Sarcasm Detection in Tweets

Christa Sparks | CSPB 4830 | April 2025

Problem Overview

Sarcasm flips sentiment—hard even for humans

Misclassifying tweets skews consumer-and-brand insights

Our task: binary classify a tweet as sarcastic or not

Examples

Hey there! Nice to see you Minnesota/ND Winter Weather

Nothing makes me happier then getting on the highway and seeing break lights light up like a Christmas tree..

Motivation & Goals

- Goal: Compare traditional ML vs. BERT fine-tuning on sarcasm
- Questions:

  How far can hand-engineered features go?

  Do contextual embeddings (BERT) capture subtlety?

- Note: Hybrid approach planned, but out of scope for this talk

Dataset Description

SemEval-2018 Task 3 (irony detection in English tweets)

Size & structure:

- ~8 000 tweets, TSV with ID, binary label, text

Built-in cleaning:

- Retweets, duplicates, non-English removed
- XML escapes resolved, emojis → text
- Irony-related hashtags stripped

My preprocessing:

- Lowercasing, URL/mention removal, hashtag symbol drop
- Contraction expansion, simple negation marking

Approach Overview

- **Traditional ML Pipeline:**

  TF-IDF n-grams + VADER sentiment + punctuation + POS ratios + emoticons

  Logistic Regression baseline

- **Transformer Pipeline:**

  Fine-tune bert-base-uncased for 2-label classification

  Evaluate via Hugging Face's Trainer

Tools & Techniques

- ➔ Data & preprocessing: pandas, NumPy, regex, spaCy
- ➔ Traditional ML: scikit-learn (Pipeline, GridSearchCV), NLTK-VADER
- ➔ Deep learning: PyTorch, Hugging Face Transformers + Accelerate
- ➔ HPO: scikit-learn's RandomizedSearchCV + Optuna for BERT
- ➔ Eval: classification_report, precision/recall/F1 curves, threshold sweep

Experiments & Analysis

Initial results:

Logistic regression (default) → 61% accuracy, F1 ≈ 0.61

BERT baseline (3 epochs at 2e-5) → 70% accuracy, F1 ≈ 0.73

Hyperparameter tuning:

ML: C, class_weight, n-gram range → +3 pts F1

BERT: Optuna search → cut HPO runtime from ~3 h to < 1 h (see next slide)

Error analysis:

Misfires on cultural references, multi-tweet context

Results

Model	Accuracy	F1	Key Hyperparameters
Logistic Regression	0.64	0.64	C = 2.54, class_weight='balanced', ngram=(1,1)
BERT (baseline)	0.70	0.73	lr = 2×10^{-5}, epochs = 3, weight_decay = 0.01

Challenges & Lessons Learned

Subtlety of sarcasm: cultural/content context matters

HPO cost vs. benefit: long BERT searches eat time

Data quirks:

- SemEval cleaning helped, but
- Tweets still contain unexpected noise

Scope creep: hybrid model remains future work

Conclusion & Future Work

Takeaways:

- Hand-engineered features give solid baseline (F1 0.64)
- BERT fine-tuning lifts performance (F1 0.73)
- Fast HPO slashes tuning time with little quality loss

Next steps:

- Incorporate user-level/contextual features (conversation history)
- Build and evaluate hybrid models (features + embeddings)
- Explore few-shot or meta-learning approaches