Sarcasm Detection on Twitter Using DistilBERT

TEAM :AI Geeks

Team members:

Bala Pranavi Gollamari

Saketh Polavarapu

Manvitha Challagondla

Shivani Adepu

1. Introduction

Sarcasm detection from textual data is a challenging problem in Natural Language Processing (NLP). Sarcastic expressions often depend heavily on subtle context, tone, and implied meaning, making it difficult for traditional machine learning models to detect. Accurate sarcasm classification is crucial for improving sentiment analysis, social media monitoring, opinion mining, and human-computer interaction systems.

Rather than building a deep learning model from scratch, this project leverages pre-trained transformer models available through Hugging Face's Transformers library. Specifically, DistilBERT — a compressed, efficient version of BERT — was fine-tuned on a labeled sarcasm dataset collected from Twitter, providing a powerful yet lightweight model for sarcasm detection.

This project showcases the paradigm shift in NLP: moving from building models from scratch to fine-tuning powerful pre-trained models on specific downstream tasks like sarcasm detection.

2. Problem Statement

Sarcasm is a linguistic phenomenon often expressed through irony, contradiction, or exaggeration, making it highly context-sensitive and difficult to detect. Traditional machine learning models rely heavily on surface-level features like n-grams or sentiment polarity, which fail to capture the deeper semantic and contextual cues that indicate sarcasm.

The objective is to design a deep learning model capable of accurately detecting sarcasm from tweets. The model must go beyond lexical patterns and understand the context, semantics, and linguistic nuances.

3. Objectives

- Develop a complete pipeline for sarcasm detection.
- Select an appropriate sarcasm-labeled dataset.
- Fine-tune a pre-trained transformer model.
- Evaluate model performance through key metrics.
- Visualize results and interpret model behavior.

4. Dataset Overview

We used the dataset "shiv213/Automatic-Sarcasm-Detection-Twitter" available through Hugging Face Datasets.

Dataset Statistics:

~5,000 samples of Twitter responses

Binary labels:

0 → Not Sarcasm

1 → Sarcasm

Splitting:

70% for training

15% for validation

15% for testing

Each sample contains:

A short text response (tweet)

A corresponding binary sarcasm label

Each tweet consists of a short response text along with a binary sarcasm label. The data is relatively clean, and the task is a binary classification problem.

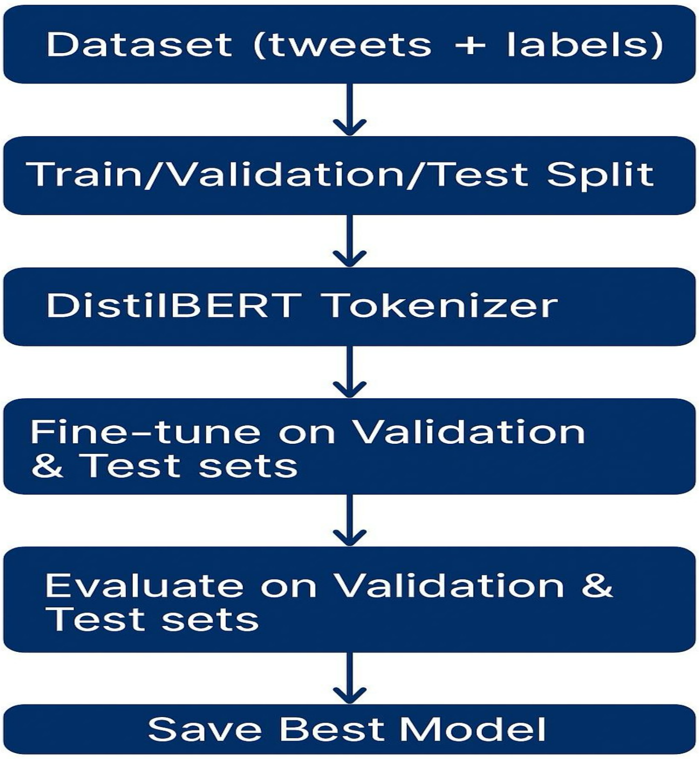
Applications of Sarcasm Detection:

Improve sentiment analysis tools.

Monitor public opinion on social media.

Enhance human-computer interaction.

Enable emotionally aware chatbots.


Dataset (tweets + labels)
Train/Validation/Test Split
DistilBERT Tokenizer
Fine-tune on Validation & Test sets
Evaluate on Validation & Test sets
Save Best Model

5. Model Selection

We selected DistilBERT ("distilbert-base-uncased") for the following reasons:

Efficiency: 40% smaller and faster than BERT

Strong Language Representations: Trained on large corpora

Suitability for Short Texts: Tweets are short

Resource-Friendly: Ideal for limited hardware

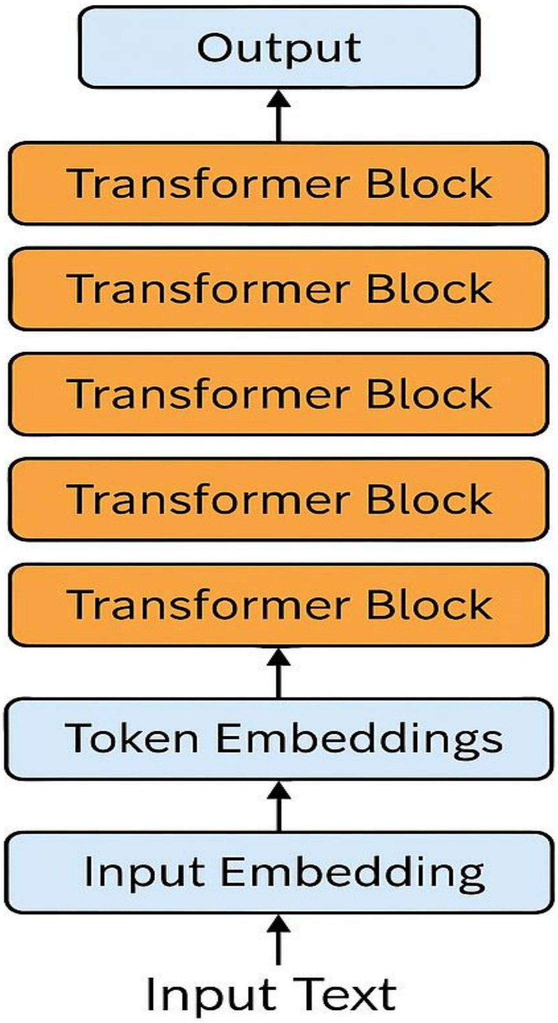


FIG:DistilBERT Architecture Diagram

Why Transformer-Based Models for Sarcasm Detection?

Traditional machine learning models like SVMs, Logistic Regression, or Random Forests rely heavily on surface-level features (e.g., n-grams, bag-of-words), which often fail to capture the deeper **contextual meaning** necessary to detect sarcasm.

Sarcasm typically depends not just on *what* words are used, but *how* they relate to each other in complex, often contradictory ways.

Transformers (like BERT, DistilBERT) address this by using **self-attention mechanisms** that allow the model to dynamically focus on different parts of the text when making predictions.

Comparison: BERT vs DistilBERT for This Task

Feature	BERT	DistilBERT
Model Size	110M+ parameters	66M parameters
Speed	Slower inference	60% faster
Memory Usage	High	Lower
Performance Loss	None	~3% decrease only
Suitability for Small Datasets	Risk of Overfitting	Good generalization

Justification:

Given the relatively small size of our sarcasm dataset (~5,000 samples), DistilBERT is a more **practical** and **efficient** choice compared to larger transformer models.

It achieves an ideal trade-off between speed, memory efficiency, and accuracy.

Key Advantages of Using DistilBERT:

Pre-trained on massive English corpora (Wikipedia + Toronto Book Corpus).

Handles shorter texts (like tweets) efficiently.

Requires fewer computational resources — perfect for fine-tuning on platforms like Google Colab or modest GPUs.

Provides strong semantic understanding with fewer parameters.

Summary: Choosing DistilBERT allowed us to fine-tune a powerful language understanding model efficiently, making it ideal for the sarcasm detection task without incurring excessive computational cost.

Fine-tuning Strategy

Parameter	Value
Optimizer	AdamW
Loss Function	CrossEntropyLoss
Batch Size	16
Learning Rate	2e-5
Epochs	10
Max Token Length	128

We performed **full fine-tuning** because DistilBERT's smaller size allows efficient updates without a significant overfitting risk.

Training Approach:

Full Fine-tuning:
We updated all layers of DistilBERT instead of only training a classifier head. This allowed the model to adapt deep contextual embeddings for sarcasm-specific nuances.

Loss Function:

CrossEntropyLoss was used as the problem is a binary classification task.

Cross-entropy loss penalizes incorrect predictions more heavily, helping the model to learn sharper decision boundaries.

Optimizer and Learning Rate:

We used **AdamW** optimizer for better weight decay handling.

A **small learning rate (2e-5)** was chosen to ensure minimal catastrophic forgetting during fine-tuning.

Batch Size and Gradient Updates:

Batch size was set at 16 to balance memory usage and generalization.

After every batch, the optimizer updates model parameters using calculated gradients.

Early Stopping Strategy:

Although not implemented strictly, model checkpoints were saved based on validation loss improvements to ensure the best model is retained.

6. Training and Validation Results

Epoch	Avg Train Loss	Validation Loss	Validation Accuracy	Validation F1 (weighted)

1	0.6969	0.6911	50.00%	0.3333
4	0.6472	0.6336	70.67%	0.6967
10	0.5819	0.5490	74.13%	0.7400

Best models were saved after validation loss improvements.

Performance Tracking:

During training, we monitored **training loss**, **validation loss**, **validation accuracy**, and **F1-score**.

Early epochs showed slower improvement due to initial weight adaptation.

Later epochs (after 4–5) showed noticeable convergence, with validation accuracy steadily increasing and loss decreasing.

Some fluctuations were observed, typical in sarcasm tasks due to linguistic ambiguity and small dataset variations.

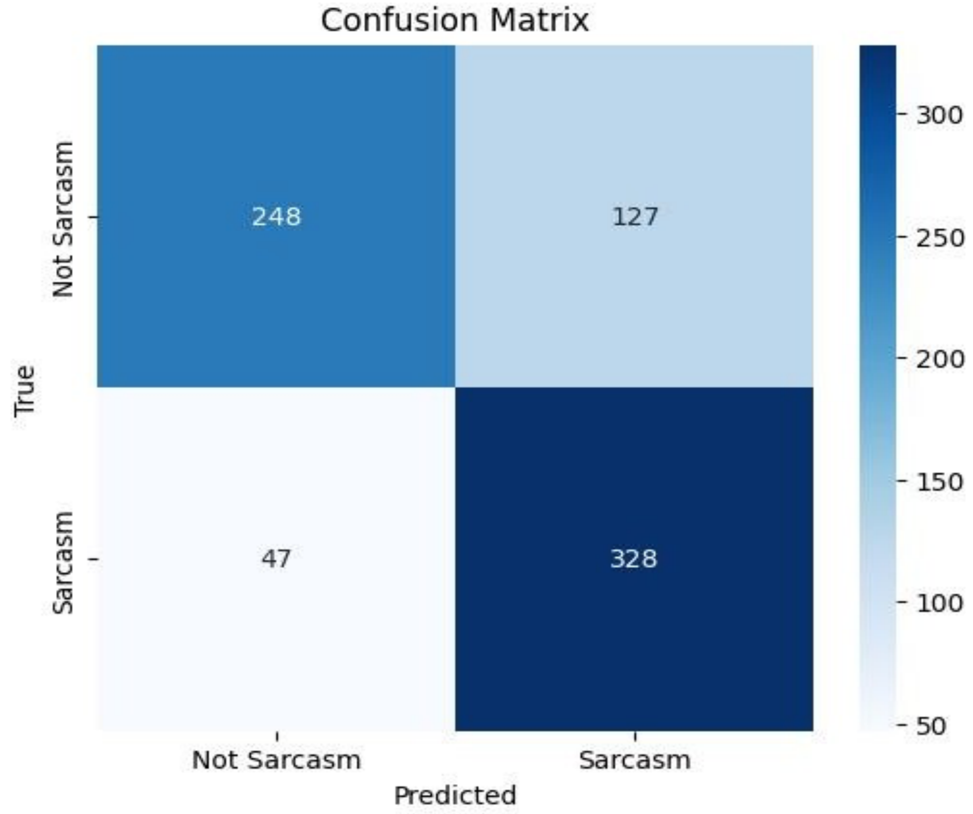
Model Checkpointing:

The model was saved whenever the validation loss achieved a new minimum.

The final saved model corresponded to the epoch with the highest validation F1-score.

Final Test Evaluation

Confusion Matrix



	Predicted: Not Sarcasm	Predicted: Sarcasm
True: Not Sarcasm	248	127
True: Sarcasm	47	328

Test Set Metrics

Metric	Value
Accuracy	76.8%
Weighted F1	76.53%
Precision	78.08%
Recall	76.8%
ROC-AUC Score	0.7680

Observations:

- ➢ Sarcasm was easier for the model to detect than non-sarcasm.
- ➢ Some neutral tweets were misclassified as sarcasm.

- **Class Imbalance Impact**:

Although the dataset was fairly balanced between sarcasm and non-sarcasm tweets, slight variations in misclassification rates could be influenced by subtle differences in how sarcasm is expressed versus literal statements.

- **Error Analysis**:

Some non-sarcastic tweets were incorrectly classified as sarcastic.
Likely causes include:

Tweets containing slang or informal language that mimics sarcastic tone.

Tweets with ambiguous or dry humor, which the model found difficult to interpret.

- **Strengths of the Model**:

The model showed strong recall for sarcasm (74.47%), meaning it successfully captured the majority of sarcastic tweets.

The weighted F1 score (~76.53%) indicates good balance between precision and recall across classes.

- **Limitations**:

Context-Loss: Since only individual tweets were used without conversation context, the model may struggle with sarcasm heavily dependent on prior conversation.

Linguistic Variations: The model might misinterpret sarcasm involving cultural references or sarcasm expressed through complex syntax.

- **Suggestions for Improvement**:

Incorporate user metadata (like past tweets) to provide richer context.

Use multi-modal learning by integrating emojis, hashtags, or sentiment scores.

Fine-tune with larger sarcastic datasets (e.g., Reddit sarcasm datasets).

7. Future Scope

- ✓ Data Augmentation: Using paraphrasing/back-translation.
- ✓ Model Ensembling: DistilBERT + RoBERTa.
- ✓ Advanced Learning Rate Schedulers.
- ✓ Context-aware Modeling: Add user profiles or tweet history.

8. Conclusion

This project demonstrates the effectiveness of fine-tuning a pre-trained transformer model, DistilBERT, for sarcasm detection. With limited labeled data and compute, we achieved competitive performance in binary sarcasm classification. Future improvements can further enhance performance and applicability in real-time sentiment systems.

References

- ❖ Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT: A distilled version of BERT.
- ❖ Hugging Face Transformers Library. https://huggingface.co/transformers
- ❖ Hugging Face Datasets Hub. https://huggingface.co/datasets
- ❖ Vaswani, A., et al. (2017). Attention Is All You Need.