

Efficient Multi-Class Toxicity Detection on Edge Devices Using Quantized Large Language Models

A THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

**MASTER OF TECHNOLOGY
IN
COMPUTER SCIENCE & ENGINEERING**

Submitted by

VINAY GREWAL (2K24/CSE/11)

Under the supervision of

PROF. ARUNA BHAT



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY**

(Formerly Delhi College of Engineering)

Bawana Road, Delhi 110042

MAY, 2026

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, **Vinay Grewal (2K24/CSE/11)**, hereby certify that the work which is being presented in the major project report II entitled **“Efficient Multi-Class Toxicity Detection on Edge Devices Using Quantized Large Language Models”** in partial fulfillment of the requirements for the award of the Degree of Master of Technology, submitted in the **Department of Computer Science and Engineering**, Delhi Technological University is an authentic record of my own work carried out during the period from Start Date, e.g., August 2024 to May 2026 under the supervision of **Prof. Aruna Bhat**.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other Institute.



Candidate's Signature

This is to certify that the student has incorporated all the corrections suggested by the examiners in the thesis and the statement made by the candidate is correct to the best of our knowledge.



Signature of Supervisor

Signature of External Examiner

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project titled “**Efficient Multi-Class Toxicity Detection on Edge Devices Using Quantized Large Language Models**”, submitted by **Vinay Grewal**, Roll No. **2K24/CSE/11**, Department of **Computer Science & Engineering**, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of **Master of Technology (M.Tech)** in Computer Science and Engineering is a genuine record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree to this University or elsewhere.



Place : Delhi

Dr. Aruna Bhat

Date : 29/05/2026

Professor

Delhi Technological University

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

I would like to express my sincere gratitude and appreciation to my supervisor, **Prof. Aruna Bhat**, Department of Computer Science and Engineering, Delhi Technological University, for her continuous guidance, valuable insights, and constant encouragement throughout the course of this project. Her expertise in the field and her willingness to support me at every stage of the research were instrumental in shaping this work.

I am also deeply thankful to the faculty members of the Department of Computer Science and Engineering at Delhi Technological University for their academic support and for fostering an environment that encourages research and inquiry. Their courses and discussions have helped me build the foundation on which this thesis is based.

I extend my heartfelt thanks to my classmates and fellow research scholars for the many stimulating conversations, technical discussions, and mutual support that made the journey of M.Tech research enriching.

Finally, I am profoundly grateful to my family for their unwavering support, patience, and encouragement throughout my studies. Without their belief in me, this work would not have been possible.

Place: Delhi

VINAY GREWAL

24/CSE/11

Date: 24.05.2026

ABSTRACT

The growth of online platforms has increased the need for automated systems that can identify harmful comments at scale. Toxic comment classification is a challenging natural language processing problem because harmful language varies in tone, context and intensity. Transformer-based language models provide strong contextual representations, but their size and computational requirements can limit deployment in resource-constrained environments. This dissertation presents a resource-efficient toxic comment classification pipeline using fine-tuned and INT-8 dynamically quantized transformer models.

The study uses the Jigsaw toxicity dataset and converts its continuous toxicity annotations into a three-class classification task: Safe, Toxic and Obscene. The original mapped dataset is highly imbalanced, containing 1,656,908 Safe comments, 138,360 Toxic comments and 9,603 Obscene comments. To reduce class imbalance, the dataset is down-sampled to 9,603 examples per class, producing a balanced dataset of 28,809 comments. The balanced dataset is split into 20,166 training samples, 4,321 validation samples and 4,322 test samples.

Three transformer models are evaluated: DistilBERT, Toxic-RoBERTa and RoBERTa. Each model is tested in three stages: base model with an untrained classification head, fine-tuned model and INT-8 dynamically quantized model. The models are compared using accuracy, macro F1-score, macro precision, macro recall, latency and model size. Toxic-RoBERTa achieves the best fine-tuned result with accuracy of 0.8987 and macro F1-score of 0.8985. After INT-8 dynamic quantization, Toxic-RoBERTa retains nearly the same performance with accuracy of 0.8984 and macro F1-score of 0.8982 while reducing size from 475.5 MB to 148.9 MB.

The results show that fine-tuning is essential for the three-class toxicity task and that dynamic quantization can significantly reduce model size with small performance loss. The thesis recommends quantized Toxic-RoBERTa as the best overall choice when accuracy and storage efficiency are both important. It also highlights that latency comparison should be performed under matched hardware

conditions because the quantized models in this experiment run on CPU while the fine-tuned models were evaluated on GPU.

Keywords: Toxicity Detection, Jigsaw Dataset, Transformer Models, DistilBERT, RoBERTa, Toxic-RoBERTa, Fine-Tuning, Dynamic Quantization, INT-8.

TABLE OF CONTENTS

Candidate's Declaration	ii
Certificate by the Supervisor	iii
Acknowledgements	iv
Abstract.....	v
List of Tables	viii
List of Figures.....	ix
List of Symbols, Abbreviations and Nomenclature	x
Chapter 1 Introduction.....	1
1.1 Background	1
1.2 Motivation	2
1.3 Problem Statement	2
1.4 Objectives of the Work.....	3
1.5 Scope of the Thesis.....	3
1.6 Organization of the Thesis.....	4
Chapter 2 Literature Review	5
Chapter 3 Dataset Preparation and Preprocessing.....	8
Chapter 4 Proposed Methodology	12
Chapter 5 Experimental Setup and Implementation.....	15
Chapter 6 Results and Discussion.....	18
Chapter 7 Conclusion and Future Scope	23
References.....	26
Appendix A Notebook Workflow Summary	27
Appendix B Full Experimental Result Table	28

LIST OF TABLES

Table 3.1	Jigsaw columns used in label mapping.
Table 3.2	Three-class label mapping strategy.
Table 3.3	Representative sanitized examples of final dataset format.
Table 3.4	Dataset distribution before and after balancing.
Table 3.5	Train, validation and test split sizes.
Table 4.1	Model registry used in the experiment.
Table 5.1	Fine-tuning hyperparameters.
Table 6.1	Full performance comparison across models and stages.
Table 6.2	Fine-tuned confusion matrix diagonal summary.
Table 6.3	Model size reduction after INT-8 quantization.

LIST OF FIGURES

Figure 6.1	Performance comparison across base, fine-tuned and quantized stages.
Figure 6.2	Radar comparison of fine-tuned and INT-8 quantized models.
Figure 6.3	Confusion matrices for fine-tuned models on the test set.
Figure 6.4	Model size comparison between FP32 fine-tuned and INT-8 quantized versions.

LIST OF SYMBOLS, ABBREVIATIONS AND NOMENCLATURE

Symbol	Meaning
AI	Artificial Intelligence
BERT	Bidirectional Encoder Representations from Transformers
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
DL	Deep Learning
F1	Harmonic mean of precision and recall
FP32	32-bit floating point representation
GPU	Graphics Processing Unit
HF	Hugging Face
INT-8	8-bit integer representation
ML	Machine Learning
NLP	Natural Language Processing
RoBERTa	Robustly Optimized BERT Pretraining Approach
TP/FP/FN	True Positive, False Positive and False Negative

CHAPTER 1

INTRODUCTION

1.1 Background

Discussion platforms have emerged as vital tools in the digital era. News sites, educational sites, social networking sites, online gaming communities and enterprise forums all depend on user commentary to facilitate discourse. While this openness serves as the strength of such platforms, the same attribute presents moderation problems. Comments that are toxic serve to hinder discourse and even put users at risk of harm. Moderation through manual means becomes impossible considering the sheer number of posts made on any platform coupled with the fact that comments are open to linguistic nuances and online jargon.

Machine learning has, therefore, become an essential part of moderation systems. Previous systems made use of handcrafted features, counts of words, lexicographic markers, or classical term frequency-inverse document frequency vectors to represent comments. They were relatively simple but did not manage to take into account context and sarcasm. Deep learning models improved representation significantly, while transformer-based language models managed to learn context-sensitive token representations by training on large amounts of text data. Examples include BERT, DistilBERT and RoBERTa, which can be fine-tuned for specific tasks.

This thesis aims at studying the classification of toxic comments via transformer models on the basis of the Jigsaw data set. In the raw data set, there exist annotations regarding toxicity with continuous labels such as target toxicity, severe toxicity, obscene, insult, threat and identity attack. For the current study, these annotations are converted into a three-way classification problem involving classes of Safe, Toxic and Obscene. Three transformer architectures are examined: DistilBERT, RoBERTa and Toxic-RoBERTa. Each architecture is evaluated in three steps: base architecture, fine-tuned architecture and INT-8 dynamically quantized architecture.

1.2 Motivation

The effectiveness of a moderation model depends on its ability to accurately predict undesirable content and be computationally feasible enough for execution in real conditions. High accuracy is usually guaranteed by large transformer-based architectures, but excessive memory usage and inference time can impede model deployment on resource-constrained machines. It may be more desirable to have a model with lower accuracy but smaller footprint than a more precise but bulky architecture. That is the reason for this research.

The raw Jigsaw dataset is significantly imbalanced. There are more safe comments than toxic comments, and the number of obscene comments is less compared to safe comments. Using the above-mentioned dataset as it is for training results in biasing the classifier, which tends to predict the majority class too often. To achieve an unbiased comparison, we have to balance, filter out and define labels properly. For this reason, equal class sampling is applied to generate 28,809 comments, each class having 9,603 samples.

A further goal is to measure the influence of post-training quantization. INT-8 dynamic quantization can shrink the model by quantizing the floating-point numbers used as weights in the linear layers to 8-bit integers at inference time. Dynamic quantization does not necessitate additional training or calibration data, which makes it easy to implement post-fine-tuning. The fundamental issue here is whether quantization results in lossless classification performance on the toxicity classification task.

1.3 Problem Statement

The main issue that this thesis focuses on is the creation and analysis of a model for the detection of toxic comments in three classes namely safe, toxic and obscene comments. In this process, a model has to be created which can convert the raw Jigsaw toxicity annotations to a single class label, create a balanced data set, fine-tuning of transformer models, applying dynamic quantization at INT-8 level and comparing these models.

Mathematically, for the given input data point x , the algorithm should predict a single output label y such that the label belongs to the set $\{\text{Safe, Toxic, Obscene}\}$. Classification of comments is done on the basis of single-label multi-class classification. For Obscenity labeling of the comment, the obscene score passes the threshold level. Similarly, for toxic labeling, other relevant scores pass the threshold level. Otherwise, the comment is classified as safe.

1.4 Objectives of the Work

Firstly, a dataset comprising only three classes that are free from any form of contamination will be created from the Jigsaw toxic data. This entails selecting pertinent columns, converting the continuous toxicity ratings to class labels, and finally reducing the number of instances of the majority class.

The second aim is to tune three transformer models based on the same experimental setting. DistilBERT denotes a small-sized transformer benchmark model, RoBERTa denotes a powerful encoder, and Toxic-RoBERTa denotes a specialized toxicity-trained model. The training and evaluation are done using the same train/validate/test split.

The third goal would be to run INT-8 dynamic quantization on the fine-tuned model and assess its accuracy, F1 score, precision, recall, latency, and size. The third goal seeks to find an answer to the practical question whether compressed transformer models can be applied to toxic text classification with minimal losses.

Objective number four will entail interpreting the findings via the use of metric tables, confusion matrices, radar charts, and size comparison charts. Objective number five entails recommending an appropriate model to deploy depending on certain criteria like maximum accuracy, minimum size, among others.

1.5 Scope of the Thesis

The scope of the research is restricted to the topic of toxicity detection in English language comments through the use of Jigsaw Toxic Comment Classification and encoder models for text based on transformers. The study considers only three types of labels, utilizing a fixed threshold label mapping

approach. It does not cover multi-label prediction, multi-lingual toxicity detection, online learning or human intervention during the process of labeling comments. For quantization, PyTorch dynamic quantization of linear layers was chosen.

However, one should note that the latency results are not directly comparable due to the fact that fine-tuned models were measured using GPU latency, whereas dynamically quantized models were tested using CPU due to their native support for dynamic quantization. However, the difference in hardware was caused by the practical needs of the experiment. The first practical deployment benefit seen in the experiment is the reduction of the size of the model.

1.6 Organisation of the Thesis

Chapter 1 introduces the motivation, problem statement and objectives.

Chapter 2 reviews the literature on toxicity detection, transformer language models and model compression.

Chapter 3 explains the Jigsaw dataset, label mapping, filtration, balancing and preprocessing steps.

Chapter 4 presents the proposed methodology and model pipeline.

Chapter 5 describes implementation details, training configuration and evaluation metrics.

Chapter 6 presents the experimental results and discusses model behavior.

Chapter 7 concludes the thesis and outlines future research directions.

CHAPTER 2

LITERATURE REVIEW

2.1 Toxicity Detection in Online Text

Toxicity detection refers to the process of detecting those comments which could potentially cause harm or abuse, such as insults or threats to the user. It is similar to hate speech detection, abusive language detection, and content moderation; however, they are not entirely the same. A comment that is toxic may be an insult to someone else without falling under the category of hate speech, and a discussion on a sensitive topic may have identity words without being classified as toxic.

The earlier toxicity classifier approaches utilized linguistic features including n-gram, term frequency inverse document frequency vector, sentiment scores, and offensive word lists. Such algorithms would train well, although they lacked robustness as a result of the changes in user's spelling, phrases, and comments that require contextual understanding. Neural models contributed to better word and sequence representation using dense embedding representations. Recurrent models like LSTM and GRU network architecture were more popular before transformer encoder architectures took over.

One problem that has been consistent with toxicity datasets is bias. Bias may occur when the model identifies unintended correlations between mentions of identity and labels for toxicity. For instance, neutral comments on a protected class could mistakenly be flagged as harmful if the term for the protected class occurs frequently in the toxic dataset.

2.2 Transformer Encoders for Text Classification

Transformers used self-attention instead of recurrence. With a transformer encoder, all tokens have attention over other tokens in the input sequence, thus enabling the model to create contextualized embeddings. The idea is especially useful in text classification, where the embedding of the same word could vary based on the surrounding words. For instance, a given word could be non-abusive in quotes or educationally related, but harmful when used directly.

BERT pioneered masked language modeling and next sentence prediction pre-training methods for obtaining bidirectional representations. In order to perform the task of classification, a classification head specific to the task is connected to the pooled representation of the sequence and the entire model is fine-tuned on labeled data. RoBERTa enhanced the training methodology used for BERT through increased batch size, more data, and pre-training changes. DistilBERT shrunk the model size by transferring knowledge from a large BERT model to a small student model.

This thesis is based on a comparison between a small-scale model, a general strong encoder, and a toxicity-focused encoder. Such a comparison is valuable since, when deciding to deploy the models, one needs to consider more than just accuracy alone.

2.3 Fine-Tuning of Pretrained Models

Fine-tuning is the process of adapting a pre-trained language model to perform a specific task by optimizing the parameters of the model using labeled data. In our current experiment, we transform each language model into a classifier with three output classes. After passing the tokenized comment into the encoder of the language model, we obtain logits corresponding to Safe, Toxic, and Obscene.

The fine-tuning approach proves helpful when the number of examples in the task-specific dataset is less than the size of the datasets in the pretraining phase. The pre-trained model has been trained on the linguistic structure. However, it is fine-tuned to learn how its internal representation can be translated into task-specific label predictions. All experiments have employed a common setup with 3 epochs, a learning rate of $2e-5$, cosine learning rate schedule, and macro F1-score.

2.4 Model Compression and INT-8 Quantization

Transformer models are known to be costly in terms of storage and execution. Techniques for model compression aim at lowering such costs. Some well-known techniques include pruning, knowledge distillation, low-rank decomposition, quantization, and efficient model design. Quantization entails reducing the level of

accuracy of weights or activations. Rather than storing the model weights using 32-bit float precision, quantization utilizes 8-bit integers, called INT-8.

Dynamic Quantization is a post-training approach where some specific layers such as linear layers are quantized using the model alone without the requirement of any extra data to train the model. In the inference process, weight parameters of the model are first stored as quantized values then dequantized whenever computations are done. The approach implemented in this project utilizes the dynamic quantization of PyTorch on linear layers.

2.5 Research Gap

Most work done in toxicity classification centers on making improvements in the predictive models, but comparatively less has been achieved regarding the actual process of data pre-processing through quantization. In the present thesis, an attempt is made to bridge this gap between theory and practice by incorporating all aspects within a single approach.

Another shortfall pertains to an honest interpretation of the results of compression. While a compressed model may be significantly reduced in size, it need not necessarily increase in speed. Both the measurements of size and latency are provided in the results, while the analysis distinguishes between size and runtime efficiencies.

2.6 Summary

There have been many pieces of literature highlighting the strength of using transformer encoders for text classification and the importance of using compression techniques for implementing these classifiers. This thesis explores both these dimensions through the comparative evaluation of fine-tuned and quantized transformers for the task of toxicity classification, which is discussed in the next chapter.

CHAPTER 3

DATASET PREPARATION AND PREPROCESSING

3.1 Source Dataset

This thesis uses the Jigsaw toxicity dataset available in train.csv in the project’s workspace. The raw data has 1,804,874 rows and 45 columns. The primary text column in the data is ‘comment_text’. Other columns that are relevant to toxicity and will be used for labeling are ‘target’, ‘severe_toxicity’, ‘obscene’, ‘identity_attack’, ‘insult’ and ‘threat’. Metadata columns like ‘article_id’, ratings and numbers of annotators are also present in the raw data but will not be used as inputs for modeling in this thesis.

The raw annotations are continuous scores rather than a ready-made three-class label. Each score represents the degree to which a comment is associated with a toxicity attribute. The project converts these continuous indicators into a single categorical label so that the downstream task becomes single-label classification. This conversion is described in the next section.

Table 3.1: Jigsaw columns used in label mapping.

Column	Use in this thesis
comment_text	Input text used by the classifier
target	General toxicity score used for Toxic mapping
severe_toxicity	Severe toxicity score used for Toxic mapping
obscene	Obscene score used for Obscene mapping
identity_attack	Identity attack score used for Toxic mapping
insult	Insult score used for Toxic mapping
threat	Threat score used for Toxic mapping

3.2 Label Mapping Strategy

The final label space contains three labels: Safe, Toxic and Obscene. The mapping gives priority to the obscene attribute. If the obscene score is greater than or equal to 0.5, the sample is assigned label 2, Obscene. If the obscene condition is not satisfied but at least one of target, severe_toxicity, insult, threat or identity_attack is greater than or equal to 0.5, the sample is assigned label 1, Toxic. If none of these conditions is met, the sample is assigned label 0, Safe.

This priority order prevents comments with strong obscene content from being merged into the broader Toxic category. It also preserves a meaningful distinction between general toxicity and obscene language. The mapping is simple, transparent and reproducible, which is useful for thesis evaluation.

Table 3.2: Three-class label mapping strategy.

Label ID	Class Name	Mapping Rule
2	Obscene	obscene $\geq$ 0.5
1	Toxic	obscene $<$ 0.5 and any of target, severe toxicity, insult, threat or identity attack $\geq$ 0.5
0	Safe	none of the toxicity conditions are satisfied

3.3 Filtration and Column Selection

Upon completion of labelling, only two fields are selected as inputs for model training: “comment_text” and “labels”. The field “comment_text” is changed to “text” for brevity in defining the structure of the dataset. Each row can be seen as a tuple of text and its corresponding label as an integer value. This is done since the objective is to classify toxicity based on comments' content.

The text will be converted into strings during tokenization, hence eliminating possible errors that may occur when there is no value or when the value is stored in a wrong data type. There will be no need for any rigorous cleansing of the text since the transformers' tokenizers are built to cope with punctuation, capitalization, and sub-word segmentation. Excessive cleansing can wash away useful information. The maximum number of tokens is 128.

3.4 Class Imbalance and Balancing

The dataset that results from mapping is extremely imbalanced. There are 1,656,908 instances in the Safe class, 138,360 instances in the Toxic class, and 9,603 instances in the Obscene class. If training is done according to this imbalance, the model would be able to obtain deceptive accuracy from the Safe class. To address this, each class is sampled to the size of the smallest class, which is 9,603.

The balanced dataset contains 28,809 samples in total, with equal representation of Safe, Toxic and Obscene. The balanced dataset is shuffled with

random_state = 42 to make the result reproducible. This balanced setting improves the reliability of macro F1-score because each class contributes equally to the final metric.

Table 3.4: Dataset distribution before and after balancing.

Class	Before balancing	After balancing
Safe	1,656,908	9,603
Toxic	138,360	9,603
Obscene	9,603	9,603
Total	1,804,871 after mapping count display	28,809

3.5 Train, Validation and Test Splits

The balanced dataset is split into training, validation and test sets using a 70:15:15 ratio. Stratified sampling is used to preserve class proportions in each split. The final split sizes are 20,166 training samples, 4,321 validation samples and 4,322 test samples. The test set is kept separate and is used only for final evaluation of the base, fine-tuned and quantized models.

A validation set is required for selecting the best checkpoint during fine-tuning. In this thesis, the best model is selected using macro F1-score. Macro F1 is appropriate because it gives equal weight to all classes rather than being dominated by the majority class.

Table 3.5: Train, validation and test split sizes.

Split	Ratio	Number of samples
Training	70%	20,166
Validation	15%	4,321
Test	15%	4,322

3.6 Representative Final Dataset Examples

The final dataset examples consist of a text field and a numeric label. Since toxic datasets can contain offensive language, the examples in this thesis are sanitized and shortened for presentation. They illustrate the final structure without reproducing harmful text in full. In the actual dataset, the model receives the original comment text and the mapped label.

Table 3.3: Representative sanitized examples of final dataset format.

Sanitized text example	Label	Class
This is so cool. It makes the discussion easier to follow.	0	Safe
Your argument is terrible and you are acting like an [insult removed].	1	Toxic
This comment contains [obscene expression removed] directed at another user.	2	Obscene

CHAPTER 4

PROPOSED METHODOLOGY

4.1 Overall Pipeline

The proposed pipeline begins with the raw Jigsaw CSV file and ends with a comparison of fine-tuned and quantized transformer models. The stages are: data loading, label mapping, column filtration, class balancing, train-validation-test split, tokenizer construction, dataset object creation, base model evaluation, fine-tuning, fine-tuned evaluation, INT-8 dynamic quantization, quantized model evaluation and visualization of results.

The same test set is used for all models and stages. This ensures that the reported differences are due to the model and stage rather than different samples. The base model stage evaluates the pretrained backbone with a fresh three-class classification head. Because this head has not learned the toxicity task, the base performance is expected to be low. This stage is included to show the value of task-specific fine-tuning.

4.2 Model Selection

Three models are selected to represent different design points. DistilBERT is included as a lightweight transformer model. It is smaller than BERT-like encoders and is useful when storage and speed are important. RoBERTa is included as a strong general-purpose transformer encoder. Toxic-RoBERTa is included because it has been pretrained or adapted for toxicity-related language, making it a strong candidate for this domain.

All three models are converted into sequence classification models with three output labels. The label mapping used in the model configuration is Safe = 0, Toxic = 1 and Obscene = 2. The same maximum sequence length and training configuration are used to maintain fairness across models.

Table 4.1: Model registry used in the experiment.

Model	Hugging Face checkpoint	Purpose
DistilBERT	distilbert-base-uncased	Compact general transformer baseline
Toxic-RoBERTa	unitary/unbiased-toxic-roberta	Toxicity-oriented RoBERTa variant

RoBERTa	roberta-base	Strong general-purpose transformer encoder
---------	--------------	--

4.3 Tokenization and Dataset Construction

Each comment is tokenized using the tokenizer corresponding to its model. Tokenization applies truncation, padding to a maximum length of 128 tokens and attention mask generation. The dataset object returns `input_ids`, `attention_mask` and `labels`. This format is compatible with Hugging Face Trainer as well as manual PyTorch evaluation loops.

The selection of max length 128 ensures information preservation without making computations too difficult. Online discussion toxic comments are usually short, and the selected max length allows capturing almost all comments without wasting unnecessary memory space. Longer comments are clipped by the tokenizer.

4.4 Fine-Tuning Procedure

The fine-tuning process happens for three epochs. The batch size for per-device training is 32 and that for evaluation is 64. The learning rate is set to $2e-5$, the warm-up ratio is 0.06, the weight decay is 0.01, and there is a cosine learning rate scheduler. If CUDA is available, then the model undergoes mixed precision training. Evaluation and checkpointing take place after each epoch ends. The checkpoint with the highest score based on macro F1-score is picked.

The training configuration is intentionally consistent across models. This makes the comparison easier to interpret and avoids introducing model-specific hyperparameter advantages. Although further tuning may improve individual results, the objective of the thesis is a controlled comparison rather than exhaustive hyperparameter search.

4.5 INT-8 Dynamic Quantization

After fine-tuning, each model is loaded in full precision on CPU and PyTorch dynamic quantization is applied to linear layers using `torch.quantization.quantize_dynamic`. The resulting INT-8 state dictionary is saved separately along with the tokenizer and configuration. For evaluation, the model

architecture is reconstructed, dynamic quantization is applied again and the saved INT-8 state dictionary is loaded.

This method reduces storage size significantly. It is also simple because it does not require calibration data or retraining. However, dynamic quantization in the implementation used is CPU-native. Therefore, latency should be interpreted relative to the CPU execution environment rather than as a direct comparison with GPU fine-tuned inference.

4.6 Evaluation Metrics

The evaluation metrics are accuracy, macro F1-score, macro precision, macro recall, latency per sample and model size. Accuracy measures the fraction of correctly classified comments. Macro precision, recall and F1-score compute the metric independently for each class and then average across classes. Macro F1-score is especially important for balanced multi-class classification because it reflects performance across all categories.

Confusion matrices are also used to study class-level errors. Each row of the confusion matrix corresponds to the true class and each column corresponds to the predicted class. The diagonal entries represent correct predictions, while off-diagonal entries show misclassifications.

CHAPTER 5

EXPERIMENTAL SETUP AND IMPLEMENTATION

5.1 Software and Hardware Environment

The notebook environment uses Python with PyTorch, Transformers, scikit-learn, NumPy, SciPy, PEFT and the Hugging Face Datasets library. The recorded versions include PyTorch 2.2.2 with CUDA 12.1 support and Transformers 4.38.2. CUDA availability was true during model training, enabling GPU execution for fine-tuning and fine-tuned evaluation.

The project was executed in a Google Colab style workflow with Google Drive mounted for saving models, plots, result summaries and the balanced dataset. The model outputs were stored in a Drive root folder named Thesis_ToxicityDetection. This structure helps preserve trained models and generated result files across notebook sessions.

5.2 Reproducibility Settings

The dataset sampling and train-test splitting use `random_state = 42`. The balanced dataset is created by sampling 9,603 examples from each class and shuffling the result. Stratified splitting preserves class balance across the training, validation and test sets. Reproducibility is important because random sampling can otherwise change the final distribution and model results.

The final thesis result files include a `comparison_table.csv` file, a `results_summary.json` file and four plots. These files are used to produce the tables and figures in Chapter 6. The notebook therefore acts not only as an experimental script but also as a reproducible record of the project workflow.

5.3 Model Training Implementation

The Hugging Face Trainer API is used for fine-tuning. The `compute_metrics` function calculates accuracy, macro F1-score, macro precision and macro recall from model logits. The best checkpoint is selected using macro F1-score. A custom

ToxicDataset class is also defined for manual PyTorch evaluation, returning tokenized input IDs, attention masks and labels.

The base model evaluation loads each pretrained model with `num_labels = 3` and `ignore_mismatched_sizes = True`. This attaches a newly initialized classification head where needed. The fine-tuning stage then updates the model parameters on the balanced training data. The final fine-tuned model and tokenizer are saved for each architecture.

Table 5.1: Fine-tuning hyperparameters.

Hyperparameter	Value
Number of epochs	3
Training batch size	32
Evaluation batch size	64
Learning rate	2e-5
Warmup ratio	0.06
Weight decay	0.01
Scheduler	Cosine
Best model criterion	Macro F1-score
Maximum sequence length	128 tokens
Mixed precision	Enabled when CUDA is available

5.4 Quantized Model Implementation

For dynamic quantization, the fine-tuned model is loaded on CPU and `quantize_dynamic` is applied to `nn.Linear` layers with `dtype = torch.qint8`. The quantized model state dictionary is saved as `model_int8.pt`. During evaluation, the same model shell is recreated, quantized and populated with the saved state dictionary. The quantized model is evaluated on the same test set using CPU.

The model size is computed and stored for the base, fine-tuned and quantized versions. Size reduction percentage is calculated as one minus the ratio of INT-8 size to fine-tuned FP32 size, multiplied by 100. This provides a clear measure of compression effectiveness.

5.5 Workflow Summary

- Load Jigsaw train.csv from Google Drive.
- Create labels using threshold-based mapping.
- Retain text and label columns.

- Sample 9,603 examples per class for balancing.
- Split into train, validation and test sets using stratification.
- Fine-tune DistilBERT, Toxic-RoBERTa and RoBERTa.
- Evaluate fine-tuned models on the test set.
- Apply INT-8 dynamic quantization to linear layers.
- Evaluate quantized models and export plots, CSV and JSON summaries.

CHAPTER 6

RESULTS AND DISCUSSION

Table 6.1: Full performance comparison across models and stages.

Model	Stage	Accuracy	F1-Macro	Precision	Recall	Latency(ms)	Size(MB)
DistilBERT	Base	0.3538	0.2791	0.2917	0.3538	0.18	255.4
DistilBERT	Finetuned	0.8609	0.8608	0.8608	0.8609	0.12	255.4
DistilBERT	Quantized	0.8445	0.8434	0.8499	0.8445	112.87	91.0
Toxic-RoBERTa	Base	0.2550	0.2089	0.1791	0.2549	0.36	475.5
Toxic-RoBERTa	Finetuned	0.8987	0.8985	0.8985	0.8987	0.19	475.5
Toxic-RoBERTa	Quantized	0.8984	0.8982	0.8984	0.8984	225.19	148.9
RoBERTa	Base	0.3334	0.1667	0.1111	0.3333	0.35	475.5
RoBERTa	Finetuned	0.8677	0.8677	0.8689	0.8676	0.19	475.5
RoBERTa	Quantized	0.8602	0.8592	0.8606	0.8603	231.77	148.9

6.1 Base Model Performance

The base models perform poorly compared with the fine-tuned models. This result is expected because the classification head is newly initialized for the three-class toxicity task. DistilBERT obtains a base accuracy of 0.3538 and macro F1-score of 0.2791. Toxic-RoBERTa obtains a base accuracy of 0.2550 and macro F1-score of 0.2089. RoBERTa obtains a base accuracy of 0.3334 and macro F1-score of 0.1667.

These results show that pretrained language representations alone are not sufficient when the task head has not learned the label mapping. Fine-tuning is necessary to align the model outputs with the Safe, Toxic and Obscene classes.

6.2 Fine-Tuned Model Performance

Fine-tuning dramatically improves all models. Toxic-RoBERTa achieves the best fine-tuned performance with accuracy of 0.8987 and macro F1-score of 0.8985. RoBERTa achieves accuracy of 0.8677 and macro F1-score of 0.8677. DistilBERT achieves accuracy of 0.8609 and macro F1-score of 0.8608. The strong result of Toxic-RoBERTa suggests that domain-oriented pretraining is beneficial for toxicity classification.

DistilBERT remains competitive despite being smaller. This makes it attractive for situations where memory and computation are constrained. RoBERTa performs better than DistilBERT but not as well as Toxic-RoBERTa on this task.

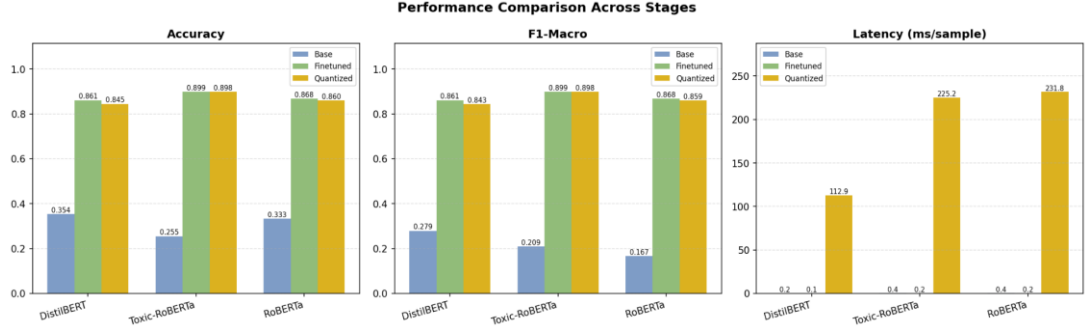


Figure 6.1: Performance comparison across base, fine-tuned and quantized stages.

6.3 Quantized Model Performance

INT-8 dynamic quantization preserves the performance of Toxic-RoBERTa almost completely. Its accuracy changes from 0.8987 to 0.8984 and macro F1-score changes from 0.8985 to 0.8982. This is a negligible loss for a 68.7 percent size reduction. RoBERTa also retains most of its performance, with accuracy changing from 0.8677 to 0.8602. DistilBERT shows a larger drop, from 0.8609 to 0.8445 accuracy.

This shows that there is variation in the effect of quantization depending on the model architecture used. RoBERTa specifically designed for toxicity detection is most resilient to dynamic quantization. In this case, DistilBERT is already compact and might be impacted by further quantization.

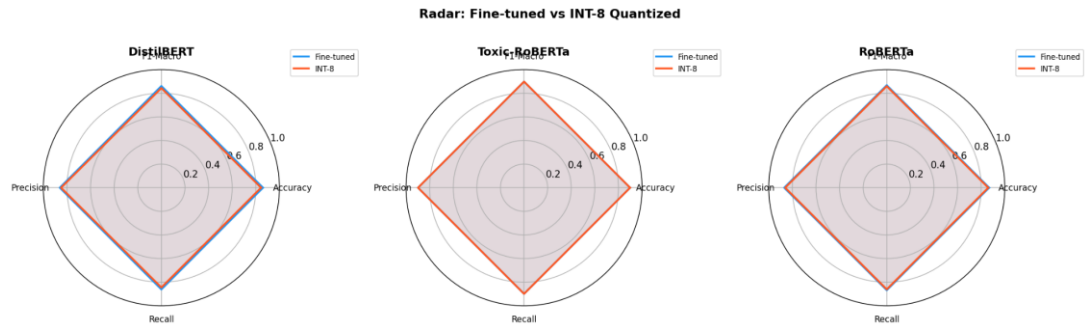


Figure 6.2: Radar comparison of fine-tuned and INT-8 quantized models.

6.4 Confusion Matrix Analysis

The confusion matrices show that the models perform well on the diagonal entries after fine-tuning. Toxic-RoBERTa has the highest correct predictions across all three classes: 1,304 Safe, 1,237 Toxic and 1,343 Obscene comments are correctly classified. DistilBERT correctly classifies 1,207 Safe, 1,189 Toxic and 1,325 Obscene comments. RoBERTa correctly classifies 1,191 Safe, 1,229 Toxic and 1,330 Obscene comments.

The most common errors involve confusion between Safe and Toxic, and between Toxic and Obscene. This is reasonable because some toxic comments may include strong language without crossing the obscene threshold, while some obscene comments may also be toxic in a broader sense. Toxic-RoBERTa reduces Safe-to-Toxic confusion compared with the other two models, which contributes to its superior overall performance.

Table 6.2: Fine-tuned confusion matrix diagonal summary.

Model	Safe correct	Toxic correct	Obscene correct
DistilBERT	1207	1189	1325
Toxic-RoBERTa	1304	1237	1343
RoBERTa	1191	1229	1330

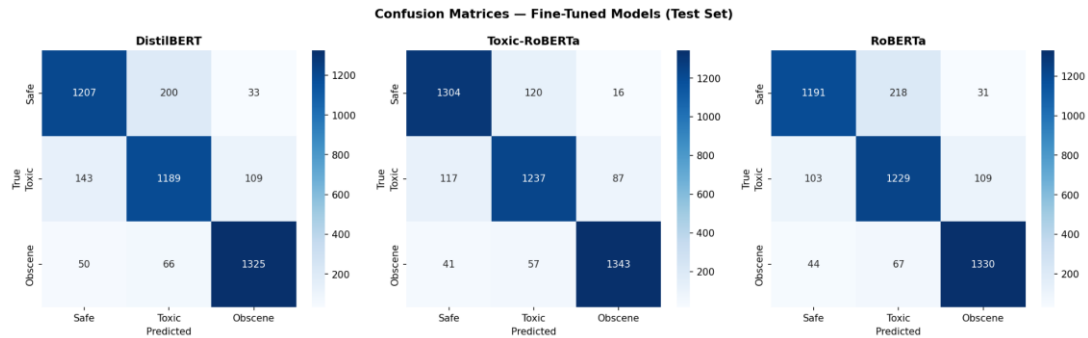


Figure 6.3: Confusion matrices for fine-tuned models on the test set.

6.5 Model Size and Deployment Trade-Off

The size comparison strongly supports quantization as a deployment technique. DistilBERT reduces from 255.4 MB to 91.0 MB, a reduction of 64.4 percent. Toxic-RoBERTa and RoBERTa reduce from 475.5 MB to 148.9 MB, a reduction of 68.7 percent. These reductions are valuable when models must be stored on limited devices, downloaded over networks or deployed as multiple model versions.

The results in terms of latencies need to be interpreted with care. Whereas fine-tuned models have been tested using GPU latency, the quantized models have been tested for their CPU latency. Therefore, the latency result here in terms of INT-8 is higher than before. This does not negate the usefulness of quantization; all this indicates is that when one wants to conduct a latency test, it should be compared on CPU using both FP32 and INT-8.

Table 6.3: Model size reduction after INT-8 quantization.

Model	FP32 fine-tuned size (MB)	INT-8 size (MB)	Reduction
DistilBERT	255.4	91.0	64.4%
Toxic-RoBERTa	475.5	148.9	68.7%
RoBERTa	475.5	148.9	68.7%

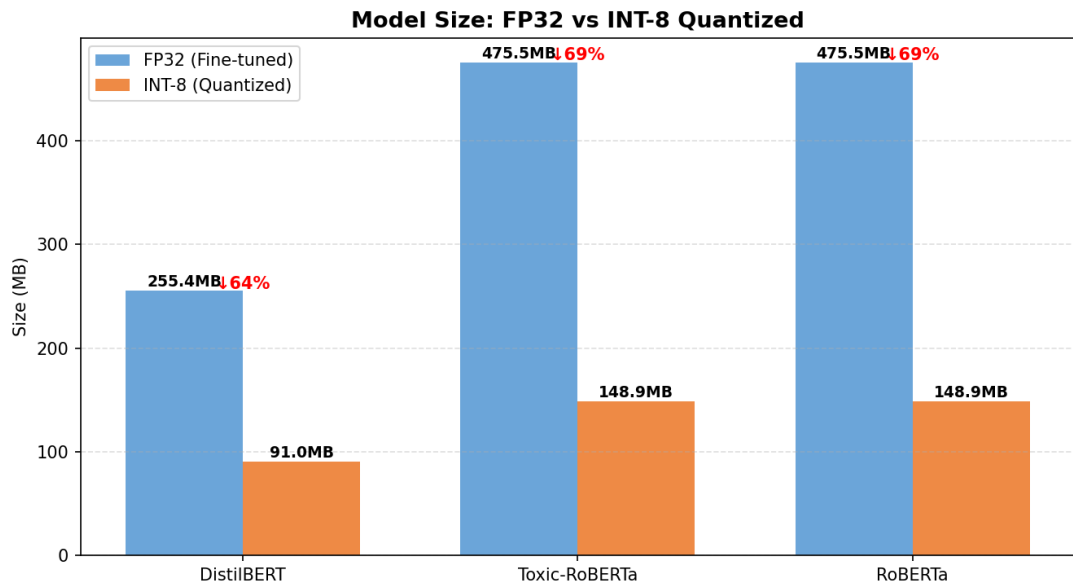


Figure 6.4: Model size comparison between FP32 fine-tuned and INT-8 quantized versions.

6.6 Overall Recommendation

With the focus being on maximum efficiency, Toxic-RoBERTa emerges as the optimal model for use in this research. This is because its quantized and fine-tuned versions attain F1-scores that are around 0.898. However, considering the case where space is the major limitation and slight sacrifices in accuracy can be made, the quantized version of Toxic-RoBERTa will be the ideal option

If a simpler and smaller architecture is preferred from the beginning, DistilBERT is still useful. Its fine-tuned model reaches macro F1-score of 0.8608 with a size of 255.4 MB, and its quantized version is only 91.0 MB. However, for this dataset and label mapping, Toxic-RoBERTa provides the best balance between accuracy and post-quantization robustness.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

This thesis presented a complete pipeline for resource-efficient toxic comment classification using fine-tuned and INT-8 quantized transformer models. The raw Jigsaw dataset was transformed into a three-class dataset with labels Safe, Toxic and Obscene. The original imbalance was addressed by sampling 9,603 comments from each class, producing a balanced dataset of 28,809 samples. The dataset was split into training, validation and test sets using a 70:15:15 ratio.

Three models were compared: DistilBERT, Toxic-RoBERTa and RoBERTa. Base model evaluation showed poor performance because the task-specific head was not trained. Fine-tuning significantly improved all models. Toxic-RoBERTa achieved the best fine-tuned result with accuracy of 0.8987 and macro F1-score of 0.8985. RoBERTa and DistilBERT also performed strongly, but below Toxic-RoBERTa.

INT-8 dynamic quantization reduced model size by 64.4 percent for DistilBERT and 68.7 percent for both RoBERTa variants. Toxic-RoBERTa retained almost all of its fine-tuned accuracy after quantization, making it the strongest candidate for deployment where storage reduction is important. The thesis also emphasized that latency must be evaluated under matching hardware conditions before drawing final speed conclusions.

7.2 Major Findings

The first major finding is that domain-oriented transformer models can provide measurable advantages for toxicity classification. Toxic-RoBERTa outperformed the general-purpose RoBERTa and the compact DistilBERT model in this experiment. The second finding is that balancing the dataset is important for meaningful macro-level evaluation because raw Jigsaw labels are highly imbalanced after mapping.

The third finding is that dynamic quantization can provide large storage reductions with small accuracy losses, especially for Toxic-RoBERTa. The fourth finding is that model size and latency must be discussed separately. Quantization reduced storage, but the CPU execution environment produced higher latency values than GPU fine-tuned inference in the recorded experiment.

7.3 Limitations

The study uses a threshold-based label mapping, and different thresholds could change the final label distribution. The task is restricted to three classes and does not perform multi-label classification. Some toxicity categories, such as threat and identity attack, are merged into the general Toxic class. The dataset examples are balanced through down-sampling, which improves class balance but discards many Safe and Toxic examples from the raw dataset.

The training configuration is consistent but not exhaustively tuned. Larger hyperparameter search, longer training or data augmentation may improve results. Latency comparison is limited by the difference between GPU evaluation for fine-tuned models and CPU evaluation for quantized models. A dedicated deployment benchmark would be required for stronger conclusions about runtime speed.

7.4 Future Scope

Future work can extend this thesis in several directions. First, the label mapping can be expanded to multi-label classification so that a comment can simultaneously be toxic, obscene, insulting or threatening. Second, threshold calibration can be performed using validation data to improve class separation. Third, fairness and bias evaluation can be added to study whether the model overpredicts toxicity for identity-related terms.

Fourth, model compression can be expanded beyond dynamic quantization. Quantization-aware training, ONNX Runtime optimization, pruning and knowledge distillation can be compared. Fifth, a proper CPU benchmark should compare FP32 CPU inference and INT-8 CPU inference using identical batch sizes. Sixth, explainability methods can be added to show which tokens influence model decisions, helping moderators understand model predictions.

Finally, the model can be integrated into a simple moderation API or dashboard. Such a deployment could demonstrate how predicted class, confidence score and moderation action can be presented to human reviewers.

REFERENCES

- [1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. and Polosukhin, I. (2017). Attention Is All You Need. *Advances in Neural Information Processing Systems*.
- [2] Devlin, J., Chang, M. W., Lee, K. and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of NAACL-HLT*.
- [3] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.
- [4] Sanh, V., Debut, L., Chaumond, J. and Wolf, T. (2019). DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. *arXiv preprint arXiv:1910.01108*.
- [5] Jigsaw and Conversation AI. Jigsaw Unintended Bias in Toxicity Classification Dataset. *Kaggle competition dataset*.
- [6] Wulczyn, E., Thain, N. and Dixon, L. (2017). Ex Machina: Personal Attacks Seen at Scale. *Proceedings of the International Conference on World Wide Web*.
- [7] Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H. and Kalenichenko, D. (2018). Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. *Proceedings of CVPR*.
- [8] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G. et al. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems*.
- [9] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A. et al. (2020). Transformers: State-of-the-Art Natural Language Processing. *Proceedings of EMNLP: System Demonstrations*.
- [10] Hugging Face. Transformers documentation and model cards for distilbert-base-uncased, roberta-base and unitary/unbiased-toxic-roberta.

APPENDIX A

NOTEBOOK WORKFLOW SUMMARY

The implementation notebook contains 24 cells. It installs compatible library versions, imports required packages, mounts Google Drive, loads the Jigsaw dataset, creates three-class labels, balances the dataset, performs a 70:15:15 split, defines tokenization and evaluation helpers, registers three models, evaluates base models, fine-tunes the models, applies INT-8 dynamic quantization, evaluates quantized models and exports plots and result summaries.

Key variables used in the notebook

```
DATA_PATH =  
"/content/drive/MyDrive/My_Thesis_Project/Jigsaw_dataset/train.csv"  
MAX_LEN = 128  
NUM_LABELS = 3  
LABEL2ID = {"Safe": 0, "Toxic": 1, "Obscene": 2}  
MIN_CLASS = 9603  
TRAIN/VALIDATION/TEST = 70/15/15  
Learning rate = 2e-5, epochs = 3, train batch size = 32
```

APPENDIX B

FULL EXPERIMENTAL RESULT TABLE

The table below repeats the full result table so that the thesis remains self-contained.

Model	Stage	Accuracy	F1-Macro	Precision	Recall	Latency(ms)	Size(MB)
DistilBERT	Base	0.353771	0.279138	0.291677	0.353847	0.180	255.416
DistilBERT	Finetuned	0.860944	0.860804	0.860843	0.860939	0.116	255.416
DistilBERT	Quantized	0.844516	0.843424	0.849916	0.844532	112.869	90.996
Toxic-RoBERTa	Base	0.254975	0.208938	0.179092	0.254916	0.358	475.494
Toxic-RoBERTa	Finetuned	0.898658	0.898539	0.898507	0.898660	0.185	475.494
Toxic-RoBERTa	Quantized	0.898427	0.898193	0.898379	0.898431	225.195	148.916
RoBERTa	Base	0.333410	0.166696	0.111137	0.333333	0.351	475.494
RoBERTa	Finetuned	0.867654	0.867679	0.868884	0.867644	0.191	475.494
RoBERTa	Quantized	0.860250	0.859249	0.860550	0.860256	231.765	148.916

M.TechThesis

by Vinay Grewal

Submission date: 20-Jun-2026 05:56PM (UTC+0530)

Submission ID: 2986490716

File name: VinayThesis.pdf (523.91K)

Word count: 7059

Character count: 41330

ABSTRACT

The growth of online platforms has increased the need for automated systems that can identify harmful comments at scale. Toxic comment classification is a challenging natural language processing problem because harmful language varies in tone, context and intensity. Transformer-based language models provide strong contextual representations, but their size and computational requirements can limit deployment in resource-constrained environments. This dissertation presents a resource-efficient toxic comment classification pipeline using fine-tuned and INT-8 dynamically quantized transformer models.

The study uses the Jigsaw toxicity dataset and converts its continuous toxicity annotations into a three-class classification task: Safe, Toxic and Obscene. The original mapped dataset is highly imbalanced, containing 1,656,908 Safe comments, 138,360 Toxic comments and 9,603 Obscene comments. To reduce class imbalance, the dataset is down-sampled to 9,603 examples per class, producing a balanced dataset of 28,809 comments. The balanced dataset is split into 20,166 training samples, 4,321 validation samples and 4,322 test samples.

Three transformer models are evaluated: DistilBERT, Toxic-RoBERTa and RoBERTa. Each model is tested in three stages: base model with an untrained classification head, fine-tuned model and INT-8 dynamically quantized model. The models are compared using accuracy, macro F1-score, macro precision, macro recall, latency and model size. Toxic-RoBERTa achieves the best fine-tuned result with accuracy of 0.8987 and macro F1-score of 0.8985. After INT-8 dynamic quantization, Toxic-RoBERTa retains nearly the same performance with accuracy of 0.8984 and macro F1-score of 0.8982 while reducing size from 475.5 MB to 148.9 MB.

The results show that fine-tuning is essential for the three-class toxicity task and that dynamic quantization can significantly reduce model size with small performance loss. The thesis recommends quantized Toxic-RoBERTa as the best overall choice when accuracy and storage efficiency are both important. It also highlights that latency comparison should be performed under matched hardware

conditions because the quantized models in this experiment run on CPU while the fine-tuned models were evaluated on GPU.

Keywords: Toxicity Detection, Jigsaw Dataset, Transformer Models, DistilBERT, RoBERTa, Toxic-RoBERTa, Fine-Tuning, Dynamic Quantization, INT-8.

TABLE OF CONTENTS

Candidate's Declaration	ii
Certificate by the Supervisor	iii
Acknowledgements	iv
Abstract.....	v
List of Tables	viii
List of Figures.....	ix
List of Symbols, Abbreviations and Nomenclature	x
Chapter 1 Introduction.....	1
1.1 Background	1
1.2 Motivation	2
1.3 Problem Statement	2
1.4 Objectives of the Work.....	3
1.5 Scope of the Thesis.....	3
1.6 Organization of the Thesis.....	4
Chapter 2 Literature Review	5
Chapter 3 Dataset Preparation and Preprocessing.....	8
Chapter 4 Proposed Methodology	12
Chapter 5 Experimental Setup and Implementation.....	15
Chapter 6 Results and Discussion.....	18
Chapter 7 Conclusion and Future Scope	23
References.....	26
Appendix A Notebook Workflow Summary	27
Appendix B Full Experimental Result Table	28

LIST OF TABLES

Table 3.1	Jigsaw columns used in label mapping.
Table 3.2	Three-class label mapping strategy.
Table 3.3	Representative sanitized examples of final dataset format.
Table 3.4	Dataset distribution before and after balancing.
Table 3.5	Train, validation and test split sizes.
Table 4.1	Model registry used in the experiment.
Table 5.1	Fine-tuning hyperparameters.
Table 6.1	Full performance comparison across models and stages.
Table 6.2	Fine-tuned confusion matrix diagonal summary.
Table 6.3	Model size reduction after INT-8 quantization.

LIST OF FIGURES

- | | |
|-------------------|---|
| Figure 6.1 | Performance comparison across base, fine-tuned and quantized stages. |
| Figure 6.2 | Radar comparison of fine-tuned and INT-8 quantized models. |
| Figure 6.3 | Confusion matrices for fine-tuned models on the test set. |
| Figure 6.4 | Model size comparison between FP32 fine-tuned and INT-8 quantized versions. |

LIST OF SYMBOLS, ABBREVIATIONS AND NOMENCLATURE

Symbol	Meaning
AI	Artificial Intelligence
BERT	Bidirectional Encoder Representations from Transformers
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
DL	Deep Learning
F1	Harmonic mean of precision and recall
FP32	32-bit floating point representation
GPU	Graphics Processing Unit
HF	Hugging Face
INT-8	8-bit integer representation
ML	Machine Learning
NLP	Natural Language Processing
RoBERTa	Robustly Optimized BERT Pretraining Approach
TP/FP/FN	True Positive, False Positive and False Negative

CHAPTER 1

INTRODUCTION

1.1 Background

Discussion platforms have emerged as vital tools in the digital era. News sites, educational sites, social networking sites, online gaming communities and enterprise forums all depend on user commentary to facilitate discourse. While this openness serves as the strength of such platforms, the same attribute presents moderation problems. Comments that are toxic serve to hinder discourse and even put users at risk of harm. Moderation through manual means becomes impossible considering the sheer number of posts made on any platform coupled with the fact that comments are open to linguistic nuances and online jargon.

Machine learning has, therefore, become an essential part of moderation systems. Previous systems made use of handcrafted features, counts of words, lexicographic markers, or classical term frequency-inverse document frequency vectors to represent comments. They were relatively simple but did not manage to take into account context and sarcasm. Deep learning models improved representation significantly, while transformer-based language models managed to learn context-sensitive token representations by training on large amounts of text data. Examples include BERT, DistilBERT and RoBERTa, which can be fine-tuned for specific tasks.

This thesis aims at studying the classification of toxic comments via transformer models on the basis of the Jigsaw data set. In the raw data set, there exist annotations regarding toxicity with continuous labels such as target toxicity, severe toxicity, obscene, insult, threat and identity attack. For the current study, these annotations are converted into a three-way classification problem involving classes of Safe, Toxic and Obscene. Three transformer architectures are examined: DistilBERT, RoBERTa and Toxic-RoBERTa. Each architecture is evaluated in three steps: base architecture, fine-tuned architecture and INT-8 dynamically quantized architecture.

1.2 Motivation

The effectiveness of a moderation model depends on its ability to accurately predict undesirable content and be computationally feasible enough for execution in real conditions. High accuracy is usually guaranteed by large transformer-based architectures, but excessive memory usage and inference time can impede model deployment on resource-constrained machines. It may be more desirable to have a model with lower accuracy but smaller footprint than a more precise but bulky architecture. That is the reason for this research.

The raw Jigsaw dataset is significantly imbalanced. There are more safe comments than toxic comments, and the number of obscene comments is less compared to safe comments. Using the above-mentioned dataset as it is for training results in biasing the classifier, which tends to predict the majority class too often. To achieve an unbiased comparison, we have to balance, filter out and define labels properly. For this reason, equal class sampling is applied to generate 28,809 comments, each class having 9,603 samples.

A further goal is to measure the influence of post-training quantization. INT-8 dynamic quantization can shrink the model by quantizing the floating-point numbers used as weights in the linear layers to 8-bit integers at inference time. Dynamic quantization does not necessitate additional training or calibration data, which makes it easy to implement post-fine-tuning. The fundamental issue here is whether quantization results in lossless classification performance on the toxicity classification task.

1.3 Problem Statement

The main issue that this thesis focuses on is the creation and analysis of a model for the detection of toxic comments in three classes namely safe, toxic and obscene comments. In this process, a model has to be created which can convert the raw Jigsaw toxicity annotations to a single class label, create a balanced data set, fine-tuning of transformer models, applying dynamic quantization at INT-8 level and comparing these models.

Mathematically, for the given input data point x , the algorithm should predict a single output label y such that the label belongs to the set {Safe, Toxic, Obscene}. Classification of comments is done on the basis of single-label multi-class classification. For Obscenity labeling of the comment, the obscene score passes the threshold level. Similarly, for toxic labeling, other relevant scores pass the threshold level. Otherwise, the comment is classified as safe.

1.4 Objectives of the Work

Firstly, a dataset comprising only three classes that are free from any form of contamination will be created from the Jigsaw toxic data. This entails selecting pertinent columns, converting the continuous toxicity ratings to class labels, and finally reducing the number of instances of the majority class.

The second aim is to tune three transformer models based on the same experimental setting. DistilBERT denotes a small-sized transformer benchmark model, RoBERTa denotes a powerful encoder, and Toxic-RoBERTa denotes a specialized toxicity-trained model. The training and evaluation are done using the same train/validate/test split.

The third goal would be to run INT-8 dynamic quantization on the fine-tuned model and assess its accuracy, F1 score, precision, recall, latency, and size. The third goal seeks to find an answer to the practical question whether compressed transformer models can be applied to toxic text classification with minimal losses.

Objective number four will entail interpreting the findings via the use of metric tables, confusion matrices, radar charts, and size comparison charts. Objective number five entails recommending an appropriate model to deploy depending on certain criteria like maximum accuracy, minimum size, among others.

1.5 Scope of the Thesis

The scope of the research is restricted to the topic of toxicity detection in English language comments through the use of Jigsaw Toxic Comment Classification and encoder models for text based on transformers. The study considers only three types of labels, utilizing a fixed threshold label mapping

approach. It does not cover multi-label prediction, multi-lingual toxicity detection, online learning or human intervention during the process of labeling comments. For quantization, PyTorch dynamic quantization of linear layers was chosen.

However, one should note that the latency results are not directly comparable due to the fact that fine-tuned models were measured using GPU latency, whereas dynamically quantized models were tested using CPU due to their native support for dynamic quantization. However, the difference in hardware was caused by the practical needs of the experiment. The first practical deployment benefit seen in the experiment is the reduction of the size of the model.

1.6 Organisation of the Thesis

Chapter 1 introduces the motivation, problem statement and objectives.

Chapter 2 reviews the literature on toxicity detection, transformer language models and model compression.

Chapter 3 explains the Jigsaw dataset, label mapping, filtration, balancing and preprocessing steps.

Chapter 4 presents the proposed methodology and model pipeline.

Chapter 5 describes implementation details, training configuration and evaluation metrics.

Chapter 6 presents the experimental results and discusses model behavior.

Chapter 7 concludes the thesis and outlines future research directions.

CHAPTER 2

LITERATURE REVIEW

2.1 Toxicity Detection in Online Text

Toxicity detection refers to the process of detecting those comments which could potentially cause harm or abuse, such as insults or threats to the user. It is similar to hate speech detection, abusive language detection, and content moderation; however, they are not entirely the same. A comment that is toxic may be an insult to someone else without falling under the category of hate speech, and a discussion on a sensitive topic may have identity words without being classified as toxic.

The earlier toxicity classifier approaches utilized linguistic features including n-gram, term frequency inverse document frequency vector, sentiment scores, and offensive word lists. Such algorithms would train well, although they lacked robustness as a result of the changes in user's spelling, phrases, and comments that require contextual understanding. Neural models contributed to better word and sequence representation using dense embedding representations. Recurrent models like LSTM and GRU network architecture were more popular before transformer encoder architectures took over.

One problem that has been consistent with toxicity datasets is bias. Bias may occur when the model identifies unintended correlations between mentions of identity and labels for toxicity. For instance, neutral comments on a protected class could mistakenly be flagged as harmful if the term for the protected class occurs frequently in the toxic dataset.

2.2 Transformer Encoders for Text Classification

Transformers used self-attention instead of recurrence. With a transformer encoder, all tokens have attention over other tokens in the input sequence, thus enabling the model to create contextualized embeddings. The idea is especially useful in text classification, where the embedding of the same word could vary based on the surrounding words. For instance, a given word could be non-abusive in quotes or educationally related, but harmful when used directly.

BERT pioneered masked language modeling and next sentence prediction pre-training methods for obtaining bidirectional representations. In order to perform the task of classification, a classification head specific to the task is connected to the pooled representation of the sequence and the entire model is fine-tuned on labeled data. RoBERTa enhanced the training methodology used for BERT through increased batch size, more data, and pre-training changes. DistilBERT shrunk the model size by transferring knowledge from a large BERT model to a small student model.

This thesis is based on a comparison between a small-scale model, a general strong encoder, and a toxicity-focused encoder. Such a comparison is valuable since, when deciding to deploy the models, one needs to consider more than just accuracy alone.

2.3 Fine-Tuning of Pretrained Models

Fine-tuning is the process of adapting a pre-trained language model to perform a specific task by optimizing the parameters of the model using labeled data. In our current experiment, we transform each language model into a classifier with three output classes. After passing the tokenized comment into the encoder of the language model, we obtain logits corresponding to Safe, Toxic, and Obscene.

The fine-tuning approach proves helpful when the number of examples in the task-specific dataset is less than the size of the datasets in the pretraining phase. The pre-trained model has been trained on the linguistic structure. However, it is fine-tuned to learn how its internal representation can be translated into task-specific label predictions. All experiments have employed a common setup with 3 epochs, a learning rate of $2e-5$, cosine learning rate schedule, and macro F1-score.

2.4 Model Compression and INT-8 Quantization

Transformer models are known to be costly in terms of storage and execution. Techniques for model compression aim at lowering such costs. Some well-known techniques include pruning, knowledge distillation, low-rank decomposition, quantization, and efficient model design. Quantization entails reducing the level of

accuracy of weights or activations. Rather than storing the model weights using 32-bit float precision, quantization utilizes 8-bit integers, called INT-8.

Dynamic Quantization is a post-training approach where some specific layers such as linear layers are quantized using the model alone without the requirement of any extra data to train the model. In the inference process, weight parameters of the model are first stored as quantized values then dequantized whenever computations are done. The approach implemented in this project utilizes the dynamic quantization of PyTorch on linear layers.

2.5 Research Gap

Most work done in toxicity classification centers on making improvements in the predictive models, but comparatively less has been achieved regarding the actual process of data pre-processing through quantization. In the present thesis, an attempt is made to bridge this gap between theory and practice by incorporating all aspects within a single approach.

Another shortfall pertains to an honest interpretation of the results of compression. While a compressed model may be significantly reduced in size, it need not necessarily increase in speed. Both the measurements of size and latency are provided in the results, while the analysis distinguishes between size and runtime efficiencies.

2.6 Summary

There have been many pieces of literature highlighting the strength of using transformer encoders for text classification and the importance of using compression techniques for implementing these classifiers. This thesis explores both these dimensions through the comparative evaluation of fine-tuned and quantized transformers for the task of toxicity classification, which is discussed in the next chapter.

CHAPTER 3

DATASET PREPARATION AND PREPROCESSING

3.1 Source Dataset

This thesis uses the Jigsaw toxicity dataset available in train.csv in the project's workspace. The raw data has 1,804,874 rows and 45 columns. The primary text column in the data is 'comment_text'. Other columns that are relevant to toxicity and will be used for labeling are 'target', 'severe_toxicity', 'obscene', 'identity_attack', 'insult' and 'threat'. Metadata columns like 'article_id', ratings and numbers of annotators are also present in the raw data but will not be used as inputs for modeling in this thesis.

The raw annotations are continuous scores rather than a ready-made three-class label. Each score represents the degree to which a comment is associated with a toxicity attribute. The project converts these continuous indicators into a single categorical label so that the downstream task becomes single-label classification. This conversion is described in the next section.

Table 3.1: Jigsaw columns used in label mapping.

Column	Use in this thesis
comment_text	Input text used by the classifier
target	General toxicity score used for Toxic mapping
severe_toxicity	Severe toxicity score used for Toxic mapping
obscene	Obscene score used for Obscene mapping
identity_attack	Identity attack score used for Toxic mapping
insult	Insult score used for Toxic mapping
threat	Threat score used for Toxic mapping

3.2 Label Mapping Strategy

The final label space contains three labels: Safe, Toxic and Obscene. The mapping gives priority to the obscene attribute. If the obscene score is greater than or equal to 0.5, the sample is assigned label 2, Obscene. If the obscene condition is not satisfied but at least one of target, severe_toxicity, insult, threat or identity_attack is greater than or equal to 0.5, the sample is assigned label 1, Toxic. If none of these conditions is met, the sample is assigned label 0, Safe.

This priority order prevents comments with strong obscene content from being merged into the broader Toxic category. It also preserves a meaningful distinction between general toxicity and obscene language. The mapping is simple, transparent and reproducible, which is useful for thesis evaluation.

Table 3.2: Three-class label mapping strategy.

Label ID	Class Name	Mapping Rule
2	Obscene	obscene ≥ 0.5
1	Toxic	obscene < 0.5 and any of target, severe_toxicity, insult, threat or identity_attack ≥ 0.5
0	Safe	none of the toxicity conditions are satisfied

3.3 Filtration and Column Selection

Upon completion of labelling, only two fields are selected as inputs for model training: “comment_text” and “labels”. The field “comment_text” is changed to “text” for brevity in defining the structure of the dataset. Each row can be seen as a tuple of text and its corresponding label as an integer value. This is done since the objective is to classify toxicity based on comments' content.

The text will be converted into strings during tokenization, hence eliminating possible errors that may occur when there is no value or when the value is stored in a wrong data type. There will be no need for any rigorous cleansing of the text since the transformers' tokenizers are built to cope with punctuation, capitalization, and sub-word segmentation. Excessive cleansing can wash away useful information. The maximum number of tokens is 128.

3.4 Class Imbalance and Balancing

The dataset that results from mapping is extremely imbalanced. There are 1,656,908 instances in the Safe class, 138,360 instances in the Toxic class, and 9,603 instances in the Obscene class. If training is done according to this imbalance, the model would be able to obtain deceptive accuracy from the Safe class. To address this, each class is sampled to the size of the smallest class, which is 9,603.

The balanced dataset contains 28,809 samples in total, with equal representation of Safe, Toxic and Obscene. The balanced dataset is shuffled with

random_state = 42 to make the result reproducible. This balanced setting improves the reliability of macro F1-score because each class contributes equally to the final metric.

Table 3.4: Dataset distribution before and after balancing.

Class	Before balancing	After balancing
Safe	1,656,908	9,603
Toxic	138,360	9,603
Obscene	9,603	9,603
Total	1,804,871 after mapping count display	28,809

3.5 Train, Validation and Test Splits

The balanced dataset is split into training, validation and test sets using a 70:15:15 ratio. Stratified sampling is used to preserve class proportions in each split. The final split sizes are 20,166 training samples, 4,321 validation samples and 4,322 test samples. The test set is kept separate and is used only for final evaluation of the base, fine-tuned and quantized models.

A validation set is required for selecting the best checkpoint during fine-tuning. In this thesis, the best model is selected using macro F1-score. Macro F1 is appropriate because it gives equal weight to all classes rather than being dominated by the majority class.

Table 3.5: Train, validation and test split sizes.

Split	Ratio	Number of samples
Training	70%	20,166
Validation	15%	4,321
Test	15%	4,322

3.6 Representative Final Dataset Examples

The final dataset examples consist of a text field and a numeric label. Since toxic datasets can contain offensive language, the examples in this thesis are sanitized and shortened for presentation. They illustrate the final structure without reproducing harmful text in full. In the actual dataset, the model receives the original comment text and the mapped label.

Table 3.3: Representative sanitized examples of final dataset format.

Sanitized text example	Label	Class
This is so cool. It makes the discussion easier to follow.	0	Safe
Your argument is terrible and you are acting like an [insult removed].	1	Toxic
This comment contains [obscene expression removed] directed at another user.	2	Obscene

CHAPTER 4

PROPOSED METHODOLOGY

4.1 Overall Pipeline

The proposed pipeline begins with the raw Jigsaw CSV file and ends with a comparison of fine-tuned and quantized transformer models. The stages are: data loading, label mapping, column filtration, class balancing, train-validation-test split, tokenizer construction, dataset object creation, base model evaluation, fine-tuning, fine-tuned evaluation, INT-8 dynamic quantization, quantized model evaluation and visualization of results.

The same test set is used for all models and stages. This ensures that the reported differences are due to the model and stage rather than different samples. The base model stage evaluates the pretrained backbone with a fresh three-class classification head. Because this head has not learned the toxicity task, the base performance is expected to be low. This stage is included to show the value of task-specific fine-tuning.

4.2 Model Selection

Three models are selected to represent different design points. DistilBERT is included as a lightweight transformer model. It is smaller than BERT-like encoders and is useful when storage and speed are important. RoBERTa is included as a strong general-purpose transformer encoder. Toxic-RoBERTa is included because it has been pretrained or adapted for toxicity-related language, making it a strong candidate for this domain.

All three models are converted into sequence classification models with three output labels. The label mapping used in the model configuration is Safe = 0, Toxic = 1 and Obscene = 2. The same maximum sequence length and training configuration are used to maintain fairness across models.

Table 4.1: Model registry used in the experiment.

Model	Hugging Face checkpoint	Purpose
DistilBERT	distilbert-base-uncased	Compact general transformer baseline
Toxic-RoBERTa	unitary/unbiased-toxic-roberta	Toxicity-oriented RoBERTa variant

RoBERTa	roberta-base	Strong general-purpose transformer encoder
---------	--------------	--

4.3 Tokenization and Dataset Construction

Each comment is tokenized using the tokenizer corresponding to its model. Tokenization applies truncation, padding to a maximum length of 128 tokens and attention mask generation. The dataset object returns input_ids, attention_mask and labels. This format is compatible with Hugging Face Trainer as well as manual PyTorch evaluation loops.

The selection of max length 128 ensures information preservation without making computations too difficult. Online discussion toxic comments are usually short, and the selected max length allows capturing almost all comments without wasting unnecessary memory space. Longer comments are clipped by the tokenizer.

4.4 Fine-Tuning Procedure

The fine-tuning process happens for three epochs. The batch size for per-device training is 32 and that for evaluation is 64. The learning rate is set to $2e-5$, the warm-up ratio is 0.06, the weight decay is 0.01, and there is a cosine learning rate scheduler. If CUDA is available, then the model undergoes mixed precision training. Evaluation and checkpointing take place after each epoch ends. The checkpoint with the highest score based on macro F1-score is picked.

The training configuration is intentionally consistent across models. This makes the comparison easier to interpret and avoids introducing model-specific hyperparameter advantages. Although further tuning may improve individual results, the objective of the thesis is a controlled comparison rather than exhaustive hyperparameter search.

4.5 INT-8 Dynamic Quantization

After fine-tuning, each model is loaded in full precision on CPU and PyTorch dynamic quantization is applied to linear layers using `torch.quantization.quantize_dynamic`. The resulting INT-8 state dictionary is saved separately along with the tokenizer and configuration. For evaluation, the model

architecture is reconstructed, dynamic quantization is applied again and the saved INT-8 state dictionary is loaded.

This method reduces storage size significantly. It is also simple because it does not require calibration data or retraining. However, dynamic quantization in the implementation used is CPU-native. Therefore, latency should be interpreted relative to the CPU execution environment rather than as a direct comparison with GPU fine-tuned inference.

4.6 Evaluation Metrics

The evaluation metrics are accuracy, macro F1-score, macro precision, macro recall, latency per sample and model size. Accuracy measures the fraction of correctly classified comments. Macro precision, recall and F1-score compute the metric independently for each class and then average across classes. Macro F1-score is especially important for balanced multi-class classification because it reflects performance across all categories.

Confusion matrices are also used to study class-level errors. Each row of the confusion matrix corresponds to the true class and each column corresponds to the predicted class. The diagonal entries represent correct predictions, while off-diagonal entries show misclassifications.

CHAPTER 5

EXPERIMENTAL SETUP AND IMPLEMENTATION

5.1 Software and Hardware Environment

The notebook environment uses Python with PyTorch, Transformers, scikit-learn, NumPy, SciPy, PEFT and the Hugging Face Datasets library. The recorded versions include PyTorch 2.2.2 with CUDA 12.1 support and Transformers 4.38.2. CUDA availability was true during model training, enabling GPU execution for fine-tuning and fine-tuned evaluation.

The project was executed in a Google Colab style workflow with Google Drive mounted for saving models, plots, result summaries and the balanced dataset. The model outputs were stored in a Drive root folder named Thesis_ToxicityDetection. This structure helps preserve trained models and generated result files across notebook sessions.

5.2 Reproducibility Settings

The dataset sampling and train-test splitting use `random_state = 42`. The balanced dataset is created by sampling 9,603 examples from each class and shuffling the result. Stratified splitting preserves class balance across the training, validation and test sets. Reproducibility is important because random sampling can otherwise change the final distribution and model results.

The final thesis result files include a `comparison_table.csv` file, a `results_summary.json` file and four plots. These files are used to produce the tables and figures in Chapter 6. The notebook therefore acts not only as an experimental script but also as a reproducible record of the project workflow.

5.3 Model Training Implementation

The Hugging Face Trainer API is used for fine-tuning. The `compute_metrics` function calculates accuracy, macro F1-score, macro precision and macro recall from model logits. The best checkpoint is selected using macro F1-score. A custom

ToxicDataset class is also defined for manual PyTorch evaluation, returning tokenized input IDs, attention masks and labels.

The base model evaluation loads each pretrained model with num_labels = 3 and ignore_mismatched_sizes = True. This attaches a newly initialized classification head where needed. The fine-tuning stage then updates the model parameters on the balanced training data. The final fine-tuned model and tokenizer are saved for each architecture.

Table 5.1: Fine-tuning hyperparameters.

Hyperparameter	Value
Number of epochs	3
Training batch size	32
Evaluation batch size	64
Learning rate	2e-5
Warmup ratio	0.06
Weight decay	0.01
Scheduler	Cosine
Best model criterion	Macro F1-score
Maximum sequence length	128 tokens
Mixed precision	Enabled when CUDA is available

5.4 Quantized Model Implementation

For dynamic quantization, the fine-tuned model is loaded on CPU and quantize_dynamic is applied to nn.Linear layers with dtype = torch.qint8. The quantized model state dictionary is saved as model_int8.pt. During evaluation, the same model shell is recreated, quantized and populated with the saved state dictionary. The quantized model is evaluated on the same test set using CPU.

The model size is computed and stored for the base, fine-tuned and quantized versions. Size reduction percentage is calculated as one minus the ratio of INT-8 size to fine-tuned FP32 size, multiplied by 100. This provides a clear measure of compression effectiveness.

5.5 Workflow Summary

- Load Jigsaw train.csv from Google Drive.
- Create labels using threshold-based mapping.
- Retain text and label columns.

- Sample 9,603 examples per class for balancing.
- Split into train, validation and test sets using stratification.
- Fine-tune DistilBERT, Toxic-RoBERTa and RoBERTa.
- Evaluate fine-tuned models on the test set.
- Apply INT-8 dynamic quantization to linear layers.
- Evaluate quantized models and export plots, CSV and JSON summaries.

CHAPTER 6

RESULTS AND DISCUSSION

Table 6.1: Full performance comparison across models and stages.

Model	Stage	Accuracy	F1-Macro	Precision	Recall	Latency(ms)	Size(MB)
DistilBERT	Base	0.3538	0.2791	0.2917	0.3538	0.18	255.4
DistilBERT	Finetuned	0.8609	0.8608	0.8608	0.8609	0.12	255.4
DistilBERT	Quantized	0.8445	0.8434	0.8499	0.8445	112.87	91.0
Toxic-RoBERTa	Base	0.2550	0.2089	0.1791	0.2549	0.36	475.5
Toxic-RoBERTa	Finetuned	0.8987	0.8985	0.8985	0.8987	0.19	475.5
Toxic-RoBERTa	Quantized	0.8984	0.8982	0.8984	0.8984	225.19	148.9
RoBERTa	Base	0.3334	0.1667	0.1111	0.3333	0.35	475.5
RoBERTa	Finetuned	0.8677	0.8677	0.8689	0.8676	0.19	475.5
RoBERTa	Quantized	0.8602	0.8592	0.8606	0.8603	231.77	148.9

6.1 Base Model Performance

The base models perform poorly compared with the fine-tuned models. This result is expected because the classification head is newly initialized for the three-class toxicity task. DistilBERT obtains a base accuracy of 0.3538 and macro F1-score of 0.2791. Toxic-RoBERTa obtains a base accuracy of 0.2550 and macro F1-score of 0.2089. RoBERTa obtains a base accuracy of 0.3334 and macro F1-score of 0.1667.

These results show that pretrained language representations alone are not sufficient when the task head has not learned the label mapping. Fine-tuning is necessary to align the model outputs with the Safe, Toxic and Obscene classes.

6.2 Fine-Tuned Model Performance

Fine-tuning dramatically improves all models. Toxic-RoBERTa achieves the best fine-tuned performance with accuracy of 0.8987 and macro F1-score of 0.8985. RoBERTa achieves accuracy of 0.8677 and macro F1-score of 0.8677. DistilBERT achieves accuracy of 0.8609 and macro F1-score of 0.8608. The strong result of Toxic-RoBERTa suggests that domain-oriented pretraining is beneficial for toxicity classification.

DistilBERT remains competitive despite being smaller. This makes it attractive for situations where memory and computation are constrained. RoBERTa performs better than DistilBERT but not as well as Toxic-RoBERTa on this task.

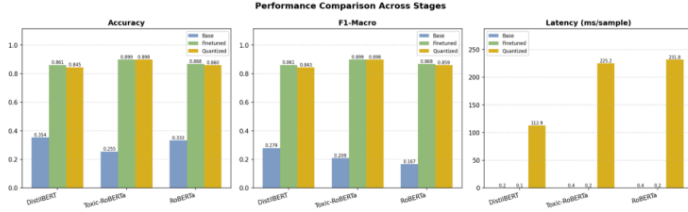


Figure 6.1: Performance comparison across base, fine-tuned and quantized stages.

6.3 Quantized Model Performance

INT-8 dynamic quantization preserves the performance of Toxic-RoBERTa almost completely. Its accuracy changes from 0.8987 to 0.8984 and macro F1-score changes from 0.8985 to 0.8982. This is a negligible loss for a 68.7 percent size reduction. RoBERTa also retains most of its performance, with accuracy changing from 0.8677 to 0.8602. DistilBERT shows a larger drop, from 0.8609 to 0.8445 accuracy.

This shows that there is variation in the effect of quantization depending on the model architecture used. RoBERTa specifically designed for toxicity detection is most resilient to dynamic quantization. In this case, DistilBERT is already compact and might be impacted by further quantization.

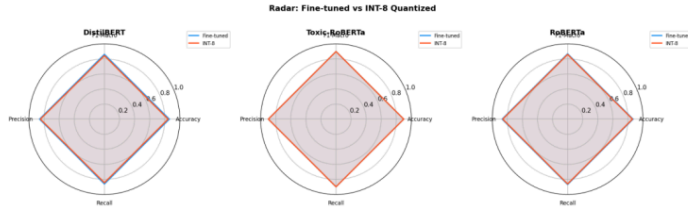


Figure 6.2: Radar comparison of fine-tuned and INT-8 quantized models.

6.4 Confusion Matrix Analysis

The confusion matrices show that the models perform well on the diagonal entries after fine-tuning. Toxic-RoBERTa has the highest correct predictions across all three classes: 1,304 Safe, 1,237 Toxic and 1,343 Obscene comments are correctly classified. DistilBERT correctly classifies 1,207 Safe, 1,189 Toxic and 1,325 Obscene comments. RoBERTa correctly classifies 1,191 Safe, 1,229 Toxic and 1,330 Obscene comments.

The most common errors involve confusion between Safe and Toxic, and between Toxic and Obscene. This is reasonable because some toxic comments may include strong language without crossing the obscene threshold, while some obscene comments may also be toxic in a broader sense. Toxic-RoBERTa reduces Safe-to-Toxic confusion compared with the other two models, which contributes to its superior overall performance.

Table 6.2: Fine-tuned confusion matrix diagonal summary.

Model	Safe correct	Toxic correct	Obscene correct
DistilBERT	1207	1189	1325
Toxic-RoBERTa	1304	1237	1343
RoBERTa	1191	1229	1330

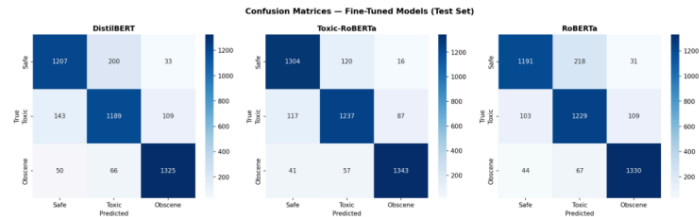


Figure 6.3: Confusion matrices for fine-tuned models on the test set.

6.5 Model Size and Deployment Trade-Off

The size comparison strongly supports quantization as a deployment technique. DistilBERT reduces from 255.4 MB to 91.0 MB, a reduction of 64.4 percent. Toxic-RoBERTa and RoBERTa reduce from 475.5 MB to 148.9 MB, a reduction of 68.7 percent. These reductions are valuable when models must be stored on limited devices, downloaded over networks or deployed as multiple model versions.

The results in terms of latencies need to be interpreted with care. Whereas fine-tuned models have been tested using GPU latency, the quantized models have been tested for their CPU latency. Therefore, the latency result here in terms of INT-8 is higher than before. This does not negate the usefulness of quantization; all this indicates is that when one wants to conduct a latency test, it should be compared on CPU using both FP32 and INT-8.

Table 6.3: Model size reduction after INT-8 quantization.

Model	FP32 fine-tuned size (MB)	INT-8 size (MB)	Reduction
DistilBERT	255.4	91.0	64.4%
Toxic-RoBERTa	475.5	148.9	68.7%
RoBERTa	475.5	148.9	68.7%

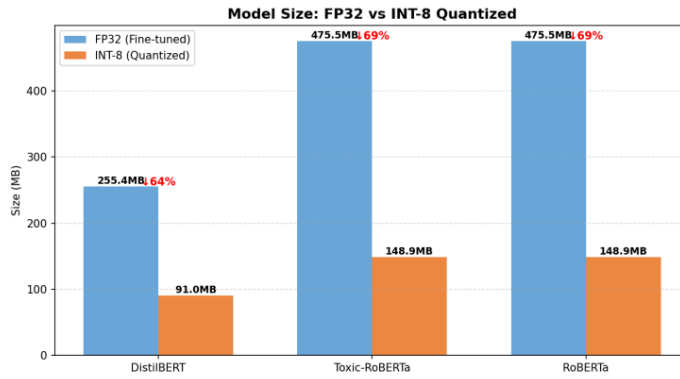


Figure 6.4: Model size comparison between FP32 fine-tuned and INT-8 quantized versions.

6.6 Overall Recommendation

With the focus being on maximum efficiency, Toxic-RoBERTa emerges as the optimal model for use in this research. This is because its quantized and fine-tuned versions attain F1-scores that are around 0.898. However, considering the case where space is the major limitation and slight sacrifices in accuracy can be made, the quantized version of Toxic-RoBERTa will be the ideal option

If a simpler and smaller architecture is preferred from the beginning, DistilBERT is still useful. Its fine-tuned model reaches macro F1-score of 0.8608 with a size of 255.4 MB, and its quantized version is only 91.0 MB. However, for this dataset and label mapping, Toxic-RoBERTa provides the best balance between accuracy and post-quantization robustness.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

This thesis presented a complete pipeline for resource-efficient toxic comment classification using fine-tuned and INT-8 quantized transformer models. The raw Jigsaw dataset was transformed into a three-class dataset with labels Safe, Toxic and Obscene. The original imbalance was addressed by sampling 9,603 comments from each class, producing a balanced dataset of 28,809 samples. The dataset was split into training, validation and test sets using a 70:15:15 ratio.

Three models were compared: DistilBERT, Toxic-RoBERTa and RoBERTa. Base model evaluation showed poor performance because the task-specific head was not trained. Fine-tuning significantly improved all models. Toxic-RoBERTa achieved the best fine-tuned result with accuracy of 0.8987 and macro F1-score of 0.8985. RoBERTa and DistilBERT also performed strongly, but below Toxic-RoBERTa.

INT-8 dynamic quantization reduced model size by 64.4 percent for DistilBERT and 68.7 percent for both RoBERTa variants. Toxic-RoBERTa retained almost all of its fine-tuned accuracy after quantization, making it the strongest candidate for deployment where storage reduction is important. The thesis also emphasized that latency must be evaluated under matching hardware conditions before drawing final speed conclusions.

7.2 Major Findings

The first major finding is that domain-oriented transformer models can provide measurable advantages for toxicity classification. Toxic-RoBERTa outperformed the general-purpose RoBERTa and the compact DistilBERT model in this experiment. The second finding is that balancing the dataset is important for meaningful macro-level evaluation because raw Jigsaw labels are highly imbalanced after mapping.

The third finding is that dynamic quantization can provide large storage reductions with small accuracy losses, especially for Toxic-RoBERTa. The fourth finding is that model size and latency must be discussed separately. Quantization reduced storage, but the CPU execution environment produced higher latency values than GPU fine-tuned inference in the recorded experiment.

7.3 Limitations

The study uses a threshold-based label mapping, and different thresholds could change the final label distribution. The task is restricted to three classes and does not perform multi-label classification. Some toxicity categories, such as threat and identity attack, are merged into the general Toxic class. The dataset examples are balanced through down-sampling, which improves class balance but discards many Safe and Toxic examples from the raw dataset.

The training configuration is consistent but not exhaustively tuned. Larger hyperparameter search, longer training or data augmentation may improve results. Latency comparison is limited by the difference between GPU evaluation for fine-tuned models and CPU evaluation for quantized models. A dedicated deployment benchmark would be required for stronger conclusions about runtime speed.

7.4 Future Scope

Future work can extend this thesis in several directions. First, the label mapping can be expanded to multi-label classification so that a comment can simultaneously be toxic, obscene, insulting or threatening. Second, threshold calibration can be performed using validation data to improve class separation. Third, fairness and bias evaluation can be added to study whether the model overpredicts toxicity for identity-related terms.

Fourth, model compression can be expanded beyond dynamic quantization. Quantization-aware training, ONNX Runtime optimization, pruning and knowledge distillation can be compared. Fifth, a proper CPU benchmark should compare FP32 CPU inference and INT-8 CPU inference using identical batch sizes. Sixth, explainability methods can be added to show which tokens influence model decisions, helping moderators understand model predictions.

Finally, the model can be integrated into a simple moderation API or dashboard. Such a deployment could demonstrate how predicted class, confidence score and moderation action can be presented to human reviewers.

REFERENCES

- [1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. and Polosukhin, I. (2017). Attention Is All You Need. *Advances in Neural Information Processing Systems*.
- [2] Devlin, J., Chang, M. W., Lee, K. and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of NAACL-HLT*.
- [3] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.
- [4] Sanh, V., Debut, L., Chaumond, J. and Wolf, T. (2019). DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. *arXiv preprint arXiv:1910.01108*.
- [5] Jigsaw and Conversation AI. Jigsaw Unintended Bias in Toxicity Classification Dataset. *Kaggle competition dataset*.
- [6] Wulczyn, E., Thain, N. and Dixon, L. (2017). Ex Machina: Personal Attacks Seen at Scale. *Proceedings of the International Conference on World Wide Web*.
- [7] Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H. and Kalenichenko, D. (2018). Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. *Proceedings of CVPR*.
- [8] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G. et al. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems*.
- [9] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A. et al. (2020). Transformers: State-of-the-Art Natural Language Processing. *Proceedings of EMNLP: System Demonstrations*.
- [10] Hugging Face. Transformers documentation and model cards for distilbert-base-uncased, roberta-base and unitary/unbiased-toxic-roberta.

APPENDIX A

NOTEBOOK WORKFLOW SUMMARY

The implementation notebook contains 24 cells. It installs compatible library versions, imports required packages, mounts Google Drive, loads the Jigsaw dataset, creates three-class labels, balances the dataset, performs a 70:15:15 split, defines tokenization and evaluation helpers, registers three models, evaluates base models, fine-tunes the models, applies INT-8 dynamic quantization, evaluates quantized models and exports plots and result summaries.

Key variables used in the notebook

```
DATA_PATH =  
"/content/drive/MyDrive/My_Thesis_Project/Jigsaw_dataset/train.csv"  
MAX_LEN = 128  
NUM_LABELS = 3  
LABEL2ID = {"Safe": 0, "Toxic": 1, "Obscene": 2}  
MIN_CLASS = 9603  
TRAIN/VALIDATION/TEST = 70/15/15  
Learning rate = 2e-5, epochs = 3, train batch size = 32
```

APPENDIX B

FULL EXPERIMENTAL RESULT TABLE

The table below repeats the full result table so that the thesis remains self-contained.

Model	Stage	Accuracy	F1-Macro	Precision	Recall	Latency(ms)	Size(MB)
DistilBERT	Base	0.353771	0.279138	0.291677	0.353847	0.180	255.416
DistilBERT	Finetuned	0.860944	0.860804	0.860843	0.860939	0.116	255.416
DistilBERT	Quantized	0.844516	0.843424	0.849916	0.844532	112.869	90.996
Toxic-RoBERTa	Base	0.254975	0.208938	0.179092	0.254916	0.358	475.494
Toxic-RoBERTa	Finetuned	0.898658	0.898539	0.898507	0.898660	0.185	475.494
Toxic-RoBERTa	Quantized	0.898427	0.898193	0.898379	0.898431	225.195	148.916
RoBERTa	Base	0.333410	0.166696	0.111137	0.333333	0.351	475.494
RoBERTa	Finetuned	0.867654	0.867679	0.868884	0.867644	0.191	475.494
RoBERTa	Quantized	0.860250	0.859249	0.860550	0.860256	231.765	148.916

ORIGINALITY REPORT

5%

SIMILARITY INDEX

5%

INTERNET SOURCES

3%

PUBLICATIONS

3%

STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Hong Kong University of Science and Technology Student Paper	1%
2	aclanthology.org Internet Source	1%
3	scholarshare.temple.edu Internet Source	1%
4	www.mdpi.com Internet Source	1%
5	dr.ur.ac.rw Internet Source	1%
6	Submitted to Universiti Sains Islam Malaysia Student Paper	<1%
7	cpcontacts.thescipub.com Internet Source	<1%
8	par.nsf.gov Internet Source	<1%
9	Submitted to De Montfort University Student Paper	<1%

10	Submitted to Malta College of Arts, Science and Technology Student Paper	<1 %
11	repositum.tuwien.at Internet Source	<1 %
12	Submitted to University of North Texas Student Paper	<1 %
13	ai.jmir.org Internet Source	<1 %
14	repositorio.ugto.mx Internet Source	<1 %
15	Uwe Engel, Anabel Quan-Haase, Sunny Xun Liu, Lars Lyberg. "Handbook of Computational Social Science - Data Science, Statistical Modelling, and Machine Learning Methods", Routledge, 2021 Publication	<1 %

Exclude quotes On

Exclude matches

< 14 words

Exclude bibliography On

FINAL GRADE

/0

GENERAL COMMENTS

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

Vinay Grewal

M.TechThesis

 Quick Submit Quick Submit Kurukshetra University

Document Details

Submission ID

trn:oid::1:3598181809

Submission Date

Jun 20, 2026, 5:55 PM GMT+5:30

Download Date

Jun 20, 2026, 5:59 PM GMT+5:30

File Name

VinayThesis.pdf

File Size

523.9 KB

34 Pages

7,059 Words

41,330 Characters

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (i.e., our AI models may produce either false positive results or false negative results), so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.