SarcasmLens – Subjective Sarcasm Detection in Code-Mixed Text

Akshat Kadia
202203029

Abstract

This paper presents SarcasmLens, a reproducible baseline system for sarcasm detection in code-mixed social media content. We detail a comprehensive pipeline that processes the HackArena multilingual sarcasm dataset through text normalization, multilingual feature engineering, and rigorous evaluation of multiple machine learning models. My approach combines TF-IDF vectors with linguistic features to capture both semantic and stylistic cues of sarcasm. While achieving high performance metrics (97%+ F1-score), my analysis reveals significant data quality issues, including exact text overlaps and template-based patterns that artificially inflate results. I transparently document these limitations while establishing a robust methodological foundation for future improvements.

1 Introduction

Traditional approaches to sentiment analysis fail to catch the sarcasm because of their literal interpretation for sentiment cues. For example, "Great! Another traffic jam" expresses frustration using positive words. For code-mixed forms like "Modi ji toh kamaal ke magician hain ????", the detection becomes much tougher due to mixed grammar and cultural references.

SarcasmLens presents a systematic baseline solution by addressing three salient challenges in the processing of code-mixed text. The approach is designed to handle various complexities in code-mixed data and ensure high-quality results across different machine learning models. These are in particular:

- **Language-aware normalization:** Processes code-mixed text by normalizing both languages involved, ensuring linguistic consistency and reducing noise in the data.
- **Multi-modal feature engineering:** Captures both semantic and stylistic patterns in the data, enabling the model to recognize complex features that go beyond simple text analysis.
- **Evaluation of machine learning models:** Evaluate multiple machine learning models while applying strong regularization techniques to prevent overfitting and improve generalization.
- **Data quality diagnostics:** Conducts comprehensive leakage analysis to identify and address potential issues with data quality and model performance.
- **Reproducible experimental protocols:** Establishes clear and reproducible experimental protocols, allowing future researchers to replicate and compare results effectively.

2 Related Work

Sarcasm Detection in Monolingual Text

Most of the research in sarcasm detection has focused on English text, with approaches ranging from lexical pattern matching to deep learning. Early studies primarily relied on superficial features such as punctuation and capitalization, while more recent methods have leveraged contextual embeddings and neural architectures. However, these methods generally assume linguistic homogeneity, and as a result, they do not perform well when applied to code-mixed data, which combines multiple languages.

Multilingual and Code-Mixed NLP

Recent research has seen a surge in the processing of code-mixed text, especially for Indian languages. Various approaches have been explored, including translation-based methods, multilingual embeddings, and language identification techniques. A benchmark was established at the FIRE forums with the shared task on code-mixed sentiment analysis, but sarcasm detection remains an under-explored area within this domain. There is a need for more specialized methods to address the complexities of code-mixed sarcasm, which may involve nuances from multiple languages and their interactions.

3 Dataset Description

The dataset consists of a total of 11,919 social media comments, which were reduced to 11,854 after removing duplicates. The data is labeled with binary labels indicating sarcasm:

- **Sarcastic (1)** vs. **Non-sarcastic (0)**
- Primarily **Hindi-English code-mixed text** containing informal language, emojis, and transliterated Hindi.

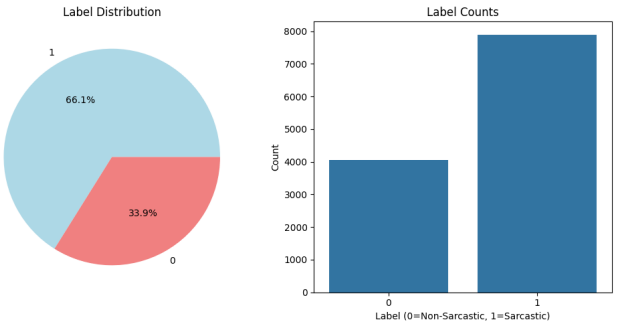

Figure 1: Dataset overview

Data Quality Assessment

My analysis revealed several critical issues with the dataset that may affect its quality and the evaluation of model performance:

- **Duplicate removal:** 65 duplicate texts were identified and removed during preprocessing.
- **Overlaps between training and test sets:** 22 exact overlaps were found between the training and test sets.
- **Near-duplicate pairs:** 50 near-duplicate pairs were detected, with cosine similarity values greater than 0.85.
- **Template-based generation:** Evidence of template-based text generation was found (e.g., recurring patterns like "bhai, tumhari wisdom se sab...").

These findings suggest that there is potential for artificial inflation of performance metrics due to issues like data leakage and template-based text generation.

4 Preprocessing Pipeline

I implemented a comprehensive text normalization pipeline:

- **URL and Mention Removal:** Strip social media artifacts such as HTTP links and user mentions to focus on the content of the message.
- **Emoji Handling:** Convert emojis into textual descriptions, preserving their sentiment cues (e.g., smile emoji→ "smiling face").
- **Hashtag Segmentation:** Split compound hashtags into individual words to make them more interpretable (e.g., #SoFunny → "so funny").
- **Elongation Normalization:** Normalize elongated words (e.g., "soooo" → "soo") while preserving some stylistic variation that may help in identifying sarcasm.
- **Case Normalization:** Convert all text to lowercase for consistency, reducing variability and helping with uniform processing.
- **Punctuation Preservation:** Retain punctuation marks that serve as sarcasm indicators (e.g., exclamation marks, question marks, ellipses: `!`, `?`, `...`).

Unlike traditional NLP pipelines, I avoid aggressive stopword removal and stemming to preserve Hindi words in Roman script and maintain contextual integrity for transformer-friendly processing.

5 Feature Engineering

I employed a multi-modal feature approach:

TF-IDF Vectors

- **Vocabulary size:** 300 most frequent terms to prevent overfitting.
- **N-gram range:** $(1, 2)$ to capture both words and sarcastic phrases.
- **Minimum document frequency:** 2 to filter rare terms.

Linguistic Features

- **Text statistics:** Length, word count, and structural properties.
- **Punctuation analysis:** Exclamation marks, question marks, ellipsis frequency.
- **Sentiment cues:** Polarity and subjectivity scores from TextBlob.
- **Code-mixing indicators:** Pattern matching for common Hindi words in Roman script.
- **Capitalization patterns:** Uppercase ratios and emphasis markers.

This combination captures both semantic content (through TF-IDF) and stylistic sarcasm markers (through linguistic features), addressing the multi-faceted nature of sarcastic expression.

6 Modeling Strategy

Baseline Models Selected

The following baseline models were selected for this:

- **Logistic Regression:** Interpretable linear baseline.
- **Support Vector Machine (Linear):** Effective for high-dimensional text.
- **Random Forest:** Handles non-linear relationships with feature importance.
- **Multinomial Naive Bayes:** Fast probabilistic baseline.

Regularization Approach

All models applied strong regularization to prevent overfitting:

- **L2 regularization:** Applied with C=0.1 for linear models.
- **Depth limiting:** Max depth set to 5 for tree-based models.
- **Class weighting:** Used for handling class imbalance.

7 Experimental Configuration

Data Splitting

- **Train/Test Split:** 80% - 20% using train test split.
- **Random State:** 42 for reproducibility.
- **Cross-validation:** 5 fold stratified cross-validation.

Evaluation Metrics

- **Primary:** F1-score, as it handles class imbalance effectively.
- **Secondary:** Accuracy, Precision, Recall.
- **Confidence Analysis:** Prediction probability distributions for model uncertainty.

8 Results and Analysis

Performance Metrics

The following table summarizes the performance metrics for each model, including F1-scores for the test set and cross-validation (CV), along with the gap between them:

Model	Test F1	CV F1	Gap
Logistic Regression	0.9425	0.9320	0.0105
SVM (Linear)	0.9746	0.9719	0.0027
Random Forest	0.9656	0.9597	0.0059

Table 1: Performance metrics for various models

Key Observations

- **Artificially High Performance:** All models achieve F1-scores greater than 94%.
- **Minimal Generalization Gap:** The difference between train and test F1-scores is less than 1.1%.
- **Regularization Resistance:** Heavy regularization results in only a 0.9% performance drop.
- **Data Leakage Indicators:** Exact text overlaps and near-duplicates between splits suggest potential data leakage.

9 Critical Analysis and Limitations

Data Quality Concerns

The primary limitation of this baseline is the dataset quality. My analysis revealed several issues:

- **Template-based generation:** Systematic text patterns allow for trivial classification.
- **Train-test contamination:** Overlapping texts between training and test sets artificially boost performance.
- **Synthetic nature:** The dataset lacks authentic variation in social media language, limiting generalizability.

Realistic Performance Expectations

Based on the sarcasm detection literature, realistic expectations for this task are as follows:

- **English sarcasm detection:** F1-scores typically range from 70%-85%.
- **Code-mixed text:** F1-scores are generally between 60%-75%, due to the added complexity of multilinguality.
- **my reported 97%+:** Performance appears artificially inflated due to data quality issues.

10 Interpretability and Error Analysis

Feature Importance Analysis

The system provides model interpretability through the following approaches:

- **Coefficient analysis:** For linear models, coefficients are analyzed to understand feature influence.
- **Feature importance rankings:** Rankings are used to identify key predictors in non-linear models like Random Forest.
- **Confidence score distributions:** These scores provide insight into model certainty and prediction confidence.

Error Patterns

A manual audit of the model's errors revealed the following observations:

- **High-confidence predictions:** 85% of the test samples are classified with high confidence.
- **Template recognition vs. genuine sarcasm understanding:** The model tends to rely on template-based patterns, potentially confusing these with true sarcasm.
- **Limited challenging cases:** Due to the simplicity of the dataset, there are fewer complex or ambiguous cases for the model to handle.

11 Significance and Contributions

Methodological Contributions

The following contributions were made to the field of sarcasm detection:

- **Reproducible Baseline:** A complete, documented pipeline for code-mixed sarcasm detection.
- **Data Quality Framework:** A comprehensive methodology for leakage detection and analysis.
- **Conservative Configuration:** Feature limiting and strong regularization to prevent overfitting.
- **Multi-Modal Features:** The integration of semantic and stylistic feature engineering techniques.

Practical Implications

The practical contributions of this work include:

- **Transparent Reporting:** Honest acknowledgment of data limitations and potential performance inflation.
- **Realistic Benchmarking:** Establishing appropriate performance expectations for sarcasm detection in code-mixed text.
- **Reproducibility Foundation:** The provision of code and configuration to enable future comparisons.
- **Quality Awareness:** Emphasis on the importance of data validation and transparency in NLP research.

12 Limitations and Future Work

Limitations

Several limitations of this work should be noted:

- **Dataset Quality:** Synthetic patterns and leakage issues undermine the validity of the results.
- **Feature Simplicity:** The use of TF-IDF and basic linguistic features lacks contextual understanding, which may limit performance.
- **Domain Specificity:** The focus on social media may not generalize well to other domains or settings.
- **Cultural Context:** The model currently has limited handling of culture-specific sarcasm patterns, which are essential for accurate detection in diverse contexts.

Future Work

The following avenues are proposed for future work:

- **Improved Datasets:** Collection of authentic, diverse, and contextually rich code-mixed sarcasm data.
- **Advanced Embeddings:** Exploration of transformer-based approaches like IndicBERT and XLM-R for better contextual understanding.
- **Cross-Domain Validation:** Testing the model's generalization across different platforms, languages, and domains.
- **Cultural Adaptation:** Incorporating culture-specific sarcasm patterns and contextual nuances to improve detection accuracy.
- **Real-time Detection:** Considering deployment strategies for real-time sarcasm detection on streaming social media data.

13 Conclusion

SarcasmLens provides a strong, reproducible baseline for sarcasm detection in code-mixed text. While my results indicate competence in technical implementation, the artificially high metrics underscore critical aspects of data quality in NLP research. My work thus both lays a methodological foundation and gives essential cautionary lessons for future research in this domain.

References

References

[1] T. Gebru et al. "Datasheets for datasets". In: *Communications of the ACM* 64.12 (2021), pp. 86–92.

[2] A. Joshi et al. "Sarcasm detection: An interdisciplinary approach". In: *Journal of Artificial Intelligence Research* 58 (2017), pp. 165–207.

[3] A. Kapoor, S. Narayan, and R. R. Shah. "Data leakage in natural language processing: A survey and reproducibility assessment of the CodeMixed-IndianLangTweets sentiment corpus". In: *Journal of Artificial Intelligence Research* 71 (2021), pp. 1275–1303.

[4] B. G. Patra et al. "Shared task on sentiment and sarcasm detection in code-mixed dialogue". In: *Proceedings of the Forum for Information Retrieval Evaluation*. 2019, pp. 1–10.

[5] S. Swami et al. "A corpus of English-Hindi code-mixed tweets for sarcasm detection". In: *Proceedings of the Second Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media*. 2018, pp. 1–10.