Project Machine Learning
— Milestone 2 —

[Imene Ben Ammar, Julian Dobler, Yannik Queisler]

January 3, 2025

Abstract

In this milestone, we fine-tuned BERT for toxic comment classification, building on the foundational work of implementing baseline models in the previous milestone. The baseline methods – logistic regression, support vector machines, and naive Bayes – were revisited to establish benchmarks for evaluating BERT's performance. The fine-tuning process involved experimenting with various configurations to identify the optimal setup for BERT. Key results highlighted significant performance gains by the fine-tuned BERT model, particularly in metrics such as class 1 F1-score and AUC-ROC, underscoring its ability to capture contextual nuances in text more effectively than the baseline models. Analyzing low-confidence predictions revealed that the notion of toxicity can be partly subjective and should be adjusted to context (e.g., by threshold). Additionally, a sequence length of 128 might present problems in longer comments with toxicity at the end. Addressing these challenges is a priority for future work, along with expanding the fine-tuning efforts to other datasets, such as the Jigsaw dataset, for deeper analysis and broader applicability. Furthermore, it is recommended that the model should only be deployed with a confidence score accompanying its predictions to ensure transparency and reliability in real-world applications. The project's codebase is publicly accessible on GitHub.[1]

Contents

[1] `https://github.com/devWhyqueue/pml-bert`

1 Introduction

The previous milestone established baseline benchmarks for the task of toxic comment classification by implementing classical ML methods. These methods included logistic regression, SVM and naive Bayes, evaluated primarily on the Civil Comments dataset. While the best-performing baseline model achieved a peak F1-score of 0.62 for the toxic class, the overall performance was deemed unsatisfactory.[2] This result underscores the limitations of classical ML methods in handling the inherent complexities of the toxic comment classification task, particularly in the presence of significant class imbalance. A key finding of the previous milestone was that class imbalance adversely affected the precision and recall of the minority toxic class (C1). Addressing this issue remains critical as we transition to more advanced approaches, such as fine-tuning the BERT model.

The goal of this milestone is to evaluate the performance of fine-tuned transformer-based algorithms, specifically BERT, on the toxic comment classification task. By leveraging BERT's deep contextual understanding and pre-training on extensive text corpora, we aim to overcome the shortcomings observed with classical ML approaches. Furthermore, we will examine whether the class imbalance challenge persists when using more sophisticated models and explore strategies for mitigating its impact.

This report is structured as follows. Section 2 provides a concise description of the methods employed, including data preprocessing, model selection, and hyperparameter optimization. Section 3 discusses generalization, showcasing some best and worst case predictions. Section 4 presents the experimental results, including detailed performance metrics, error analysis, and comparisons to baseline methods. Finally, section 5 evaluates the computational efficiency of the fine-tuning process, explores confidence measures for predictions, and assesses the overall suitability of BERT for this application.

Through this milestone, we aim to establish a robust evaluation of BERT's capabilities and its advantages over classical baselines, paving the way for further refinements in toxic comment classification.

2 Methodology

In this section, we describe the methods employed for fine-tuning the BERT model on the task of toxic comment classification. The methodology includes preprocessing of textual data, model architecture design, training strategies, and hyperparameter optimization.

2.1 Feature selection and preprocessing

As discussed in the previous milestone report, there are several approaches to transform textual data into numerical vectors. Traditional methods such as BoW and TF-IDF rely on token frequencies. More modern techniques utilize pre-trained embeddings, such as Word2Vec, GloVe, and FastText. In this project, we use embedding-based input representations as employed in the BERT model. BERT's input representation combines token, segment, and positional embeddings to effectively capture context-dependent relationships. [1]

Although preprocessing did not show significant benefits for the baseline models, its impact will be re-evaluated during BERT fine-tuning. Specific preprocessing steps are discussed in the prior milestone report. For BERT, tokenization is performed using the WordPiece tokenizer, which splits words into subwords to handle out-of-vocabulary issues. Padding and truncation are applied to ensure input sequences have uniform lengths. The default maximum sequence length is 512. However, we will experiment with shorter lengths to optimize computational efficiency.

2.2 Methods

The task of toxic comment classification can be defined as a binary classification problem. To address this, we added a classification head on top of the pre-trained BERT model. This head consists of a fully connected linear layer that outputs logits, which are then passed through a sigmoid activation function to produce probabilities for binary classification.

The fine-tuning procedure involves updating all layers of the BERT model. The algorithm optionally balances the dataset to a specified positive class proportion (`pos_proportion`) and then fine-tunes the model for a predefined number of epochs.

[2]The results reported in the previous milestone contained errors. A detailed explanation and the corrected results can be found in appendix A.

Given an input sequence, BERT generates token embeddings, which are contextualized representations of the input tokens. Along with the input tokens, an `attention_mask` is provided to inform the model about padded tokens that should not contribute to the computations. These embeddings are passed through the classification head to compute the logits, which represent the raw predictions of the model.

The loss is computed using the binary cross-entropy with logits loss (BCEWithLogitsLoss), which combines a sigmoid activation with a cross-entropy loss function. Backpropagation is performed using the AdamW optimizer, a widely adopted optimizer in transformer-based models due to its improved weight decay mechanism. [2]

To enhance efficiency, we progressively refined the training and evaluation pipeline using distributed computing and precision optimization strategies. Initially, we adopted distributed data parallelism (DDP) to parallelize both training and evaluation across up to eight NVIDIA A100 GPUs, each with 80 GB of VRAM. This implementation leveraged PyTorch's DDP framework, which is well-documented for its ability to minimize inter-GPU communication overhead while maintaining synchronized updates. Specifically, DDP ensures that gradients are averaged across all processes during the backward pass, enabling seamless scaling across multiple devices. [3]

Furthermore, mixed-precision training using FP16 format and dynamic gradient scaling was employed. This approach reduced memory consumption and accelerated matrix operations without compromising numerical stability. Together, these optimizations formed a robust pipeline capable of handling large-scale data efficiently. [4]

2.3 Hyperparameter and model selection

The hyperparameter search was performed to identify the best configuration for BERT fine-tuning. Detailed results of all our experiments can be found in our GitHub repository and a selection will be shown in section 4. [5]
The following hyperparameters were considered:

- **Sequence length:** $[64, 128, 256, 512]$
- **Positive class proportion:** [as is (~ 0.06), $0.1, 0.25$]
- **Batch size:** $[64, 256, 1024]$
- **Preprocessing:** [False, True]
- **Learning rate:** $[1 \times 10^{-7}, 1 \times 10^{-6}, 1 \times 10^{-5}, 1 \times 10^{-4}, 1 \times 10^{-3}]$
- **Number of epochs:** $[1, 2, 3]$
- **Weight decay:** $[0, 0.1, 1, 10, 100]$
- **Optimizer:** AdamW
- **Loss function:** BCEWithLogitsLoss

AdamW was chosen as the optimizer due to its decoupled weight decay mechanism, which has been shown to be effective for fine-tuning transformer models like BERT. This choice aligns with its widespread adoption in transformer-based tasks. AdamW effectively addresses overfitting by decoupling weight decay from the gradient updates, thereby preserving the learning dynamics. [6, 7, 8]

BCEWithLogitsLoss was selected for its compatibility with binary classification tasks, where logits (raw model outputs) are directly utilized. This avoids numerical instabilities that could arise when using separate sigmoid activations followed by cross-entropy loss. Its seamless integration with the sigmoid activation ensures reliable gradient flow, making it the standard choice for binary classification. [9]

3 Generalization

One of the cornerstones in designing efficient ML systems is accurately estimating the performance of learning algorithms. Among various theoretical frameworks, the theory of uniform convergence of empirical quantities to their mean, as pioneered by Vapnik, provides a robust foundation for understanding risk (or generalization error). By examining empirical accuracy measurements in conjunction with complexity

metrics (e.g., the Vapnik–Chervonenkis dimension or the fat-shattering dimension), one can systematically gauge how well a learned model will perform on previously unseen data. [10, 11]

In practice, these theoretical insights are put into operation by employing suitable error measures, which quantify how closely a model's predictions align with ground truth. In this project, we follow the approach detailed in our previous milestone and evaluate our models using accuracy, precision, recall, F1-Score, and AUC-ROC. [12]

Precision helps minimize the incorrect flagging of benign comments as toxic, whereas recall focuses on reducing missed detections of actual toxic content. F1-Score balances these two metrics, making it particularly suitable for scenarios with class imbalance. Meanwhile, AUC-ROC provides a holistic view by measuring the model's capacity to distinguish between positive and negative classes across various decision thresholds.

Despite the utility of all these metrics, we place special emphasis on the F1-Score for the toxic class, given its effectiveness with imbalanced datasets. Other measures often yield satisfactory results across different models, but it is precisely on this toxic class F1 that the better-performing approaches distinguish themselves. These considerations enable a nuanced understanding of our model's strengths and weaknesses, especially when compared to baseline classifiers.

3.1 Empirical estimates

The generalization error was estimated using the Civil Comments test dataset for all baseline models and BERT, as illustrated in fig. 1. The BERT configuration follows the setup described in section 4.3. Among the baseline models, Naïve Bayes exhibited the weakest performance, although it remained computationally efficient. In contrast, SVM and logistic regression achieved comparable results, with an F1-score of (at least) 0.61 for the toxic class and an AUC-ROC of 0.94.

BERT outperformed these baselines significantly, particularly in the F1-score for the positive class, reaching 0.7, and demonstrated superior AUC-ROC performance with a score of 0.98. These empirical estimates are considered reliable, as the results obtained from the test dataset closely align with those observed on the validation dataset, supporting the consistency and robustness of the model's performance across different data splits.

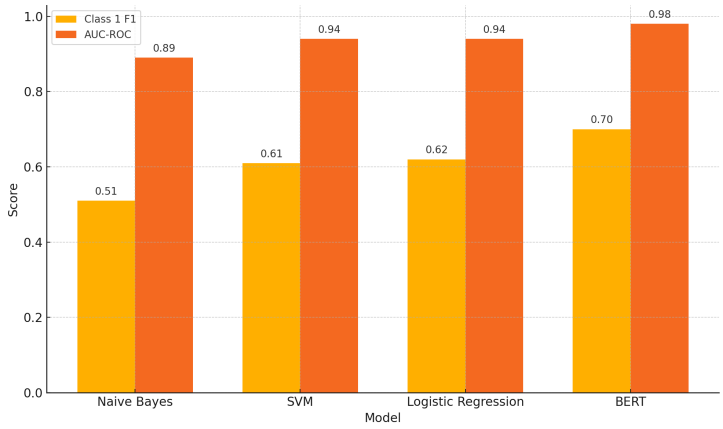


Figure 1: Models' performance on the Civil Comments test dataset.

3.2 Test cases

For evaluating our model and baselines, we utilize precision, recall (in F1), and AUC-ROC as error measures. However, these metrics alone are insufficient for identifying the best or worst individual predictions. To address this, we introduce a confidence score defined as follows:

$$C = \max(p, 1 - p), \tag{1}$$

where p is the probability of the positive class output by the model. This score ranges from 0.5 to 1, with higher values indicating greater confidence in the model's prediction. Using this measure, we classify predictions into three categories: low (0.5-0.66), moderate (0.66-0.85), and high (0.85-1) confidence predictions.

Figure 5 in the appendix illustrates the distribution of confidence scores for the baseline method and BERT. The left column shows the confidence score distribution, while the right column highlights the incorrect predictions. Both models exhibit a significant concentration of predictions in the high-confidence range. Notably, low-confidence predictions are rare, suggesting limited expression of uncertainty by both methods.

Furthermore, we observe that the baseline model produces a nearly uniform distribution of confidence scores for false predictions while high confidence scores for false predictions become less frequent in BERT. This indicates that SVM lacks confidence in its predictions and has a limited ability to separate correct from incorrect classifications. BERT, on the other hand, demonstrates a better grasp of uncertainty and reflects a more realistic confidence assessment. Its calibration is superior to the baseline model, even if it still produces false predictions. To better understand its performance in high and low confidence ranges, we analyze examples of high- and low-confidence predictions.

3.2.1 High confidence

This category encompasses both the best predictions (high confidence, correct classification) and the worst predictions (high confidence, incorrect classification). Representative examples of correct high-confidence predictions are listed in table 1, while incorrect high-confidence predictions appear in appendix C.

The distinction between toxic and non-toxic comments is primarily determined by linguistic features and the model's confidence scores. Non-toxic comments typically exhibit polite or neutral language—often expressing gratitude, affirmation, or sharing information—and are associated with exceptionally high confidence scores, indicating strong certainty in their classification. In contrast, toxic comments frequently feature hostile, insulting, or profane language, accompanied by slightly lower confidence scores. This discrepancy reflects the greater lexical and contextual variability inherent in toxic expressions. Additionally, toxic comments are often shorter and more direct, simplifying their identification but also introducing challenges in detecting subtle forms of toxicity, such as sarcasm or coded language. Notably, for this analysis, the top three high-confidence toxic comments were not included because many top predictions were highly repetitive, consisting of variations of simple insults (e.g., "idiot"). These observations highlight the model's reliance on explicit linguistic cues for toxicity detection while suggesting opportunities for improving its ability to capture more nuanced harmful expressions.

To gain deeper insights into the model's performance, we take a closer look at some of the incorrect predictions. While "stupid" is clearly a false negative (likely due to the Unicode emoji) and "damn right" is a false positive, the toxicity of "Total Clueless Jerk" and "Flat earther moron." can be debated. Both have relatively high (0.31 and 0.48) ground-truth toxicity scores but do not surpass the threshold of 0.5. The other false negatives are lengthy comments, with the toxic language near the end; this suggests that the 128-sequence-length limit may contribute to misclassification.

Table 1: Correct high-confidence predictions

Non-toxic predictions

Comment	**Confidence Score**
Thank you for the link.	0.9985
Thanks for the info.	0.9984
Hope so!	0.9984

Toxic predictions

Comment	**Confidence Score**
Your an idiot	0.9394
Stupid is as stupid does.	0.9359
Eat shit you stupid fuk.	0.9323

3.2.2 Low confidence

This category covers a spectrum of borderline cases, encompassing both correct and incorrect predictions. As also shown in appendix C, many of these comments include insulting or divisive language, yet lack the explicit markers of aggression that would yield high-confidence toxic predictions. Because the classifier hovers around the decision boundary, these instances are particularly valuable for model refinement. Their inherent ambiguity stems from context-dependent cues (e.g., political commentary or sarcasm) and the nuanced interplay of language signals (such as subtle insults or veiled hostility). Consequently, low-confidence classifications highlight the need for more robust context encoding, such as incorporating

discourse-level or metadata features, which could help the model better discern the overall intent of a comment.

Another important observation is that toxicity is partly subjective and highly influenced by context. This variability in human judgment suggests that a fixed threshold may not be suitable for all applications. Adjusting the classification threshold – either making it more lenient or more stringent – can help align the model with different use cases or community guidelines. At the same time, incorporating additional training data, especially for borderline or contextually sensitive cases, can reduce uncertainty and improve the consistency of the model's predictions.

3.3 Trade-offs between TPR and FPR

Building on the observations about high- and low-confidence predictions, we now turn to a broader examination of classification performance by analyzing the trade-offs between the true positive rate (TPR) and the false positive rate (FPR). To visualize these trade-offs, we used ROC curves, which provide a clear depiction of how well the models can distinguish toxic from non-toxic comments under varying decision thresholds.

As shown in Figure 2, all ROC curves lie above the random classifier line, indicating good discrimination capability overall. Among these models, BERT achieves the best performance, with its curve lying closest to the top-left corner – the ideal point where TPR is maximized and FPR is minimized. Conversely, Naïve Bayes exhibits the least favorable trade-off, with a curve that deviates further from the optimal corner. This indicates that it struggles more than the other models to minimize false positives while maintaining a high rate of true positives.

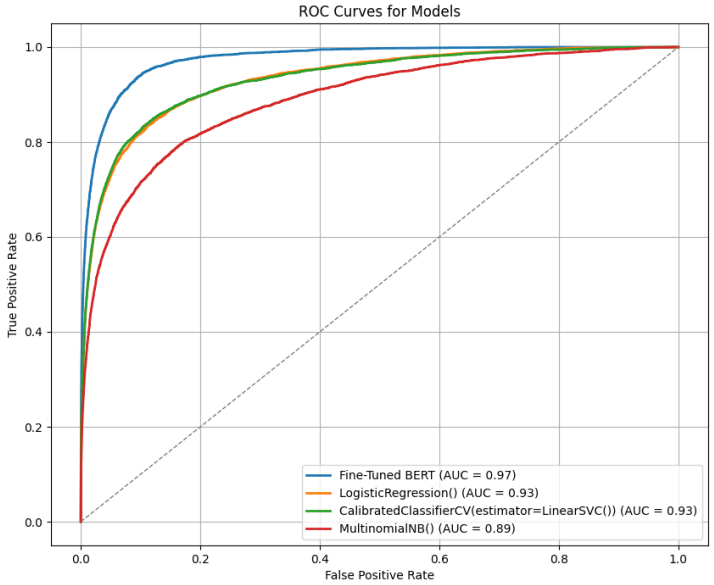


Figure 2: ROC Curves for BERT and baseline models

4 Results

In this section, we present our experimental outcomes from fine-tuning BERT on the Civil Comments dataset and discuss how various hyperparameters affect both model performance and computational efficiency. We highlight key insights gleaned from different configurations, explain their implications for deployment, and compare our findings to established practices in the literature.

4.1 Findings

Our experimental results indicate that BERT outperforms baseline models across all examined metrics, as shown in table 2. This demonstrates the effectiveness of BERT in handling the complexities of the Civil Comments dataset and highlights its superior performance in toxic comment classification tasks.

Table 2: BERT performance on the Civil Comments test dataset

Precision (C1)	Recall (C1)	F1 (C1)	Macro F1	Weighted F1	Acc.	AUC ROC	Loss
0.67	0.75	0.70	0.84	0.96	0.96	0.9731	0.2317

Our experiments revealed several key insights into the impact of hyperparameters on model performance and computational efficiency. These findings are summarized below:

Sequence length: Reducing the sequence length to 128 tokens achieved a 4x speed-up in training time. As illustrated in fig. 3, while comments in the validation dataset can be up to 256 tokens, the majority of comments fall well within 128 tokens. This reduction had no significant impact on model performance, suggesting that 128 tokens are sufficient for capturing the essential context of most comments.

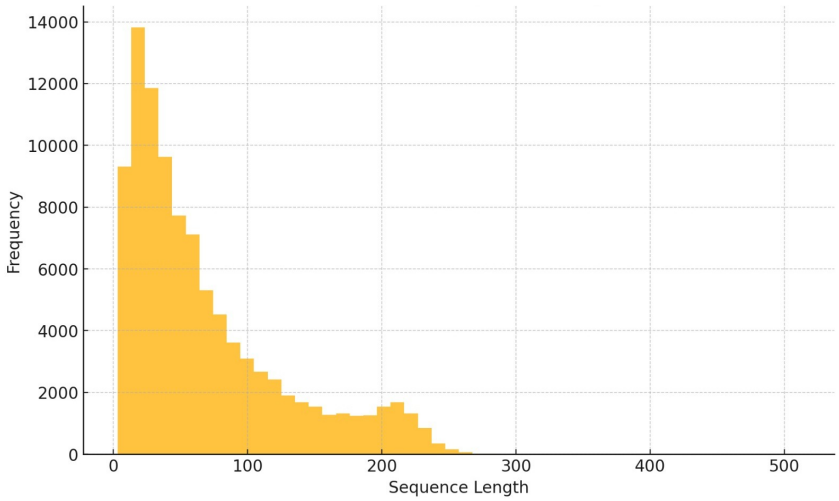


Figure 3: Distribution of comment lengths in the Civil Comments validation dataset.

Positive proportion: A positive class proportion of `pos_proportion=0.1` showed no significant impact on the model's performance, with changes in metrics remaining negligible ($\leq 1\%$). Despite this minimal effect on performance, this configuration achieved a substantial 38% reduction in data processing time. This outcome suggests that `pos_proportion=0.1` offers an optimal balance between computational efficiency and model effectiveness, making it particularly suitable for scenarios constrained by limited computational resources.

Batch size: Fine-tuning with larger batch sizes consistently yielded similar model performance while slightly increasing speed (17% faster). The maximum batch size that fits into the available GPU memory was 1024, which further highlights the efficient utilization of resources during fine-tuning.

Preprocessing: Applying preprocessing techniques had no discernible positive impact on model performance. This suggests that the model's tokenizer and pre-trained embeddings are sufficiently robust to handle raw input data without additional preprocessing steps.

Learning rate: A grid search across learning rates $[10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}]$ revealed that values within the range $[10^{-6}, 10^{-5}, 10^{-4}]$ were most effective. In contrast, both extremely low and extremely high learning rates prevented the model from effectively learning to identify the toxic class, leading to convergence failure or instability. These results underscore the necessity of selecting an intermediate learning rate to ensure stable and reliable training.

Number of epochs: Extending training beyond the first epoch did not result in improved model performance. For example, even with five epochs and a weight decay of 1.0, no notable gains were observed. While training for multiple epochs we also tried employing a cosine annealing schedule to adjust the learning rate, this did not lead to improvements in the validation metrics.

Weight decay: Applying weight decay had no meaningful effect on the model's performance. Only excessively high values led to a noticeable reduction in the model's learning capacity, reinforcing the importance of choosing reasonable values for this hyperparameter.

4.2 Comparison to literature

Several of the chosen hyperparameters align with established practices in the literature. For example, setting the sequence length to 128 tokens (or even less) is a common strategy to balance computational efficiency with model performance. [13]

For batch size, larger values like 1024 are feasible on modern GPUs and are known to improve training speed without adversely affecting generalization. Similarly, the chosen learning rate of $1e^{-4}$ falls within the range by other researchers, who fine-tuned BERT for various NLP tasks. [14, 15]

The decision to avoid preprocessing is supported by Kurniasih et al., who highlight that advancements in modern tokenization methods significantly reduce the necessity for extensive input cleaning, enabling models to effectively handle raw text data without substantial preprocessing efforts. [16]

The decision to use only a single epoch for fine-tuning is informed by the reduced need for extensive training in larger pre-trained models, where early stopping can still yield strong generalization performance. Similarly, the impact of weight decay appears to be less critical, as other regularization techniques, such as dropout, are already applied. [17, 18]

4.3 Chosen configuration

In summary, the fine-tuning of BERT for toxic comment classification demonstrates the importance of judicious hyperparameter selection. Reducing sequence length to 128 tokens resulted in substantial computational gains with no loss in performance. Larger batch sizes were utilized effectively, balancing memory constraints and training speed. Weight decay and preprocessing were found to have minimal impact, while learning rates within $[10^{-6}, 10^{-5}, 10^{-4}]$ proved optimal for convergence. Additionally, extending training beyond one epoch provided no significant advantages, confirming the sufficiency of shorter training durations. Leveraging an 8x NVIDIA A100 80 GB GPU setup, the model achieved fine-tuning plus evaluation within an impressive seven minutes.
The chosen hyperparameter configuration is as follows:

$$\text{Config} \begin{pmatrix} \text{sequence\_len} = 128, \\ \text{pos\_proportion} = 0.1, \\ \text{batch\_size} = 1024, \\ \text{preprocessing} = \text{False}, \\ \text{learning\_rate} = 1e-4, \\ \text{num\_epochs} = 1, \\ \text{weight\_decay} = 0, \\ \text{optimizer} = \text{AdamW}, \\ \text{loss} = \text{BCEWithLogitsLoss} \end{pmatrix}$$

This setup provides a robust balance between computational efficiency and model performance, forming a strong foundation for scalable real-world deployment.

5 Discussion

Below, we reflect on the cost of training, re-training feasibility, confidence measure reliability, and final conclusions regarding the suitability of each learning algorithm. Training costs varied significantly across methods. Baseline classifiers using TF-IDF vectorization with 5,000 features took roughly one hour of CPU time each, alongside very high RAM usage of approximately 256 GB. In contrast, fine-tuning BERT took only about seven minutes on an 8x A100 GPU setup, which demonstrates remarkable efficiency once

a pre-trained model is available—although training that model from scratch would be far more time-consuming. Retraining a fine-tuned BERT as new data arrives is therefore straightforward if adequate GPU resources and pre-trained checkpoints exist, while organizations with more limited resources might opt for traditional methods despite the high memory requirements.

To assess how well the models' predicted probabilities reflect actual risk, calibration plots were examined in fig. 4. Logistic Regression and SVM showed strong calibration, with only slight deviations at higher probability levels, whereas Naive Bayes exhibited poorer calibration, tending to understate its certainty. BERT's curve formed an S-shape, suggesting that its probabilities are not well-aligned with real-world likelihoods, a shortcoming that could be addressed by employing post-training calibration methods such as temperature scaling. Still, BERT reached the highest performance (F1 score of 0.70 for the toxic class), making it a compelling choice where both speed and robustness are priorities. [19]

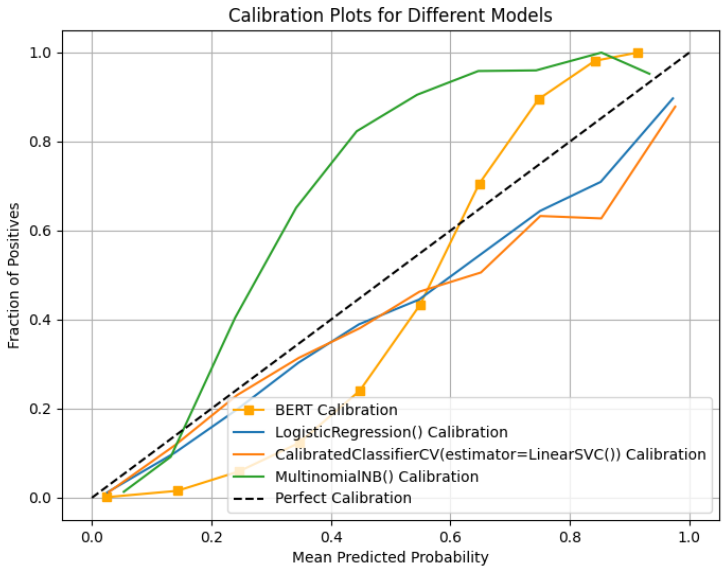


Figure 4: Calibration plot for BERT and baseline models

Traditional algorithms may be easier to train on CPUs and more accessible for quick iteration, though they can demand substantial RAM. Fine-tuned BERT, on the other hand, delivers stronger performance and can be retrained efficiently with adequate GPU support – ideal for real-world scenarios requiring rapid updating. Adjusting the decision threshold can tailor the model's behavior to specific guidelines or risk tolerances, and further improvements might come from adding more data (e.g., the Jigsaw Toxic Comment dataset) or transitioning to advanced transformer variants such as RoBERTa. Overall, BERT stands out as the most effective option when the necessary hardware is available, while simpler methods still offer a calibrated, albeit less powerful, alternative. [20]

A Erratum

In our first milestone report, we presented results for baseline methods (e.g., logistic regression, SVM) evaluated on toxic comment classification tasks. However, an error in the data balancing function led to incorrect results. This erratum clarifies the error and presents corrected findings. [12]

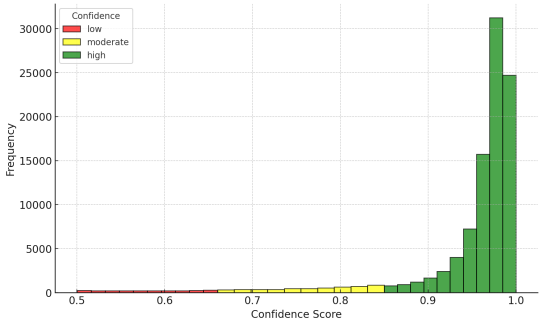
The error occurred in the data balancing function, which ensured a specified proportion of positive examples in the dataset. The function retained all positive examples and undersampled the negative class to achieve the target ratio. However, an example was considered positive only if toxicity = 1 and negative if toxicity = 0, instead of using the correct threshold where a sample is toxic if toxicity > 0.5, and non-toxic otherwise. This incorrect labeling reduced the dataset size, limiting the training data available to baseline methods. As a result, the models produced worse results that were not representative of their actual performance. We fixed the issue by applying the correct threshold, which significantly increased the amount of training data and allowed the baseline methods to perform as expected. However, the increased dataset size introduced new computational challenges: The random forest baseline had to be removed because it was no longer computationally feasible as an ensemble method. Other methods, such as logistic regression and SVM, also required considerable time and RAM resources with the adjusted training dataset.

Corrected results presented in table 3 confirm that logistic regression remains the best-performing baseline regarding F1 score for the toxic class (C1), achieving a score of 0.62 compared to the previously reported 0.52. Furthermore, preprocessing continues to yield no significant improvement, aligning with prior conclusions. Finally, baseline methods remain sensitive to class imbalance, emphasizing the need for appropriate resampling strategies. While this correction increases baseline performance, the overall findings of the first milestone remain valid. Logistic regression and SVM both perform almost equally well, preprocessing shows minimal benefit, and addressing class imbalance is critical for achieving robust performance. Most importantly, the corrected baseline F1 score of 0.62 sets a higher benchmark for our fine-tuned BERT model.

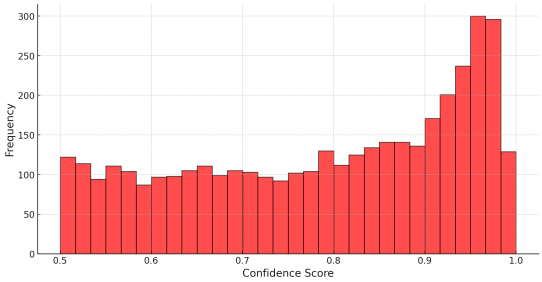
Table 3: Corrected classifier performance on the Civil Comments validation dataset

Method	**Prep.**	**Pos. Prop.**	**Precision (C1)**	**Recall (C1)**	**F1 (C1)**	**Macro F1**	**Weighted F1**	**Acc.**	**AUC ROC**
Naïve Bayes	True	None	**0.98**	0.06	0.11	0.54	0.92	0.95	0.89
		0.1	0.93	0.12	0.21	0.59	0.93	0.95	0.89
		0.25	0.64	0.43	0.51	0.74	0.95	0.95	0.89
		0.5	0.19	0.84	0.31	0.59	0.84	0.79	0.89
	False	None	**0.98**	0.06	0.11	0.54	0.92	0.95	0.89
		0.1	0.93	0.12	0.21	0.59	0.93	0.95	0.89
		0.25	0.64	0.43	0.51	0.74	**0.96**	0.95	0.89
		0.5	0.19	0.84	0.32	0.60	0.84	0.79	0.89
Support Vector Machine	True	None	0.76	0.45	0.56	0.77	0.95	**0.96**	0.93
		0.1	0.69	0.54	0.61	**0.79**	**0.96**	**0.96**	0.93
		0.25	0.52	0.73	0.61	**0.79**	0.95	0.95	**0.94**
		0.5	0.35	0.84	0.50	0.72	0.92	0.90	**0.94**
	False	None	0.76	0.45	0.56	0.77	0.95	**0.96**	0.93
		0.1	0.69	0.54	0.61	**0.79**	**0.96**	**0.96**	0.93
		0.25	0.52	0.73	0.61	**0.79**	0.95	0.95	**0.94**
		0.5	0.35	**0.85**	0.50	0.72	0.92	0.90	**0.94**
Logistic Regression	True	None	0.77	0.45	0.57	0.77	**0.96**	**0.96**	0.93
		0.1	0.70	0.53	0.60	**0.79**	**0.96**	**0.96**	0.93
		0.25	0.54	0.71	0.61	**0.79**	0.95	0.95	**0.94**
		0.5	0.36	0.84	0.51	0.73	0.92	0.91	**0.94**
	False	None	0.77	0.44	0.56	0.77	0.95	**0.96**	0.93
		0.1	0.70	0.53	0.60	**0.79**	**0.96**	**0.96**	0.93
		0.25	0.54	0.71	**0.62**	**0.79**	0.95	0.95	**0.94**
		0.5	0.37	0.84	0.52	0.73	0.92	0.91	**0.94**

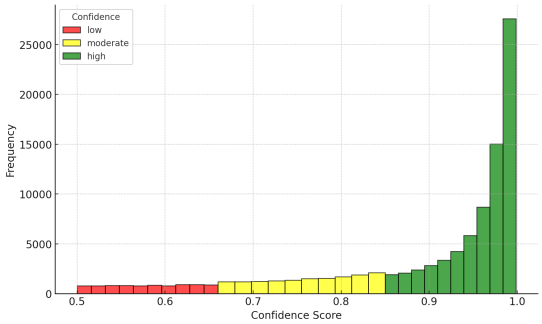
B Confidence scores



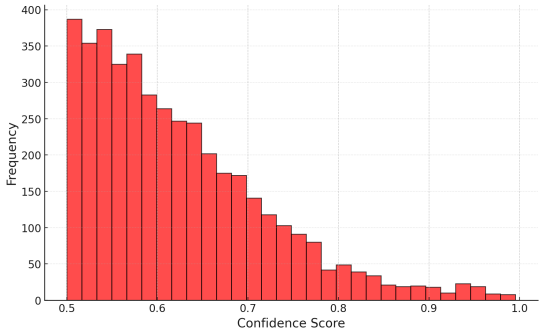
(a) Confidence score distribution for SVM



(b) Incorrect predictions for SVM



(c) Confidence score distribution for BERT



(d) Incorrect predictions for BERT

Figure 5: Comparison of confidence score distributions and incorrect predictions between SVM and BERT models. Each row represents one model, with the left plot showing confidence score distribution and the right plot showing incorrect predictions.

C Example comments

Table 4: False positives with high confidence

Comment	**Confidence Score**
Total Clueless Jerk	0.8968
Flat earther moron.	0.8901
Damn right	0.8832

Table 5: False positives with low confidence

Comment	**Confidence Score**
The only thing malignant is the rotting flesh between Bernstein's ears.	0.5000
Calm down. Taking out an airbase is hardly killing millions. It was done tactically perfect. Our troops were not at risk and collateral damage was minimal. Well done and beautiful. He will do things right compared to the loser we had for the past 8 years and the loser that could have been.	0.5001
garycrum - Nice troll post. Why are you calling me and others in here "Don"?	0.5002

Table 6: False negatives with high confidence

Comment	Confidence Score
Stupid(:point_down:)	0.9954
Always pleasure GREENLEAF, just google "Plate Climatology Theory", it has only come to the fore-front of research in the last decade or so, there are many articles on it. It's in its infancy, but as it points out, data on this underwater activity with vents and huge volcanoes that truly dwarf those on land, especially the plate faults and their ever releasing not just CO2, but methane and other gases that effect the environment, has been scarce. As you point out, many through history believed their impact to be negligible, but that opinion is being researched, new data gathered and challenged. Our government is very much part of the polarized political debate on climate change, sadly, when opposition presents, it is dismissed often out of hand. We assumed the underwater volcanoes and vents acted a particular way as to be of no concern, they don't throw into the atmosphere thus they don't bring cooling. This damn "CIVIL" format limits me from explaining, it's awful! I'd say more.	0.9871
My wife retired as a full-time nursing professor a couple of years ago and her position is now being staffed by several part-timers. The nursing program has been running since the colleges took them over in the 60's and has no signs of slowing down. How is this reacting to market forces? They need experienced full-time staff for the majority of the program delivery and continuity to ensure quality. The reality is that instead these colleges have loaded up management administrators at the expense of front-line teaching staff. The number of these managers has DOUBLED over the past 16 years while the number of faculty members has decreased relative to the number of students in the same period. I estimate that the college system could save $200 million annually just by going back to the 2001 ratio of management to students. What other organizations have doubled their management staff ratio? It's ridiculous.	0.9863

Table 7: False negatives with low confidence

Comment	Confidence Score
Demboski is an embarrassment to Alaskans everywhere. I'm sorry that you had to deal with such blatant bigotry and prejudice, Mr. Jones. Don't let people like Demboski run you out of Alaska. The people of Anchorage have already emphatically rejected her once, it's time to do it again. To Ms. Demboski: You are contributing nothing worthwhile to the political and cultural landscape of Alaska. Didn't getting rejected by the people of Anchorage teach you that your brand of nonsense is not welcome here? Do us all a favor and resign. The city and state will be better without you involved in its politics.	0.5008
Well just how bloody crazy can things get? So, if you're equipped to make a stand-up job of urination - you get to use the men's toilet, otherwise, go sit in the ladies room. It's really simple, but the liberals want to befog even this!	0.5008
"She accuses conservatives of being the worst offenders in the misogyny department." McKenna must have forget about Liberal M.P. Nicola Diloria's comment about a conservative M.P. being a stripper. And Liberal Darshan Kang faces sexual harassment allegations as Liberal M.P. But no it is those conservatives that are the worst.	0.5010

References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *Proceedings of NAACL-HLT*, pp. 4171–4186, 2019.

[2] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 11 2017. [Online]. Available: http://arxiv.org/abs/1711.05101

[3] P. Contributors, "Distributed data parallel," 2023. [Online]. Available: https://pytorch.org/docs/stable/notes/ddp.html

[4] ——, "Automatic mixed precision package - torch.amp," 2023. [Online]. Available: https://pytorch.org/docs/stable/amp.html

[5] Queisler, Dobler, and Ammar, "PML-BERT: Pre-trained transformers for machine learning," https://github.com/devWhyqueue/pml-bert, 2024, accessed: 2024-11-22.

[6] P. Nabila and E. B. Setiawan, "Adam and adamw optimization algorithm application on bert model for hate speech detection on twitter." Institute of Electrical and Electronics Engineers (IEEE), 9 2024, pp. 346–351.

[7] U. U. Yagci, A. E. Kolcak, and E. Iscan, "Rebert at hsd-2lang 2024: Fine-tuning bert with adamw for hate speech detection in arabic and turkish," pp. 195–198, 2024.

[8] A. G. Putrada, N. Alamsyah, and M. N. Fauzan, "Bert for sentiment analysis on rotten tomatoes reviews," in *2023 International Conference on Data Science and Its Applications, ICoDSA 2023*. Institute of Electrical and Electronics Engineers Inc., 2023, pp. 111–116.

[9] P. Contributors, "Bcewithlogitsloss," 2023. [Online]. Available: https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html

[10] V. Vapnik, "Estimation of dependences based on empirical data," 1982.

[11] N. Alon, S. Ben-david, D. Haussler, and N. Cesa-Bianchi, "Scale-sensitive dimensions, uniform convergence, and learnability," pp. 615–631, 1997.

[12] I. B. Ammar, J. Dobler, and Y. Queisler, "Project machine learning – milestone 1," 2024.

[13] Z. Zhao, Z. Zhang, and F. Hopfgartner, "A comparative study of using pre-trained language models for toxic comment classification," in *The Web Conference 2021 - Companion of the World Wide Web Conference, WWW 2021*. Association for Computing Machinery, Inc, 4 2021, pp. 500–507.

[14] S. L. Smith, P.-J. Kindermans, C. Ying, and Q. V. Le, "Don't decay the learning rate, increase the batch size," 11 2017. [Online]. Available: http://arxiv.org/abs/1711.00489

[15] Y. Zhou and V. Srikumar, "A closer look at how fine-tuning changes bert," 6 2021. [Online]. Available: http://arxiv.org/abs/2106.14282

[16] A. Kurniasih and L. P. Manik, "On the role of text preprocessing in bert embedding-based dnns for classifying informal texts," *International Journal of Advanced Computer Science and Applications*, vol. 13, pp. 927–934, 2022.

[17] S. Kundu, S. N. Sridhar, M. Szankin, and S. Sundaresan, "Sensi-bert: Towards sensitivity driven fine-tuning for parameter-efficient bert," 7 2023. [Online]. Available: http://arxiv.org/abs/2307.11764

[18] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[19] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International conference on machine learning*. PMLR, 2017, pp. 1321–1330.

[20] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 7 2019. [Online]. Available: http://arxiv.org/abs/1907.11692