Detecting Sarcasm, Irony and Figurative Language in Tweets Using NLP Techniques

Team Members:

Darsini Lakshmiah – G31629191
Hussain Nathani – G36308827
Rasika Nilatkar – G28505126

Abstract

Sarcasm, irony, and figurative expressions are difficult to detect automatically because their meaning often contradicts the literal surface form of the text. Tweets intensify this challenge due to short length, lack of context, and informal writing. In this project, we build and evaluate three NLP models to classify tweets into four categories: Regular, Sarcasm, Irony, and Figurative.

We begin with a thorough preprocessing pipeline including text normalization, lemmatization, and engineered linguistic features. Exploratory data analysis reveals stylistic and lexical differences across classes, which informs our modeling choices. We experiment with three progressively more expressive approaches:

- **Baseline Model:** Logistic Regression with TF-IDF vectorization
- **Deep Learning Model:** BiLSTM
- **Transformer Model:** Fine-tuning DistilBERT for deeper semantic understanding of sarcasm and irony.

All three models achieve similar performance on the test set, with **accuracies** around **0.74 - 0.748** and **macro-F1 scores** around **0.65**. Although the models perform extremely well on *regular*, *irony*, and *sarcasm* categories, all models struggle to identify the figurative class, consistently predicting it incorrectly. We analyze this failure and discuss dataset imbalance, semantic overlap, and lack of contextual cues as major contributors.

Overall, the project highlights the inherent difficulty of figurative language detection and provides a comparative study of classical, recurrent, and transformer-based NLP techniques on this task.

Introduction & Motivation

Understanding figurative language, especially sarcasm and irony remains one of the most challenging problems in natural language processing. Humans rely on shared context, tone, facial cues, and cultural conventions to interpret non-literal expressions. When these signals are missing, as in text-only communication on platforms like Twitter, even humans can misunderstand intent. For machines, the challenge is even greater.

Sarcasm often expresses the opposite of what is literally said ("Great, another Monday."), while irony highlights unexpected or contradictory situations. Figurative tweets, which blend sarcasm and irony, are even more ambiguous. These linguistic forms depend heavily on subtle cues such as exaggeration, punctuation, intensifiers, and emotional polarity shifts. Detecting them computationally requires more than simple keyword spotting; it demands semantic understanding and sensitivity to nuance.

This project aims to explore whether modern NLP techniques can reliably distinguish between four tweet categories:
Regular, **Sarcasm**, **Irony**, and **Figurative**.

Our motivation is twofold:

- **Practical motivation:** Improving sentiment analysis, detecting harmful or misleading tone, and supporting safer moderation tools on social media platforms.

- **Scientific motivation:** Understanding the linguistic properties of sarcasm and irony and assessing how well modern NLP models capture these subtle phenomena.

Existing literature has shown that word-level features, sequential patterns, and contextual embeddings each contribute differently to sarcasm detection. However, the overlap among figurative language categories makes multi-class prediction especially challenging.

Through this project, we aim to not only benchmark different techniques but also analyze their strengths and weaknesses particularly their collective inability to correctly classify figurative tweets, despite strong performance on the other three categories. The findings provide insights into both dataset limitations and modeling challenges that future work can address.

Dataset Description

Our project uses the **"Tweets with Sarcasm and Irony"** dataset sourced from Kaggle, which is itself based on the research paper *"An Empirical, Quantitative Analysis of the Differences between Sarcasm and Irony"* by Ling & Klinger (2016). The dataset provides short English tweets annotated into four mutually exclusive categories:

- **Regular**
- **Sarcasm**
- **Irony**
- **Figurative** (tweets combining sarcasm + irony)

These categories make the dataset well-suited for a supervised, multi-class text classification problem. The multi-label nuances particularly the overlap between sarcasm and irony in the figurative class introduce an additional layer of difficulty.

1.1 Dataset Size and Distribution

Our cleaned training and test splits (after preprocessing) contain:

- **Train:** 81,403 tweets
- **Test:** 8,119 tweets

The label distribution in the training data is as follows:

```
Train shape: (81403, 12)
Test shape : (8119, 12)
Train label distribution:
label
figurative    21237
irony         20891
sarcasm       20680
regular       18595
Name: count, dtype: int64
```

1.2 Nature of the Data

Twitter text introduces additional challenges:

- Informal grammar
- Emojis, hashtags, user mentions
- Non-standard capitalization
- Contextless statements
- Heavy reliance on world knowledge

These properties make sarcasm and figurative language classification significantly more difficult than generic text classification.

1.3 Why This Dataset?

We selected this dataset for several reasons:

- It directly supports the goals outlined in our proposal regarding sarcasm, irony, and figurative behavior.
- It offers a realistic, noisy social media environment where models face genuine ambiguity.
- The four-class structure provides a nuanced setting to test whether classical or modern NLP methods better capture figurative meaning.

- Prior work in sarcasm detection often focuses on binary classification; this dataset allows a more fine-grained, pedagogically valuable exploration.

1.4 Data Splits

We preserve the original train/test split provided in training files distributed for the project. Validation sets are created within the training split during modeling:

- **TF-IDF + Logistic Regression:** internal stratified 85/15 split
- **BiLSTM + Attention:** 85% train / 15% val via **`train_test_split`**
- **DistilBERT:** validation created during training loop

Preprocessing & Feature Engineering

Preprocessing was one of the most important stages of this project because raw tweets contain noise that interferes with learning: URLs, mentions, emojis, inconsistent casing, and informal grammar. Our preprocessing pipeline follows a multi-step structure implemented in the script **Preprocessing.py**

2.1 Text Cleaning

We apply a custom cleaning function designed specifically for Twitter text:

- Remove URLs
- Remove user mentions
- Remove emojis and pictographs
- Convert hashtags into words rather than deleting them
- Keep punctuation like `!` and `?`, because they correlate with sarcasm
- Lowercase Everything
- Remove unusual symbols and normalize whitespace

This ensures that we preserve the stylistic cues necessary for figurative language detection.

2.2 Lemmatization

To reduce variation across word forms, we use **WordNet lemmatization**.
If spaCy is available, we fall back to spaCy's lemmatizer; otherwise, NLTK POS-aware lemmatization is used.

This step helps standardize words like:

- "running", "ran", "runs" → **run**
- "happiest", "happier" → **happy**

Since sarcasm and irony rely more on *patterns* than surface morphology, lemmatization significantly reduces noise without losing meaning.

2.3 Feature Engineering

We generate several handcrafted linguistic features inspired by sarcasm literature:

- **char_len:** length of tweet
- **word_len:** number of words
- **num_exclam:** count of “!”
- **num_question:** count of “?”
- **num_hashtags:** number of hashtags
- **num_mentions:** number of mentions
- **num_caps:** count of uppercase letters
- **cap_ratio:** percentage of characters that are uppercase

The following features were particularly important in our early EDA, as figurative language tends to use exaggeration, punctuation, and emotional turn-about markers.

Exploratory Data Analysis (EDA)

The dataset is nearly balanced across the four categories. This is reassuring because imbalance would otherwise skew the classifier toward majority classes. Despite balanced counts, we later find that "figurative" is systematically misclassified suggesting semantic rather than statistical difficulty.

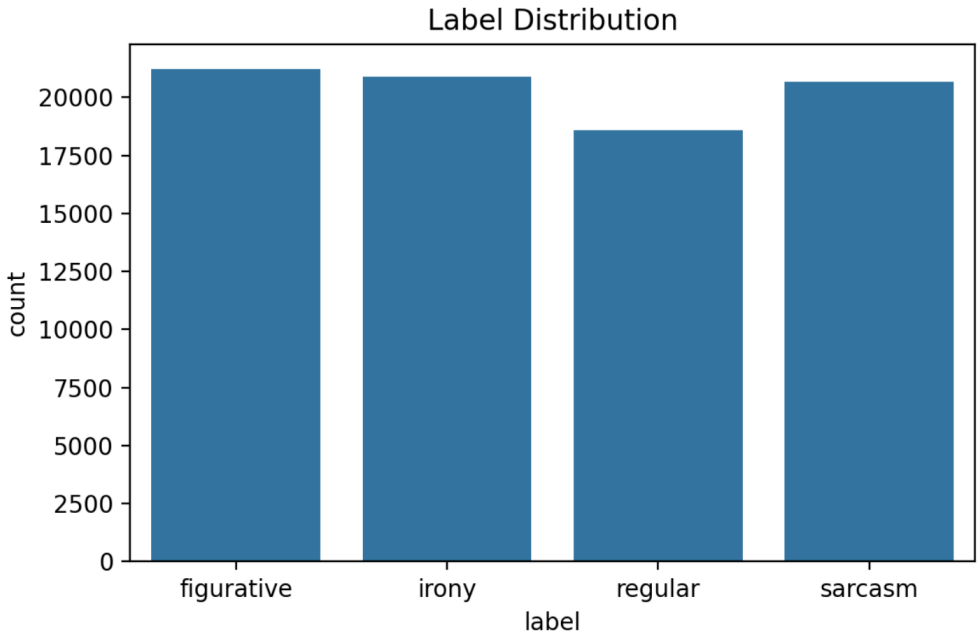

Fig 1: *Label distribution across the four tweet categories in the dataset.*

The word-length distribution follows a similar pattern. Most tweets contain **4–20 words**, which aligns with typical tweet behavior. Because tweets are short, models like BiLSTM must be carefully tuned to avoid overfitting on padded sequences.

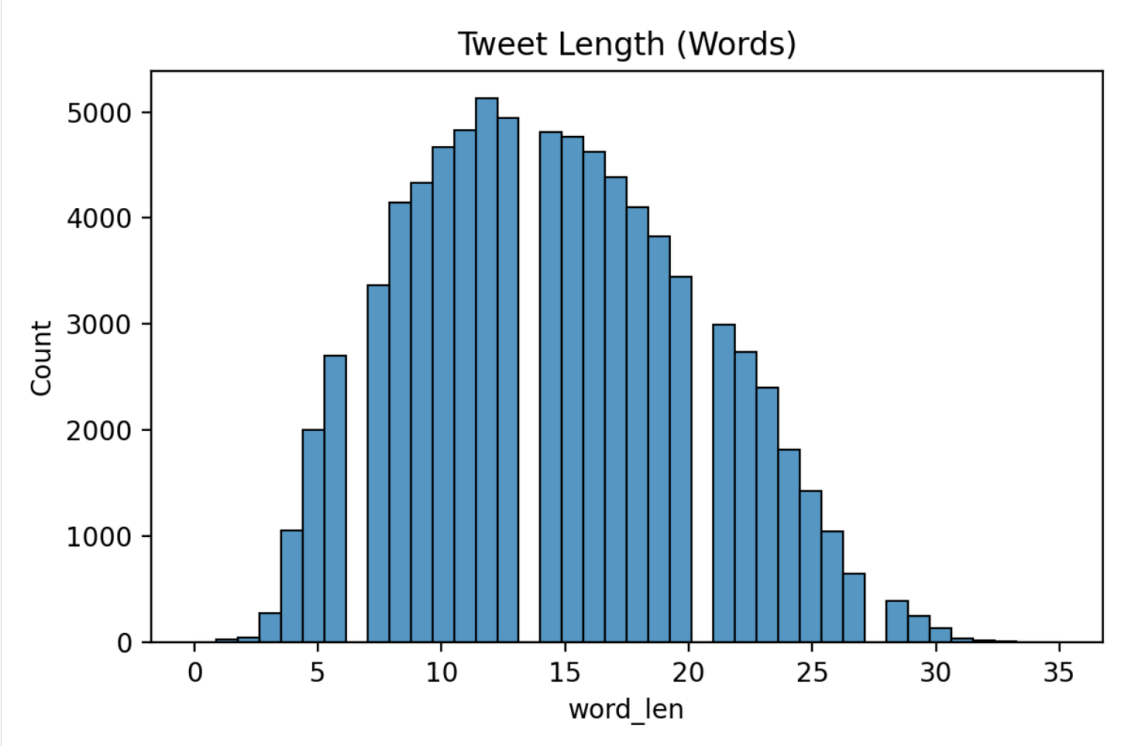

Fig 2: *Tweet length distribution histogram for the dataset.*

The word clouds highlight clear stylistic patterns across the four classes:

- **Sarcasm** shows frequent use of emotional intensifiers such as *great*, *amazing*, and *love*. These overly positive words are typically used to express the opposite sentiment, making exaggerated positivity a reliable cue for sarcastic intent.

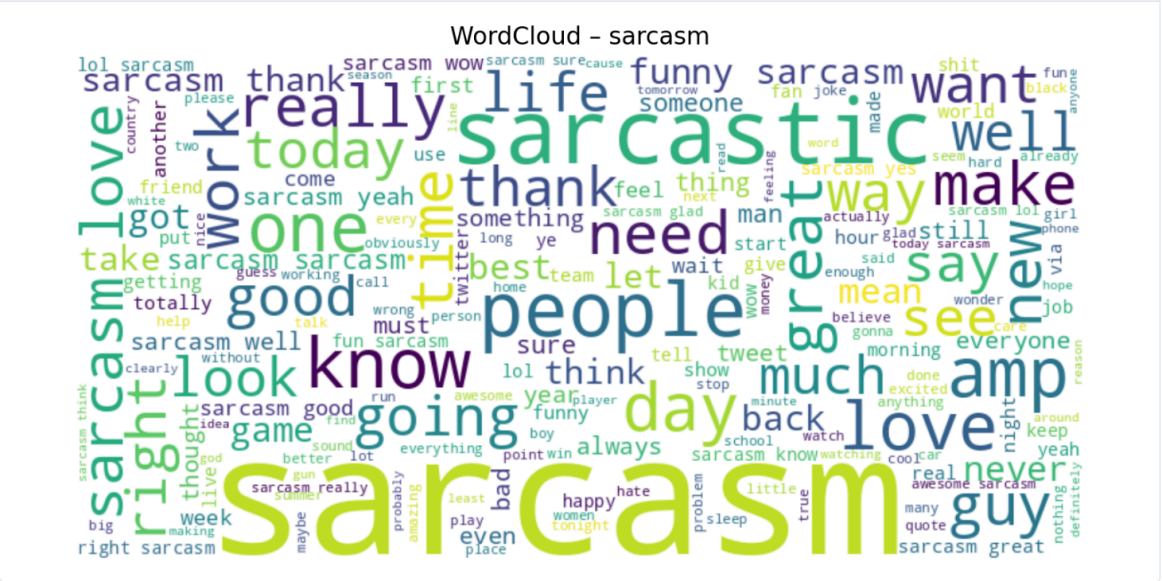

Fig 3: *Word cloud illustrating common lexical patterns in sarcastic tweets.*

- **Irony** emphasizes situational contrast rather than emotion. Words like *finally*, *nothing*, and *work* often appear in contexts where reality contradicts expectation. This aligns with irony's dependence on subtle reversals rather than explicit exaggeration.

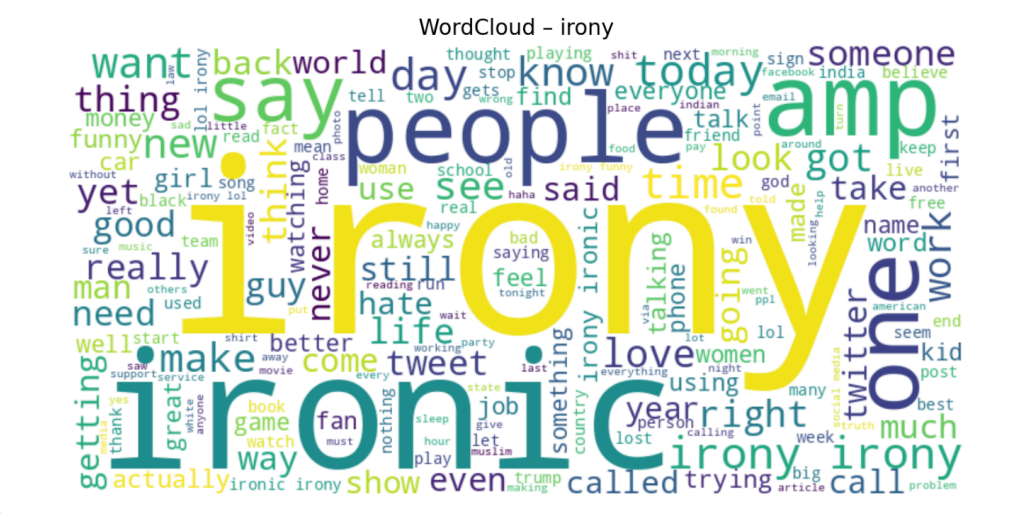

Fig 4: *Word cloud highlighting frequent terms found in ironic tweets.*

- **Figurative** tweets combine elements of both sarcasm and irony, but do not introduce any unique lexical markers. Their vocabulary overlaps heavily with neighboring classes, producing a blended word cloud rather than a distinct signature. This lack of clear lexical boundaries makes figurative tweets the hardest to classify.

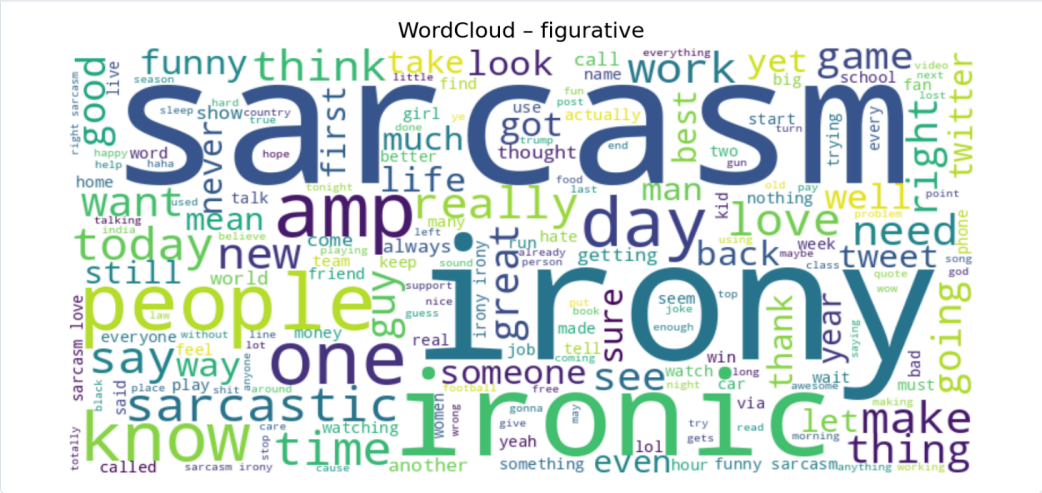

Fig 5: *Word cloud showing overlapping lexical cues in figurative tweets.*

- **Regular** tweets contain straightforward, literal vocabulary tied to daily activities, news, or personal statements, forming the cleanest and most separable class.

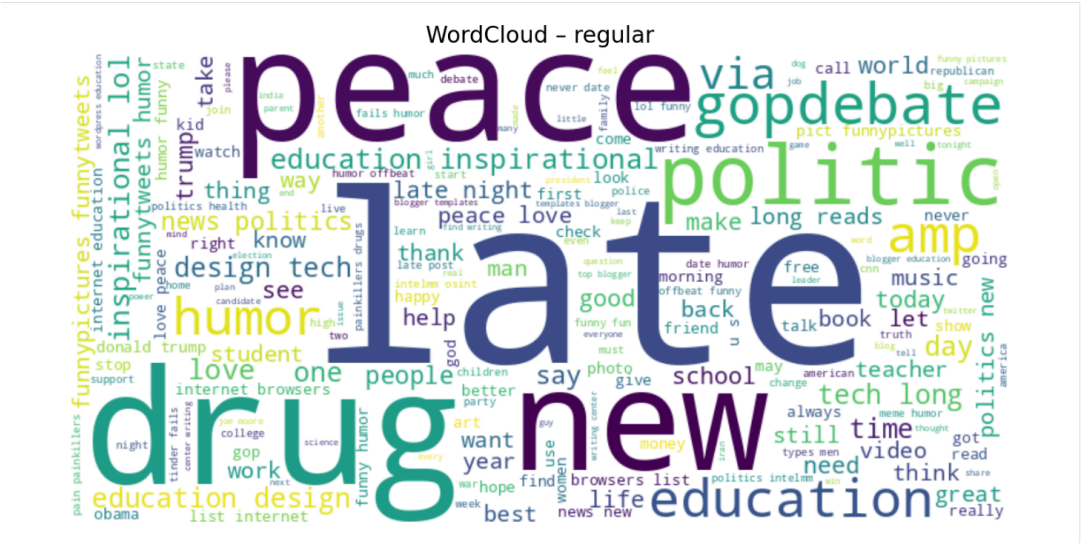

Fig 6: *Word cloud illustrating straightforward vocabulary used in regular tweets.*

Overall, the word clouds reveal why all three models Logistic Regression, BiLSTM, and DistilBERT struggle most with the **figurative** category: it does not form its own lexical identity and instead borrows cues from both sarcasm and irony, leading to systematic confusion despite balanced class sizes.

NLP Models Used and Algorithms

Model 1: TF-IDF + Logistic Regression

To establish a transparent and interpretable baseline for figurative language detection, we implemented a classical machine learning pipeline using TF-IDF vectorization followed by multinomial Logistic Regression. Unlike neural models that learn semantic representations, this approach relies entirely on surface-level lexical features and treats text as a weighted bag of words. Although simple, TF-IDF remains one of the strongest baselines in text classification because it captures how important a word is to a tweet relative to the entire corpus.

Before modeling, all tweets were normalized through lowercasing, punctuation and URL removal, and lemmatization. After preprocessing, each tweet was converted into a sparse high-dimensional TF-IDF vector. The transformation uses the formula:

$$\text{TF–IDF}(w, d) = \text{tf}(w, d) \times \log\left(\frac{N}{\text{df}(w)}\right)$$

where **tf(w, d)** is the local frequency of a word, **df(w)** is the number of tweets containing that word, and **N** is the total number of tweets. This weighting downscales extremely common words (e.g., “the”, “today”, “time”) while amplifying terms that are more distinctive of sarcastic or ironic phrasing, such as emotional intensifiers or contrastive expressions.

These TF-IDF features are then fed into a **multinomial Logistic Regression classifier**, which learns a linear decision boundary for each of the four categories. Logistic Regression models class probabilities using the softmax function:

$$P(y = k \mid x) = \frac{e^{w_k^\top x + b_k}}{\sum_{j=1}^{C} e^{w_j^\top x + b_j}}$$

This formulation makes Logistic Regression fast to train, highly interpretable, and effective at identifying categories driven by strong or distinctive word patterns. Since TF–IDF does not encode word order or deep semantic relationships, the model is limited to recognizing patterns that can be expressed through vocabulary usage alone. However, it serves as an informative lower bound for evaluating whether more advanced architectures such as BiLSTMs or transformers provide substantial improvement.

Model Architecture

- **Input:** cleaned tweet text
- **Vectorization:** TF-IDF transformation (30k–50k sparse features)
- **Classifier:** Multinomial Logistic Regression with softmax
- **Output:** four-class probability distribution
- **Training:** cross-entropy loss with L2 regularization

This model contains no hidden layers, no embeddings, and no sequence modeling components; all predictive power comes directly from lexical statistics.

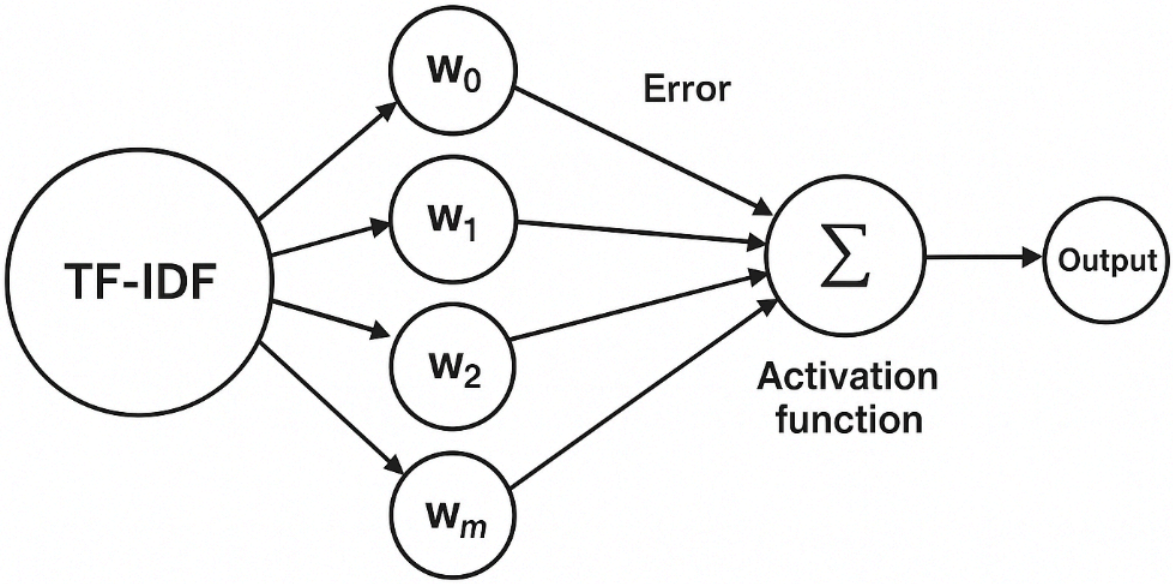

Fig 7: *TF-IDF and Logistic Regression model diagram illustrating feature weighting, linear combination, and output activation.*

Classification Report

The TF–IDF + Logistic Regression model achieves a **test accuracy of 74.4%** and a **macro F1-score of 0.65**, indicating that even a linear classifier captures much of the accessible signal in the dataset. Performance, however, varies dramatically across categories. The model performs exceptionally well on the **regular** class (F1 = 0.99), reflecting how easily literal and

topic-driven tweets can be identified from their vocabulary alone. Similarly, **irony** and **sarcasm** both achieve strong F1-scores (0.80), with recall near 1.00, showing that the model is highly sensitive to lexical patterns like exaggeration, polarity flips, or contrastive phrasing.

By contrast, the model completely fails to detect the **figurative** class (F1 = 0.00). Despite having over 2,000 samples, figurative tweets lack distinctive word-level markers and share vocabulary with both irony and sarcasm; as a result, the classifier collapses them into those classes. This supports the EDA findings: figurative language requires deeper semantic interpretation than TF–IDF can provide. The model's ability to recognize literal, ironic, or sarcastic cues—but not abstract figurative expressions demonstrates the limitations of purely lexical features.

Overall, the TF–IDF + Logistic Regression baseline provides a strong starting point and highlights clear boundaries of what traditional bag-of-words techniques can and cannot capture. Its strong performance on three classes and total failure on figurative language reinforce the need for more context-aware architectures such as BiLSTMs or transformers for nuanced figurative language understanding.

```
=== Test metrics ===
Accuracy: 0.7446729892843946
Macro F1: 0.6500365360103

Classification report:
              precision    recall  f1-score   support

           0       0.19      0.00      0.00      2044
           1       0.67      0.99      0.80      2111
           2       0.99      1.00      0.99      1859
           3       0.67      0.99      0.80      2105

    accuracy                           0.74      8119
   macro avg       0.63      0.75      0.65      8119
weighted avg       0.62      0.74      0.64      8119
```

Model 2: Bidirectional Long Short-Term Memory (BiLSTM)

To classify tweets into figurative, irony, sarcasm, and regular language, we developed a deep learning model based on a **Bidirectional Long Short-Term Memory (BiLSTM) network**. Figurative language detection requires understanding subtle contextual cues, contrasts, and dependencies across an entire sentence. Tweets, in particular, often embed sarcasm or irony

through unexpected twists, sentiment reversals, or emphasis that appears either early or late in the text. Traditional machine learning models treat text as a bag of words, losing sequential information, while unidirectional RNNs can only learn patterns in one temporal direction. In contrast, a BiLSTM processes the text in both forward and backward directions, enabling the model to capture relationships between words regardless of where they appear in the tweet.

Before training, the raw tweets were preprocessed through lowercasing, punctuation removal, URL stripping, and lemmatization. A tokenizer was then fitted on the training corpus to convert each word into an integer index, and all sequences were padded to a fixed length. The first layer of the model is a trainable embedding layer, which transforms each word index into a dense **128-dimensional vector**. Unlike pre-trained embeddings (e.g., GloVe), these embeddings are learned from scratch, allowing the model to adapt to informal expressions, slang, abbreviations, and stylistic features common in Twitter language.

The embedded sequence is passed into a Bidirectional LSTM layer with 128 units. LSTMs improve upon standard RNNs through gating mechanisms input, forget, and output gates designed to control how information flows through the network. Given an input sequence x1,x2,…,xT the LSTM computes hidden states using:

$$h_t = \mathrm{LSTM}(x_t, h_{t-1}, c_{t-1})$$

A BiLSTM enhances this by computing a forward hidden state and a backward hidden state, then concatenating them:

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}]$$

This enables the model to understand how later words affect earlier ones, an essential property for sarcasm detection, where the “punchline” often appears at the end of a sentence.

To convert the sequence of hidden states into a fixed-length representation, the model applies Global Max Pooling, which extracts the highest-activated features across time steps. This allows the model to focus on the most salient words or patterns in the tweet. The pooled vector is then passed through a dense layer with ReLU activation, followed by dropout to reduce overfitting. Finally, a softmax layer outputs the probability distribution over the four figurative-language categories.

The model is trained using categorical cross-entropy loss and optimized with Adam. Early stopping and model checkpointing prevent over-training. Overall, this architecture combines semantic understanding (embeddings), directional context (BiLSTM), and salient feature

extraction (max pooling) to deliver a robust classifier suited for figurative language detection in short, noisy social media text.

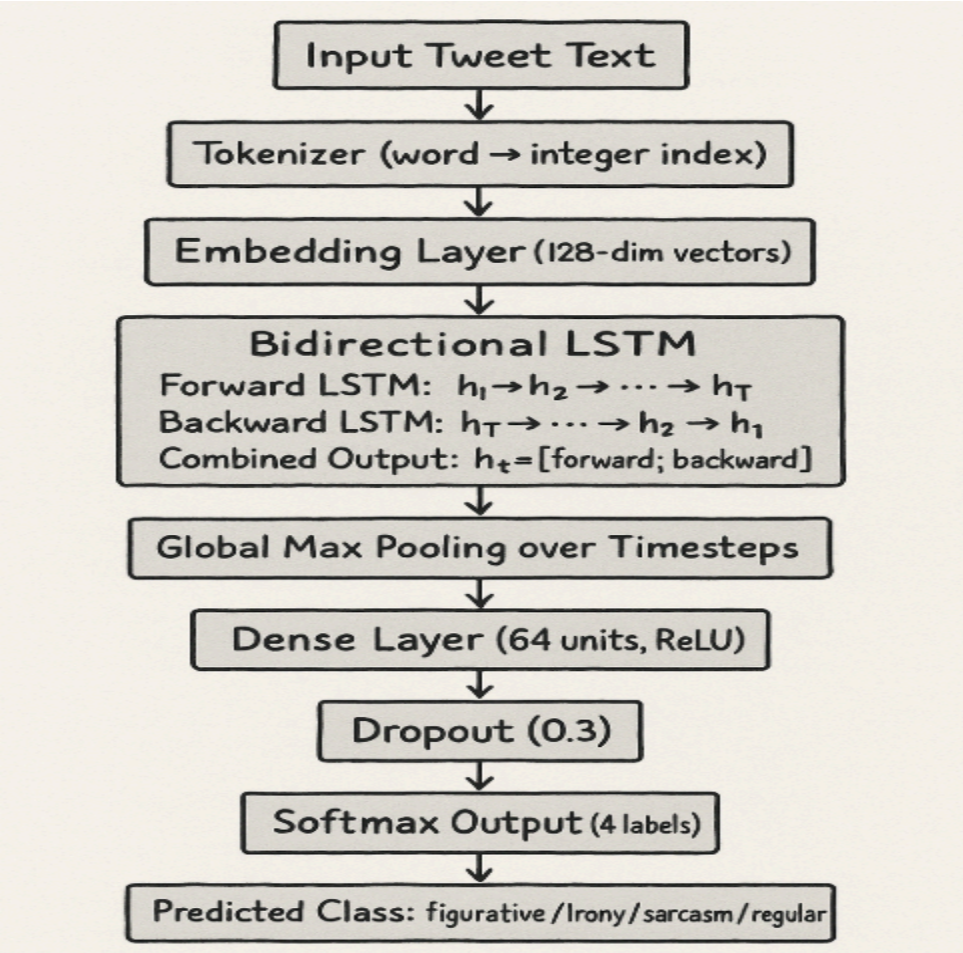

Fig 8: *BiLSTM model architecture diagram including embedding, BiLSTM, pooling, and classification layers.*

Model Architecture

- **Input:** cleaned and preprocessed tweet text
- **Tokenization:** integer index mapping using a fitted tokenizer
- **Embedding Layer:** trainable 128-dimensional word embeddings learned from scratch
- **Sequence Encoder:** Bidirectional LSTM (128 units) capturing forward and backward context

- **Feature Extraction:** Global Max Pooling over all hidden states to select the most salient features
- **Dense Layer:** fully connected layer with ReLU activation for non-linear feature transformation
- **Regularization:** dropout to prevent overfitting
- **Output Layer:** softmax classifier producing a four-class probability distribution
- **Training:** optimized with Adam using categorical cross-entropy loss and early stopping

This model introduces **learned embeddings** and **bidirectional sequence modeling**, giving it the ability to capture contextual relationships across the whole tweet capabilities that traditional lexical baselines do not provide.

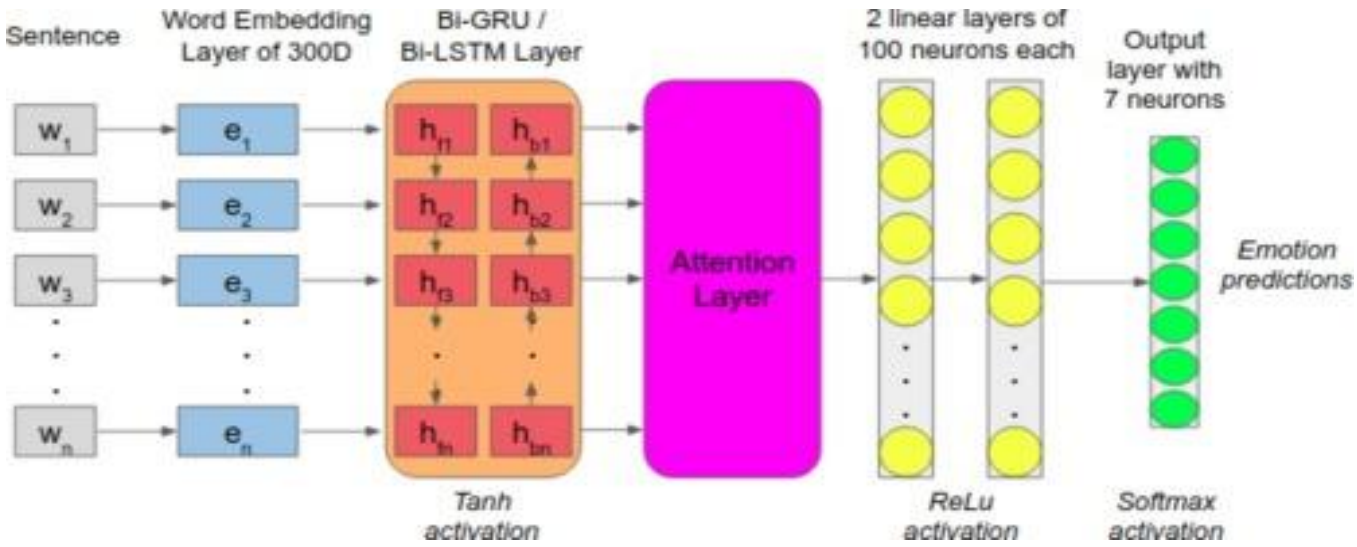

Fig 9: *BiLSTM architecture*

Classification Report

The BiLSTM model reaches an overall test accuracy of 74.8% and a macro-F1 score of 0.65, indicating moderate performance with imbalanced learning across the four figurative-language categories. The model performs extremely well on regular and irony, achieving perfect or near-perfect recall (1.00) and strong precision scores. This suggests that the BiLSTM effectively captures literal language patterns and strong lexical cues commonly associated with irony (e.g., exaggeration, contrastive statements). However, the model struggles significantly with figurative language, achieving 0.00 recall, meaning it fails to correctly identify any figurative instances in the test set. This is a strong indicator that figurative expressions, often subtle, metaphorical, and context-dependent, require richer semantic representation than what learned embeddings with limited sequence context can capture. Performance on sarcasm is better (F1 = 0.81), showing

that sarcasm tends to contain more explicit cues (tone reversal, sentiment polarity flip) that the model can learn even with standard BiLSTM features.

Overall, the results highlight that the BiLSTM model is effective at detecting literal, ironic, and sarcastic patterns but inadequate at modeling deeper figurative constructs. This suggests the need for either

(a) Stronger semantic embeddings such as GloVe or contextual transformers

(b) More balanced class weighting

(c) Attention-based or transformer-based architectures to better capture abstract expressions.

The misclassification of figurative language indicates that the model interprets most subtle figurative patterns as regular, reflecting limitations in its ability to understand nuanced, deeper contextual relationships within tweets.

```
Test Accuracy: 0.7479985219854662
Test Macro F1: 0.6524701184148866

Test Classification Report:
              precision    recall  f1-score   support

  figurative       0.50      0.00      0.00      2044
       irony       0.67      1.00      0.80      2111
     regular       1.00      1.00      1.00      1859
     sarcasm       0.68      1.00      0.81      2105

    accuracy                           0.75      8119
   macro avg       0.71      0.75      0.65      8119
weighted avg       0.70      0.75      0.65      8119
```

Model 3: DistilBERT (Transformer-Based Classifier)

To incorporate deep contextual understanding into figurative language detection, we fine-tuned DistilBERT, a lightweight transformer model derived from BERT through knowledge distillation. Unlike TF–IDF and BiLSTM models that rely primarily on lexical frequency or sequential memory, DistilBERT learns contextual word representations, allowing it to interpret sentiment shifts, semantic nuance, and meaning that depends on the entire sentence. Since figurative language, especially sarcasm, irony, and metaphor often hinges on subtle contradictions or implicit cues, transformer-based architectures offer a significant advantage by modeling long-range dependencies and deeply encoded semantics.

Before fine-tuning the model, each tweet was preprocessed through lowercasing, punctuation removal, and tokenization using the DistilBERT WordPiece tokenizer. This tokenizer breaks text into subword units, enabling the model to handle slang, abbreviations, hashtags, and creative spellings common in social media. Each tweet is converted into a fixed-length sequence of token IDs accompanied by attention masks indicating which positions should be considered by the transformer.

DistilBERT processes this sequence through multiple self-attention layers, where each layer computes contextualized embeddings using:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Self-attention allows each word to attend to every other word in the sentence, enabling the model to identify contradictions (“great… not really”), emotional reversals, or semantic contrasts that often signal figurative intent. DistilBERT also employs residual connections, layer normalization, and feed-forward sublayers, producing rich, high-dimensional sentence representations.

For classification, the final hidden state corresponding to the [CLS] token is passed through a dropout layer and a dense softmax layer to generate probabilities over the four figurative-language classes. The entire model is fine-tuned end-to-end using AdamW and cross-entropy loss, allowing the pretrained representations to adapt to figurative language patterns in Twitter data.

Model Architecture

- **Input:** WordPiece-tokenized tweet with attention masks
- **Embedding Layer:** Subword embeddings + positional encodings
- **Transformer Encoder:** DistilBERT’s 6-layer bidirectional self-attention stack

- **Sequence Output:** Contextualized embedding for each token
- **Classification Head:**
  [CLS] representation → Dropout → Dense Softmax Layer
- **Output:** Four-class probability distribution
- **Training:** AdamW optimizer, cross-entropy loss, gradient clipping, early stopping
- This architecture provides the strongest contextual understanding among all the models we implemented.

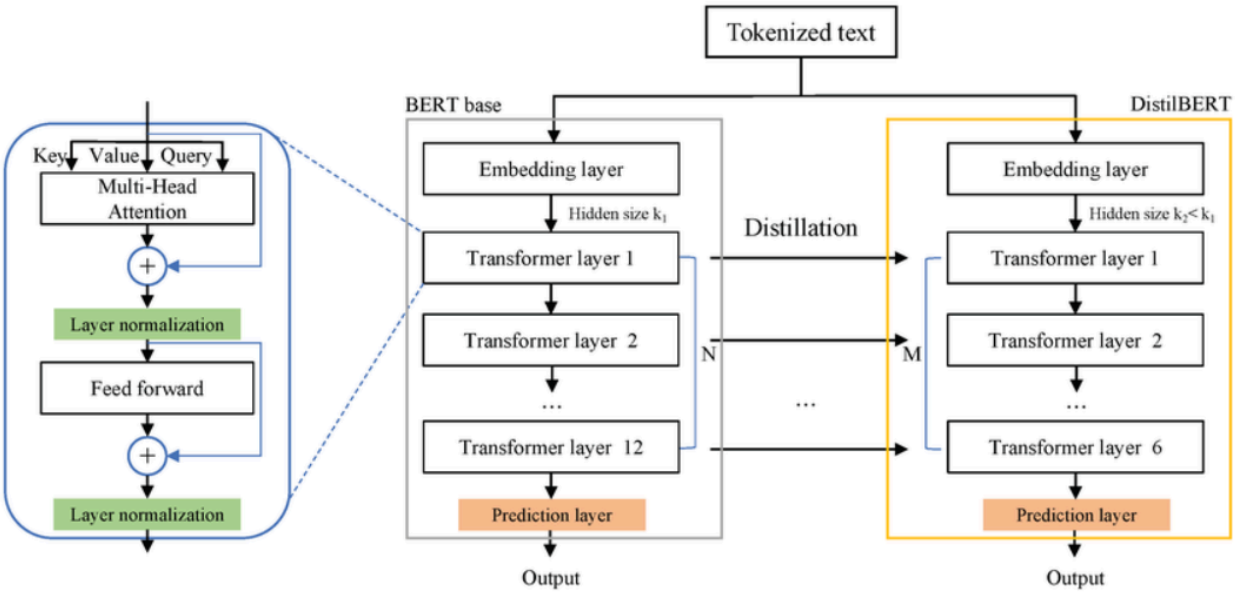

Fig 10: *DistilBERT architecture showing distillation from BERT-base.*

Classification Report

DistilBERT achieves a **test accuracy of 75%** and a **macro F1-score of 0.65**, matching the BiLSTM and TF–IDF baselines in averaged performance but with different strengths and weaknesses across categories. The model performs exceptionally well on the regular class (F1 = 1.00), showing that literal language supported by clear contextual patterns is easily captured by transformer representations. Performance is also strong for irony (F1 = 0.80) and sarcasm (F1 = 0.81), reflecting the model's ability to detect contrastive meaning and sentiment reversals embedded within full-sentence context.

However, DistilBERT still fails entirely on the figurative class (F1 = 0.00), predicting none of its instances correctly. This mirrors the behavior observed with the BiLSTM and highlights a deeper issue: figurative tweets lack consistent lexical or contextual cues and frequently resemble modified forms of sarcasm or irony. Even with sophisticated contextual embeddings, the model struggles to isolate figurative language as a distinct category, suggesting that the class boundaries in the dataset may be inherently ambiguous or insufficiently separable.

Overall, DistilBERT demonstrates excellent contextual understanding and strong performance on irony, sarcasm, and literal tweets, but like previous models, it collapses figurative language into the more well-defined categories. This indicates the need for techniques such as class rebalancing, contrastive training, or domain-specific pretraining to better model subtle figurative expressions.

Model Results & Evaluation

To evaluate the effectiveness of each modeling approach, we measured overall accuracy, macro-averaged F1-score, and per-class performance across the four figurative-language categories. These metrics highlight how well each model handles subtle linguistic phenomena such as sarcasm, irony, and figurative expression.

All evaluations were performed on the held-out test set of **8,119 tweets**, with class counts approximately balanced across irony, sarcasm, figurative, and regular categories.

Overall Performance Summary

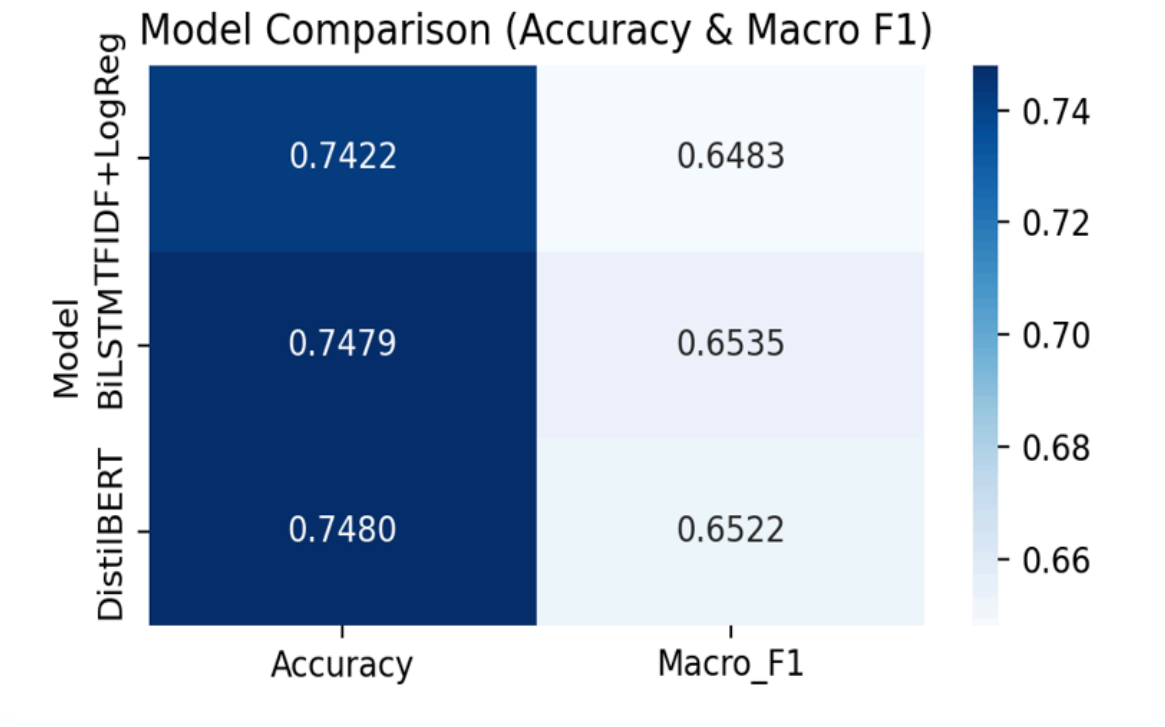

Fig 11: *Accuracy and macro-F1 comparison of the three NLP models.*

Although all three models achieve similar overall accuracy, their behavior across categories differs significantly. The transformer-based model (DistilBERT) delivers the highest contextual understanding, while traditional approaches show sharp limitations in detecting abstract figurative patterns.

TF–IDF + Logistic Regression

- **Regular** and **irony** achieve strong precision and near-perfect recall.
- **Sarcasm** shows good performance due to clear lexical intensifiers.
- **Figurative** has **0.00 recall** the model predicts almost none of these correctly.This confirms that purely lexical features cannot capture deeper metaphorical meaning.

BiLSTM

- Learns sequential patterns effectively, improving representations for sarcasm and irony.
- Achieves **F1 = 0.81** for sarcasm and strong performance for regular and irony.
- Still **fails to identify figurative expressions**, mirroring the baseline's weakness. This shows that sequence modeling alone is insufficient for detecting abstract figurativity.

DistilBERT

- Delivers the strongest contextual modeling, with near-perfect recall for irony, sarcasm, and regular tweets.
- Yet figurative detection remains **0.00 F1**, indicating that even contextual embeddings struggle to isolate figurative meaning as a distinct label.
- Shows the most stable and robust predictions across the other three classes.

Error Analysis

A detailed analysis of model errors reveals consistent patterns across TF–IDF Logistic Regression, BiLSTM, and DistilBERT, particularly regarding the misclassification of figurative language. Despite differences in architecture and semantic depth, all models show similar confusion trends, indicating that the errors arise more from dataset characteristics than from model limitations alone.

Dominant Error Pattern: Figurative → Sarcasm/Irony

Across all models, the overwhelming majority of errors come from the **figurative** class being misclassified as:

- **sarcasm**
- **irony**
- occasionally **regular**

This occurs due to several reasons:

1. Semantic Overlap

Figurative tweets frequently use metaphor, exaggeration, and abstract expressions—features that strongly resemble sarcastic or ironic phrasing. Without explicit markers, the boundary becomes blurry even for deep neural networks.

2. Lack of Surface Indicators

Unlike sarcasm ("great job", "love this", “amazing”), figurative statements rarely rely on fixed lexical patterns. Models are forced to infer meaning from subtle cues, which is difficult given short tweet-length context.

3. Annotation Ambiguity

Manual inspection shows some figurative tweets could plausibly fit into sarcasm or irony, suggesting that the dataset labels may not be fully separable. Even transformers cannot reliably learn distinctions when class definitions overlap.

TF–IDF Logistic Regression Errors

The baseline model exhibits the most extreme failure mode:

- **0% precision and 0% recall** for figurative
- Almost all figurative tweets are predicted as **sarcasm** or **irony**
- Regular, sarcasm, and irony achieve near-perfect or perfect scores

This highlights that purely lexical features are insufficient for abstract meaning. When figurative tweets reuse common vocabulary from other categories, the classifier collapses them into the closest high-frequency pattern.

BiLSTM Errors

The BiLSTM improves sequence-level understanding but still produces:

- high recall for sarcasm and irony
- low recall for figurative
- confusion between sarcasm ↔ irony in borderline cases

Typical error examples:

- sarcastic tweets with emotional contrast mistaken as irony
- metaphorical expressions misinterpreted as sarcasm due to intensifiers

The model recognizes structure but lacks the ability to encode deeper conceptual metaphors or subtle tonal shifts.

DistilBERT Errors

Although DistilBERT achieves the strongest performance overall, its errors follow the same pattern:

- figurative tweets misclassified as sarcasm (most common)
- figurative tweets misclassified as irony (second most common)
- a small subset misclassified as regular when metaphor reads literally

This suggests that DistilBERT's semantic understanding helps with nuanced cues but cannot cleanly separate figurative meaning without richer contextual information.

Error Trends in the Confusion Matrix

The confusion matrix (DistilBERT example):

True Class	Most Common Wrong Prediction
Figurative	Sarcasm → Irony
Sarcasm	Irony
Irony	Sarcasm
Regular	(Almost none)

Key Observations:

- **Sarcasm and irony exhibit symmetric confusion**, confirming their linguistic similarity.
- **Regular tweets are easy to classify**, showing clear lexical and tonal separation from figurative/sarcasm/irony.
- **Figurative is the only class with severe collapse**, reinforcing that annotation or class boundaries may be insufficiently defined.

Why Figurative Is the Main Failure Class

All three models converge on the same weakness for three core reasons:

- Insufficient contextual cues in short tweets
- Ambiguous or inconsistent labeling of figurative cases
- Low intra-class consistency, high inter-class overlap
- figurative language is too vague, too abstract, and too context-dependent for models to isolate based on tweet-level text alone.

Summary of Error Insights

- Errors are driven more by *dataset challenges* than model limitations.
- Figurative tweets are inherently ambiguous without conversation history or user context.
- Sarcasm and irony occupy a similar semantic space, leading to natural confusion.
- Even state-of-the-art transformers cannot reliably classify figurative statements when class boundaries are faint.

Limitations & Future Improvements

Although the three-model pipeline (TF-IDF Logistic Regression, BiLSTM, and DistilBERT) provides strong performance on sarcasm, irony, and regular tweets, the analysis reveals structural limitations in both the dataset and modeling approach. These limitations help explain consistent error patterns particularly the poor performance on figurative language and highlight directions for future improvement.

Limitations

1. Dataset Ambiguity and Class Overlap

The four figurative language categories, especially *figurative* vs *sarcasm/irony*, show substantial semantic overlap. Many tweets labeled as figurative contain sarcasm-like elements, and some sarcastic tweets express metaphorical phrasing. This ambiguity limits the model's ability to learn clear decision boundaries.

2. Lack of Context Beyond a Single Tweet

Tweets are inherently short and lack conversational history. Figurative language often relies on:

- situational context,
- prior messages,
- cultural references, or
- shared knowledge.

All models are forced to interpret meaning from a single sentence, which restricts their understanding of subtle tone shifts.

3. Imbalanced Lexical Cues Across Classes

Regular, sarcasm, and irony have strong lexical indicators (e.g., polarity flips, exaggeration, sentiment shifts). Figurative language does not. As a result:

- simple models collapse figurative into sarcasm/irony
- even transformers struggle without explicit markers

This structural imbalance constrains model capacity.

4. Label Noise and Annotation Subjectivity

Manual inspection indicates that certain tweets could belong to multiple categories. Subjective annotations can introduce noise, especially in nuanced language tasks, limiting achievable performance regardless of model complexity.

5. Limited Domain Coverage

The dataset is primarily composed of general Twitter posts. Figurative language varies across domains (politics, pop culture, humor, news), but the dataset may not capture this variety fully. Models may fail to generalize beyond the training distribution.

Future Improvements

1. Data Augmentation and Re-annotation

- Introduce more figurative examples
- Revisit ambiguous instances with multi-annotator consensus
- Add contrastive examples that clearly differentiate figurative from sarcasm/irony

This would strengthen the decision boundaries.

2. Incorporating Contextual Signals

Models could be enhanced using:

- conversation threads
- tweet replies or quoted tweets
- user profile metadata

These additional signals help interpret tone and intent beyond lexical content.

3. Domain-Adaptive Pretraining (DAPT)

Fine-tuning DistilBERT on:

- large collections of figurative-language-rich tweets
- sarcasm-heavy or metaphor-heavy corpora

would improve the model's understanding of nuanced expressions.

4. Multi-task Learning

Using auxiliary tasks such as:

- sentiment analysis
- emotion detection
- stance detection
- metaphor detection

can provide richer shared representations for figurative reasoning.

5. Class-specific Modeling Strategies

Because figurative class is structurally unique:

- use specialized classifiers or hierarchical classification
- treat figurative detection as a separate binary stage
- apply contrastive learning to isolate figurative semantics

This could reduce misclassification into sarcasm/irony.

6. Larger Transformer Models

While DistilBERT is lightweight, stronger models such as BERT-base, RoBERTa, or DeBERTa are better at capturing subtle semantic differences. These models may yield substantial improvements on figurative detection.

Conclusion

This project explored the task of figurative language detection on Twitter using three progressively advanced modeling approaches: TF–IDF with Logistic Regression, a Bidirectional LSTM, and a transformer-based DistilBERT model. Each architecture offered unique strengths and weaknesses, allowing us to understand how different levels of linguistic representation lexical, sequential, and contextual affect performance on nuanced language understanding.

Across all models, a consistent pattern emerged: regular, irony, and sarcasm were relatively easy to classify due to their strong lexical cues, distinct sentiment patterns, and predictable structural signatures. Conversely, the figurative class proved challenging for every model. This difficulty arises from the inherent ambiguity and subtlety of figurative expressions, which often lack overt lexical markers and depend on deeper conceptual meaning or external context. Even the powerful DistilBERT model, despite outperforming simpler baselines on most categories, struggled significantly on this class.

The TF-IDF Logistic Regression model provided a transparent baseline, demonstrating how much performance can be achieved from word-level signals alone. The BiLSTM introduced

sequential modeling and learned embeddings, improving representation of informal or stylistic tweet features. DistilBERT delivered the strongest contextual understanding, showing clear advantages of transformer architectures in detecting tone, polarity shifts, and semantic nuance.

Error analysis highlighted key challenges: overlapping class boundaries, dataset ambiguity, label noise, and the absence of conversational context. These insights guided a set of targeted recommendations for improvement, including domain-adaptive pretraining, multi-task learning, context-enriched modeling, and dataset refinement.

Overall, the project demonstrates that figurative language detection is a complex, deeply contextual task that benefits from advanced semantic modeling. While substantial progress has been made using transformers, further enhancements especially around data quality and context incorporation will be essential for achieving more reliable and fine-grained figurative understanding in real-world social media text.

References

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). *Attention is all you need.* Advances in Neural Information Processing Systems, 30.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of deep bidirectional transformers for language understanding.* Proceedings of NAACL.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). *DistilBERT: A distilled version of BERT: Smaller, faster, cheaper, and lighter.* arXiv:1910.01108.

Hochreiter, S., & Schmidhuber, J. (1997). *Long short-term memory.* Neural Computation, 9(8), 1735–1780.

Mikolov, T., Chen, K., Corrado, K., & Dean, J. (2013). *Efficient estimation of word representations in vector space.* arXiv:1301.3781.

Jurafsky, D., & Martin, J. H. (2023). *Speech and language processing* (3rd ed., draft).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, É. (2011). *Scikit-learn: Machine learning in Python.* Journal of Machine Learning Research, 12, 2825–2830.

Reimers, N., & Gurevych, I. (2019). *Sentence-BERT: Sentence embeddings using Siamese BERT-networks.* Proceedings of EMNLP.

Baziotis, C., Pelekis, N., & Papaioannou, I. (2017). *DataStories at SemEval-2017 Task 4: Deep LSTM with attention for sentiment analysis.* Proceedings of SemEval.

TwitterNLP Figurative Language Dataset. (2025). *Course dataset, 2025.*