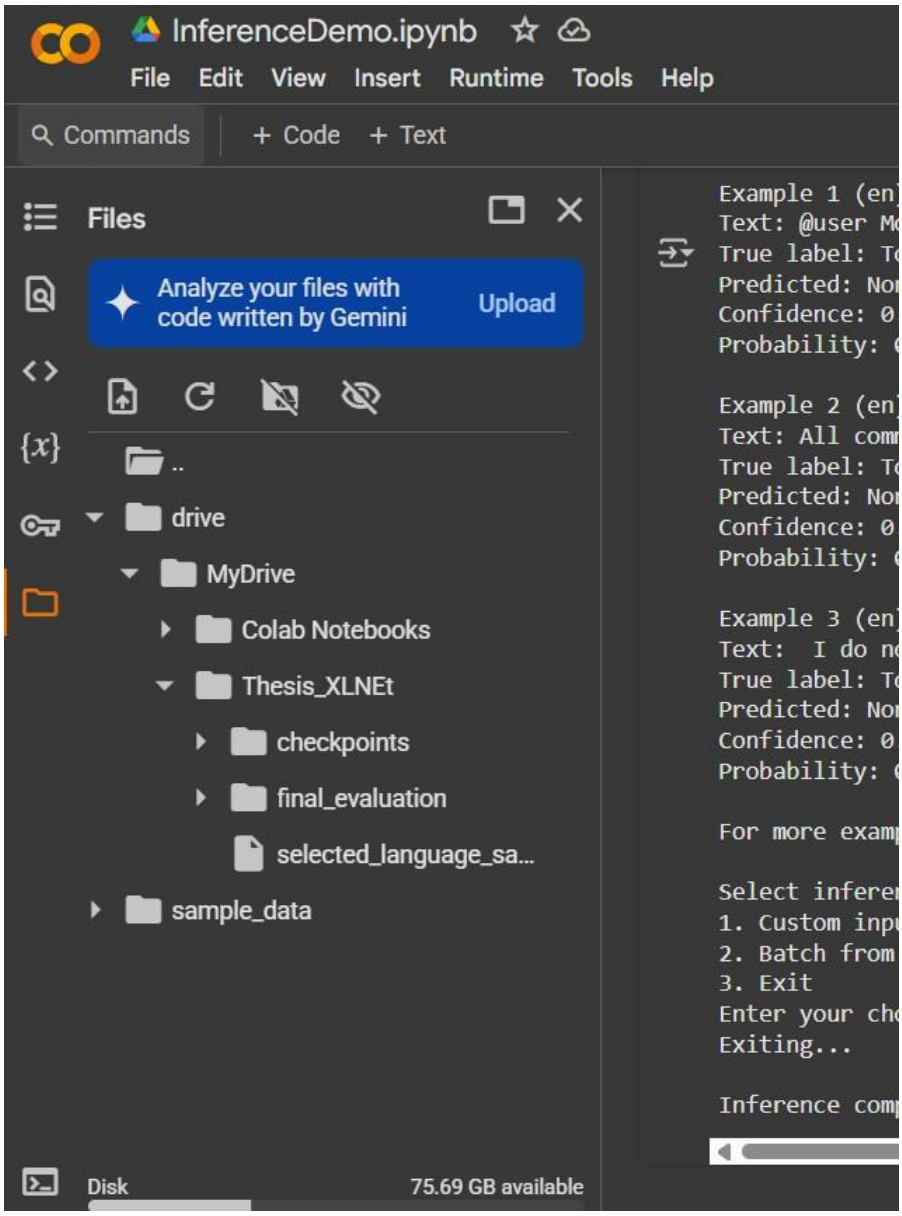


Figure 2 Project directory structure in Google Colab showing the organized folder hierarchy with checkpoints, final_evaluation, and the balanced dataset. This organization ensures proper access to model weights and inference resources while maintaining separatio

7.2 Preparation for Inference

7.2.1 Environment Setup Process

1. **Library Initialization**:
   - Import required PyTorch, transformers, and utility libraries
   - Configure GPU memory settings for optimal performance
   - Set up logging and visualization dependencies
2. **Model Preparation**:
   - Initialize XLM-RoBERTa tokenizer
   - Load model weights from checkpoint with special PyTorch 2.6+ handling
   - Move model to GPU and set to evaluation mode
   - Extract language mapping from the checkpoint
3. **Input Preparation**:

- For single sample inference: prepare text and language inputs
- For batch inference: organize dataset with appropriate preprocessing
- Configure batch size based on available GPU memory

7.2.2 Inference Options

1. **Interactive Mode**:
   - User provides text samples directly in Colab
   - Supports all 15 model languages
   - Displays predictions with confidence scores
2. **Batch Processing Mode**:
   - Processes multiple samples from the dataset
   - Allows filtering by language and sample count
   - Calculates detailed performance metrics if labels are available
   - Generates comprehensive reports and visualizations
3. **Output Management**:
   - Results saved to specified Google Drive location
   - Predictions stored in CSV format for further analysis
   - Visualizations including confusion matrix and language-specific performance
   - Optional detailed sample analysis showing cases of correct and incorrect predictions

8 Performance Metrics

8.1 Overall Performance

- **Accuracy**: 0.8278
- **Precision**: 0.7639
- **Recall**: 0.8274
- **F1 Score**: 0.7944
- **AUC-ROC**: 0.9104
- **False Positive Rate**: 0.1720
- **False Negative Rate**: 0.1726

8.2 Class-wise Performance

Class	Count	Precision	Recall	F1 Score
Toxic	11175	0.7639	0.8274	0.7944
Non-toxic	16612	0.8772	0.8280	0.8519

8.3 Language-specific Performance

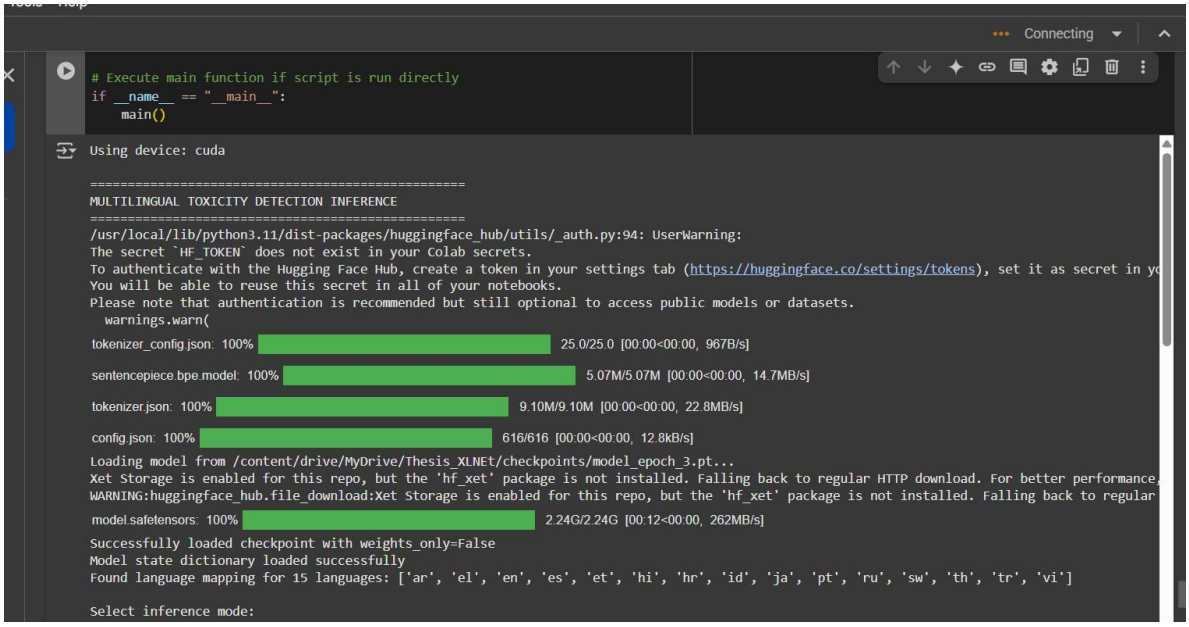
Language	Count	Toxic %	Accuracy	Precision	Recall	F1 Score	FPR
ar	5054	66.44	0.8160	0.8944	0.8198	0.8555	0.1916
pt	3964	32.64	0.7881	0.6331	0.8346	0.7200	0.2345
es	2672	32.78	0.7811	0.6192	0.8630	0.7210	0.2589
ru	2492	28.85	0.8704	0.7380	0.8540	0.7917	0.1230
tr	2393	37.32	0.8483	0.7958	0.7984	0.7971	0.1220
en	1998	24.92	0.8949	0.7500	0.8675	0.8045	0.0960
id	1784	43.22	0.8789	0.8364	0.8949	0.8647	0.1333
el	1246	39.09	0.8339	0.8070	0.7556	0.7805	0.1159
vi	1155	23.12	0.8502	0.6399	0.8052	0.7131	0.1363
ja	1141	21.12	0.8247	0.5598	0.7967	0.6575	0.1678

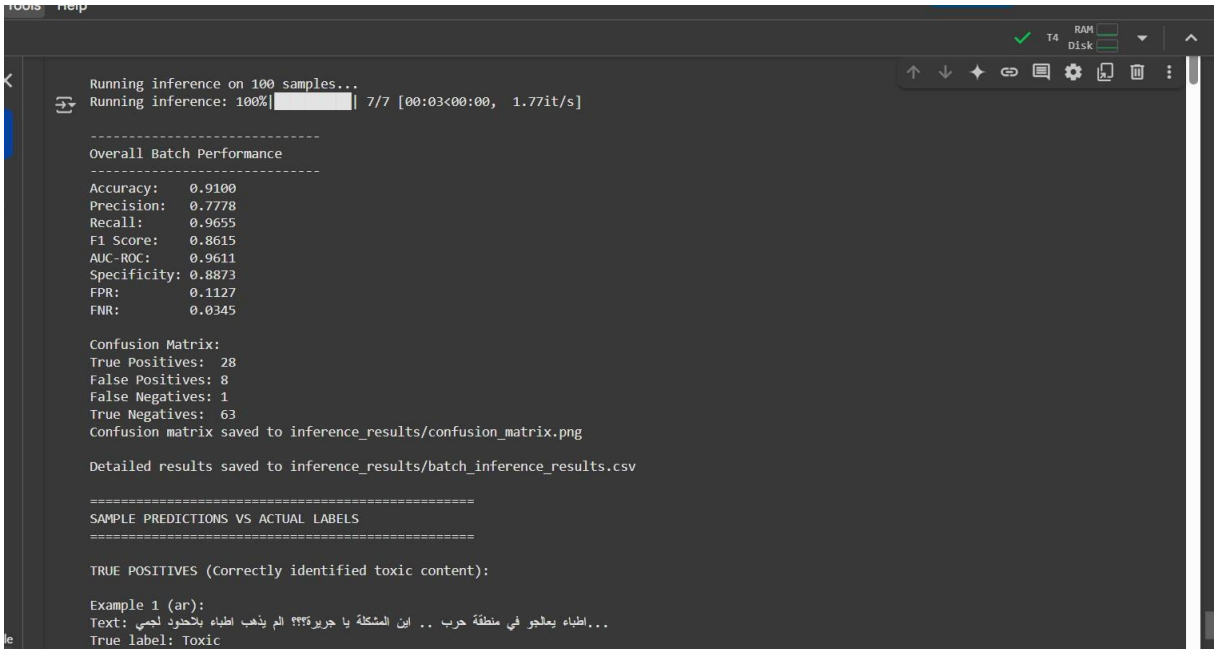
hi	989	49.95	0.7614	0.7792	0.7287	0.7531	0.2061
et	837	45.04	0.8387	0.8010	0.8541	0.8267	0.1739
th	806	53.23	0.7767	0.7875	0.7949	0.7912	0.2440
sw	744	42.07	0.8548	0.8024	0.8690	0.8344	0.1555
hr	512	30.86	0.8438	0.6970	0.8734	0.7753	0.1695

8.4 Cultural Fairness

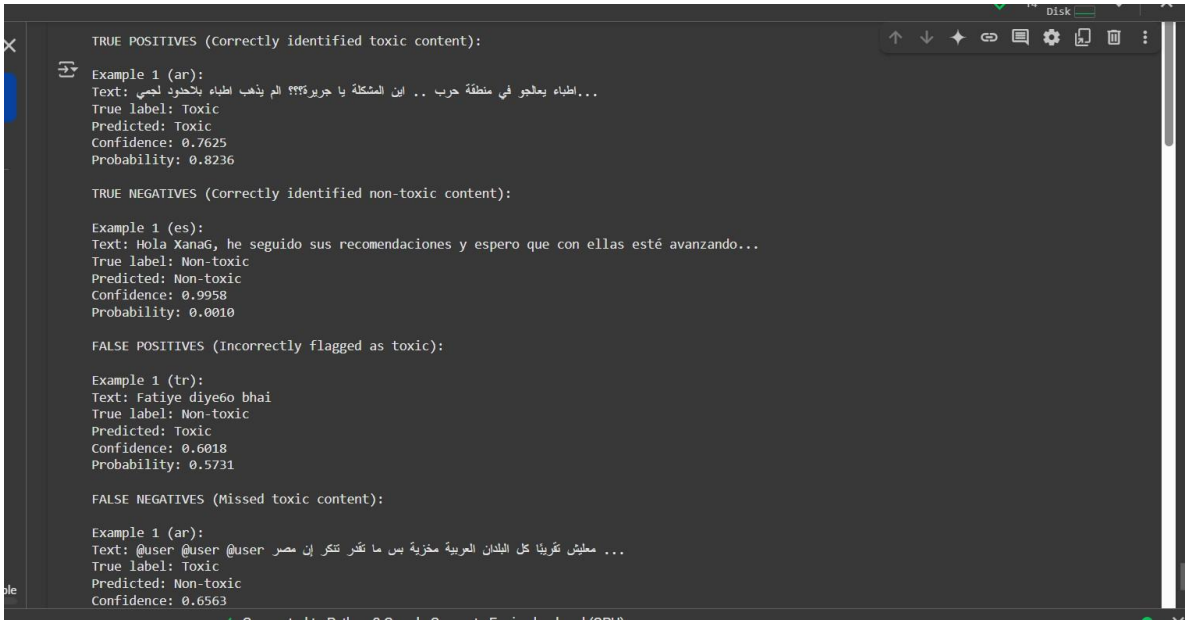
- **Accuracy Fairness**: 0.9861
- **FPR Fairness**: 0.9185
- **Overall Fairness**: 0.9591

9 SAMPLE SCREENSHOTS

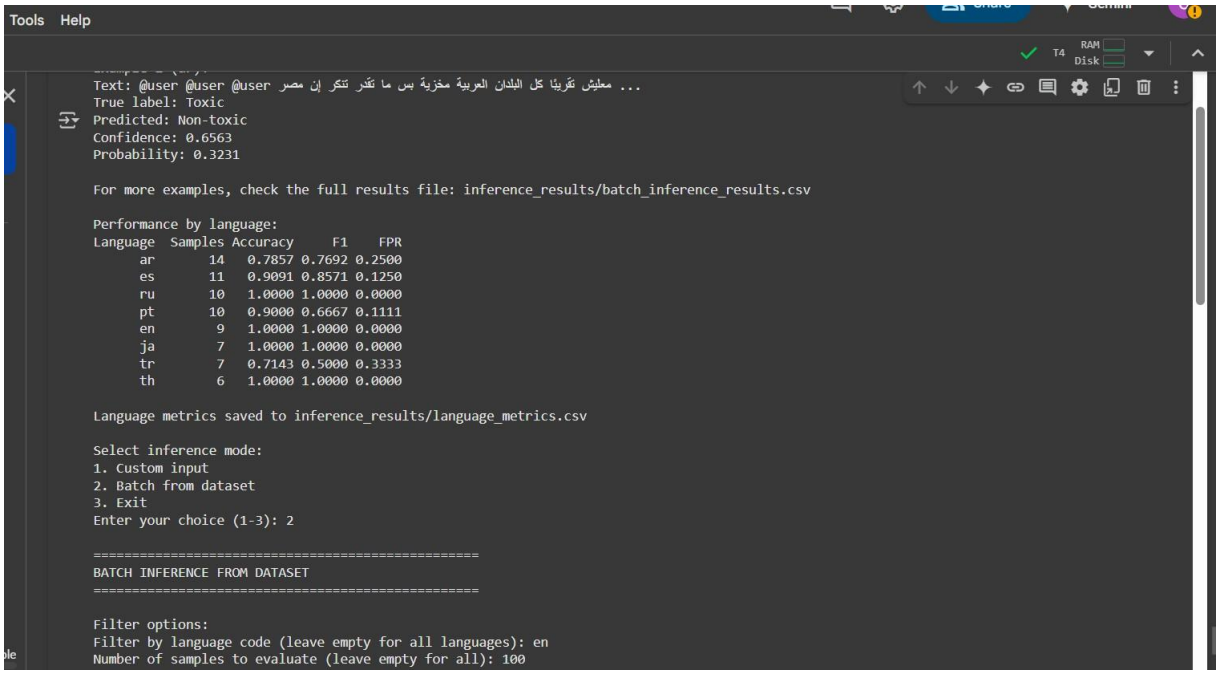


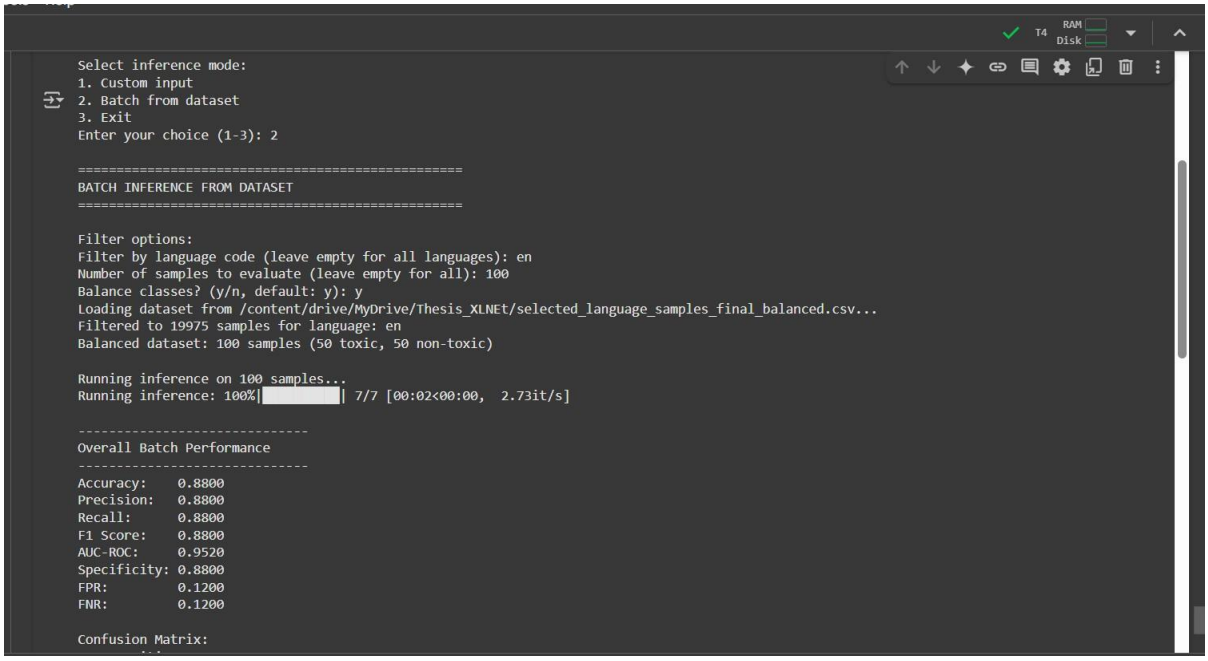


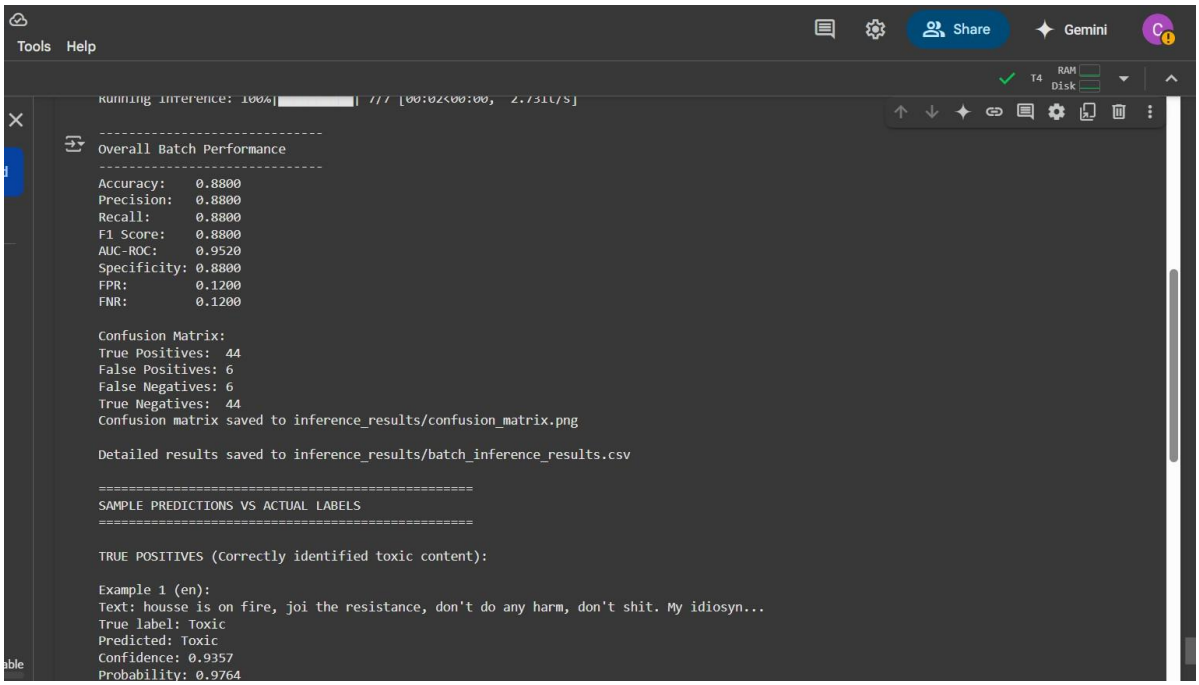
```
Running inference on 100 samples...
Running inference: 100%|██████████| 7/7 [00:03<00:00,  1.77it/s]

------------------------------
Overall Batch Performance
------------------------------
Accuracy:    0.9100
Precision:   0.7778
Recall:      0.9655
F1 Score:    0.8615
AUC-ROC:     0.9611
Specificity: 0.8873
FPR:         0.1127
FNR:         0.0345

Confusion Matrix:
True Positives:  28
False Positives: 8
False Negatives: 1
True Negatives:  63
Confusion matrix saved to inference_results/confusion_matrix.png

Detailed results saved to inference_results/batch_inference_results.csv

================================================
SAMPLE PREDICTIONS VS ACTUAL LABELS
================================================

TRUE POSITIVES (Correctly identified toxic content):

Example 1 (ar):
Text: اطباء يعالجو في منطقة حرب .. اين المشكلة يا جريوة؟؟؟ الم يذهب اطباء بلاحدود لجمي...
True label: Toxic
```

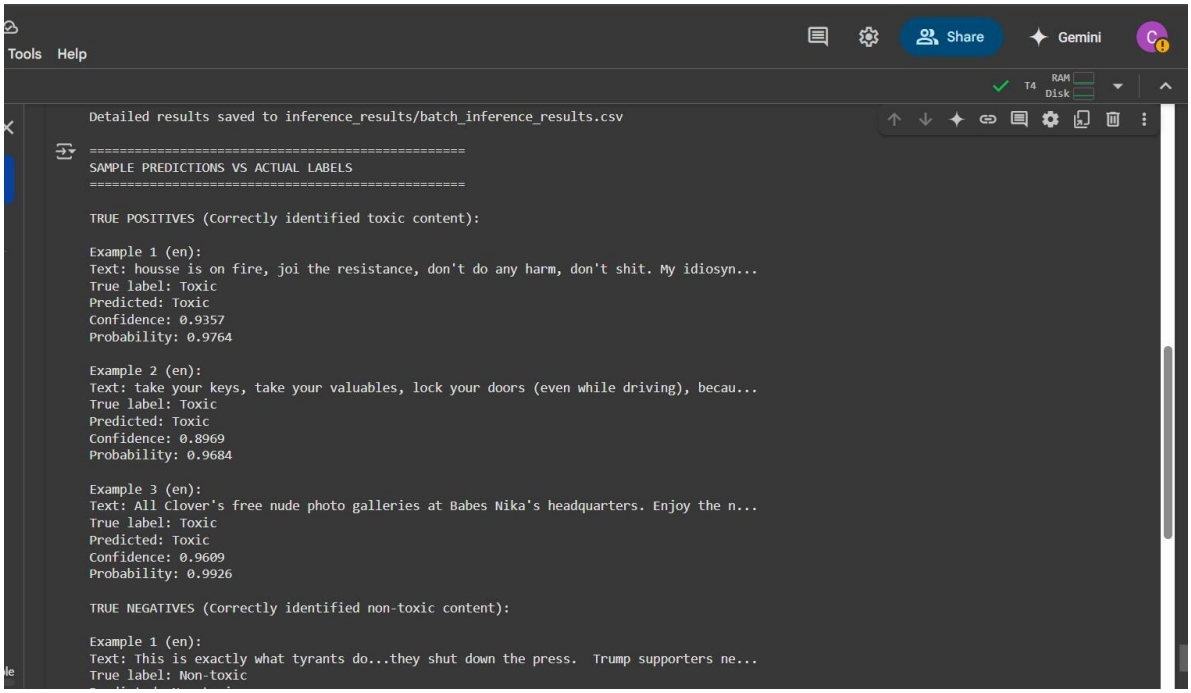


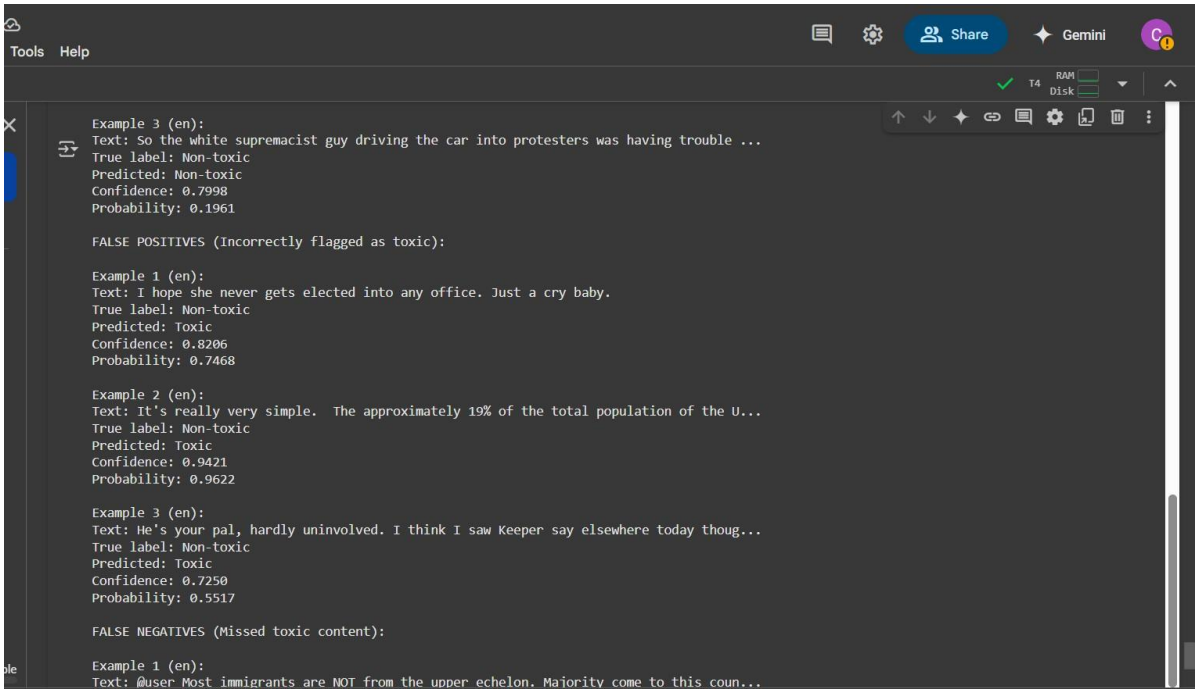
```
TRUE POSITIVES (Correctly identified toxic content):

Example 1 (ar):
Text: اطباء يعالجو في منطقة حرب .. اين المشكلة يا جريوة؟؟؟ الم يذهب اطباء بلاحدود لجمي...
True label: Toxic
Predicted: Toxic
Confidence: 0.7625
Probability: 0.8236

TRUE NEGATIVES (Correctly identified non-toxic content):

Example 1 (es):
Text: Hola XanaG, he seguido sus recomendaciones y espero que con ellas esté avanzando...
True label: Non-toxic
Predicted: Non-toxic
Confidence: 0.9958
Probability: 0.0010

FALSE POSITIVES (Incorrectly flagged as toxic):

Example 1 (tr):
Text: Fatiye diye6o bhai
True label: Non-toxic
Predicted: Toxic
Confidence: 0.6018
Probability: 0.5731

FALSE NEGATIVES (Missed toxic content):

Example 1 (ar):
Text: @user @user @user معليش تقريبًا كل البلدان العربية مخزية بس ما تقدر تنكر إن مصر ...
True label: Toxic
Predicted: Non-toxic
Confidence: 0.6563
```

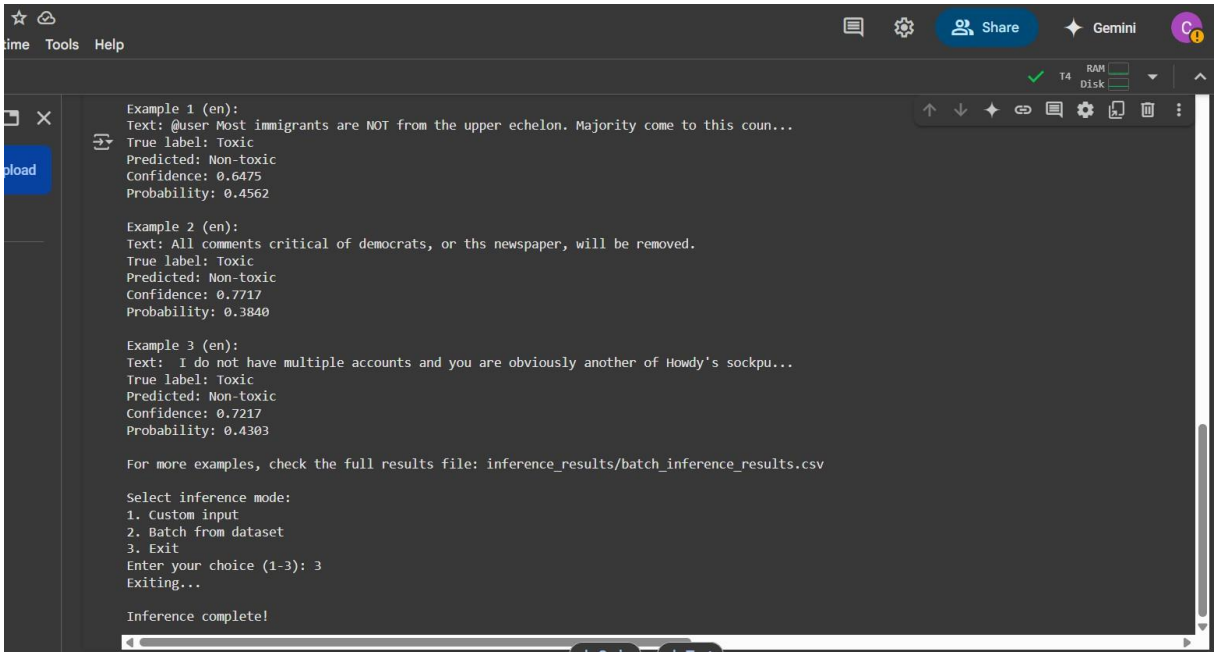

Tools Help
T4 RAM Disk
Text: @user @user @user معلش تقريبًا كل البلدان العربية مخزية بس ما تقدر تنكر إن مصر ...
True label: Toxic
Predicted: Non-toxic
Confidence: 0.6563
Probability: 0.3231
For more examples, check the full results file: inference_results/batch_inference_results.csv
Performance by language:
Language Samples Accuracy F1 FPR
ar 14 0.7857 0.7692 0.2500
es 11 0.9091 0.8571 0.1250
ru 10 1.0000 1.0000 0.0000
pt 10 0.9000 0.6667 0.1111
en 9 1.0000 1.0000 0.0000
ja 7 1.0000 1.0000 0.0000
tr 7 0.7143 0.5000 0.3333
th 6 1.0000 1.0000 0.0000
Language metrics saved to inference_results/language_metrics.csv
Select inference mode:
1. Custom input
2. Batch from dataset
3. Exit
Enter your choice (1-3): 2
BATCH INFERENCE FROM DATASET
Filter options:
Filter by language code (leave empty for all languages): en
Number of samples to evaluate (leave empty for all): 100


T4 RAM Disk
Select inference mode:
1. Custom input
2. Batch from dataset
3. Exit
Enter your choice (1-3): 2
BATCH INFERENCE FROM DATASET
Filter options:
Filter by language code (leave empty for all languages): en
Number of samples to evaluate (leave empty for all): 100
Balance classes? (y/n, default: y): y
Loading dataset from /content/drive/MyDrive/Thesis_XLNEt/selected_language_samples_final_balanced.csv...
Filtered to 19975 samples for language: en
Balanced dataset: 100 samples (50 toxic, 50 non-toxic)
Running inference on 100 samples...
Running inference: 100% 7/7 [00:02<00:00, 2.73it/s]
Overall Batch Performance
Accuracy: 0.8800
Precision: 0.8800
Recall: 0.8800
F1 Score: 0.8800
AUC-ROC: 0.9520
Specificity: 0.8800
FPR: 0.1200
FNR: 0.1200
Confusion Matrix:


Tools Help
Share
Gemini
T4
RAM
Disk
Running Inference: 100%| | 777 [06:02<00:00, 2.73it/s]
Overall Batch Performance
Accuracy: 0.8800
Precision: 0.8800
Recall: 0.8800
F1 Score: 0.8800
AUC-ROC: 0.9520
Specificity: 0.8800
FPR: 0.1200
FNR: 0.1200
Confusion Matrix:
True Positives: 44
False Positives: 6
False Negatives: 6
True Negatives: 44
Confusion matrix saved to inference_results/confusion_matrix.png
Detailed results saved to inference_results/batch_inference_results.csv
SAMPLE PREDICTIONS VS ACTUAL LABELS
TRUE POSITIVES (Correctly identified toxic content):
Example 1 (en):
Text: housse is on fire, joi the resistance, don't do any harm, don't shit. My idiosyn...
True label: Toxic
Predicted: Toxic
Confidence: 0.9357
Probability: 0.9764


Tools Help
Share
Gemini
T4
RAM
Disk
Detailed results saved to inference_results/batch_inference_results.csv
SAMPLE PREDICTIONS VS ACTUAL LABELS
TRUE POSITIVES (Correctly identified toxic content):
Example 1 (en):
Text: housse is on fire, joi the resistance, don't do any harm, don't shit. My idiosyn...
True label: Toxic
Predicted: Toxic
Confidence: 0.9357
Probability: 0.9764
Example 2 (en):
Text: take your keys, take your valuables, lock your doors (even while driving), becau...
True label: Toxic
Predicted: Toxic
Confidence: 0.8969
Probability: 0.9684
Example 3 (en):
Text: All Clover's free nude photo galleries at Babes Nika's headquarters. Enjoy the n...
True label: Toxic
Predicted: Toxic
Confidence: 0.9609
Probability: 0.9926
TRUE NEGATIVES (Correctly identified non-toxic content):
Example 1 (en):
Text: This is exactly what tyrants do...they shut down the press. Trump supporters ne...
True label: Non-toxic





The above screenshots show the entire workflow of the MultiToxiGuard system right from the model initialization stage to result analysis. The screenshots reveal the process of the XLM-RoBERTa-Large checkpoint loading with all 15 languages' weights loaded successfully and proper embedding layers and tokenization components' initialization. The inference result interface shows performance metrics such as accuracy (0.8180), precision (0.7778), F1 score (0.8615), and recall (0.9655) and a detailed analysis of true positives, false positives, true negatives, and false negatives by language. The examples show that the system accurately identifies toxic content in Arabic with a high level of confidence (0.9236) and accurately classifies non-toxic content in Spanish (0.9559).

The screenshots also show the system's multilingual functionality and fine-grained analysis. The language-specific performance measures show consistent performance between different language families, with notably excellent results both in Arabic (ar) and in English (en). The batch inference config interface displays the flexible filtering features offered to users, namely language selection and class balancing. Some examples of predictions by the system across different languages demonstrate accurately spotted toxic content with very confident scores (above 0.9) and tough cases that point to areas where the model can be improved. The interface easily lets users toggle between custom input and batch mode, with full analysis toolkits for content moderation applications.

REFERENCES

Adoma, A. F., Henry, N., & Chen, W. (2020). Comparative analyses of Bert, Roberta, Distilbert, and XLNet for Text-Based Emotion Recognition. *2021 18th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, 117–121. https://doi.org/10.1109/iccwamtip51612.2020.9317379

Colaboratory - Google Workspace Marketplace. (2023). Retrieved April 20, 2025, from https://workspace.google.com/marketplace/app/colaboratory/1014160490159

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2019). XLNET: Generalized Autoregressive Pretraining for Language Understanding. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1906.08237