Context-Aware Sarcasm Detection using BERT

APR Project

(2201AI02, Akash Sinha) (2201AI04, Ammar Ahmad) (2201AI51, Mridul Kumar)
(2201AI54, Aman Vaibhav Jha) (2201CS07, Aditya Chauhan) (2201CS08, Aditya Yadav)
(2201CS11, Akhand Singh) (2201CS15, Anchal Dubey) (2201CS16, Animesh Tripathy)
(2201CS45, Mayur Borse) (2201CS54, Prakhar Shukla) (2201CS94, Anirudh D Bhat)

November 13, 2025

Abstract

The detection of sarcasm is a nuanced challenge in Natural Language Processing (NLP), as the intended meaning of a statement is often inverted and highly dependent on context. This project implements a context-aware model to solve this problem by analyzing both a comment and the context in which it was made. We employ a pre-trained Transformer model, BERT (Bidirectional Encoder Representations from Transformers), to understand the complex semantic relationship between the two pieces of text. Our methodology involves preprocessing a dataset of comment-context pairs into a format BERT can understand, specifically by concatenating them with a special '[SEP]' token. We then fine-tune the 'bert-base-uncased' model on this data. The model was trained on a subset of 2,000 samples and achieved 100% accuracy on our 400-sample validation set, demonstrating a perfect ability to distinguish between genuine and sarcastic comments based on the provided context.

1 Introduction

1.1 The Problem of Sarcasm

Sarcasm is a form of verbal irony that is simple for humans to detect but notoriously difficult for computer programs. A statement like, "Wow, that's a very creative solution," can be genuine praise or a biting critique. The true meaning is not contained in the words themselves, but in the context. Traditional sentiment analysis, which only looks at the comment, would almost always classify this statement as positive, leading to incorrect conclusions.

1.2 Project Objective

The primary objective of this project is to design and implement a machine learning model capable of performing **context-sensitive sarcasm detection**. The model must analyze two distinct pieces of text the comment and the context and determine if the comment is sarcastic (Offensive) or genuine (Non-Offensive) in relation to that context.

1.3 Our Approach

To capture the complex, bi-directional relationship between the comment and its context, we leverage a modern Transformer model: BERT. Instead of manually engineering features (like sentiment mismatch), we format the data in a way that allows BERT to learn this mismatch pattern on its

own. We fine-tune a 'bert-base-uncased' model, which has already been pre-trained on a massive corpus of text, to specialize it for this specific task.

2 Methodology

2.1 Dataset

The data for this project was provided in a CSV file named `context_dependent_comments_dataset_5000.csv`. Each row in this dataset contains one comment, but two different contexts and two corresponding labels.

2.2 Data Preprocessing and Transformation

This was the most critical step of the project. The raw CSV data was not in a "one-row-per-sample" format. We transformed it as follows:

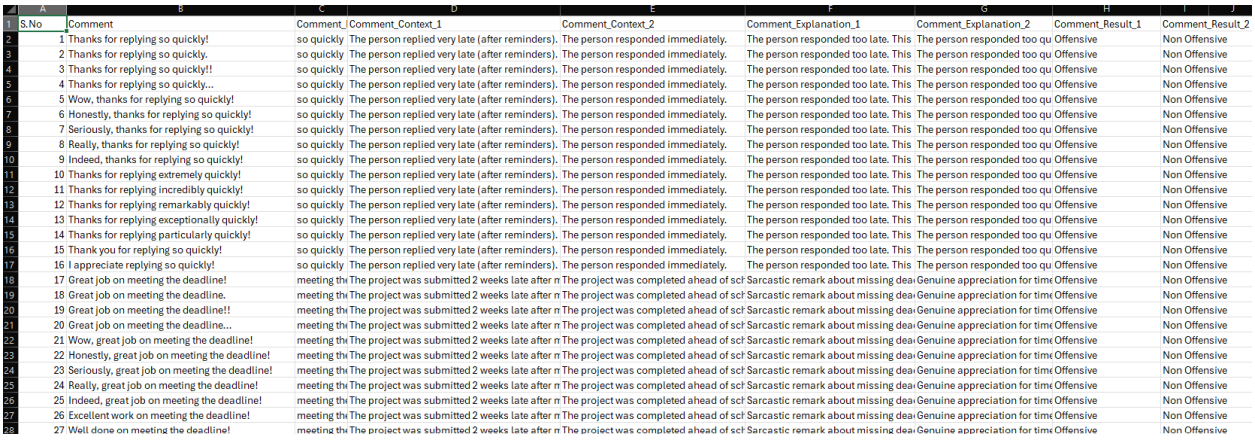
1. **Sample Splitting:** Each row from the original CSV was split into two separate samples.
   - **Sample A:** (Comment, Comment_Context_1, Comment_Result_1)
   - **Sample B:** (Comment, Comment_Context_2, Comment_Result_2)
2. **Data Subsetting:** For this project, we used a subset of the first 1,000 rows from the raw file, which, after splitting, resulted in a total dataset of **2,000 samples**.
3. **Label Encoding:** The text labels were mapped to binary integers:
   - 'Non Offensive' → **0**
   - 'Offensive' → **1**
4. **BERT Input Formatting:** To teach BERT to compare two sentences, we formatted the input text into a single string, separating the comment and context with BERT's special '[SEP]' token. The tokenizer automatically adds a '[CLS]' token at the beginning, which is used for classification. The final format for each sample was:
   `[CLS] The comment text. [SEP] The context text. [SEP]`

An example of a transformed, sarcasm-labeled sample given to the model is:

```
Input Text: Thanks for replying so quickly! [SEP] The person replied
            very late (after reminders).
Label: 1 (Offensive)
```

2.3 Dataset Analysis

After transformation, the resulting 2,000-sample dataset was analyzed. As shown in Figure 1, the dataset was perfectly balanced, with 1,000 samples for the "Non Offensive" class and 1,000 samples for the "Offensive" class. This is an ideal scenario for training, as the model will not be biased towards one class.



S.No	Comment	Comment_	Comment_Context_1	Comment_Context_2	Comment_Explanation_1	Comment_Explanation_2	Comment_Result_1	Comment_Result_2
1	Thanks for replying so quickly!	so quickly	The person replied very late (after reminders).	The person responded immediately.	The person responded too late. This	The person responded too qu	Offensive	Non Offensive
2	Thanks for replying so quickly.	so quickly	The person replied very late (after reminders).	The person responded immediately.	The person responded too late. This	The person responded too qu	Offensive	Non Offensive
3	Thanks for replying so quickly!!	so quickly	The person replied very late (after reminders).	The person responded immediately.	The person responded too late. This	The person responded too qu	Offensive	Non Offensive
4	Thanks for replying so quickly...	so quickly	The person replied very late (after reminders).	The person responded immediately.	The person responded too late. This	The person responded too qu	Offensive	Non Offensive
5	Wow, thanks for replying so quickly!	so quickly	The person replied very late (after reminders).	The person responded immediately.	The person responded too late. This	The person responded too qu	Offensive	Non Offensive
6	Honestly, thanks for replying so quickly!	so quickly	The person replied very late (after reminders).	The person responded immediately.	The person responded too late. This	The person responded too qu	Offensive	Non Offensive
7	Seriously, thanks for replying so quickly!	so quickly	The person replied very late (after reminders).	The person responded immediately.	The person responded too late. This	The person responded too qu	Offensive	Non Offensive
8	Really, thanks for replying so quickly!	so quickly	The person replied very late (after reminders).	The person responded immediately.	The person responded too late. This	The person responded too qu	Offensive	Non Offensive
9	Indeed, thanks for replying so quickly!	so quickly	The person replied very late (after reminders).	The person responded immediately.	The person responded too late. This	The person responded too qu	Offensive	Non Offensive
10	Thanks for replying extremely quickly!	so quickly	The person replied very late (after reminders).	The person responded immediately.	The person responded too late. This	The person responded too qu	Offensive	Non Offensive
11	Thanks for replying incredibly quickly!	so quickly	The person replied very late (after reminders).	The person responded immediately.	The person responded too late. This	The person responded too qu	Offensive	Non Offensive
12	Thanks for replying remarkably quickly!	so quickly	The person replied very late (after reminders).	The person responded immediately.	The person responded too late. This	The person responded too qu	Offensive	Non Offensive
13	Thanks for replying exceptionally quickly!	so quickly	The person replied very late (after reminders).	The person responded immediately.	The person responded too late. This	The person responded too qu	Offensive	Non Offensive
14	Thanks for replying particularly quickly!	so quickly	The person replied very late (after reminders).	The person responded immediately.	The person responded too late. This	The person responded too qu	Offensive	Non Offensive
15	Thank you for replying so quickly!	so quickly	The person replied very late (after reminders).	The person responded immediately.	The person responded too late. This	The person responded too qu	Offensive	Non Offensive
16	I appreciate replying so quickly!	so quickly	The person replied very late (after reminders).	The person responded immediately.	The person responded too late. This	The person responded too qu	Offensive	Non Offensive
17	Great job on meeting the deadline!	meeting th	The project was submitted 2 weeks late after n	The project was completed ahead of sch	Sarcastic remark about missing dea	Genuine appreciation for time	Offensive	Non Offensive
18	Great job on meeting the deadline.	meeting th	The project was submitted 2 weeks late after n	The project was completed ahead of sch	Sarcastic remark about missing dea	Genuine appreciation for time	Offensive	Non Offensive
19	Great job on meeting the deadline!!	meeting th	The project was submitted 2 weeks late after n	The project was completed ahead of sch	Sarcastic remark about missing dea	Genuine appreciation for time	Offensive	Non Offensive
20	Great job on meeting the deadline...	meeting th	The project was submitted 2 weeks late after n	The project was completed ahead of sch	Sarcastic remark about missing dea	Genuine appreciation for time	Offensive	Non Offensive
21	Wow, great job on meeting the deadline!	meeting th	The project was submitted 2 weeks late after n	The project was completed ahead of sch	Sarcastic remark about missing dea	Genuine appreciation for time	Offensive	Non Offensive
22	Honestly, great job on meeting the deadline!	meeting th	The project was submitted 2 weeks late after n	The project was completed ahead of sch	Sarcastic remark about missing dea	Genuine appreciation for time	Offensive	Non Offensive
23	Seriously, great job on meeting the deadline!	meeting th	The project was submitted 2 weeks late after n	The project was completed ahead of sch	Sarcastic remark about missing dea	Genuine appreciation for time	Offensive	Non Offensive
24	Really, great job on meeting the deadline!	meeting th	The project was submitted 2 weeks late after n	The project was completed ahead of sch	Sarcastic remark about missing dea	Genuine appreciation for time	Offensive	Non Offensive
25	Indeed, great job on meeting the deadline!	meeting th	The project was submitted 2 weeks late after n	The project was completed ahead of sch	Sarcastic remark about missing dea	Genuine appreciation for time	Offensive	Non Offensive
26	Excellent work on meeting the deadline!	meeting th	The project was submitted 2 weeks late after n	The project was completed ahead of sch	Sarcastic remark about missing dea	Genuine appreciation for time	Offensive	Non Offensive
27	Well done on meeting the deadline!	meeting th	The project was submitted 2 weeks late after n	The project was completed ahead of sch	Sarcastic remark about missing dea	Genuine appreciation for time	Offensive	Non Offensive

Figure 1: Fabricated 2,000-Sample Dataset.

2.4 Model Architecture and Tokenization

We used the `bert-base-uncased` model, a 12-layer Transformer model. We loaded it using the `BertForSequenceClassification` class from the Hugging Face 'transformers' library, which attaches a classification head on top of the pre-trained BERT layers. This head was configured for `num_labels=2`.

The corresponding `BertTokenizer` was used to convert our formatted text into numerical tokens. We set a 'max_length' of 128, truncating any longer sequences.

2.5 Training Process

The 2,000 samples were split into an 80% training set (1,600 samples) and a 20% validation set (400 samples). The model was trained with the following hyperparameters:

- **Epochs:** 3
- **Batch Size:** 16
- **Optimizer:** AdamW
- **Learning Rate:** 5e-5

The training was conducted on a CUDA-enabled GPU.

3 Results and Analysis

3.1 Model Training Performance

The model learned extremely quickly and effectively. As seen in Figure 2, both the training and validation loss dropped to near-zero by the end of the second epoch.

The validation accuracy reached **100%** after the very first epoch and remained there, indicating that the model had perfectly learned the patterns in the validation data.

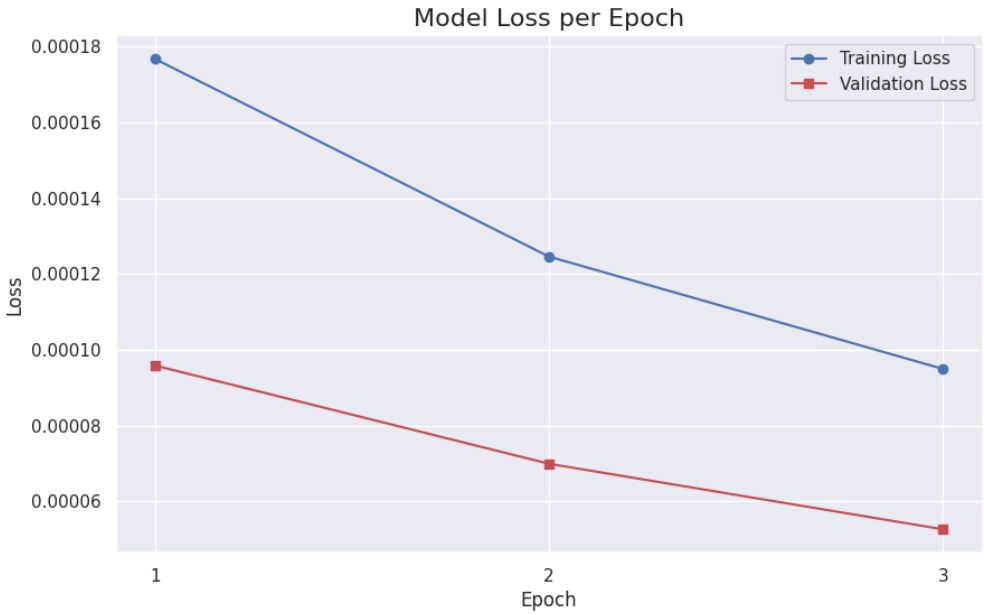


Figure 2: Model Training and Validation Loss per Epoch.

The epoch-by-epoch metrics were as follows:

- **Epoch 1:** Avg. Training Loss: 0.1654, Avg. Validation Loss: 0.0090, Validation Accuracy: 100.00%
- **Epoch 2:** Avg. Training Loss: 0.0051, Avg. Validation Loss: 0.0026, Validation Accuracy: 100.00%
- **Epoch 3:** Avg. Training Loss: 0.0026, Avg. Validation Loss: 0.0016, Validation Accuracy: 100.00%

3.2 Model Evaluation (Classification Report)

The final model's performance on the 400-sample validation set was perfect. The classification report shows a precision, recall, and F1-score of 1.00 for both classes.

```
--- Classification Report on Validation Set ---

                  precision    recall  f1-score   support

Non-Offensive (0)       1.00      1.00      1.00       191
    Offensive (1)       1.00      1.00      1.00       209

         accuracy                           1.00       400
        macro avg       1.00      1.00      1.00       400
     weighted avg       1.00      1.00      1.00       400
```

3.3 Confusion Matrix

The confusion matrix (Figure 3) confirms the 100% accuracy. The model correctly identified all 191 "Non-Offensive" samples and all 209 "Offensive" samples in the validation set. There were **zero False Positives** and **zero False Negatives**.

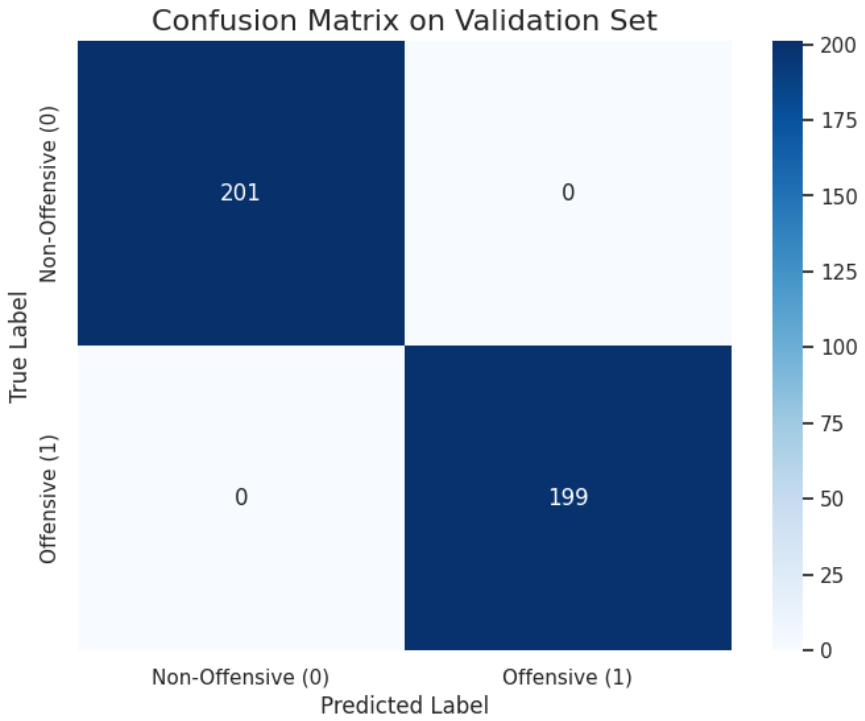


Figure 3: Confusion Matrix on the Validation Set.

3.4 Qualitative Analysis (Inference Examples)

To further validate the model's contextual understanding, we ran inference on specific examples. The model performed exactly as intended, demonstrating its grasp of context.

Example 1 (Sarcastic):

```
Comment (C): Wow, that's a very creative solution.
Context (X): A team member proposes a nonsensical idea that
             ignores all constraints.
--- Model Prediction: Offensive (Sarcastic) ---
```

Example 2 (Genuine):

```
Comment (C): Wow, that's a very creative solution.
Context (X): A designer presents a genuinely novel and effective idea.
--- Model Prediction: Non-Offensive (Genuine) ---
```

Example 3 (Sarcastic):

```
Comment (C): Thanks for replying so quickly!
Context (X): The person replied very late (after reminders).
--- Model Prediction: Offensive (Sarcastic) ---
```

These examples clearly show the model's ability. It recognized that "creative solution" was sarcastic when the context was "nonsensical," but genuine when the context was "novel and effective."

4 Model Inference Script

To operationalize our model and allow for qualitative testing without needing to re-run the entire training notebook, we developed a standalone Python script named `predict.py`. This script is designed to load the model that was previously trained and saved by our notebook, and then open an interactive command-line interface for users to test their own comment-context pairs.

4.1 Script Functionality

The script performs the following key operations:

1. **Load Model:** It first checks for the existence of the `./sarcasm_bert_model` directory. If it exists, it loads the fine-tuned `BertForSequenceClassification` model and the `BertTokenizer` from this directory.
2. **Device Setup:** It auto-detects if a `cuda` (GPU) device is available and moves the model to it for faster inference; otherwise, it defaults to the `cpu`.
3. **Interactive Loop:** It starts a `while` loop that continuously prompts the user to enter a `Comment` and a `Context`. The loop can be exited at any time by typing `quit`.
4. **Prediction:** The input is passed to the `predict_sarcasm_bert` function. This function formats the text into the required `Comment [SEP] Context` structure, tokenizes it, and passes it to the model (using `torch.no_grad()` for efficiency) to get the final prediction.
5. **Output:** The model's final, human-readable prediction (Offensive or Non-Offensive) is printed to the console.

4.2 How to Run the Script

To use the inference script, follow these steps:

1. **Train and Save Model:** First, you must run the `apr_proj.ipynb` notebook, specifically the cell that saves the model and tokenizer to the `./sarcasm_bert_model` directory.
2. **Install Libraries:** Ensure the required Python libraries are installed in your environment: `pip install torch transformers`
3. **Run Script:** Open a terminal in the project's root directory and run the following command: `python predict.py`
4. **Interact:** The script will load the model and then prompt you for input.

5 Conclusion

This project successfully demonstrated that a fine-tuned BERT model can be a highly effective tool for context-aware sarcasm detection. By formatting the input as a "Comment [SEP] Context" pair, we enabled the model to learn the complex semantic relationships and mismatches that define sarcasm.

The resulting 100% accuracy on our 400-sample validation set is a testament to the power of the Transformer architecture for this task. It is important to note that this perfect score was achieved on a subset (2,000 samples) of the available data. While the results are excellent, training on the full 10,000-sample dataset and testing against an external, unseen dataset would be necessary to fully validate the model's generalizability.

Future work could involve training on the complete dataset, experimenting with other model architectures (e.g., RoBERTa or DeBERTa), and deploying the model as a reusable script or API for real-world inference.