Sarcasm Detection in News Headlines: Comparing Rule-Based, Embedding-Based, and LLM Methods

Deogyong Kim (2021134015)

1 Introduction

This project investigates sarcasm detection in news headlines. The task distinguishes satirical headlines from The Onion (a satirical news publication) from real news headlines from HuffPost. This task requires understanding the intent formed by word combinations, not just interpreting individual word meanings. This capability is not limited to news headline classification. It represents a core ability for machines to truly read and understand text rather than simply process patterns.

Prior work in sarcasm detection has evolved from manual feature engineering to deep learning approaches like CNNs [1] and attention-based models [2]. While recent studies utilize pre-trained encoders like BERT, this project extends the comparison to include generative Large Language Models (LLMs) to benchmark their inference capabilities against efficient embedding baselines.

Specifically, we implements and compares three approaches. First is rule-based baselines using keyword dictionaries. Second is embedding-based classifiers that capture semantic representations. Third is LLMs that have shown strong performance on various natural language tasks recently.

2 Task Definition

Task description: Classify news headlines into two categories:

- **Real**: Factual news from HuffPost
- **Satire**: Satirical headlines from The Onion

Input: A news headline (string, typically 5-20 words)
Output: Binary label (0 = Real, 1 = Satire)
Motivation: This task requires understanding meaning created by word combinations. Consider "Great job breaking it!" The words "great" and "job" are positive, but the sentence expresses harsh criticism. Word meanings change based on their combinations.

Such limitations cause real-world problems. When companies automatically analyze customer reviews, sarcastic complaints get misclassified as praise, corrupting the dataset. Customer service systems that fail to detect sarcasm cannot understand customer intent, leading to inappropriate responses.

Sarcasm detection is about understanding the true meaning of sentences beyond surface-level words. It requires capturing context, intent, and implicit meaning. This capability is fundamental for natural language understanding systems.

Success criteria: F1 score and accuracy on held-out test set.
F1 is preferred over pure accuracy as it balances precision and recall. This is important because labeling real news as satire and missing satire have different implications.

3 Methods

We implement five approaches across three paradigms.

3.1 Naive Baseline

Our baseline uses a scoring function with three components:

1. Intensifiers (+1.5 each): "absolutely", "totally", "perfect"
2. Satirical triggers (+2.0 each): "area man", "nation", "study finds"
3. Punctuation (+0.5 each): !, ?, or ""

We scan the headline and sum all matched scores. Headlines scoring ≥ 1.5 are classified as satire. This approach is naïve because it relies solely on word-level pattern matching without understanding semantic meaning. It is expected to fail on headlines that express sarcasm through semantic contradictions or absurdity without using explicit trigger words.

3.2 Embedding-based Classification

Both methods use Sentence-BERT (all-mpnet-base-v2) to encode headlines into 768-dimensional vectors. The pipeline consists of three stages:

1. **Preprocessing**: None (raw headlines used),
2. **Representation**: Sentence-BERT encoding to 768-dim vectors,
3. **Decision**: Centroid similarity or Logistic Regression.

Centroid Distance. We compute class centroids (mean vectors) from the training set. For each test headline, we compute cosine similarity to both centroids and assign it to the class with higher similarity. No training required.

Logistic Regression. We train sklearn's LogisticRegression (max_iter=1000) on the training set embeddings. The input is the 768-dimensional vector from Sentence-BERT. The output is a binary label (0=Real, 1=Satire). At test time, we encode the headline with Sentence-BERT and feed it to the trained classifier.

3.3 LLM-based Inference

We use Qwen2.5-32B-Instruct, selected as the largest dense (non-MoE) model that fits in a single RTX 3090 GPU. Dense models provide comprehensive world knowledge beneficial for understanding nuanced sarcasm.

Zero-shot. We provide a system prompt defining satire characteristics (absurd situations, "area man" phrases, mundane-as-news) and ask the model to classify each headline as "Satire" or "Real".

Few-shot. We add four manually-selected examples per class to the prompt. The model then classifies each test headline using the same output format.

4 Experiments

4.1 Dataset

Source: Sarcasm Headlines Dataset v2 [3], publicly available on Kaggle and GitHub.

Total size: 28,619 headlines

- Real (HuffPost): 14,985 (52.4%)
- Satire (The Onion): 13,634 (47.6%)

Split: 80/20 train-test split with fixed random seed (42)

- Training: 22,895 samples
- Test: 5,724 samples

Preprocessing: None. Headlines were used as-is. Sentence-BERT and LLM tokenizers handle encoding. Headlines average ∼11 words in both classes, so length alone cannot separate them.

4.2 Metrics

Primary metrics:

- **Accuracy**: Overall correctness
- **F1 Score**: Harmonic mean of precision and recall
- **Precision**: Correct predictions among all predicted satire
- **Recall**: Detected satire among all actual satire

F1 is our main metric because it balances false positives (labeling real news as satire) and false negatives (missing actual satire).

4.3 Overall Performance

4.3.1 Results

Table 1: Performance comparison across all methods

Method	Accuracy	F1	Precision	Recall
Naive Baseline	56.46%	0.24	0.71	0.15
Embedding + Centroid	74.72%	0.73	0.74	0.73
Embedding + LR	**84.71%**	**0.84**	**0.85**	0.83
LLM Zero-shot	81.87%	0.80	**0.85**	0.76
LLM Few-shot	82.95%	**0.84**	0.76	**0.94**

4.3.2 Analysis

The naive baseline achieves 56.46% accuracy with a distinctive pattern: high precision (0.71) but very low recall (0.15). This means it only predicts satire when obvious markers like "area man" or "report:" are present, missing most subtle cases. The low recall shows the fundamental limitation of rule-based approaches. They cannot understand semantic meaning beyond surface patterns.

Embedding-based methods show significant improvement. The centroid method achieves 74.72% accuracy without any training, simply by computing class centroids and measuring cosine similarity. The 18 percentage point improvement over naive baseline confirms that semantic embeddings capture meaningful patterns that keyword matching cannot. When we train a logistic regression classifier on these embeddings, performance jumps to 84.71% accuracy with balanced precision (0.85) and recall (0.83). This shows that learning a task-specific decision boundary in embedding space is highly effective.

LLM methods were expected to perform best given their strong performance on language understanding tasks. Zero-shot prompting achieves 81.87% accuracy. Few-shot prompting with 8 examples improves this to 82.95%, with very high recall (0.94) but lower precision (0.76). The LLM tends to over-predict satire, catching most actual satire but also mislabeling many real headlines.

Surprisingly, Embedding + LR outperforms the 32B parameter LLM Few-shot by 1.76 percentage points (84.71% vs 82.95%). This suggests that task-specific training on semantic embeddings can be more effective than general-purpose LLM inference for this dataset. Sentence-BERT already encodes rich semantic relationships, and the trained logistic regression learns dataset-specific decision boundaries that few-shot prompting cannot easily replicate.

4.3.3 Qualitative Analysis

Table 2: Distribution of prediction patterns across test set

Naive	Embedding + LR	LLM Few-shot	Percentage
✓	✓	✓	38.1%
✗	✓	✓	33.5%
✓	✓	✗	10.8%
✓	✗	✓	5.1%
✗	✗	✓	6.3%
✓	✗	✗	2.5%
✗	✓	✗	2.3%
✗	✗	✗	1.4%

To understand when each method succeeds or fails, we analyzed prediction patterns across the test set by comparing three methods: Naive Baseline, Embedding + LR, and LLM Few-shot. We present four representative cases that together cover 79% of the test set.

Case 1: Clear Satire (38.1%)

These headlines have obvious satirical markers that all methods detect. They typically contain explicit satire keywords ("report:", "area man", "nation"), absurd causal claims, or physically impossible scenarios. No semantic reasoning is needed.

Example:

> "report: majority of instances of people getting lives back on track occur immediately after visit to buffalo wild wings" (*Satire*)

Case 2: Semantic Understanding Required (33.5%)

These headlines lack obvious satirical keywords, requiring understanding of semantic contradictions. Naive baseline fails because it only checks surface patterns. Embedding and LLM detect category violations or incongruous word combinations. This is the most common failure mode for rule-based approaches.

Example:

> "eighth-grader drinks at twelfth-grade level" (*Satire*)

Case 5: Complex Reasoning Required (6.3%)

These headlines require multi-step reasoning about contradictions or escalating absurdity. Embedding captures semantic similarity but misses logical contradictions. Only LLM can reason about layered irony, quote-action contradictions, or absurdity escalation.

Example:

> "'just take it slow, and you'll be fine,' drunk driver assures self while speeding away in stolen police car" (*Satire*)

Case 8: Ambiguous Reality (1.4%)

These headlines are satirical but use factual reporting tone, or describe real events that sound absurd. All methods fail because the linguistic surface is indistinguishable from real news. These represent the fundamental limitation of text-only classification without source metadata.

Example:

> "fbi quickly follows up on tip about potentially dangerous man who killed 17 in school shooting" (*Satire*)

4.4 Computational Efficiency

4.4.1 Results

We measured average inference time per sample on a single GPU (RTX 3090) using 10 test samples after 1 warmup.

Table 3: Inference time comparison

Method	**Avg Time (ms)**	**Relative Speed**
Naive Baseline	0.030 ± 0.006	1× (baseline)
Embedding + Centroid	10.383 ± 2.017	351× slower
Embedding + LR	8.877 ± 0.058	301× slower
LLM Zero-shot	1398.769 ± 6.472	47,352× slower
LLM Few-shot	2017.215 ± 21.448	68,287× slower

4.4.2 Analysis

The efficiency difference is dramatic. LLM methods are approximately 227× slower than embedding-based methods and 68,287× slower than the naive baseline. Embedding + LR achieves the best trade-off: it matches LLM Few-shot's F1 score (0.84) while being 227× faster. This demonstrates that Sentence-BERT's semantic encoding is not only more accurate but also far more efficient than using large language models for this task.

5 Reflection and Limitations

This project revealed that defining "sarcasm" itself was the hardest challenge. This made selecting keywords for the naive baseline and writing prompts for LLM both difficult. Because of this definitional problem, our attempt with a SOTA-level NLI model achieved only ∼60% accuracy. This suggests that simply providing more examples to LLMs cannot solve the task when the underlying concept is fundamentally ambiguous. Real news can sound absurd, and satire can sound factual. While using a larger LLM might help, the embedding method already achieved 84.71% accuracy with much faster speed, making further LLM optimization unnecessary. F1 effectively measured overall quality, but aggregate metrics alone cannot capture distinct failure modes, which motivated our qualitative analysis.

Our qualitative analysis showed that each method's strengths cover different case types rather than forming subset relationships. Because only 1.4% of cases caused all three methods to fail, ensemble approaches could achieve significantly higher performance by routing cases appropriately. Beyond news headlines, this approach could improve systems that detect user intent, such as customer service applications identifying sarcasm or frustration.

Most surprisingly, the small embedding model outperformed the 32B LLM while being 227× faster. Despite LLMs being treated as universal solutions, task-appropriate models often perform better. Matching model capabilities to task requirements matters more than model size.

References

[1] Aniruddha Ghosh and Tony Veale. Fracking sarcasm using neural network. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 161–169, 2016.

[2] Avinash Kumar, Vishnu Teja Narapareddy, Veerubhotla Aditya Srikanth, Aruna Malapati, and Lalita Bhanu Murthy Neti. Sarcasm detection using multi-head attention based bidirectional LSTM. *IEEE Access*, 8:6388–6397, 2020.

[3] Rishabh Misra and Prahal Arora. Sarcasm detection using news headlines dataset. *AI Open*, 4:13–18, 2023.