Verified Patterns and Interpretations

Based Strictly on Experimental Findings

This document lists patterns observed in the baseline experiments with interpretations supported directly by the data. Each interpretation includes the specific evidence from the experiments.

Pattern 1: TF-IDF Models Use Character Names as Features

Evidence

- TF-IDF + LogReg on gold: 75.00% (original) → 60.71% (normalized) = **-14.29%**
- TF-IDF + SVM on gold (context_text): 60.71% (original) → 50.00% (normalized) = **-10.71%**
- TF-IDF + SVM on test (text_only): 64.18% (original) → 55.22% (normalized) = **-8.96%**
- Mean accuracy change across all TF-IDF conditions: negative

Interpretation

TF-IDF vectorization converts character names (CHANDLER:, SHELDON:, etc.) into features with associated weights. When names are removed, these features disappear, and accuracy drops substantially. The larger drop in TF-IDF compared to XLM-R (see Pattern 2) indicates TF-IDF models directly encode character names as predictive features.

Pattern 2: XLM-R is More Robust to Name Removal Than TF-IDF

Evidence

- XLM-R (text_only, mean, gold): 82.14% (original) → 78.57% (normalized) = **-3.57%**
- XLM-R (text_only, cls, gold): 71.43% (original) → 71.43% (normalized) = **0.00%**
- TF-IDF LogReg (text_only, gold): 75.00% (original) → 60.71% (normalized) = **-14.29%**

Interpretation

XLM-R's pre-trained semantic representations capture meaning beyond individual tokens. When character names are normalized, XLM-R retains more predictive information from the remaining text than TF-IDF, which loses the character name features entirely.

Pattern 3: Mean Pooling Outperforms CLS Pooling

Evidence

On gold set (text_only, original):

- Mean pooling: **82.14%** accuracy
- CLS pooling: **71.43%** accuracy
- Difference: **+10.71%** for mean pooling

On gold set (text_only, normalized):

- Mean pooling: **78.57%** accuracy
- CLS pooling: **71.43%** accuracy
- Difference: **+7.14%** for mean pooling

Interpretation

Mean pooling aggregates information from all tokens in the sequence. CLS pooling uses only the first token's representation. For this sarcasm detection task, distributing attention across the entire utterance captures more relevant signal than relying on a single summary token.

Pattern 4: Context Does Not Improve Frozen Model Performance

Evidence

XLM-R mean pooling on gold (original names):

- text_only: **82.14%**
- context_text: **67.86%**
- Difference: **-14.28%** (context hurts)

TF-IDF LogReg on gold (original names):

- text_only: **75.00%**
- context_text: **64.29%**
- Difference: **-10.71%** (context hurts)

Overall context effect:

- Mean accuracy change: **+0.32%** (negligible)
- Conditions where context helps: 8
- Conditions where context hurts: 5

Interpretation

Adding dialogue context does not help and sometimes hurts frozen baseline models. Without task-specific training, the models cannot learn to use context for detecting incongruity between what is said and what is meant. The additional text from context may introduce noise rather than useful signal.

Pattern 5: Surface Sarcasm Markers Correlate with Incorrect Predictions

Evidence (from correct_vs_incorrect.csv)

Features **higher** in incorrect predictions (statistically significant, $p < 0.05$):

- all_caps_count: +1440% ($p < 0.001$)
- ellipsis_count: +165.52% ($p < 0.001$)
- sentiment_polarity: +113.73% ($p < 0.001$)
- sarcasm_marker_count: +64.12% ($p < 0.001$)

- starts_with_interjection: +60.67% (p < 0.001)
- positive_word_count: +56.90% (p < 0.001)
- intensifier_count: +36.94% (p = 0.021)

Interpretation

Examples containing features commonly associated with sarcasm (exclamations, "oh", "great", positive sentiment, intensifiers) are **harder** to classify correctly, not easier. The baseline models cannot distinguish between genuine use of these markers and ironic use.

Pattern 6: Negative Language and Contrast Help Classification

Evidence (from correct_vs_incorrect.csv)

Features **lower** in incorrect predictions (statistically significant, p < 0.05):

- negative_word_count: -38.08% (p = 0.024)
- has_contrast: -37.87% (p < 0.001)
- negation_count: -24.82% (p = 0.019)

Interpretation

When text contains explicit negative words, negation, or contrast markers (but, however), the models perform better. These features provide clearer signals because they are less ambiguous — negative sentiment expressed with negative words is more likely to be genuine than positive sentiment expressed sarcastically.

Pattern 7: False Negatives Have More Sarcasm Markers Than False Positives

Evidence (from fp_vs_fn.csv)

Features **higher** in FN than FP (statistically significant, p < 0.05):

- intensifier_count: +582.72% (p < 0.001)
- has_intensifier: +465.27% (p < 0.001)
- exclamation_count: +121.14% (p < 0.001)
- sarcasm_marker_count: +116.91% (p < 0.001)
- has_sarcasm_marker: +101.88% (p < 0.001)
- starts_with_interjection: +105.67% (p < 0.001)

Error counts:

- False Positives: 365
- False Negatives: 452

Interpretation

The model misses sarcasm (FN) more often in examples that have **more** sarcasm markers, not fewer. Sarcastic utterances with heavy marker use (many exclamations, interjections, intensifiers) are being

classified as literal. This indicates the model cannot distinguish between exaggerated genuine enthusiasm and sarcastic exaggeration.

Pattern 8: Chandler Dominates the Error Set

Evidence (from both clustering analyses)

Character mentions in 78 unique errors:

- Chandler: **21 mentions (26.9%)**
- Howard: 10 mentions (12.8%)
- Leonard: 9 mentions (11.5%)
- Person: 7 mentions (9.0%)
- Sheldon: 6 mentions (7.7%)

Semantic clustering Cluster 1:

- Size: 34 examples (43.6% of all errors)
- Chandler mentions: 21 (61.8% of cluster)
- FN count: 19, FP count: 15

Interpretation

Chandler appears in over a quarter of all unique errors and dominates the largest error cluster. Despite being a famously sarcastic character, the model fails to correctly classify many Chandler utterances. The model may have learned an association between Chandler and sarcasm but fails when Chandler's sarcasm lacks obvious markers.

Pattern 9: Different Characters Have Different Error Types

Evidence (from clustering analyses)

Characters associated with **FN clusters** (missed sarcasm):

- Chandler: Clusters 1, 3 (TF-IDF) / Cluster 1 (Semantic)
- Howard: Cluster 7 (TF-IDF) / Cluster 5 (Semantic)
- Dorothy: Cluster 9 (TF-IDF) / Cluster 4 (Semantic)

Characters associated with **FP clusters** (false sarcasm):

- Leonard: Cluster 5, 8 (TF-IDF) / Cluster 3 (Semantic)
- Sheldon: Cluster 4 (TF-IDF) / Cluster 7 (Semantic)

Interpretation

The error type (FN vs FP) correlates with character identity. Chandler, Howard, and Dorothy errors are predominantly missed sarcasm. Leonard and Sheldon errors are predominantly false sarcasm. This indicates character-specific patterns that the baseline models handle differently.

Pattern 10: Semantic Clustering Produces More Coherent Groups Than TF-IDF

Evidence

TF-IDF clustering:

- Number of clusters: 12
- Silhouette score: **0.0153**

Semantic clustering:

- Number of clusters: 8
- Silhouette score: **0.0860**

Silhouette score ratio: 0.0860 / 0.0153 = **5.62x higher** for semantic

Interpretation

Semantic embeddings group errors into fewer, more coherent clusters than TF-IDF. The higher silhouette score indicates that semantic similarity produces tighter, better-separated groups. Errors that share meaning (but not necessarily words) are grouped together in semantic clustering.

Pattern 11: Dry/Deadpan Sarcasm is Consistently Missed

Evidence (from cluster examples)

Cluster 4 (Semantic) - Dorothy, 100% FN, 0% exclamation, 0% question:

- "Show them your slides of Hawaii"
- "Lords of Arabia"
- "No, she is going to sit here where it's a 112 degrees and eat enchiladas."

Cluster 1 (Semantic) - Chandler examples with low exclamation (29.4%):

- "If that doesn't keep kids in school, what will?"
- "Uh, so how many cameras are actually on you."
- "Yeah either that or gloria estefan was right eventually the rhythm is going to get you"

Interpretation

Sarcastic utterances with no exclamation marks, no questions, and no obvious markers (dry/deadpan delivery) are consistently classified as literal. The baseline models have no mechanism to detect sarcasm that relies purely on semantic incongruity or absurdity without surface-level cues.

Pattern 12: Test and Gold Performance Do Not Always Correlate

Evidence

XLM-R text_only_original_mean:

- Test: 68.66%
- Gold: **82.14%**
- Difference: +13.48% on gold

TF-IDF LogReg context_text_normalized:

- Test: 67.16%
- Gold: **50.00%**
- Difference: -17.16% on gold

Interpretation

Model rankings differ between test and gold sets. The gold set (28 examples) may have different characteristics than the test set (67 examples) drawn from the orange dataset. The small size of the gold set (28 examples) also means individual examples have high impact on accuracy.

Summary Table of Verified Patterns

#	Pattern	Key Evidence	Confidence
1	TF-IDF uses character names as features	-14.29% accuracy when normalized	High
2	XLM-R more robust to name removal	-3.57% vs -14.29% drop	High
3	Mean pooling > CLS pooling	+10.71% on gold	High
4	Context doesn't help frozen models	-14.28% for best model with context	High
5	Sarcasm markers correlate with errors	+64% sarcasm_marker_count in errors	High
6	Negative language helps classification	-38% negative_word_count in errors	High
7	FN have more markers than FP	+583% intensifier_count in FN	High
8	Chandler dominates errors	21/78 errors (27%)	High
9	Error type correlates with character	Chandler/Howard=FN, Leonard=FP	Medium
10	Semantic clustering more coherent	5.62x higher silhouette score	High
11	Dry sarcasm consistently missed	Dorothy cluster: 100% FN, 0% markers	High
12	Test/Gold performance differs	+13.48% to -17.16% differences	High

All interpretations are derived directly from experimental results. No external assumptions have been made.