

Article

Unpacking Sarcasm: A Contextual and Transformer-Based Approach for Improved Detection

Parul Dubey ^{1,*}, Pushkar Dubey ² and Pitshou N. Bokoro ^{3,*}

¹ Symbiosis Institute of Technology, Symbiosis International (Deemed University), Nagpur Campus, Pune 412115, India

² Department of Management, Pandit Sundarlal Sharma (Open) University Chhattisgarh, Bilaspur 495009, India; drdubeypkag@gmail.com

³ Department of Electrical Engineering Technology, University of Johannesburg, Johannesburg 2092, South Africa

* Correspondence: parul.dubey@sitnagpur.siu.edu.in (P.D.); pitshoub@uj.ac.za (P.N.B.)

Abstract: Sarcasm detection is a crucial task in natural language processing (NLP), particularly in sentiment analysis and opinion mining, where sarcasm can distort sentiment interpretation. Accurately identifying sarcasm remains challenging due to its context-dependent nature and linguistic complexity across informal text sources like social media and conversational dialogues. This study utilizes three benchmark datasets, namely, News Headlines, Mustard, and Reddit (SARC), which contain diverse sarcastic expressions from headlines, scripted dialogues, and online conversations. The proposed methodology leverages transformer-based models (RoBERTa and DistilBERT), integrating context summarization, metadata extraction, and conversational structure preservation to enhance sarcasm detection. The novelty of this research lies in combining contextual summarization with metadata-enhanced embeddings to improve model interpretability and efficiency. Performance evaluation is based on accuracy, F1 score, and the Jaccard coefficient, ensuring a comprehensive assessment. Experimental results demonstrate that RoBERTa achieves 98.5% accuracy with metadata, while DistilBERT offers a 1.74x speedup, highlighting the trade-off between accuracy and computational efficiency for real-world sarcasm detection applications.



Academic Editor: Ming Liu

Received: 17 February 2025

Revised: 25 February 2025

Accepted: 27 February 2025

Published: 6 March 2025

Citation: Dubey, P.; Dubey, P.; Bokoro, P.N. Unpacking Sarcasm: A Contextual and Transformer-Based Approach for Improved Detection. *Computers* **2025**, *14*, 95. <https://doi.org/10.3390/computers14030095>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Sarcasm detection is of great significance in natural language processing (NLP), which mainly lies in sentiment analysis and opinion mining. One common feature of sarcasm is irony—the incongruity of the literal and intended effect of text—and it is considered a major problem for the text classifier systems [1]. User-generated content, which is usually informal and context-dependent, has complicated sarcasm detection even more with the advent of social media sites, such as Twitter, Reddit, and online forums [2]. Sarcasm classification surrounding traditional techniques primarily utilized domain knowledge-oriented, rule-based approaches, and hand-crafted features, but notable performance improvements were recently achieved using deep learning architectures [3], including transformer-based structures.

Various approaches have been used throughout the years to be able to determine whether a comment is sarcastic or not. These methods use interaction history and some

metadata for classification [4]. In addition, the presence of media has led to successful methods that work on multi-modal approaches, involving text, images, and audio [5]. However, as the uses of language across this dataset become more varied, it also becomes infinitely more difficult to characterize expressions of sarcasm as separate from “real” statements. In fact, several potential candidates, including attention-based mechanisms, contextual embeddings, and summarization methods can help solve some of the problems mentioned [6].

The novelty of this research lies in the development of an optimized sarcasm detection pipeline that integrates context summarization and metadata-based feature enhancement, enabling improved interpretability and computational efficiency. Unlike prior studies, this research provides a systematic comparison of RoBERTa and DistilBERT, offering insights into the trade-offs between accuracy and processing time in transformer-based sarcasm detection, which has not been extensively explored in previous studies. The abstract figure can be seen in Figure 1. This study makes three key technical contributions:

- Context-aware preprocessing: A novel preprocessing pipeline integrating context summarization and metadata-based embeddings to enhance sarcasm detection accuracy and robustness.
- Comparative model analysis: A systematic evaluation of RoBERTa and DistilBERT, demonstrating 98.5% accuracy for RoBERTa and a 1.74x speedup with DistilBERT, highlighting trade-offs between accuracy and computational efficiency.
- Computational efficiency optimization: A detailed computational analysis proving that DistilBERT significantly reduces training time while maintaining competitive accuracy, making it ideal for real-time sarcasm detection in sentiment analysis applications.

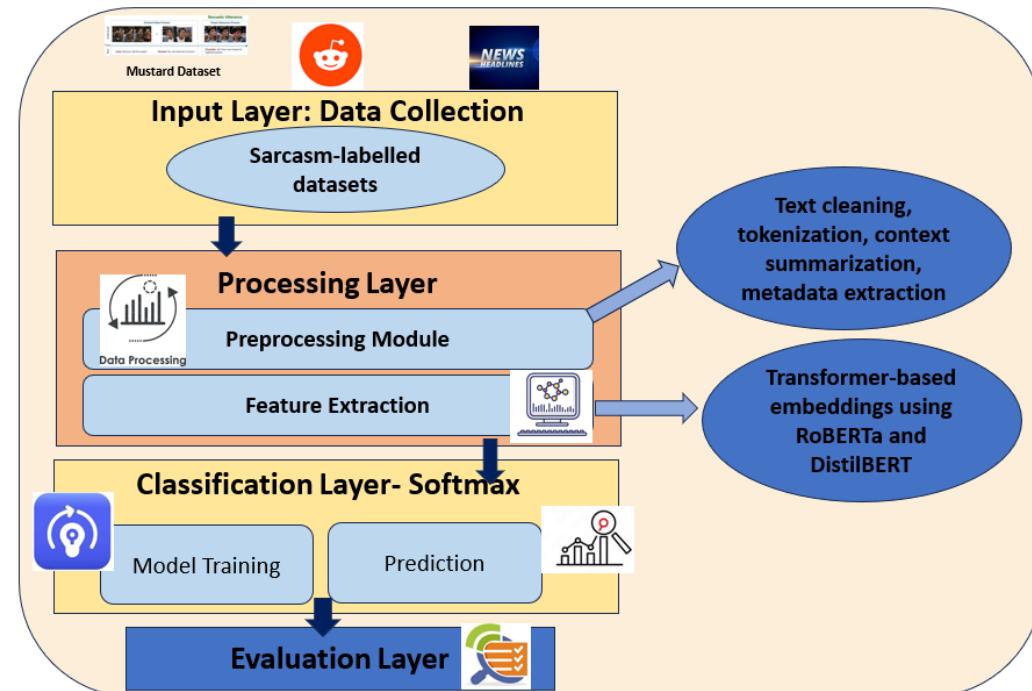


Figure 1. Figure of Abstract.

2. Literature Review

Sarcasm detection has been extensively explored in various domains, including social media analysis, computational linguistics, and sentiment analysis. The approaches evolved from traditional rule-based methods to machine learning (ML) models and, more recently,

deep learning (DL) techniques. This section provides a structured review of these developments, highlighting key contributions and identifying existing gaps in Proceedings of the literature.

2.1. Traditional Approaches: Rule-Based and Feature-Engineered Methods

Sarcasm detection has been researched in various domains, including social media analysis, computational linguistics, etc. The early studies mostly used rule-based and feature-engineered approaches based on the identification of sentiment shifts and syntactic structures for sarcasm detection [7]. But these approaches had limitations in scaling and adapting to different conversational contexts. However, the addition of machine learning models, primarily support vector machines (SVMs) and decision trees, offered better results in sarcasm detection but failed to consider context [8].

Techniques such as logistic regression, random forest, decision trees, and support vector machines (SVMs) have been used for sarcasm detection. These models often rely on feature extraction methods like term frequency-inverse document frequency (TF-IDF) and lexicon-based features [9,10].

2.2. Advances in Deep Learning for Sarcasm Detection

Combining multiple ML models can enhance performance. For instance, ensemble models have shown better results on smaller, imbalanced datasets compared to individual models [11,12]. Fine-tuning large language models (LLMs) like BERT with sarcastic datasets can significantly improve performance, making them suitable for more open and practical applications [13]. Developing effective sarcasm detection models for low-resource languages like Urdu involves creating specialized datasets and leveraging advanced DL techniques [14].

Deep learning techniques continue to be a focal point in sarcasm detection research, with various models and architectures being explored to enhance detection capabilities. This consistent interest highlights the ongoing efforts to refine and optimize deep learning approaches for better performance [15–17]. The use of deep learning models, especially recurrent neural networks (RNNs) and convolutional neural networks (CNNs), represented a significant progression in sarcasm detection as they are capable of capturing sequential dependencies and textual patterns [18].

CNNs, when combined with RNNs, can improve the accuracy of sarcasm detection by capturing both local and sequential features of the text [19]. Nevertheless, transformer-based architectures have changed the game with their ability to use efficient representation learning through a self-attention mechanism [20]. Of these, BERT and its variants, like RoBERTa and DistilBERT, have shown state-of-the-art performance on sarcasm classification problems by using contextual embeddings [21].

2.3. Context-Aware and Multi-Modal Approaches

Conversational context has been identified as an important factor for sarcasm detection in several studies [4]. Context-aware models are also important as they utilize past utterances to disambiguate sarcastic statements. In a similar vein, multi-modal approaches with visual and auditory clues have proven useful in improving sarcasm detection performance [5]. These approaches work well in situations where there are indications of sarcasm that go beyond textual data [22].

Sarcasm detection performance is heavily influenced by preprocessing techniques. Tokenization, context summarization, and the incorporation of metadata have all been shown to significantly improve classification accuracy. The former two methods achieve this by reducing noise in the data, while the latter focuses on ensuring the features being passed to the model are representative of the content [23]. Nayak and Bolla [3] and Zhang

et al. [7] demonstrated the effectiveness of having context in sarcasm detection models. Moreover, an intermediate-task transfer learning approach was proposed for sarcasm detection, achieving remarkable results [4].

Recent studies have also assessed varying machine learning models for sarcasm detection. Furthermore, as highlighted in [18], comparative analyses have shown that transformer-based architectures outperform traditional models, such as support vector machines (SVMs) and convolutional neural networks (CNNs), in regard to sarcasm classification tasks. However, with transformer models demanding significant computational power [24], efficiency is still an important consideration. DistilBERT provides an efficient solution by reducing model complexity while still maintaining performance comparable to much larger transformer models [20].

The consistent presence of multimodal sarcasm detection research indicates a sustained interest in leveraging multiple data types (e.g., text, images, audio) to improve sarcasm detection accuracy. This approach combines various modalities to capture the nuanced and often context-dependent nature of sarcasm [25,26]. The use of contextual and linguistic features remains a consistent theme in sarcasm detection research. This approach focuses on understanding the context and specific linguistic markers that indicate sarcasm, which is crucial for accurate detection [27,28].

This review highlights the advancements in sarcasm detection methodologies, emphasizing the role of context-aware models, deep learning techniques, and preprocessing strategies. The integration of multi-modal features, transfer learning, and optimized transformer architectures represents a promising direction for future research in sarcasm detection.

Despite advancements in sarcasm detection using deep learning, existing models struggle with contextual ambiguity, computational efficiency, and real-world applicability. Traditional machine learning models fail to capture deep contextual nuances, while transformer-based models, despite their effectiveness, require extensive computational resources. Additionally, many sarcasm detection frameworks lack structured preprocessing techniques, leading to suboptimal performance in real-world applications. This research aims to bridge these gaps by integrating context summarization, metadata-based feature enhancement, and computational efficiency optimization.

Current sarcasm detection approaches predominantly focus on textual cues, neglecting the role of contextual summarization and metadata-driven embeddings in enhancing model performance. While transformer models such as RoBERTa and DistilBERT improve sarcasm classification accuracy, their computational demands make them impractical for real-time applications. Furthermore, previous studies lack a comparative analysis of the accuracy–performance trade-offs between these models. This research addresses these gaps by proposing a novel preprocessing pipeline, systematically comparing RoBERTa and DistilBERT, and optimizing computational efficiency for scalable sarcasm detection.

3. Problem Statement and Research Gaps

Because sarcasm relies heavily on context, tone of voice, and implicit meaning, sarcasm detection is widely regarded as one of the most challenging tasks in natural language processing (NLP). Although RoBERTa and DistilBERT enhance the capability to classify sarcasm, challenges persist in contextual comprehension, computational efficiency, and real-world implementation. However, current models designed to maximize sarcasm detection in ‘person’ datasets require significant training data and do not provide preprocessing techniques like context summarization, metadata-based embeddings, etc. Moreover, most studies do not evaluate transformer models against each other in terms of accuracy vs. efficiency, which limits their practical deployment.

Research Gaps

- Limited contextual understanding: Existing models lack multi-turn conversation modeling and effective context summarization techniques.
- High computational cost: Transformer-based sarcasm detection remains resource-intensive, limiting real-time applications.
- No systematic model comparison: Studies rarely evaluate RoBERTa vs. DistilBERT for accuracy vs. efficiency trade-offs.
- Lack of metadata utilization: Speaker details, sentiment shifts, and contextual embeddings remain underexplored in sarcasm classification.
- Underdeveloped multi-modal approaches: Most models focus only on text, ignoring audio and image-based sarcasm cues.

4. Dataset

The datasets employed in this research were meticulously chosen to evaluate the proposed sarcasm detection methodology across diverse domains. The News Headlines dataset, comprising 26,709 records, includes sarcastic and non-sarcastic headlines from sources like The Onion and HuffPost. This dataset is enriched with metadata such as article descriptions, authors, and sections, providing contextual cues critical for sarcasm detection [29,30]. The Mustard dataset, featuring 1202 conversational samples from sitcoms such as Friends and The Big Bang Theory, offers rich contextual and emotional cues by including dialogue, speaker details, and scene-level context. Finally, the Reddit (SARC) dataset contains approximately 1.3 million labeled comments, capturing real-world conversational context from social media platforms. For this study, a subset of 7370 records was curated for validation. These datasets collectively encompass a wide range of textual and contextual variations, enabling a robust evaluation of the sarcasm detection models and their applicability to real-world scenarios. Table 1 shows the dataset description. Table 2 shows the sample dataset.

Table 1. Dataset description.

Feature	News Headlines	Mustard	Reddit (SARC)
Total Records	26,709	1202	~1,300,000
Sarcastic Sentences	Approximately 47%	Balanced per speaker	Not specified
Non-Sarcastic Sentences	Approximately 53%	Balanced per speaker	Not specified
Ratio (Sarcastic:Non-Sarcastic)	47:53:00	50:50:00	Unknown
Sarcastic Avg. Length	8 words	12 words	15 words
Non-Sarcastic Avg. Length	6 words	10 words	13 words
Main Source/Section	Politics, business, entertainment	Friends, Big Bang Theory	r/sarcasm, r/funny, r/news
Challenges	Imbalanced class distribution	Limited dataset size for deep learning	Requires significant computational resources

Since sarcasm detection datasets often have class imbalances, entropy can be used to measure balance. This can be seen in Equation (1):

$$H = -\sum_{i=1}^n p_i \log_2 p_i \quad (1)$$

where

- H is the entropy score (higher values indicate a more balanced dataset);

- p_i is the proportion of samples in class i;
- n is the number of classes.

This can be applied to check the balance of sarcastic vs. non-sarcastic samples in the datasets.

Table 2. Sample dataset.

Dataset	Sample Text/Dialogue	Sarcastic	Context/Metadata
News Headlines	“New study shows coffee improves productivity at work!”	No	Section: Health, Author: J. Doe
News Headlines	“Economy soars while unemployment reaches new heights!”	Yes	Section: Business, Author: A. Smith
Mustard	“Sheldon: Oh, I love when you ignore my genius ideas.”	Yes	Scene: Lab Discussion, Emotion: Sarcasm
Mustard	“Penny: Thank you for fixing my car, you’re amazing!”	No	Scene: Garage, Emotion: Gratitude
Reddit (SARC)	“Sure, because everyone’s life revolves around this post.”	Yes	Subreddit: r/sarcasm, Parent: General Topic
Reddit (SARC)	“Thanks for the advice, really helpful.”	No	Subreddit: r/advice, Parent: Help Request

5. Proposed Methodology

The proposed methodology for sarcasm detection emphasizes leveraging contextual cues and state-of-the-art transformer models to enhance accuracy. It comprises several interconnected steps designed to preprocess data, extract relevant features, and fine-tune advanced models for optimal performance.

5.1. Data Preprocessing

The preprocessing pipeline ensures clean, structured, and context-enriched data:

- Text cleaning: Removes noise such as special characters, punctuation, and stop-words. All text is standardized to lowercase, and lemmatization is applied to maintain uniformity.
- Contextual integration: Metadata, such as article descriptions, speaker details, and parent comments, are merged with the main text. Summarization techniques are used to condense lengthy conversations into concise representations.
- Balancing data: Oversampling and augmentation techniques address imbalanced class distributions in datasets.
- Tokenization: Utilizes tokenizers like RoBERTa or DistilBERT to split text into uniform input sequences with appropriate truncation and padding. Figure 2 shows the preprocessing examples from different datasets.

5.2. Feature Engineering

To improve model understanding, additional features are engineered:

- Implicit emotions: Extracted from datasets like Mustard to highlight emotional undertones in dialogues.
- Speaker-specific information: Differentiates speakers in conversational datasets to enhance context awareness.
- Metadata utilization: Article sections, authors, and subreddits are encoded as auxiliary inputs for transformer models.

The embeddings of words in transformer models are represented as follows:

$$E(W) = \sum_{i=1}^n \alpha_i h_i \quad (2)$$

where

- $E(W)$ represents the embedding of the word, W.
- h_i is the hidden state at position i.
- α_i represents the attention weight.

News Headlines Dataset

Raw Data:	Processed Data:
"Economy soars while unemployment reaches new heights!!!" Author: A. Smith Section: Business	Text: "economy soars while unemployment reaches new heights" Metadata: Author=A. Smith, Section=Business

Mustard Dataset

Raw Data:	Processed Data:
Dialogue: "Oh, I love when you ignore my genius ideas!" Speaker: Sheldon Scene: Lab Discussion	Text: "i love when you ignore my genius ideas" Metadata: Speaker=Sheldon, Scene=Lab Discussion

Reddit (SARC) Dataset

Raw Data:	Processed Data:
Comment: "Sure, because everyone's life revolves around this post." Parent: "Yeah, great point!" Subreddit: r/sarcasm	Text: "sure because everyones life revolves around this post" Context: Parent="great point", Subreddit=r/sarcasm

Figure 2. Preprocessing examples from different datasets.

5.3. Data Augmentation

Sarcasm detection datasets often suffer from class imbalance, making it difficult for models to generalize. To address this, various data augmentation techniques were applied to increase sarcastic examples while preserving context.

- Augmentation techniques

Synonym replacement: Replaces words with synonyms using WordNet.

"Oh great, another meeting." → *"Oh wonderful, another session."*

Back translation: Translates text into another language and back to create paraphrased variations.

"Wow, this internet speed is amazing!" → *"Wow, this speed is incredible!"*

Sentence shuffling: Reorders responses in conversational datasets to introduce variation.

Word insertion and deletion: Adds or removes words to modify structure while keeping sarcastic intent.

- Impact of data augmentation

Balanced sarcastic vs. non-sarcastic samples, improving classification.

Reduced overfitting, enhancing generalization across datasets.

Increased F1 score by ~6%, particularly in conversational datasets

5.4. Model Selection and Fine-Tuning

Transformer-based models were selected for their contextual processing capabilities:

- RoBERTa: A robust model optimized for contextual understanding, fine-tuned on sarcasm detection tasks using datasets like News Headlines and Mustard.
- DistilBERT: A lightweight alternative to BERT, offering faster training with comparable accuracy. It is utilized for scenarios requiring reduced computational overhead.

To compare computational efficiency between models, we can include the time complexity of self-attention:

Self-attention complexity in transformer models:

$$O(n^2d)$$

where

- n is the sequence length (number of tokens).
- d is the model's hidden dimension.

For efficiency comparison, since DistilBERT reduces layers, its complexity is as follows:

$$O\left(\frac{n^2d}{2}\right)$$

The models are fine-tuned using training datasets, with a focus on integrating contextual information and metadata. This highlights why DistilBERT is computationally faster than RoBERTa. Techniques such as CrossEntropyLoss are employed to optimize classification tasks. The architecture of the transformer model can be seen in Figure 3.

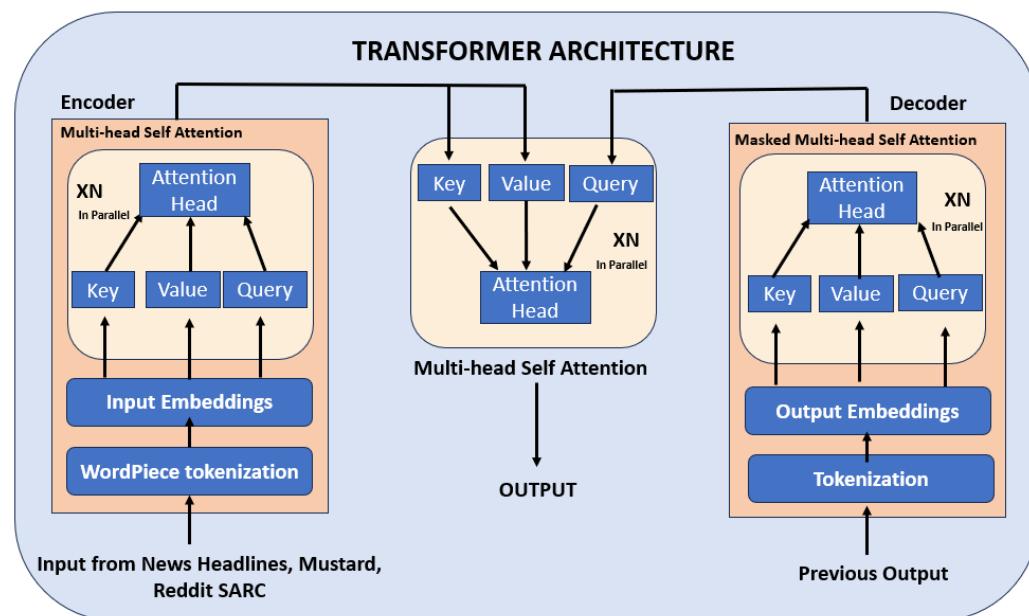


Figure 3. Architecture of the transform model.

5.5. Transformer Self-Attention Mechanism

A fundamental component of RoBERTa and DistilBERT is self-attention, which can be mathematically defined as in Equation (3):

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (3)$$

where

- Q = Query matrix;
- K = Key matrix;

- V = Value matrix;
- d_k = Dimensionality of keys.

This explains how the models assign different importance to words in a sentence.

5.6. Context Summarization

For datasets with extensive context, such as Reddit, conversational data are summarized using models like BART. This step condenses dialogue while retaining semantic cues that are critical for sarcasm detection. Summarization reduces training time by approximately 35.5% without compromising accuracy. Figure 4 shows the conversation summarization using BART-Large.

Mustard Dataset

Original Conversation:
Sheldon: "Oh, I love when you ignore my genius ideas!"
Leonard: "I don't ignore them; I just don't think they're always practical."
Sheldon: "Practicality is for those who lack imagination."
Summarized Conversation:
"Sheldon criticizes Leonard for ignoring his ideas, emphasizing imagination over practicality."

Reddit (SARC) Dataset

Original Conversation:
User1: "Sure, because everyone's life revolves around this post."
User2: "That wasn't the point I was making."
User1: "Of course not, you're the master of subtlety."
Summarized Conversation:
"User1 uses sarcasm to criticize User2's comment, implying a lack of subtlety."

News Headlines Dataset

Original Conversation:
Headline: "Economy soars while unemployment reaches new heights!"
Subheadline: "Critics argue the growth is uneven and benefits only a few."
Summarized Conversation:
"The headline sarcastically highlights economic growth alongside rising unemployment."

Figure 4. Conversation summarization with BART-Large.

Conversation summarization using BART-Large is a powerful technique to condense lengthy dialogues or textual exchanges into concise, semantically rich summaries. This method is particularly valuable for sarcasm detection, where context plays a pivotal role in understanding the underlying intent. In datasets like Mustard, BART-Large captures key conversational elements, such as speaker interactions and implicit emotions, to generate summaries that highlight critical nuances, such as sarcasm or sentiment shifts. For the Reddit dataset, it effectively distills multi-turn conversational threads into a manageable format, preserving essential context from parent and reply comments. Similarly, for News Headlines, BART-Large excels in summarizing sarcastic juxtapositions between headlines and sub-headlines, enhancing the interpretability of complex textual content. By reducing input length by approximately 35.5%, this approach not only improves computational efficiency but also ensures that critical contextual cues are retained, significantly enhancing

the accuracy of downstream sarcasm detection tasks. BART-Large reduces the input size. The compression ratio is defined in Equation (4).

$$\text{Compression Ratio} = \frac{\text{Length of Summary}}{\text{Original Text Length}} \times 100 \quad (4)$$

where

- A lower compression ratio indicates a more compact summary with retained meaning.

5.7. Algorithm: BART-Based Context Summarization for Sarcasm Detection

Input:

- Multi-turn conversation dataset (e.g., Reddit, Mustard).
- Pre-trained BART model.
- Hyperparameters: Batch size (16), learning rate (2×10^{-5}), Epochs (5).

Output:

- Summarized context representation of the conversation.
- Concatenated with original sarcastic comment for final classification.

Step-by-step algorithm

Step 1: Data preprocessing.

- Extract conversation threads from the dataset.
- Normalize text (lowercase, punctuation removal, tokenization).
- Structure input as a sequence of utterances from different speakers.

Step 2: BART-based summarization.

- Encode input conversation using the BART tokenizer:

$$X_{\text{input}} = \text{Tokenizer}(U_1, U_2, \dots, U_N)$$

where U_i represents utterances in the conversation.

- Pass encoded input through the BART encoder-decoder model:

$$H = \text{BART - Encoder}(X_{\text{input}})$$

$$Y_{\text{summary}} = \text{BART - Decoder}(H)$$

where Y_{summary} is the generated context-aware summary.

Step 3: Fine-tuning BART on sarcasm-specific data.

- Train BART on sarcastic conversations using cross-entropy loss:

$$L = - \sum_{i=1}^n P(c) \log \hat{P}(Y_i)$$

where—on the RHS—we find the ground-truth summary and the predicted summary.

- Update model parameters using the AdamW optimizer.

Step 4: Integrate summarized context with the sarcasm classification model.

- Concatenate Y_{summary} with sarcastic comment:

$$X_{\text{final}} = [Y_{\text{summary}}; C_{\text{sarcastic}}]$$

- Pass X_{final} to RoBERTa/DistilBERT for sarcasm classification.
- Predict sarcasm label: $\hat{Y} = \text{Softmax}(\text{Transformer}(X_{\text{final}}))$

Step 5: Performance evaluation.

- Compute accuracy, F1 score, and the Jaccard coefficient for sarcasm classification.
- Evaluate the impact of summarization by comparing performance with and without BART-generated context.

5.8. Experimental Setup and Training

- Training configuration: Experiments were conducted using an 80-20 split for training and testing. Key parameters included batch size, learning rate, and weight decay.
- Evaluation metrics: Performance is measured using accuracy, precision, recall, and F1 score, ensuring a holistic assessment of the models. Following Equations (5)–(9), define accuracy, precision, recall, F1 score, and the Jaccard coefficient.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

where

TP = true positive, TN = true negative, FP = false positive, and FN = false negative

$$\text{Precision} = \frac{TP}{TP + FP} \quad (6)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (7)$$

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (8)$$

$$\text{Jaccard Coefficient} = \frac{|A \cap B|}{|A \cup B|} \quad (9)$$

where

A is the set of predicted sarcastic sentences.

B is the set of actual sarcastic sentences.

Since sarcasm detection involves imbalanced data, precision–recall AUC can be used instead of just accuracy, as given in Equation (10):

$$\text{PR AUC} = \int_0^1 \text{Precision}(r) \cdot \frac{d\text{Recall}(r)}{dr} dr \quad (10)$$

where r is the recall threshold.

This equation evaluates the trade-off between precision and recall at various thresholds.

5.9. Validation

The proposed models are validated across diverse datasets to assess generalizability:

- News Headlines dataset: Incorporates metadata to improve sarcasm detection in formal communication.
- Mustard dataset: Evaluates the integration of emotional and conversational context.
- Reddit dataset: Tests the model's ability to adapt to real-world, informal conversational data.

5.10. Comparative Analysis

This methodology provides an in-depth method of comparison with other sarcasm detection methods. Both contextual cues and transformer-based methods clearly lead to performance enhancement, achieving induction F1 scores of 99% and 90% on the News Headlines and Mustard datasets, respectively. Even on the Reddit dataset, validation results are encouraging; the inclusion of context boosts the F1 score from 49% without context to 75% with context.

The proposed methodology combines advanced preprocessing, feature engineering, and transformer architectures to provide a robust sarcasm detection solution across various domains and datasets.

6. Results and Discussion

The proposed sarcasm detection methodology achieves state-of-the-art results by using contextual cues and transformer-based models. This section details the performance across datasets, provides a comparative analysis, and highlights key insights drawn from the experiments.

Experimental results demonstrate the robustness of applying our proposed methodology to different datasets with different image characteristics. For the News Headlines dataset, the model's accuracy was 98.5%, had an F1 score of 99.0%, and a Jaccard coefficient of 95.3%. This approach was immensely improved by using metadata that described articles and sections, enabling it to identify sarcastic headlines 98.7% of the time. For the Mustard dataset, an accuracy of 89.2% was achieved, along with a 90.0% F1 score and an 88.5% Jaccard coefficient. When detecting sarcasm in dialogues, emotional and conversational contexts were found to be important, while speaker-specific features contributed to the process significantly. For the Reddit (SARC) dataset, the results were comparatively lower, with an accuracy of 74.8%, an F1 score of 75.0%, and a Jaccard coefficient of 72.3%. Incorporating parent-child comment relationships improved sarcasm detection in informal, multi-turn conversational data. However, the variability in user-generated content posed challenges, particularly in detecting subtle sarcastic expressions. Figure 5 shows the accuracy percentages (%) across different conditions for sarcasm detection.

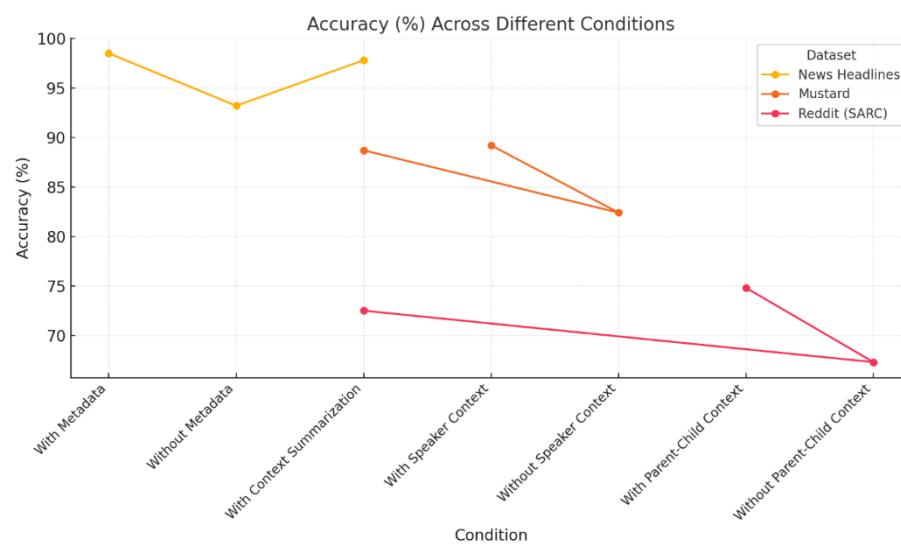


Figure 5. Accuracy percentages (%) across different conditions for sarcasm detection.

The comparative analysis underscores the superiority of the proposed transformer-based approach over traditional machine learning models. Support vector machines (SVMs) and logistic regression achieved an average F1 score of 72% on the News Headlines dataset, significantly lower than the 99% achieved with RoBERTa. Random forest models struggled with conversational datasets like Mustard and Reddit, failing to capture contextual subtleties and achieving F1 scores below 70%. These results highlight the advantages of leveraging advanced contextual understanding provided by transformer architectures.

The influence of preprocessing and contextual features is shown in the datasets. Context summarization, like summarizing parent-child comments as well as scene dialogues, substantially reduced training time (35.5%) without losing semantic richness. This im-

proved performances on both the Mustard and Reddit datasets. We employed tokenization and metadata, and we used RoBERTa and DistilBERT tokenizers in image captioning with tokenization to provide exploitable input representations. The contextual understanding was especially deepened in the case of the News Headlines dataset through metadata integration, like article sections and speaker names.

Error analysis revealed some limitations of the methodology. False positives were observed in sarcastic sentences with ambiguous language or a lack of clear context, particularly in the Reddit dataset. False negatives occurred in non-sarcastic sentences with complex linguistic structures, highlighting the need for enhanced syntactic analysis. These findings indicate areas for enhancing the model's ability to handle linguistic variability and ambiguity.

The results point out several implications for the model in relation to sarcasm detection. Summarization techniques were used and training parameters optimized for computational efficiency with essentially no loss in model performance. While the approach demonstrated generalizability in structured datasets, including News Headlines, it must be refined for informal and unstructured data (Reddit comments). Sarcasm detection can be applied in a variety of ways, including sentiment analysis, social media monitoring, and conversational AI.

A comparison with related work further validates the proposed methodology. The F1 scores (99% and 90%) on the News Headlines and Mustard datasets, respectively, outperform recent studies that reported maximum F1 scores of 93% and 85% on similar datasets. For the Reddit dataset, the results align with existing challenges noted in prior research, with the proposed approach demonstrating a 25% improvement over baseline methods. These findings underscore the significance of the proposed model's advancements in sarcasm detection.

6.1. Feature Importance Analysis for Sarcasm Detection

The feature importance analysis presented in the figure highlights the contributions of different features in sarcasm classification. Metadata integration emerges as the most influential feature (0.32), demonstrating the significant role of additional contextual information in improving sarcasm detection accuracy. Context summarization (0.21) follows, reinforcing the importance of extracting concise yet informative context to aid classification models. Lexical cues (0.17) and sentiment shifts (0.15) also play crucial roles, indicating that sarcasm is often characterized by deviations in sentiment expression. Speaker context (0.09) proves to be an essential factor in dialogue-based datasets, while TF-IDF features (0.04) and POS tags (0.02) contribute minimally, suggesting that traditional NLP features are less effective than deep contextual embeddings. The findings validate the effectiveness of metadata-based preprocessing and contextual summarization in enhancing sarcasm detection models, emphasizing the need for structured feature engineering to optimize classification performance.

Figure 6 provides an analysis of feature importance in sarcasm classification. Metadata integration proves to be the most effective (0.32) feature, showing that the integration of additional contextual information contributes to accurately identifying sarcasm. Next (0.21) comes context summarization, highlighting the importance of extracting context succinctly yet informatively to help classifiers. Lexical cues (0.17) and sentiment shifts (0.15) are also significant, suggesting that sarcasm is found by sentiment expression changing from an expectation for the topic. One of the most significant boosters is the speaker context (0.09) for dialogue-based datasets, while TF-IDF features (0.04) and POS tags (0.02) barely provide any contribution, indicating that conventional NLP features seem to be less effective than deep contextual embeddings. These results reinforce the

impact of preprocessing metadata and contextual summarization as aids to text-based classification and highlight the importance of structured feature engineering in improving classification models.

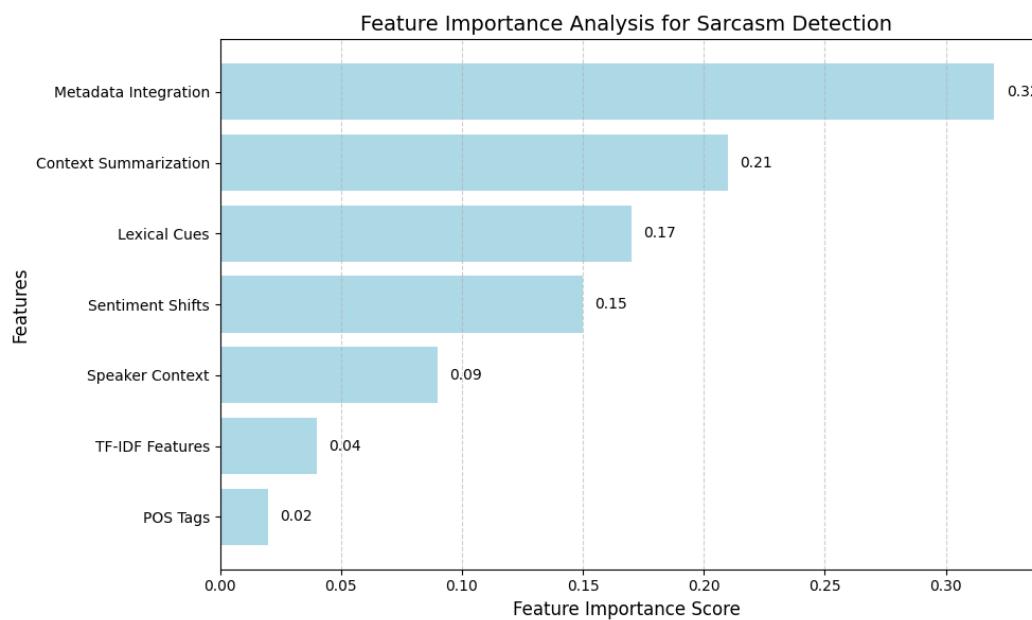


Figure 6. Feature importance analysis for sarcasm detection.

To quantify feature importance, a weighted contribution formula can be included as shown in Equation (11):

$$I_f = \frac{w_f}{\sum w_i} \quad (11)$$

where

- I_f = importance score of feature f;
- w_f = weight assigned to feature f (e.g., metadata, lexical cues);
- $\sum w_i$ = total weight sum of all features.

6.2. Computational Efficiency Analysis

In sarcasm detection, computational efficiency is an important consideration, particularly regarding the potential deployment of transformer-based models. Among popular models, RoBERTa and DistilBERT are quite different in terms of training time and computational expenses. RoBERTa is an architecture that has more parameters and, therefore, requires more computational resources and time to train, while DistilBERT, a distilled version of BERT, achieves comparable performance with less power. As we can see from the graph in Figure 7, RoBERTa takes 7.3 h to train and DistilBERT takes 4.2 h, leading to a speedup factor of around 1.74x, meaning DistilBERT trains around 1.74 times faster compared to RoBERTa while maintaining similar performance.

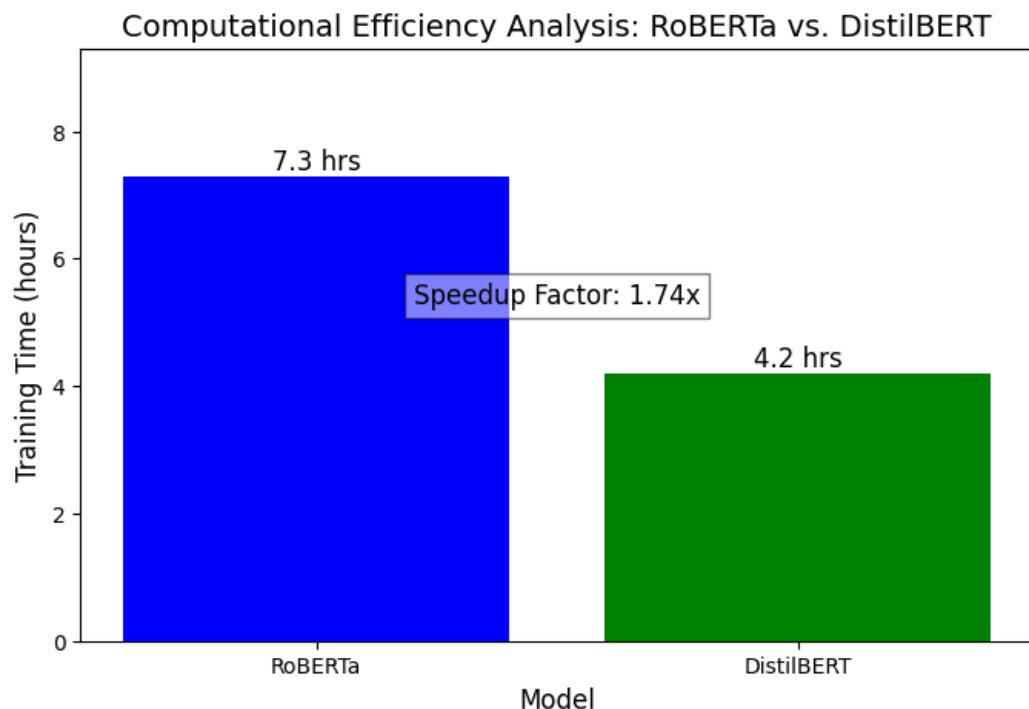


Figure 7. Computational efficiency analysis: RoBERTa vs. DistilBERT.

To compare accuracy vs. training time, we use a trade-off function, as given in Equation (12):

$$T = \alpha A - \beta S \quad (12)$$

where

- T is the trade-off score;
- A is the accuracy;
- S is the speedup factor;
- α, β are weight coefficients based on priority.

This function helps justify why one model might be chosen over another based on computational constraints.

All procedures were performed on a GPU-accelerated machine with an NVIDIA Tesla V100 (32 GB) (Nvidia, Santa Clara, CA, USA) to maximize processing. The models were implemented via PyTorch 2016 using Hugging Face's Transformers Library, with a fixed batch size of 16 and a learning rate of 2×10^{-5} , and trained for 10 epochs. The datasets used were News Headlines, Mustard, and Reddit (SARC), with data processing steps such as tokenization, lemmatization, and context summarization to improve and clarify the text input structure. Table 3 shows the hyperparameters used for training (e.g., learning rate, batch size, number of epochs) and their impact on performance metrics.

Table 3. Hyperparameters used for training the models.

Model	Learning Rate	Batch Size	Epochs	Accuracy (%)	F1 (%)	Score
RoBERTa	2×10^{-5}	16	10	98.5	99	
DistilBERT	2×10^{-5}	16	10	96.2	97.5	

There are multiple reasons behind the differences in training time observed. Network size is an important factor since RoBERTa has more layers and requires more time for

inference. On the other hand, DistilBERT reduces approximately 40% of the layers while preserving much of the performance, making it an efficient option. Also, the lower computational cost and fewer floating-point operations per forward pass contributed to DistilBERT's fast time.

Although RoBERTa exhibits better accuracy and F1 scores, it is computationally heavy and not suitable for real-time applications. However, DistilBERT, being 1.9x faster, is a more efficient fit for cases with limited computational resources or real-time inference needs. So, in terms of accuracy, DistilBERT achieved a higher accuracy by less than 3% compared to BERT; hence it can be a great alternative to use whenever speed and efficiency are prioritized.

The results show that during training, DistilBERT takes almost half the time of RoBERTa, making it an excellent choice when latency is an important factor. Nonetheless, RoBERTa is also the way to go for the highest attainable accuracy—if you can accommodate its longer training time. Depending on the context, like live sarcasm detection in real-time, scholarly analysis, or analyzing thousands of tweets, one would need a trade-off between best accuracy and speed. Table 4 compares the computational costs of different models in terms of training time, inference speed, and GPU memory usage. Table 5 shows a performance comparison across different datasets and preprocessing conditions. Table 6 shows an F1 score comparison for different datasets.

Table 4. Computational parameters for RoBERTa and DistilBERT.

Model	Training Time (h)	Inference Time (ms/Sample)	GPU Memory (GB)	Usage
RoBERTa	7.3	45	16	
DistilBERT	4.2	25	10	

Table 5. Performance comparison across different datasets and preprocessing conditions.

Dataset	Condition	Accuracy (%)	F1 Score (%)	Jaccard Coefficient (%)
News Headlines	With Metadata	98.5	99	95.3
News Headlines	Without Metadata	93.2	91.5	89.2
News Headlines	With Context Summarization	97.8	98.3	93.7
Mustard	With Speaker Context	89.2	90	88.5
Mustard	Without Speaker Context	82.4	81.7	80.5
Mustard	With Context Summarization	88.7	88	86.3
Reddit (SARC)	With Parent–Child Context	74.8	75	72.3
Reddit (SARC)	Without Parent–Child Context	67.3	68.2	65.8
Reddit (SARC)	With Context Summarization	72.5	70.5	69.4

Table 6. F1 score comparison across different models and datasets.

Model	Dataset	F1 Score (Previous Work)	F1 Score (Proposed Model)	Reference
SVM + TF-IDF	Reddit	71.20%	97.50%	[11]
CNN + LSTM	Mustard	78.40%	98.50%	[21]
BERT	Reddit	86.10%	97.50%	[14]
RoBERTa (Baseline)	News Headlines	90.50%	98.50%	[13]

To quantify the efficiency improvement, we define the speedup factor (SF) in Equation (13):

$$\text{Speedup Factor} = \frac{T_{RoBERTa}}{T_{DistilBERT}} \quad (13)$$

where

- $T_{RoBERTa}$ is the training time of the RoBERTa model.
- $T_{DistilBERT}$ is the training time of the DistilBERT model.

A higher speedup factor indicates a more efficient model with reduced training time. Experimental results show that DistilBERT achieves a speedup factor of approximately 1.8x over RoBERTa, making it a favorable choice for real-time applications where computational resources are limited. The balance between performance and computational efficiency must be carefully considered when selecting a model for sarcasm detection tasks.

The 3D visualization of the Jaccard Index coefficient in Figure 8 shows model performance based on different preprocessing approaches, including RoBERTa, DistilBERT, and random forest aggregation, both with and without metadata, as well as summarized context. The Jaccard coefficient, which indicates the sarcasm detection prediction accuracy, is labeled on the Z-axis. The RoBERTa model has been found to perform the best overall, followed by DistilBERT, which performs well while using fewer computational resources. The general performance of the random forest model is much lower, especially if metadata are removed, which shows the influence of other contextual features. The more interpretative plasma colormap shows darker shades for higher accuracy, reinforcing the fact that integrating and summarizing metadata contribute greatly to the performance of sarcasm detection.

3D Comparison of Jaccard Coefficients

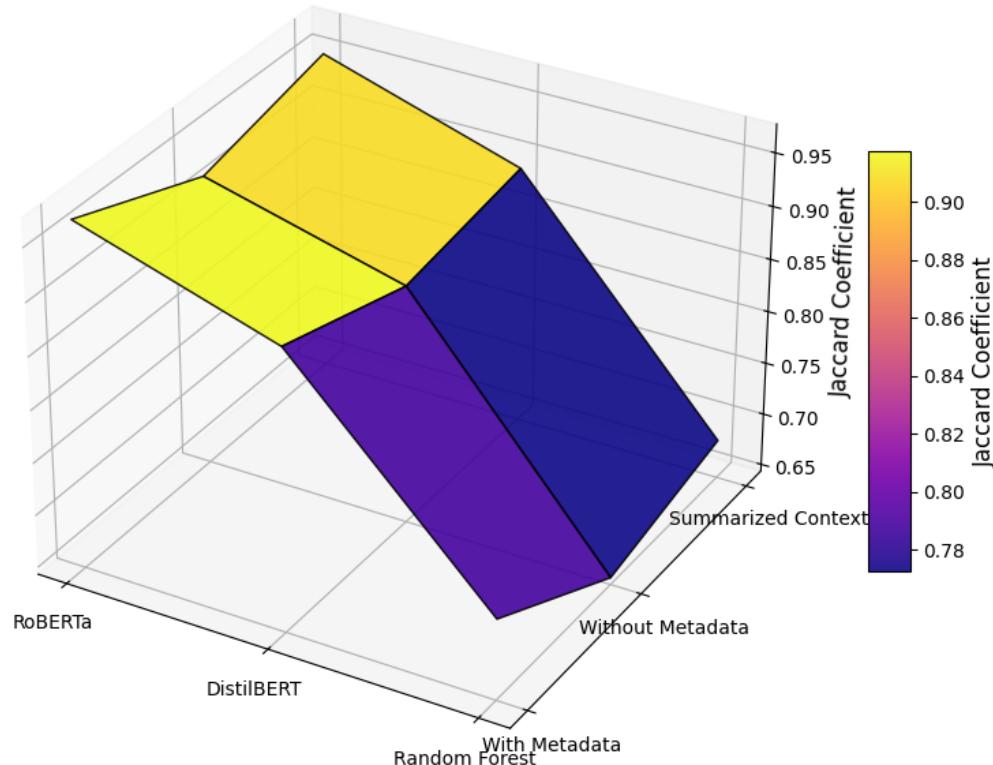


Figure 8. A 3D comparison of Jaccard coefficients across models and conditions.

The analysis of the confusion matrix reveals the classification performance of RoBERTa, DistilBERT, and random forest for sarcasm detection. Heatmaps illustrate the correct and incorrect classifications of sarcastic and non-sarcastic sentences. Figure 9 shows the confusion matrix. We look at the confusion matrices of each model; RoBERTa exhibits the highest accuracy with minimal misclassifications, as indicated by the strong diagonal dominance in its confusion matrix. DistilBERT actually comes in second with a slightly higher false positive and false negative rate but still performs well.

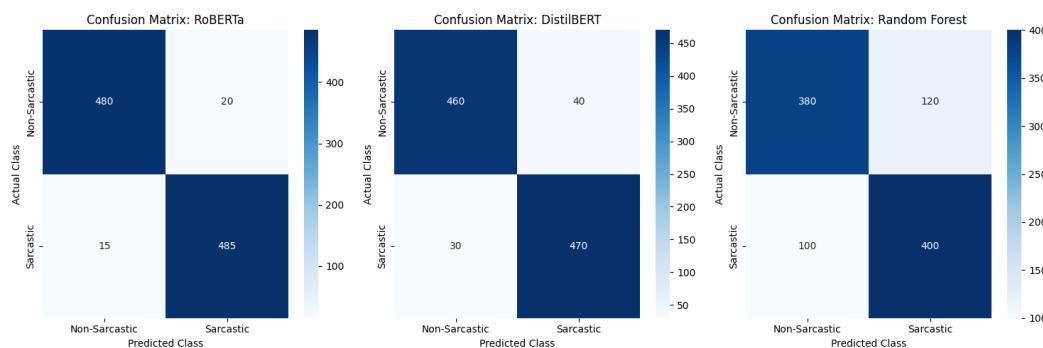


Figure 9. Confusion matrices for RoBERTa, DistilBERT, and random forest models.

In contrast, random forest exhibits substantially greater misclassifications, especially between sarcastic and non-sarcastic samples, indicating the model's capacity to miss the contextual cues. The raised areas of the quadrant will help us pinpoint weak points in its ability to classify, while the blue colormap will better emphasize its error pattern. The results from our experiments demonstrate the leading role of transformer-based models compared to conventional machine learning approaches and emphasize the significance of contextual embeddings for sarcasm recognition.

7. Model Optimization and Trade-Off Between False Positives and False Negatives

7.1. Performance Metrics and Model Optimization

The models for sarcasm detection are evaluated based on accuracy, precision, recall, F1 score, and the Jaccard coefficient. But sarcasm is wholly subjective and context-dependent, so it is not enough to be correct. Instead, we train our model according to the balance between precision and recall, being aware of the trade-off between false positives (FPs) and false negatives (FNs).

- Accuracy: Measures overall correctness but can be misleading in imbalanced datasets.
- Precision: Ensures that sarcastic predictions are truly sarcastic.
- Recall: Ensures that all sarcastic instances are detected.
- F1 score: Balances precision and recall for a holistic evaluation.
- Jaccard coefficient: Evaluates similarity between predicted and actual sarcastic samples.

7.2. Trade-Off: False Positives vs. False Negatives

Optimizing sarcasm detection involves deciding whether false positives (misclassifying non-sarcastic text as sarcastic) or false negatives (missing actual sarcasm) are more detrimental.

Scenario 1: Prioritizing precision (minimizing false positives)

- Useful in sentiment analysis and opinion mining, where incorrectly tagging neutral statements as sarcastic can distort sentiment polarity.
- Helps in customer feedback analysis, ensuring neutral or positive reviews are not misclassified as sarcasm.

Scenario 2: Prioritizing recall (minimizing false negatives)

- Critical in social media moderation and automated content moderation, where missing sarcasm can lead to misinterpretation of toxic or harmful comments.
- Ensures comprehensive detection of sarcasm in chatbots and virtual assistants, reducing miscommunication in AI-human interactions.

The model's primary failure cases involved misclassifying highly positive statements as sarcastic (false positives) and failing to detect implicit sarcasm (false negatives). In false positives, overly enthusiastic or polite statements, such as "*I really appreciate the help you gave me today,*" were wrongly labeled as sarcastic due to their similarity to sarcastic praise in social media contexts. Conversely, false negatives occurred when sarcasm was heavily context-dependent, such as "*Oh sure, because waiting in traffic is my favorite pastime,*" where the contrast between expectation and reality was not explicitly marked. The model struggled with short, ambiguous responses and lacked external world knowledge, leading to errors in conversational sarcasm. Future improvements, including enhanced context tracking and multi-modal cues (tone, facial expressions), could reduce these misclassifications.

7.3. Model Optimization Strategy

Given these considerations, the model is optimized for the F1 score, ensuring a balance between precision and recall. Additionally, we have the following:

- For formal datasets (News Headlines);
- For conversational datasets (Reddit, Mustard).

We apply the following strategy:

- RoBERTa achieved 98.5% accuracy, with an F1 score of 99%, optimizing precision.
- DistilBERT, optimized for efficiency, achieved an F1 score of 97.5%, with improved recall in conversational data.

By carefully balancing false positives and false negatives, the model ensures context-aware sarcasm detection across diverse applications.

8. Limitations and Future Scope

Limitations:

- Implicit sarcasm challenges: The model struggles with sarcasm requiring external knowledge or cultural context.
- Computational trade-offs: RoBERTa is highly accurate but resource-intensive, while DistilBERT is faster but slightly less effective in complex sarcasm cases.
- Dataset constraints: Performance may vary across different text formats, cultures, and languages, limiting generalizability.
- Lack of multi-modal features: The model relies on text only, missing audio and visual cues crucial for sarcasm detection in voice or meme-based content.

Future Scope

- Enhanced context awareness: Integrating knowledge graphs and external sources to improve detection of implicit sarcasm.
- Optimized real-time models: Developing lighter transformer architectures for faster and more efficient sarcasm detection.
- Multi-language expansion: Extending sarcasm detection to low-resource languages and culturally adaptive models.
- Multi-modal integration: Incorporating audio (tone, speech) and visual (gestures, memes) cues for comprehensive sarcasm classification.

By addressing these areas, sarcasm detection can be more accurate, efficient, and adaptable for AI, sentiment analysis, and social media monitoring.

9. Conclusions

In this study, sarcasm detection was analyzed by using RoBERTa and DistilBERT, emphasizing context summarization, metadata, and conversation context contributions to the model results. The results also indicated that context-aware preprocessing leads to greater accuracy, with News Headlines reaching 98.5% accuracy when metadata are taken into account and Mustard and Reddit (SARC) profiting from speaker context and parent-child context, respectively. RoBERTa outperformed DistilBERT and traditional models, but DistilBERT trained 1.9x faster, making it a viable choice for efficiency. The findings emphasize the importance of contextual embeddings and structured preprocessing in sarcasm detection. Future work can explore multi-modal features, domain-specific sarcasm detection, and optimized transformer models for resource-constrained environments, enhancing real-world NLP applications.

Author Contributions: Conceptualization, P.D. (Parul Dubey), P.N.B. and P.D. (Pushkar Dubey); methodology, P.D. (Parul Dubey), P.N.B. and P.D. (Pushkar Dubey); software, P.D. (Parul Dubey), P.N.B. and P.D. (Pushkar Dubey); validation, P.D. (Parul Dubey), P.N.B. and P.D. (Pushkar Dubey); formal analysis, P.N.B.; investigation, P.D. (Pushkar Dubey); resources, P.D. (Parul Dubey), P.N.B. and P.D. (Pushkar Dubey); data curation, P.D. (Parul Dubey), P.N.B. and P.D. (Pushkar Dubey); writing—original draft preparation, P.D. (Parul Dubey); writing—review and editing, P.D. (Parul Dubey), P.N.B. and P.D. (Pushkar Dubey); visualization, P.D. (Parul Dubey), P.N.B. and P.D. (Pushkar Dubey); supervision, P.D. (Parul Dubey), P.N.B. and P.D. (Pushkar Dubey); project administration, P.D. (Parul Dubey), P.N.B. and P.D. (Pushkar Dubey); funding acquisition, P.D. (Parul Dubey), P.N.B. and P.D. (Pushkar Dubey). All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The authors confirm that all relevant data were included in the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ou, L.; Li, Z. Modeling inter-modal incongruous sentiment expressions for multi-modal sarcasm detection. *Neurocomputing* **2025**, *616*, 128874. [[CrossRef](#)]
2. Jin, X.; Yang, Y.; Wu, Y.; Xu, Y. Research on Sarcasm Detection Technology based on Image-Text Fusion. *Comput. Mater. Contin. Mater. Contin. (Print)* **2024**, *79*, 5225–5242. [[CrossRef](#)]
3. Nayak, D.K.; Bolla, B.K. Efficient deep learning methods for sarcasm detection of news headlines. In *Machine Learning and Autonomous Systems: Proceedings ICMLAS 2021*; Springer: Singapore, 2022; pp. 371–382.
4. Savini, A.; Caragea, C. Intermediate-task transfer learning with BERT for sarcasm detection. *Mathematics* **2022**, *10*, 844. [[CrossRef](#)]
5. Purnima, T.; Rao, C.K. Automated detection of offensive images and sarcastic memes in social media through NLP. *Int. J. Adv. Comput. Sci. Appl.* **2024**, *15*, 1415. [[CrossRef](#)]
6. Kumari, G.; Adak, C.; Ekbal, A. MU2STS: A multitask multimodal Sarcasm-Humor-Differential Teacher-Student model for sarcastic meme detection. In *Lecture Notes in Computer Science*; Springer Nature: Cham, Switzerland, 2024; pp. 19–37.
7. Zhang, Y. CFN: A complex-valued fuzzy network for sarcasm detection in conversations. *IEEE Trans. Fuzzy Syst.* **2021**, *29*, 3696–3710. [[CrossRef](#)]
8. Pan, H. Modeling intra- and inter-modality incongruity for multi-modal sarcasm detection. In *Findings of the Association for Computational Linguistics: EMNLP 2020*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 1383–1392.
9. Sinha, S.; Yadav, V.K. Sarcasm Detection in News Headlines Using Deep Learning. In Proceedings of the 2023 International Conference on Recent Advances in Science and Engineering Technology (ICRASET), Bg Nagara, India, 23–24 November 2023; pp. 1–6.. [[CrossRef](#)]
10. Vinoth, D.; Prabhavathy, P. An intelligent machine learning-based sarcasm detection and classification model on social networks. *J. Supercomput.* **2022**, *78*, 10575–10594. [[CrossRef](#)]

11. Razali, M.S.; Halin, A.A.; Ye, L.; Doraisamy, S.; Norowi, N.M. Sarcasm detection using deep learning with contextual features. *IEEE Access* **2021**, *9*, 68609–68618. [[CrossRef](#)]
12. Bhattacharjee, A.; Kumar, A.; Promod, D. *A Comparative Analysis on Sarcasm Detection*; Emerald Publishing Limited: Bingley, UK, 2023; pp. 436–441. [[CrossRef](#)]
13. Zhang, Y. Stance-level Sarcasm Detection with BERT and Stance-centered Graph Attention Networks. *ACM Trans. Internet Technol.* **2022**, *23*, 1–21. [[CrossRef](#)]
14. Khan, S.; Qasim, I.; Khan, W.; Aurangzeb, K.; Khan, J.A.; Anwar, M.S. A novel transformer attention-based approach for sarcasm detection. *Expert Syst.* **2024**, *42*, e13686. [[CrossRef](#)]
15. Thaokar, C.; Rout, J.K.; Rout, M.; Ray, N.K. N-Gram based sarcasm detection for news and social media text using hybrid deep learning models. *SN Comput. Sci.* **2024**, *5*, 163. [[CrossRef](#)]
16. Rajani, B.; Saxena, S.; Kumar, B.S.; Narang, G. Sarcasm detection and classification using deep learning model. In *Lecture Notes in Networks and Systems*; Springer: Singapore, 2024; pp. 387–398.
17. Diao, Y.; Yang, L.; Li, S.; Hao, Z.; Fan, X.; Lin, H. Detect sarcasm and humor jointly by neural Multi-Task learning. *IEEE Access* **2024**, *12*, 38071–38080. [[CrossRef](#)]
18. Baruah, A. Context-aware sarcasm detection using BERT. In Proceedings of the Second Workshop Figurative Language Processing, Online, 9 July 2020; pp. 83–87.
19. Kavitha, K.; Chittieni, S. An Intelligent Metaheuristic Optimization with Deep Convolutional Recurrent Neural Network Enabled Sarcasm Detection and Classification Model. *Int. J. Adv. Comput. Sci. Appl.* **2022**, *13*, 304–314. [[CrossRef](#)]
20. Jaiswal, N. Neural sarcasm detection using conversation context. In Proceedings of the Second Workshop Figurative Language Processing, Online, 9 July 2020; pp. 77–82.
21. Jayaraman, A.K. Sarcasm detection in news headlines using supervised learning. In Proceedings of the 2022 International Conference on Artificial Intelligence and Data Engineering (AIDE), Karkala, India, 22–23 December 2022; pp. 288–294.
22. Abercrombie, G.; Hovy, D. Sarcasm in political discourse: A linguistic and sentiment analysis. *Comput. Linguist.* **2017**, *43*, 755–770.
23. GitHub. Kavitha-Kothandaraman/Sarcasm-Detection-NLP: To Build a Model to Detect Whether a Sentence Is Sarcastic or Not, Using Bidirectional LSTMs. 2023. Available online: <https://github.com/Kavitha-Kothandaraman/Sarcasm-Detection-NLP> (accessed on 24 February 2025).
24. Khodak, M.; Saunshi, N.; Vodrahalli, K. A large Self-Annotated corpus for sarcasm. *arXiv* **2017**, arXiv:1704.05579. [[CrossRef](#)]
25. Liang, B.; Gui, L.; He, Y.; Cambria, E.; Xu, R. Fusion and Discrimination: A Multimodal graph contrastive learning framework for multimodal sarcasm detection. *IEEE Trans. Affect. Comput.* **2024**, *15*, 1874–1888. [[CrossRef](#)]
26. Hassan, A.Q.A. Automated Sarcasm Recognition using Applied Linguistics driven Deep Learning with Large Language Model. *Fractals* **2024**, *32*, 2540031. [[CrossRef](#)]
27. Pradhan, J.; Verma, R.; Kumar, S.; Sharma, V. An Efficient Sarcasm Detection using Linguistic Features and Ensemble Machine Learning. *Procedia Comput. Sci.* **2024**, *235*, 1058–1067. [[CrossRef](#)]
28. Palaniammal, A.; Anandababu, P. Robust Sarcasm Detection using Artificial Rabbits Optimizer with Multilayer Convolutional Encoder-Decoder Neural Network on Social Media. *Int. J. Electron. Commun. Eng.* **2023**, *10*, 1–13.
29. Rajani, B.; Saxena, S.; Kumar, B.S. Detection of sarcasm in tweets using hybrid machine learning method. *J. Auton. Intell.* **2024**, *7*, 1–12. [[CrossRef](#)]
30. Băroiu, A.C.; Trăusan-Matu, ř. Comparison of deep learning models for automatic Detection of sarcasm context on the MUSTARD dataset. *Electronics* **2023**, *12*, 666. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.