TF-IDF vs Transformer Sentiment Analysis

TweetEval / sentiment — Notebook-derived results & figures

Albert Vempala | Capstone Two

Purpose & Success Criteria

- Compare TF-IDF baseline vs fine-tuned Transformer
  - Primary metrics: Accuracy + Macro F1 (class-balanced)
  - Use confusion matrices + error analysis for explainability
  - Select final model for production deployment

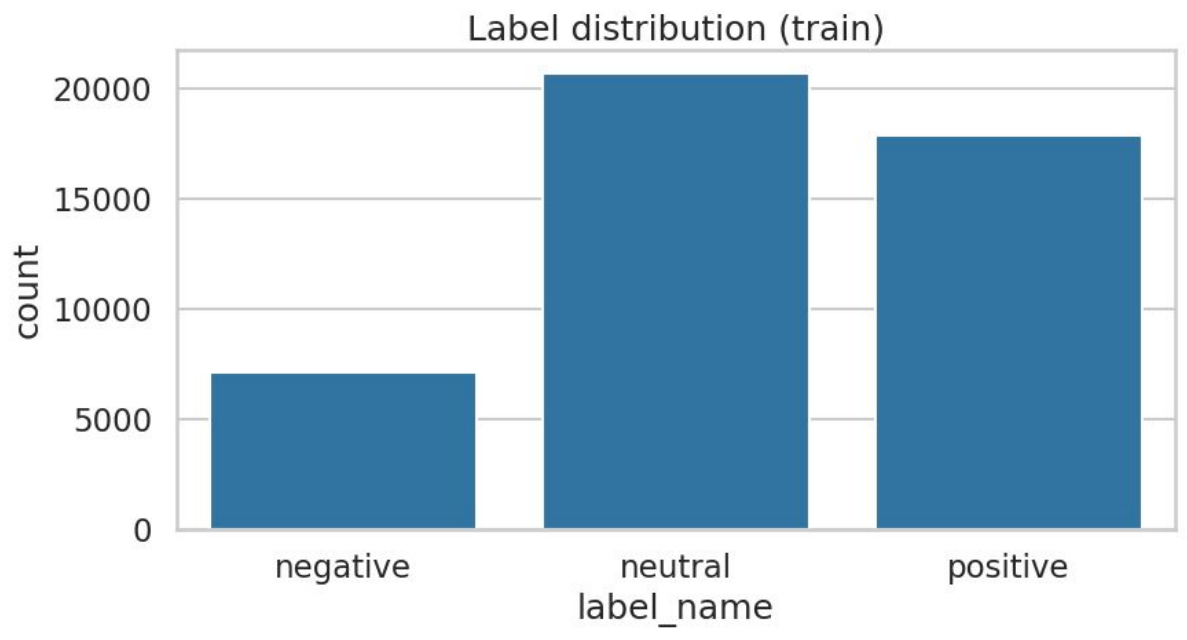
Data Overview (TweetEval / sentiment)

- Splits: Train=45,615 | Val=2,000 | Test=12,284 (notebook output)
  - Labels: 0=negative | 1=neutral | 2=positive
  - Short tweets: slang, emojis, abbreviations
  - Objective: robust generalization on noisy text

Preprocessing (Two Tracks)

- TF‑IDF track: normalize and clean for vectorization
  - Transformer track: minimal normalization to preserve semantics
  - Shared checks: missing/duplicates + label mapping
  - Ensures fair apples-to-apples comparison

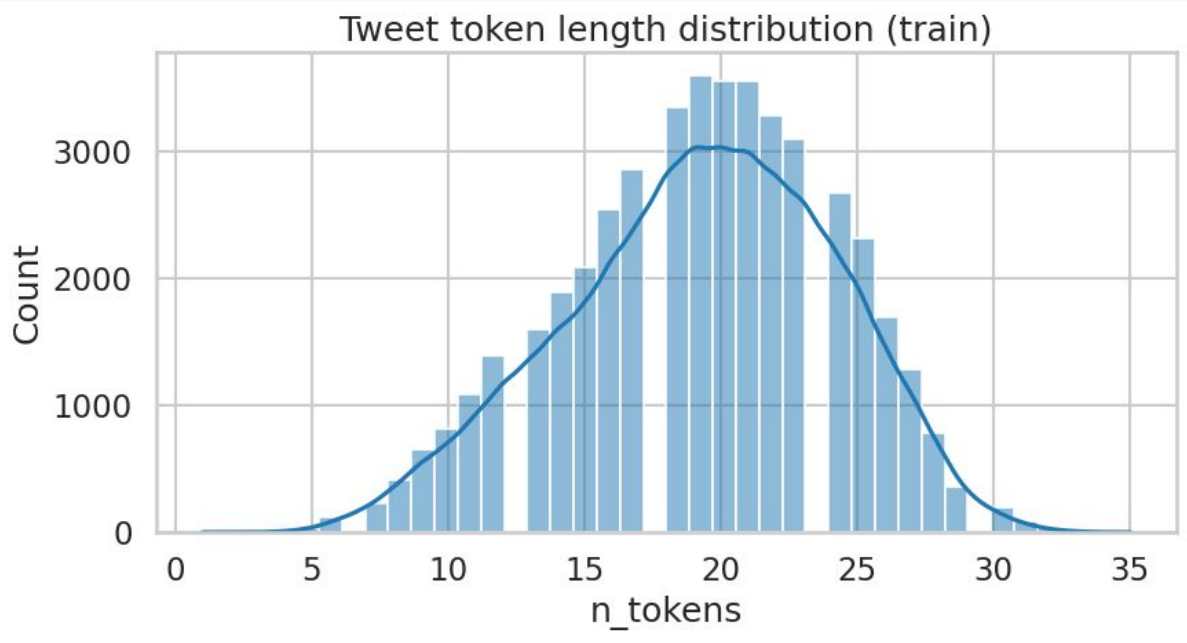
EDA: Label Distribution (Train)



Class balance informs macro metrics selection Supports class-focused error analysis

Notebook: eda_label_distribution.png

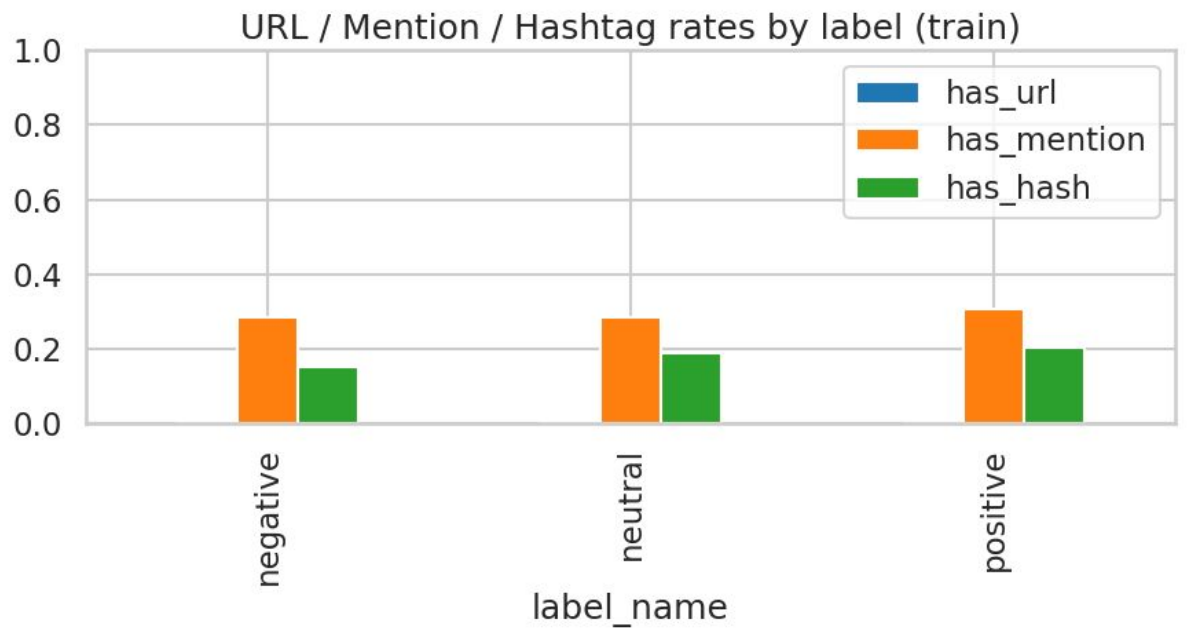
EDA: Token Length Distribution (Train)



Tweets are short; context can flip sentiment, motivates contextual modeling beyond bag-of-words

Notebook: eda_token_length.png

EDA: URL/Mention/Hashtag Rates by Label (Train)



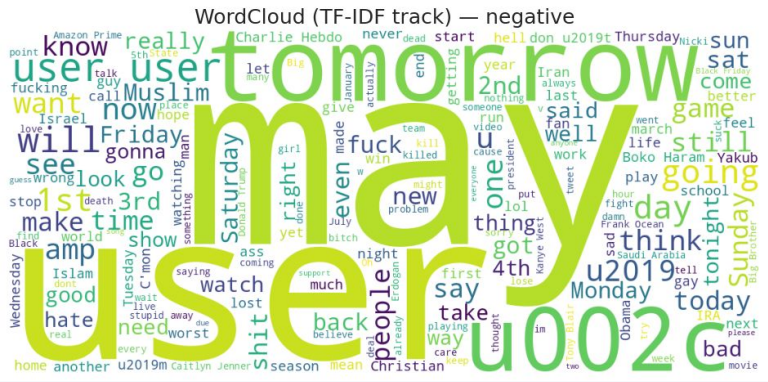
Behavioral signals differ across sentiment labels
Used for interpreting class-specific errors

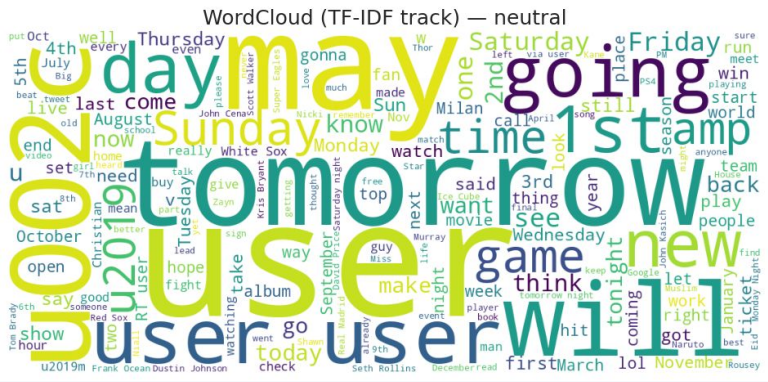
Notebook: eda_rates_by_label.png

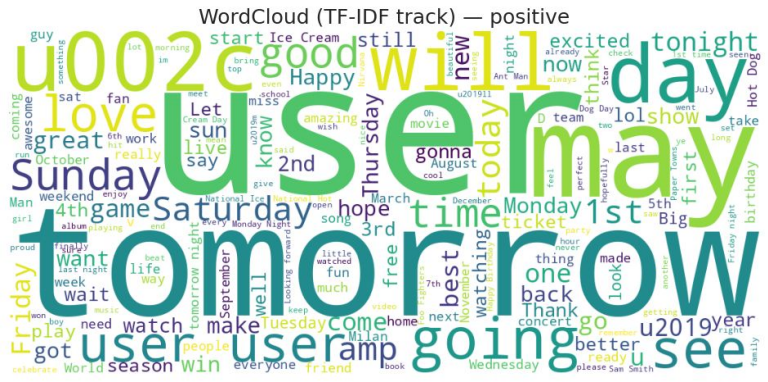
label	has_url	has_mention	has_hash	n_emojis
negative	0.30%	28.65%	15.14%	0.0
neutral	0.23%	28.53%	18.73%	0.0
positive	0.13%	30.81%	20.28%	0.0

EDA: Word Clouds by Class (TF-IDF Track)

Qualitative scan of discriminative language by sentiment class







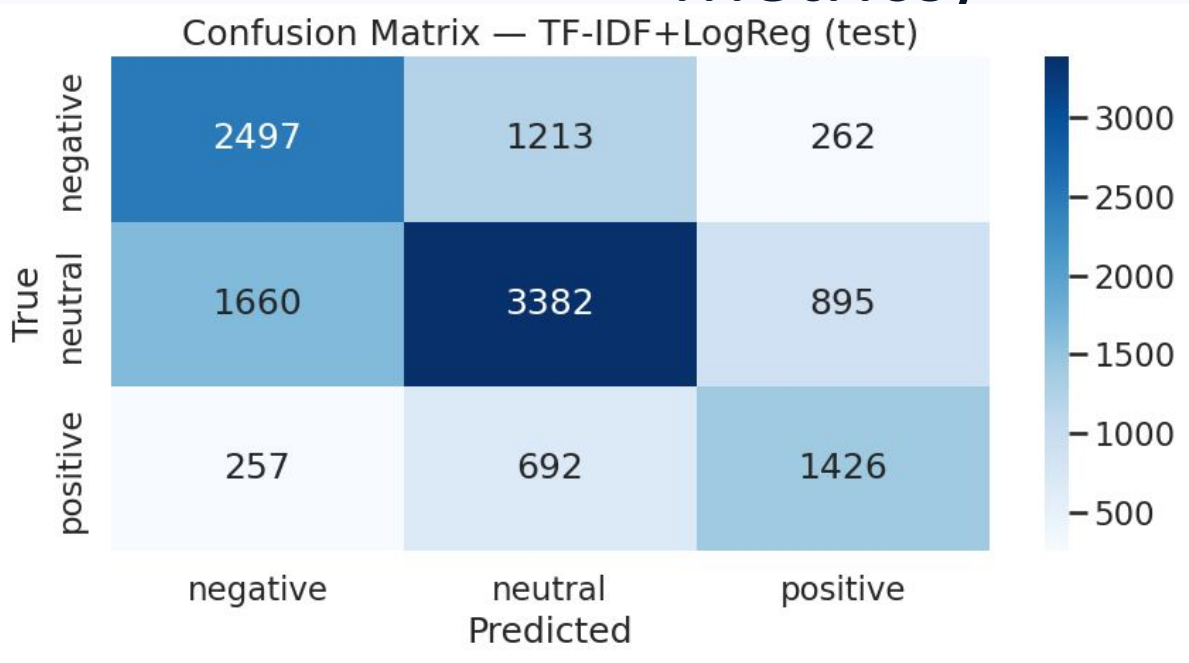
Methodology (Rubric-Aligned)

- Train on Train, tune on Val, report final metrics on Test
  - Compare models using Accuracy + Macro Precision/Recall/F1
  - Use confusion matrices to diagnose error modes
  - Document results and provide business recommendations

Model A: TF-IDF + Logistic Regression (Baseline)

- Vectorizer: n-grams (1,2), sublinear TF, min_df=2, max_df=0.95
  - Classifier: Logistic Regression, class_weight=balanced, max_iter=3000
  - Grid search → Best C = 2.0 (notebook output)
  - Validation: Acc=0.6675 | Macro F1=0.6455

TF-IDF Test Results (Confusion Matrix + Metrics)

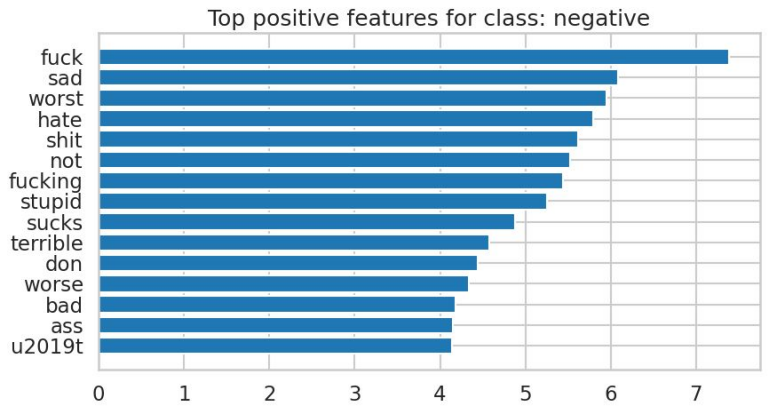


Test: Acc=0.5947 | Macro P=0.5858 | Macro R=0.5996 | Macro F1=0.5911 Error mode: neutral boundary cases + context loss

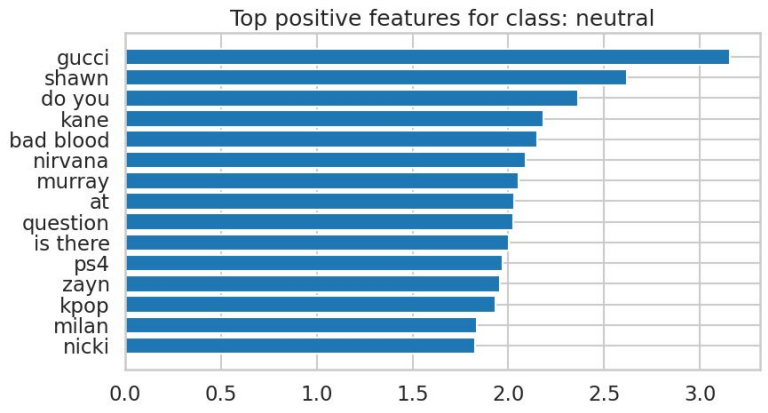
Notebook: cm_tfidf_test.png

TF-IDF Interpretability: Top Features by Class

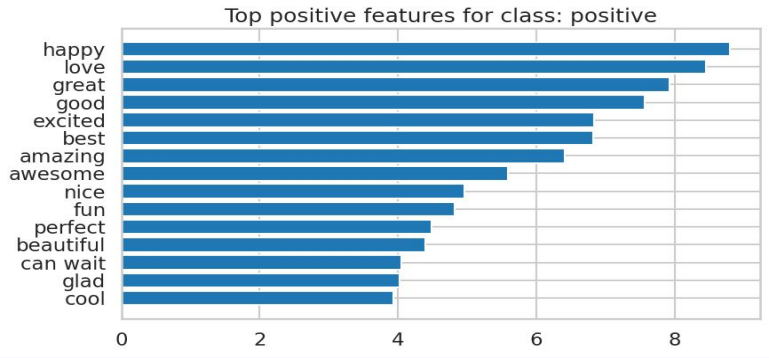
Notebook-derived feature-weight bar charts (model explainability)



Negative



Neutral

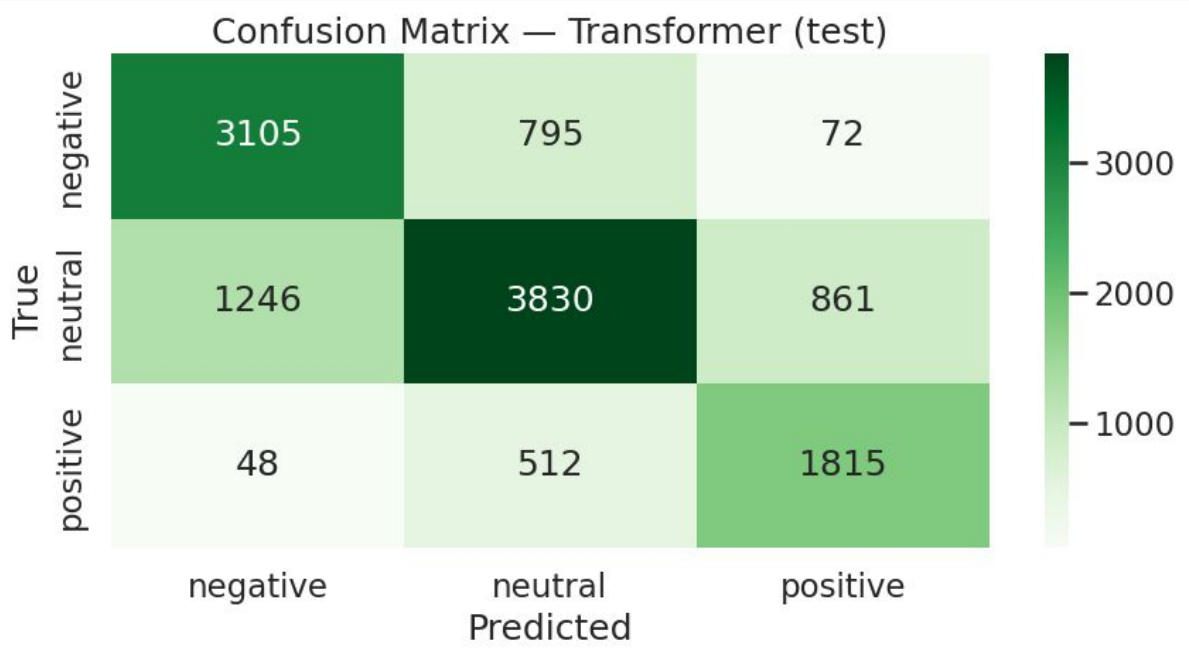


Positive

Model B: Fine-Tuned Transformer (Contextual)

- Fine-tuned on TweetEval sentiment (3 epochs; notebook)
  - Test: Acc=0.7123 | Macro P=0.7040 | Macro R=0.7303 | Macro F1=0.7140
  - Strength: captures word order + semantic context
  - Improves class-balanced recall (macro recall)

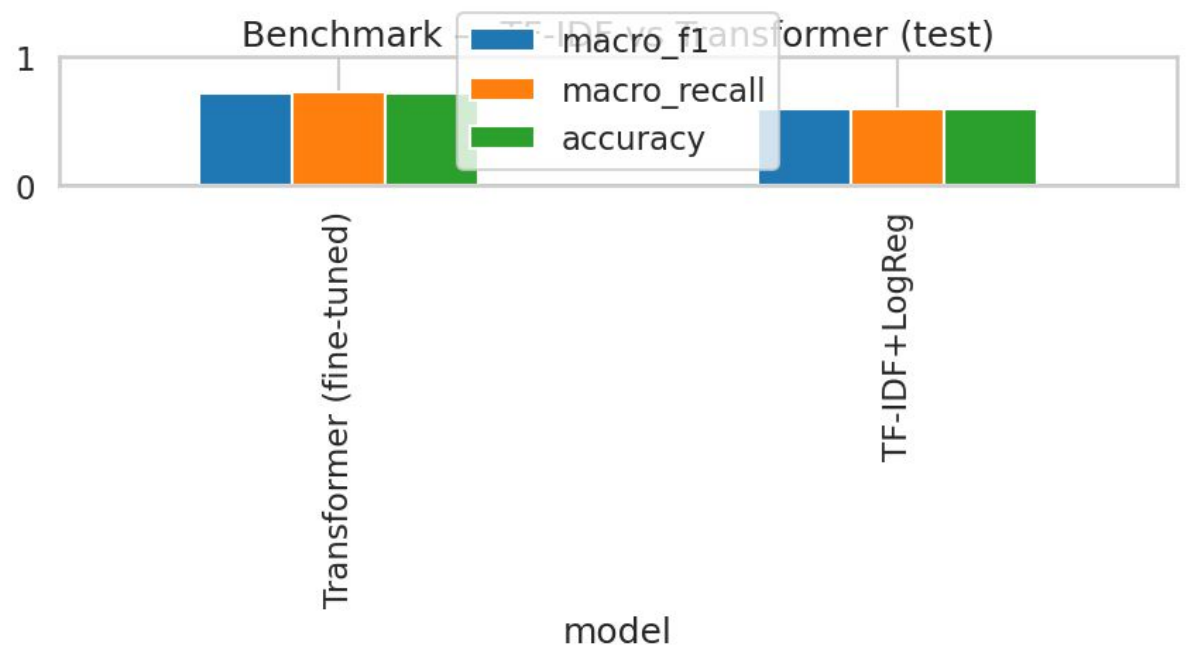
Transformer Test Results (Confusion Matrix)



Fewer cross-class confusions vs TF-IDF
Better coverage of nuanced sentiment cases

Notebook: cm_transformer_test.png

Benchmark Comparison (Notebook Bar Plot)



Accuracy lift: +0.1176 absolute

- Macro F1 lift: +0.1229 absolute
- Macro Recall lift: +0.1307 absolute
- Macro Precision lift: +0.1182 absolute

Notebook: benchmark_bar.png

Error Analysis: TF-IDF Correct, Transformer Wrong

Failure cases highlight ambiguity and ultra-short tweets (notebook output).

text	true	tfidf	transformer
@user @user @user @user @user @user take away	negative	negative	neutral
Ben Carson awoke this morning to find out that.....	neutral	neutral	positive
Next?#draintheswamp #TimeForChange#Trump Transi.....	neutral	neutral	negative
Grayson Allen just gave dude such a sick move…	negative	negative	positive
Download UBER app, Register with this promo co.....	neutral	neutral	positive

Recommendations (Executive + Technical)

- Deploy Transformer for decision-critical sentiment intelligence
  - Keep TF‑IDF baseline for low-cost fallback and regression checks
  - Operationalize with batching, monitoring, and periodic re‑tuning
  - Report macro metrics to ensure class-balanced quality

High-Impact Conclusion

- Accuracy improved: 0.5947 → 0.7123 (Δ=0.1176)
  - Macro F1 improved: 0.5911 → 0.7140 (Δ=0.1229)
  - Transformers reduce costly sentiment misreads on ambiguous text
  - Final selected model: Transformer (fine-tuned)