Performance Analysis Report

Project Overview

This project aims to build a multilingual toxicity detection model. Fist model was built leveraging **FastText embeddings**, a **Bidirectional LSTM architecture**, and language-aware preprocessing.
The second one used **BERT** to leverage more accuracy and performance to the model. The training data includes various types of toxic behavior (toxic, abusive, vulgar, menace, offense, bigotry), which were combined into a binary classification target.

Model Performance Summary

Overall Accuracy

- **Test Accuracy**: `77.38%` **`(For LSTM+FastText)`**
  This means that the model is correctly classifying approximately 3 out of every 4 samples. Due to limited memory on the device, only 4 epochs were run. With more epochs, it's likely that we could have achieved greater accuracy.
- **Test Accuracy**: `77.28%` **`(For Bert based multilingual model)`**
  This means it has almost the same accuracy as the last model. Approximately 3 out of 4 samples can be detected in this case. In this case I've run 3 epoch for limited memory issue. It's likely to gain more accuracy if the epoch count was bigger.

Classification Metrics

(For LSTM+FastText)

Metric	Non-Toxic (0)	Toxic (1)
Precision	0.77	0.64
F1-Score	0.87	0.02
Support	4637	1363

(For BERT)

Metric	Non-Toxic (0)	Toxic (1)
Precision	0.77	0.62
F1-Score	0.89	0.02
Support	4633	1367

Key Observations:

- **Precision for toxic (1)** is moderate for both, indicating that when the model predicts that it is toxic, it is often correct.
- The low **F1-score** for the toxic class highlights an opportunity to further enhance the model's ability to detect toxic content, especially in more nuanced or less frequent cases.

Weighted & Macro Averages:

- **Macro F1-score**: `0.45` → Low due to imbalance and recall drop in toxic class.
- **Weighted F1-score**: `0.68` → Heavily skewed by performance on non-toxic class.

Class Imbalance Effect

The dataset is imbalanced: toxic samples are the minority. The model has learned to predict **non-toxic** most of the time to optimize accuracy, which inflates performance metrics but fails to solve the real problem (toxic detection).

Performance by Language

Language	Samples	Accuracy
Turkish (`tr`)	1341	**89.04%**
Portuguese (`pt`)	1059	82.53%
Italian (`it`)	799	80.48%
Russian (`ru`)	1018	76.33%
French (`fr`)	1026	69.88%
Spanish (`es`)	757	**57.86%**

Insights: (For LSTM)

- Best performance: **Turkish (tr)** – likely due to cleaner, more distinguishable patterns in the training set or lower noise.
- **Spanish (es)** and **French (fr)** lag significantly — likely due to:
    - Vocabulary mismatch with FastText embeddings.
    - Less representativeness in training data.
    - It can be a tokenization issue with accented characters or different linguistic structures.

Strengths (For both)

- Successfully integrated **FastText** multilingual embeddings.
- Cleaned and engineered several useful **textual features** (e.g., word/char counts, stopwords).
- Used **TPU strategy** and **EarlyStopping** to speed up training and avoid overfitting.
- Achieved **good accuracy** on non-toxic comments.

Weaknesses / Areas for Improvement (for both)

1. **Class Imbalance**:
    - The model shows a strong bias toward non-toxic predictions, which contributes to a high overall accuracy.
    - This gives a valuable opportunity to fine-tune the model for better sensitivity to toxic content in imbalanced scenarios.
2. **Language-Specific Performance Gaps**:
    - Performance drops in French and Spanish indicate room for more language-specific preprocessing or model fine-tuning.
3. **Fixed FastText Embeddings**:
    - `trainable=False` limits adaptation to dataset-specific contexts.
    - May hinder toxic context detection, especially in less frequent phrases or languages.

Future Upgradation

If time was not an issue followed modifications could be done to leverage the performance of both models.

1. **Handling Class Imbalance**:
    - Using **oversampling** (e.g., SMOTE) or **class weighting** during training might solve most issues.
    - Trying **focal loss** instead of binary cross-entropy to emphasize harder-to-classify toxic samples may increase the overall performance
2. **Multi-Channel or Attention Models**:
    - Incorporate **attention mechanisms** or **Transformer-based layers**.
    - Experiment with **Multilingual BERT (mBERT)** or **XLM-RoBERTa** for contextual, language-specific understanding. **BERT** model can be upgraded with more fine tuning.
3. **Language-Aware Training**:
    - To `lang` as an **additional input feature** may increase the model potential.
    - Alternatively, **training separate models per language** if enough data is available.
4. **Language-Specific Tokenization**:
    - Integrate libraries like **spaCy** or **Stanza** to tokenize and process language-specific rules better.