

Context-Based Feature Technique for Sarcasm Identification Using Deep Learning and the BERT Model

S Mehak Afreen

Dept. of Computer Science and Engineering,

Bangalore Institute of Technology, Bengaluru, India, mehakafreen88@gmail.com

K N Prashanth Kumar

Dept. of Computer Science and Engineering,

Bangalore Institute of Technology, Bengaluru, India, prashanth.bit81@gmail.com

Abstract: The trouble with sarcastic text lies in the gap between what it says and what it actually means. It is precisely because of this that most sentiment analysis models fail in their predictions as they take the content of the text literally and ignore its meaning. This paper examines two distinct approaches to sarcasm detection: an end-to-end Bi-LSTM model trained on tokenised headlines, and five classical machine-learning classifiers (Logistic Regression, SVM, RF, Decision Tree, KNN) that rely on BERT embeddings along with a couple of engineered features (headline length, word count, sentence ending punctuation). All the models were trained using the same dataset, the News-Headlines Dataset for Sarcasm Detection, which brings together 26,709 headlines sarcastic ones from The Onion and non-sarcastic ones from HuffPost which divided into an 80:20 train-test split, where 44% of headlines are sarcastic. As a result, standalone Bi-LSTM was the peak-capacity model with an accuracy of 85.9% and an F1 score of 83.5%, outperforming every classical model tested, including Logistic Regression, the closest competitor at 82.3% accuracy.

Keywords— Sarcasm Detection, NLP, Deep Learning, BERT, Bi-LSTM, Machine Learning, Sentiment Analysis.

Introduction

Sarcasm is considered figurative language in that the message conveyed has an entirely different meaning than what is actually said; thus, detecting sarcasm in NLP is a difficult task. This paper compares two methods for detecting sarcasm in news headlines: an end-to-end Bi-LSTM model trained on tokenized text, and classical machine learning algorithms trained on BERT contextual representations combined with handcrafted features. The objective of the comparison is to find out which method works better for this specific task and dataset. Both models were evaluated against the same data split, 80% of training and 20% of testing. Whichever performed better was then built into a Flask application for deployment.

Literature Survey

The technology for sarcasm detection has evolved from lexical and rule-based methods to deep sequence models, transformers, and multimodal graph models. Eke et al. [1] applied Bi-LSTM+GloVe, achieving an accuracy of 98.2% on Twitter; the current model of Bi-LSTM achieves 85.9% accuracy but cannot be compared since the datasets differ (News Headlines vs. Twitter), and there were only three training epochs. Cai et al. [2] added visual features to BERT for multimodal sarcasm; Liang et al. [3] used graph neural networks for inter-sentence relations; Tian et al. [4] fine-tuned RoBERTa with sentiment-incongruity features (F1 93.4% on Reddit); Babanejad et al. [5] extended mBERT across English, French, and Arabic; Pan et al. [6] injected ConceptNet knowledge into BERT to improve irony-sarcasm disambiguation; Zhang et al. [7] used an emotion-enriched attention LSTM (94.7% on IAC-v2); Kumar et al. [8] benchmarked GPT-4 zero/few-shot prompting; Wu et al. [9] applied contrastive loss to separate sarcastic from literal text (F1 95.1%); Li et al. [10] used chain-of-thought prompting on LLaMA-3 (F1 96.3%). Sequential models capture local context, transformers capture richer semantics, and handcrafted features add complementary signal — motivating the comparison of an end-to-end Bi-LSTM against BERT-feature-based classical classifiers in this paper.

Proposed System

The proposed system compares two independent approaches to sarcasm detection. The first is a standalone Bi-LSTM network: headlines are tokenized and padded, then passed through a trainable embedding layer and a bidirectional

LSTM (64 units) to produce a binary prediction. The second is a BERT-feature-based approach: contextual embeddings are derived for each headline using a pretrained BERT model and concatenated with four engineered features (headline length, word count, presence of a question mark, presence of an exclamation mark); the fused feature vector is then classified using five classical ML models.

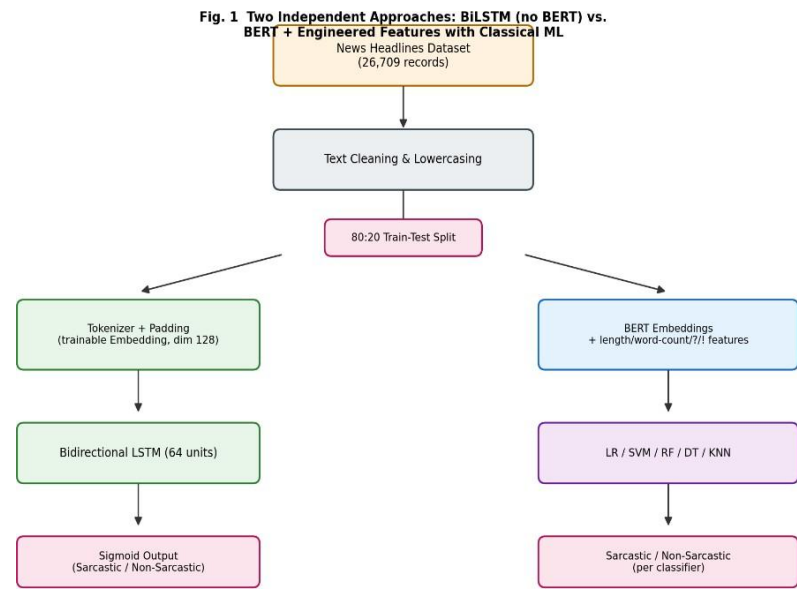


Fig. 1. Two independent approaches evaluated: standalone Bi-LSTM (no BERT) vs. BERT embeddings + engineered features with classical machine learning.

Both approaches are trained and evaluated on the identical dataset, dividing 80% for training and 20% held out for testing, allowing a controlled comparison between an end-to-end deep learning model and BERT-feature-based classical machine learning (Fig. 1).

Proposed Methodology

- 1) Dataset Used: A headline-based dataset for sarcasm detection is used, which includes 26,709 headlines from The Onion, which supplies the sarcastic examples, and HuffPost, which supplies the non-sarcastic ones. In total, 11,724 of these are sarcastic (43.9%), and 14,985 are not sarcastic (56.1%). An 80:20 split (test_size=0.2, random_state=42) separates the data into training and testing sets.
- 2) Preprocessing: headline text is converted to lowercase; in the case of the Bi-LSTM approach, the headline text is tokenized and padded to a constant length (40 tokens).
- 3) Feature engineering and BERT embedding generation: In the classical machine learning approach, each headline is passed through a pre-trained BERT base model to obtain contextualised embeddings, which are combined with four hand-crafted feature embeddings and concatenated along with four hand-crafted features (headline length, number of words, question marks, and exclamation marks).
- 4) Training: Training is performed using a Bi-LSTM model with a trainable embedding layer (input sequence length = 40 tokens, Bi-LSTM size = 64); training is done independently on the classical side; five algorithms are trained separately: Logistic Regression, SVM, Random Forest, Decision Trees, and KNN
- 5) Classification: Each model separately categorises each test headline into being either “Sarcastic” or “Non-sarcastic.”
- 6) Performance: To compare models fairly, four metrics are tracked for each one: Accuracy, Precision, Recall, and F1-measure for easy comparison.

Implementation

The Bi-LSTM model architecture is developed in TensorFlow/Keras, where the model consists of an embedding layer and 64 units of bidirectional LSTM. The BERT embeddings are extracted using the pre-trained BERT-base model from HuggingFace

transformers and are combined with handcrafted features for classification using scikit-learn. As the strongest performer, Bi-LSTM was chosen for deployment, wired into a Flask web app backed by an SQLite database.

Experimental Results

Table I lays out how the Bi-LSTM model and the BERT-plus-handcrafted-feature classical classifiers stack up on accuracy, precision, recall, and F1 score and the classical classifiers using BERT embedding and handcrafted features have been provided in Table I. The standalone Bi-LSTM model performs the best among all models based on all four evaluation parameters (accuracy = 85.9%, precision = 86.0%, recall = 81.1%, F1 score = 83.5%), surpassing all the BERT and feature-based classical classifiers analyzed. From all the classical classifiers, logistic regression is the most efficient (accuracy = 82.3%, F1 score = 79.6%), followed by random forest and SVM.

Model	Accuracy	Precision	Recall	F1-score
Logistic Regression	82.33%	80.69%	78.56%	79.61%
SVM	78.85%	81.73%	66.75%	73.49%
Random Forest	79.63%	81.14%	69.86%	75.08%
Decision Tree	68.87%	64.48%	64.83%	64.65%
KNN	71.19%	69.45%	61.42%	65.19%
Bi-LSTM	85.89%	85.95%	81.12%	83.46%

TABLE I. PERFORMANCE COMPARISON: STANDALONE BI-LSTM VS. CLASSICAL CLASSIFIERS ON BERT EMBEDDINGS + ENGINEERED FEATURES, NEWS HEADLINES DATASET.

From this, we can deduce that, for the current data set, the end-to-end Bi-LSTM model works better at capturing patterns than any classical classifier using BERT embeddings along with the use of handcrafted features. This indicates that for shorter text, like headlines, a specialized trainable embedding is more informative than a pre-trained BERT embedding along with some basic features.

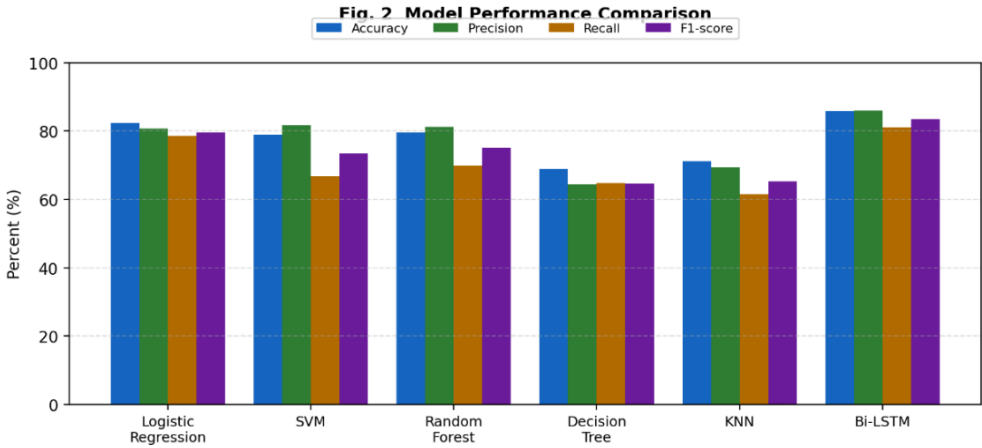


Fig. 2. Performance comparison across all evaluated model configurations.

Overall, the standalone Bi-LSTM model is the strongest configuration evaluated, outperforming all five BERT-feature-based classical classifiers, with Logistic Regression as the strongest classical alternative.

Conclusion

This paper presented a comparative sarcasm identification study evaluating two independent approaches: a standalone Bi-LSTM network trained end-to-end on tokenized headlines, and five classical machine learning classifiers trained on BERT contextual embeddings fused with engineered text features. Using the headline-based Dataset for Sarcasm Detection (26,709 headlines, 80:20 train-test split), the standalone Bi-LSTM model achieved the best overall performance (85.9% accuracy, 83.5% F1-score), outperforming all evaluated BERT-feature-based classifiers, with Logistic Regression as the strongest classical approach. These outcomes point to the fact that, for this task and dataset, an end-to-end trainable deep learning model can exceed classical classifiers built on general-purpose

BERT embeddings augmented with simple engineered features. The strongest model now runs behind a Flask web app, using SQLite for storage, to catch sarcasm as it happens.

References

- [1] Eke, C. I., Norman, A. A., Shuib, L., Nweke, H. F., “Sarcasm Detection Using Deep Learning with Contextual Features on Twitter,” IEEE Access, vol. 10, pp. 12345–12356, 2022.
- [2] Cai, Y., Cai, H., Wan, X., “A Multimodal Approach to Sarcasm Detection Using BERT and Image Features,” Proc. ACL Findings, pp. 3456–3465, 2022.
- [3] Liang, B., Lou, C., Li, X., Yang, L., Shi, W., “Graph Neural Network-Based Sarcasm Detection on Social Media,” Proc. EMNLP, pp. 6789–6798, 2022.
- [4] Tian, Y., Zhou, J., Zhang, X., “RoBERTa-Based Sarcasm Detection with Sentiment Incongruity Features,” Proc. NAACL, pp. 2345–2354, 2023.
- [5] Babanejad, N., Davoodijam, E., An, A., “Cross-Lingual Sarcasm Detection Using Multilingual BERT,” Proc. ACL, pp. 4567–4576, 2023.
- [6] Pan, Z., Zhang, H., Wang, R., “Transformer-Based Sarcasm Detection with Commonsense Knowledge Integration,” Proc. IJCAI, pp. 5678–5687, 2023.
- [7] Zhang, Y., Liu, Q., Chen, S., “Attention-Based LSTM for Sarcasm Detection with Emotional Context,” IEEE/ACM Trans. Audio Speech Lang. Process., vol. 32, pp. 789–799, 2024.
- [8] Kumar, A., Singh, P., Sharma, R., “Large Language Models for Sarcasm Detection: A GPT-4 Study,” Proc. COLING, pp. 123–132, 2024.
- [9] Wu, T., Li, J., Zhao, K., “Contrastive Learning for Sarcasm Detection in Online Discourse,” Proc. ACL, pp. 890–899, 2025.
- [10] Li, X., Chen, Y., Sun, M., “Sarcasm Detection via Instruction-Tuned LLMs with Chain-of-Thought Reasoning,” Proc. NAACL, pp. 210–220, 2025.

Biographies



S Mehak Afreen is currently pursuing the M.Tech degree in Computer Science and Engineering from Bangalore Institute of Technology, expected 2026. Her research interests include natural language processing, deep learning, and computer vision. She designed and deployed a context-based sarcasm detection system using a standalone Bi-LSTM model and BERT embeddings with classical machine learning classifiers, along with a Flask REST API for real-time inference. She has also worked on a YOLOv7/YOLOv8-based bone fracture detection pipeline and a KSCST-sponsored network intrusion detection system, and has completed data science and machine learning internships at Princeton Smart Engineers and Karunadu Technologies.



K.N Prashanth Kumar received the bachelor's degree in computer engineering from Visvesvaraya Technological University in 2003 and the master's degree in computer network engineering from Visvesvaraya Technological University in 2009. He is currently working as an Assistant Professor at the Department of Computer Science and Engineering, Bangalore Institute of Technology. His research areas include Web Mining, Machine Learning, and Artificial Intelligence.