SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V12I5P103
Volume 12 Issue 5, 19-32, May 2025

Original Article

Context-Aware Models with Rule-Based System Incorporating Historical and Situational Context to Improve the Understanding and Detection of SARCASM

R. Babubalaji[1], N. Subalakshmi[2]

[1, 2]Department of Computer and Information Science, Annamalai University, Chidambaram, Tamil Nadu, India.

[1]Corresponding Author : babubalajimsc@gmail.com

Received: 02 March 2025 Revised: 04 April 2025 Accepted: 04 May 2025 Published: 27 May 2025

Abstract - *The detection of SARCASM in text presents a significant challenge in natural language processing due to its reliance on contextual subtleties and the interplay between literal and intended meanings. This research aims to develop context-aware models with the rule-based system that incorporate both historical and situational context to enhance the understanding and detection of SARCASM. Propose a multi-faceted approach that integrates linguistic cues, user-specific historical data, and situational information to capture the nuances of sarcastic expressions. The historical context encompasses users' prior interactions and communication patterns, while the situational context involves the immediate conversational environment and external factors influencing the dialogue. The proposed context-aware models are evaluated on benchmark SARCASM detection datasets and real-world social media data to assess their effectiveness and robustness. This research contributes to the broader sentiment analysis and conversational AI field, offering potential applications in social media monitoring, customer service automation, and human-computer interaction.*

Keywords - *SARCASM detection, Context-Aware Models, Historical context, Situational context, Natural Language Processing, Machine Learning, Sentiment analysis, Conversational AI, Social media analysis.*

1. Introduction

Identifying SARCASM in Natural Language Processing (NLP) remains a challenging task. Unlike simple emotions, SARCASM often involves a complex interplay between literal and intended meanings, making it difficult for traditional sentiment analysis algorithms to detect. Sarcastic expressions can be nuanced and context-specific, requiring a deep understanding of both linguistic subtleties and contextual factors [1]. There is much to explore in developing more advanced models that can accurately recognize and understand SARCASM, especially in user-generated content on social networking sites where such expressions are common. SARCASM is a type of figurative speech where the writer's intended meaning differs from the statement's literal meaning [2, 3].

Initial studies on automated SARCASM detection focused on salient textual features such as pragmatics, n-grams, and lexical aspects. Textual recognition of SARCASM can be enhanced by explicit cues within phrases, such as sentiment lexicons, user mentions, emoticons, and contradictions [4]. These methods often overlook implicit traits that contribute to the ambiguity of sarcastic expressions, focusing instead on highlighting obvious characteristics [5].

The field of satirical sentiment assessment in NLP is expanding rapidly, with studies ranging from concept-level categorization to record-keeping, word, wording, and phrase-level categorization [6]. Researchers are making progress in analyzing sentiments in written language more accurately and efficiently in understanding SARCASM, amusement, and irony in social media content. Based on the characteristics of text used for categorization, sentiment identification is divided into three groups: lexical, pragmatic, and hyperbolic, as shown in Figure 1. The amount of information social media generates today is enormous [7]. Businesses have used sentiment analysis for years to strengthen their positions in their chosen marketplaces.

On the other hand, SARCASM is defined as a positive statement or phrase with an underlying negative intention. Since SARCASM can alter the polarity of a sentence, it is crucial for automatic natural language processing techniques to identify and handle it appropriately [8] existing research on SARCASM identification, such as the studies conducted by rule-based methods.

More recent research has shifted towards using deep learning to recognize distinctive traits autonomously. The conceptual framework is illustrated in Figure 2, focusing on two primary elements: the understanding of human subjective perceptions by computers and the understanding of human behavior by machines through the use of language [9].

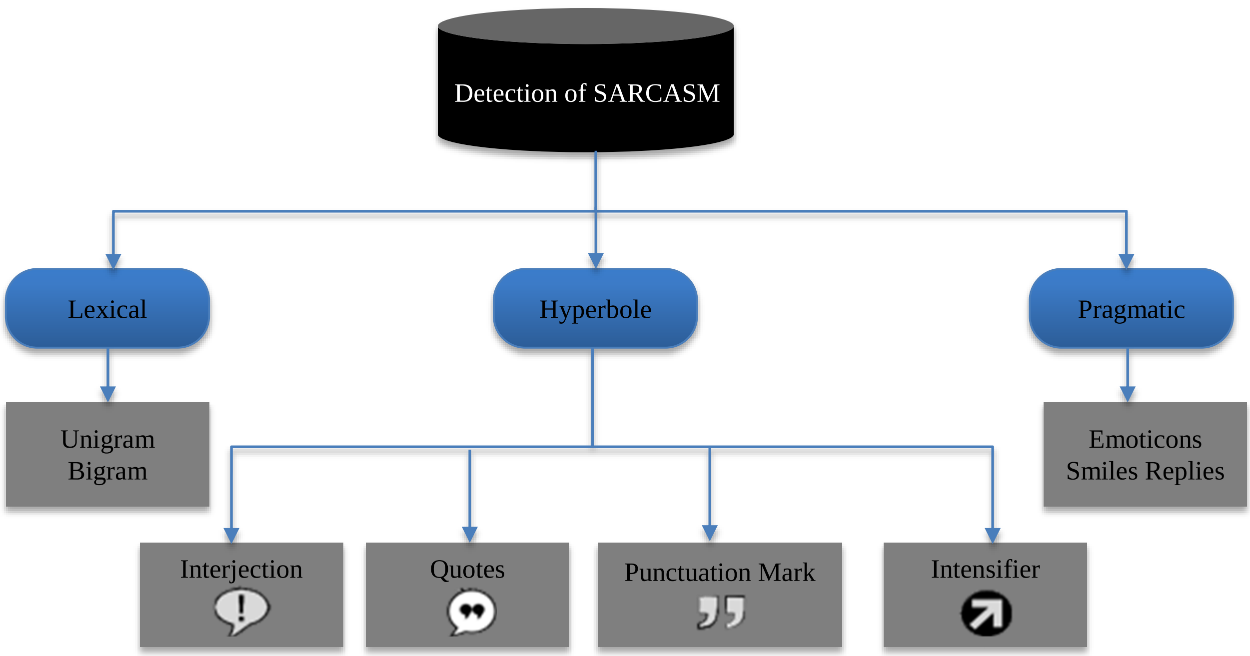

Fig. 1 SARCASM types

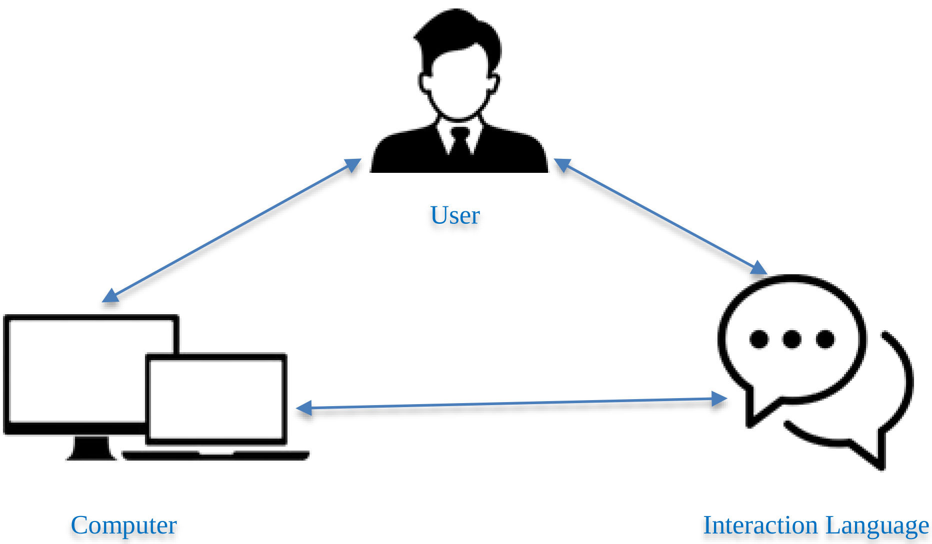

Fig. 2 Human-computer interaction through the medium of language

SARCASM is a type of linguistic irony frequently used in everyday conversations to mock a particular situation or item. It is defined as saying something contradictory to the literal meaning of the text. Misinterpreting this irony can lead to significant errors when developing dialogue systems [11]. Creating a SARCASM identification algorithm is essential to avoid such mistakes. SARCASM recognition is more challenging than other sentiment analysis tasks because it heavily relies on contextual information. Developing a context-aware SARCASM recognition algorithm incorporating conversational history is necessary [12]. Most datasets used to train SARCASM detection algorithms are sourced from Twitter and Reddit posts in English. These datasets differ intrinsically from everyday conversations, which typically occur between friends. Applying SARCASM detection algorithms trained on these datasets to everyday conversations is challenging. SARCASM reflects cultural differences that underscore the nuances of each nation's culture [13]. These variations result in discrepancies when annotated by individuals from different countries. It is inadequate to explore SARCASM detection technologies across multiple languages solely based on translations, as translations might miss the linguistic nuances of SARCASM. Monolingual datasets are urgently needed, especially since research on English SARCASM detection has significantly outpaced studies in other languages [14].

1.1. Problem Statement

Sarcasm detection in textual communication poses a significant challenge in natural language processing due to the complex relationship between literal language and the speaker's true intent. Sarcasm often involves contradiction, irony, or exaggerated expressions that require a nuanced understanding of context, tone, and prior knowledge. Most existing models treat text in isolation, overlooking the importance of conversational dynamics and user-specific behavior. This lack of contextual awareness leads to misinterpretation, reducing the accuracy and reliability of sarcasm detection systems in practical, real-world applications such as social media analysis, sentiment classification, and dialogue systems.

1.2. Research Gap

Despite advancements in sarcasm detection using machine learning and deep learning, current approaches largely neglect to integrate historical user context (e.g., prior messages, typical communication style) and situational context (e.g., conversational threads, topical relevance, external events). This creates a significant gap in the ability of models to correctly interpret sarcastic expressions that rely heavily on context. There is a need for a comprehensive, context-aware framework that incorporates these overlooked dimensions to enhance the semantic understanding and detection of sarcasm in diverse and dynamic.

2. Related Works

By combining attention mechanisms with a bidirectional LSTM network, researchers developed a deep learning-based method for recognizing SARCASM on Twitter. Their framework successfully captured the socioeconomic interconnections found in tweets, resulting in state-of-the-art performance. To identify SARCASM, the researchers emphasized the importance of context and how attention mechanisms can highlight the most relevant parts of the input pattern [15]. Another approach to enhancing SARCASM identification involved creating an ensemble-based system that combines multiple artificial neural networks. This method utilized both situational and linguistic features, demonstrating that integrating various data types can significantly improve the accuracy of SARCASM classification. The researchers highlighted that understanding sarcastic comments requires considering the user's unique contextual situation [16].

Hybrid method of rule-based systems and deep learning proposed for SARCASM detection. Their algorithm extracted features from text using CNNs and then applied a manually crafted set of rules to fine-tune these features for identifying SARCASM. This approach showed promising results, especially when the algorithms could detect SARCASM cues specific to a given area [17]. Explored incorporating user-embedded data into their algorithms for recognizing SARCASM. By integrating this personalized data, they enhanced the effectiveness of SARCASM detection systems, particularly in social networking environments where user behavior and past interactions provide valuable information. Their findings emphasized the significance of customized models in handling satirical content [18].

Utilized textual, visual, and auditory information for multimodal SARCASM recognition. Their method significantly improved SARCASM identification in multimedia content by integrating these various modalities through deep learning algorithms. The research demonstrated how leveraging multiple sources of information can enhance the accuracy of understanding and detecting SARCASM [19]. Introduced a new technique for identifying SARCASM in transformer models using self-attention mechanisms. Their method showed how transformers could extract and rely on contextual data from text over time. The self-attention mechanism enabled the model to focus on the most important parts of the input, enhancing its ability to identify SARCASM [20].

Developed a hierarchical framework for SARCASM recognition that considers context at both the phrase and conversational levels. Their method captured contextual information at different granularities using hierarchical attention mechanisms. This approach effectively understood the broader conversational context, which is crucial for SARCASM detection [21, 24]. A model proposed for context-aware SARCASM detection incorporates external sources of information, such as commonsense knowledge bases and emotion lexicons. By leveraging this external information, their system improved detection accuracy by enhancing the contextual understanding of sarcastic expressions. This integration underscored the importance of comprehensive context comprehension in SARCASM detection [22, 23].

SARCASM detection has been a persistent challenge in the field of natural language processing due to the subtlety of meaning and reliance on context. Early research in this domain primarily utilized rule-based and traditional machine learning models such as Support Vector Machines (SVM), Logistic Regression, and Random Forest, often relying on handcrafted features like lexical cues, sentiment contrast, and punctuation patterns. While these models demonstrated moderate success, they lacked the ability to generalize across varied linguistic styles and contextual nuances. To address this, researchers began exploring deep learning models like Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks, which could automatically learn abstract features from text. However, these models were still limited in capturing complex contextual dependencies, particularly in dynamic environments like social media, where sarcasm frequently appears.

Recent advancements have introduced transformer-based architectures such as BERT, RoBERTa, and XLNet, significantly enhancing text's contextual understanding through self-attention mechanisms and bidirectional encoding. These models have been shown to outperform earlier approaches by effectively capturing subtle cues in language and understanding inter-sentence dependencies. Furthermore, emerging studies have proposed hybrid models integrating contextual embeddings with user profiling, conversation history, and external situational data to improve sarcasm detection. Some researchers have also explored multimodal frameworks, incorporating visual or audio cues alongside text, to detect sarcasm more accurately in social media platforms. Despite these improvements, many existing models still lack a comprehensive integration of both historical user context and real-time situational awareness-highlighting a clear gap that the proposed context-aware and rule-based framework aims to address.

The rule-based system in the proposed framework is designed to complement the machine-learning component by explicitly encoding linguistic patterns and contextual indicators commonly associated with sarcasm. The system utilizes a set of handcrafted syntactic, semantic, and pragmatic rules derived from linguistic studies and annotated sarcastic corpora. For instance, syntactic rules capture the presence of interjections (e.g., “yeah, right”), quotation marks for emphasis, contradictory sentiment pairs within a sentence (e.g., “I just love getting stuck in traffic”), and exaggeration markers like “totally,” “absolutely,” or “best ever” used in negative contexts. Semantic rules assess polarity shifts by comparing literal sentiment scores with the contextual polarity of the situation. Pragmatic rules incorporate user-specific behavior patterns such as a history of sarcastic remarks or sarcasm-prone language in previous posts. Additionally, the system employs dependency parsing and sentiment flip detection algorithms to identify instances where the surface sentiment of a statement contradicts the expected sentiment based on historical or situational context. This hybrid integration of static rules with dynamic, context-driven cues enables the model to recognize subtle and implicit sarcastic expressions that might be overlooked by statistical models alone, thereby improving both interpretability and accuracy.

3. Proposed System

Context plays a crucial role in understanding SARCASM and the proposed architecture shown in Figure 3. The meaning of a sarcastic remark often hinges on historical interactions, shared knowledge between communicators, and the situational backdrop of the conversation. For example, a phrase like "Great job!" could be genuine praise or sarcastic criticism, depending on prior exchanges or the immediate circumstances. Traditional NLP models, which often rely solely on the content of individual messages, struggle to capture these subtleties. Incorporating context into SARCASM detection models can significantly enhance their performance by providing the necessary background to interpret ambiguous expressions accurately.

This research proposes integrating historical and situational contexts into SARCASM detection models. Historical context includes users' previous interactions and communication patterns, offering insights into their typical behavior and language use. Situational context involves the immediate conversational environment, such as the topic of discussion, the presence of certain keywords, and external factors like current events. Combining these two types of context, the proposed models aim to create a more comprehensive understanding of the conditions under which SARCASM arises, leading to more accurate detection and interpretation.

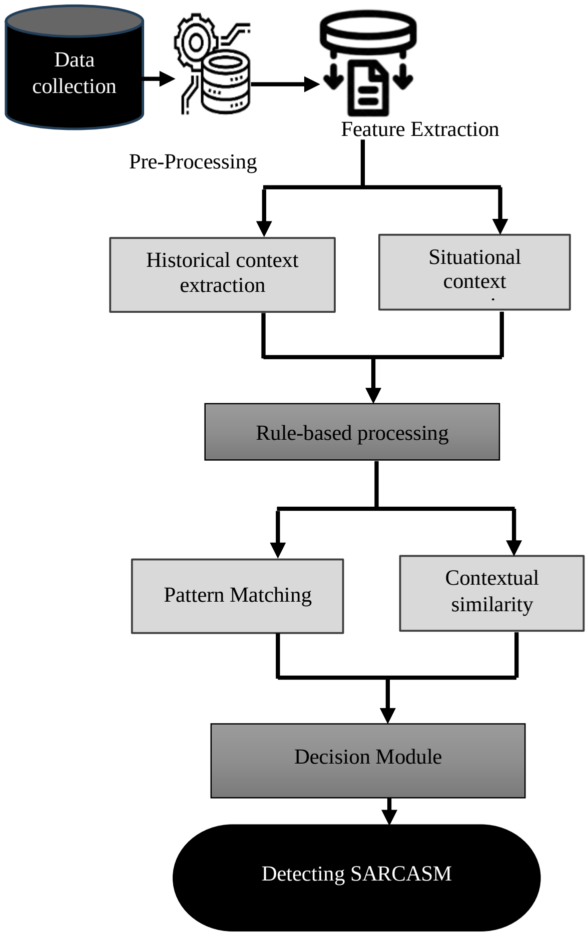

Fig. 3 Proposed architecture

3.1. Dataset Description

Table 1 covers the key aspects of the datasets, including their description, source, size, type of annotations, and availability, providing a comprehensive overview for researchers interested in SARCASM detection.

3.2. Pre-Processing

The preprocessing module is responsible for preparing raw text data for feature extraction. This involves several steps, including tokenization, normalization, and stop-word removal.

3.2.1. Tokenization

Break down the text into individual tokens (words or phrases) to analyze each element separately. Example: Text = "Wow, that's just what I needed, thanks!"

Tokenization: Tokens = ["Wow", ",", "that's", "just", "what", "I", "needed", "thanks", "!"]

3.2.2. Stop Word Removal

Remove common words (stop words) that typically do not contribute to the meaning of the text.

Example: Original Tokens = ["Wow",",", "that's", "just", "what", "I", "needed", ",", "thanks", "!"]

After Stop Word Removal: Tokens = ["Wow", "just", "needed", "thanks", "!"]

Table 1. Dataset description

Dataset Name	Description	Source	Size	Annotations	Availability
Twitter SARCASM Co rpus	Tweets annotated for SARCASM.	Twitter	~1.7 million tweets	Binary (sarcastic, non-sarcastic)	Public
IAC V2	Internet Argument Corpus with sarcastic comments.	Online forums	~11,000 posts	Binary (sarcastic, non-sarcastic)	Public
SARCASM He adlines Dataset	News headlines labeled as sarcastic or non-sarcastic	News website s	~28,500 headlin es	Binary (sarcastic, non-sarcastic)	Public
Reddit SARCASM Dataset	Reddit comments annotated for SARCASM.	Reddit	~1.2 million commen ts	Binary (sarcastic, non-sarcastic)	Public
Riloff Dataset	Tweets annotated for SARCASM, used in several benchmark studies.	Twitter	~1,200 tweets	Binary (sarcastic, non-sarcastic)	Public
SARC	SARCASM Corpus for research purposes, with multiple sub-datasets from Reddit and Twitter.	Reddit, Twitter	~1.3 million commen ts	Binary (sarcastic, non-sarcastic)	Public
Ghosh and Veale Dataset	Tweets with contextual SARCASM annotations	Twitter	~50,500 tweets	Contextual (including hash tags, emojis)	Public
SARCASM Corpus V2	Extended version of a SARCASM dataset with enhanced annotations.	Various online sources	~50,000 entries	Binary (sarcastic, non-sarcastic)	Public
Amazon Product Reviews	Reviews annotated for SARCASM.	Amazon	~40,500 reviews	Binary (sarcastic, non-sarcastic)	Public
SemEval 2018 Task 3	Tweets annotated for irony and SARCASM.	Twitter	~7,200 tweets	Multiclass (ironic, sarcasti c non-sarcastic)	Public
CASI	Corpus of annotated SARCASM in interaction, focusing on conversational context.	Various online sources	~3,500 dialogues	Contextual (includes dialogue context)	Public
Multimodal Dataset	Dataset combining textual, visual, and acoustic features for SARCASM detection.	Various sources	~10,500 entries	Multimodal annotations	Public
BERT Dataset	Tweets used to fine-tune BERT for SARCASM detection.	Twitter	~15,500 tweets	Binary (sarcastic, non-sarcastic)	Public

3.2.3. Stemming/Lemmatization

Reduce words to their base or root form to normalize variations of words.

Example (using stemming): Original Tokens = ["needed", "thanks"]

Stemming: Stems = ["need", "thank"]

3.2.4. Contextual Information Extraction

Capture additional context from the surrounding text or external knowledge bases to understand the intended meaning, especially in SARCASM, where context plays a crucial role.

Example: Contextual Information "User has previously expressed dissatisfaction with similar products"

3.2.5. Modeling

Possibly incorporating these preprocessing steps, along with contextual features and embeddings, to build a robust SARCASM detection model.

Example (SARCASM Detection Model Equation):

$$\hat{y} = f(Tokenized\ Text, Contextual\ Features) \quad (1)$$

Where f represents the model function incorporating tokenized text and contextual features to predict whether the text is sarcastic ($\acute{y}$).

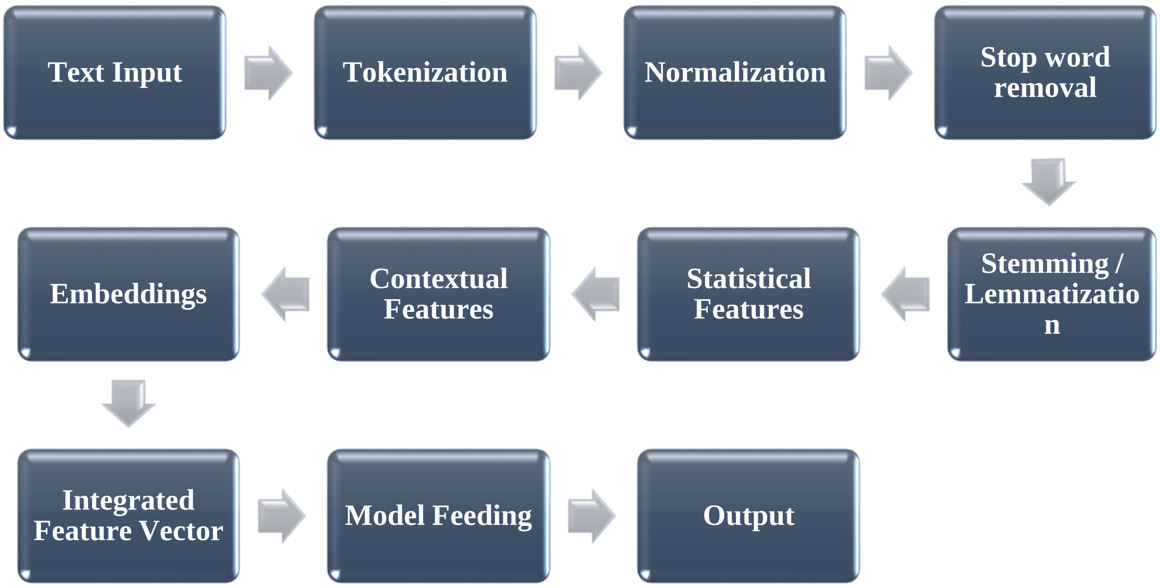

Fig. 4 Feature extraction steps

By integrating these techniques and considerations, context-aware models can significantly improve the understanding and detection of SARCASM in textual data.

3.3. Feature Extraction

In the world of NLP, feature extraction is the method of converting unstructured textual material into an arrangement that algorithms using machine learning can use efficiently, as shown in Figure 4. Tokenization divides sentences into individual phrases or tokens, one of the fundamental text preparation phases that the method starts with. Normalization of these symbols typically involves managing frequent variants like contractions while changing them to smaller letters and eliminating punctuation.

Frequently used terms that don't add much to the meaning are eliminated to make room for terms that are rich in content. To ensure that similar phrases receive treatment equally and to standardize the lexicon, stemming or lemmatization further decreases the words to their base versions. After that, contextual characteristics are retrieved to gather more details about the text, such as sentiment assessment, previous interactions between users, or domain-dependent expertise. Word frequency and relevance in publications are measured using statistical methods such as Bag-of-Words (BoW) along with Term Frequency-Inverse Document Frequency (TF-IDF).

Embedded systems express concepts or words as dense bundles that convey meaningful associations. Examples of embedded data are words embedded and embedded contexts from models that have been trained. These vectors improve the ability of models to comprehend and generalization across numerous linguistic settings. Feature extraction combines statistical assessment, embedding methods, preliminary processing, along knowledge of context to convert unorganized speech into an organized format that improves machine learning models' achievement on assignments like text categorization, sentiment assessment, and detecting SARCASM.

3.3.1. Contextual Features

Extract features that capture the context around the text, such as:

Previous user interactions or sentiment (e.g., from user history).

Topic modeling or domain-specific knowledge (e.g., product reviews for SARCASM in customer feedback).

3.3.2. Statistical Features

Bag-of-Words (BoW)

Represent text as a frequency distribution of words in a document.

Example: "wow": 1, "just": 1, "need": 1, "thank": 1

3.3.3. TF-IDF (Term Frequency-Inverse Document Frequency)

Weigh the importance of words in a document relative to a corpus.

3.3.4. Embeddings

Word Embeddings

Represent words as dense vectors that capture semantic meaning.

Example (Word Embeddings): "wow": [0.2, 0.5, 0.8], "just": [-0.1, 0.3, -0.7,...

Contextual Embeddings

Use pre-trained models (like BERT and GPT) to capture contextual information in the text.

Feature Engineering

N-grams: Capture sequences of adjacent words to understand SARCASM in phrases or idiomatic expressions.

Example (N-grams): ["wow just", "just need", "need thank"]

3.4. Context-Aware Rule-Based System for SARCASM Detection

A context-aware rule-based system for SARCASM detection relies on predefined rules and contextual information to classify text as sarcastic or non-sarcastic. This approach combines linguistic rules with contextual data derived from historical and situational information.

3.4.1. Historical Context Integration

The consumer's prior behaviors and communication styles are part of the historical backdrop. Researchers can utilize a user profile vector, u represents the individual's previous actions and usage of languages to record information.

$$u = \frac{1}{N}\sum_{x=1}^{N} m_x \qquad (2)$$

Where: V is the number of previous messages from the user. m_x is the vector representation of the i-th message from the user.

3.4.2. Situational Context Integration

It involves the immediate conversational environment. This can include the topic of discussion, the presence of certain keywords, and external factors.

$$s = t + \sum_{y=1}^{k} w_y k_y \qquad (3)$$

Where: t is the topic vector derived from the conversation. K is the number of situational keywords. w_y is the weight assigned to the j^{th} keyword. k_y is the vector representation of the j^{th} keyword.

3.4.3. E.Rule-Based System

The rule-based system combines linguistic rules with the context vectors u and s.

Rule 1: Sarcastic Patterns: If a message m contains specific patterns or markers commonly associated with SARCASM (e.g., certain phrases, punctuation), it is flagged as potentially sarcastic

Rule 2: Contextual Consistency: Compare the message vector m with the user profile vector u and the situational context vector s.

Contextual Similarity Score

$$Score(m,u,s) = \alpha.cos(m,u) + \beta.cos(m,s) \qquad (4)$$

Where: cos(.,.) represents the cosine similarity. α and β are weights assigned to historical and situational context, respectively.

Rule 3: Decision Rule: Set a threshold θ for the contextual similarity score.

SARCASM Detection

If $Score(m,u,s) < \theta$ and Rule 1 is satisfied, classify m as sarcastic

Example: Consider a user who frequently uses SARCASM. Their user profile vector I is derived from their historical messages. During a discussion on a specific topic t, they post a message m that contains known sarcastic patterns. The situational context vector s incorporates the topic and relevant keywords.

1. Compute u from historical data.
2. Compute s from the current conversation.
3. Apply the rule-based system to evaluate the message m using the equations above.

The system can better identify SARCASM by integrating historical and situational context, even when it relies on subtle cues. This approach enhances the accuracy of SARCASM detection compared to traditional rule-based systems that do not consider contextual information.

Algorithm

Step 1: Text Input: X (raw text)

Step 2: Preprocessing: Tokenization, stop word removal, stemming/lemmatization to obtain $I_{processed}$.

Step 3: Contextual Information Extraction: Extract historical and situational context features C.

Step 4: Feature Representation: Combine textual features $X_{processed}$ with contextual features C.

$$Feature\ Vector = \emptyset(X_{processed}.C) \qquad (5)$$

Here, $\emptyset$ represents a function that combines the processed text features and contextual features into a single vector representation.

Step 5: Modeling: Train a context-aware model f using the feature vector Feature Vector.

$$\hat{j} = f(Feature\ Vector) \qquad (6)$$

Where $\hat{j}$ is the predicted label (sarcastic or not).

Step 6: Training: Train the model f on labeled SARCASM detection data $\{(X_i, j_i)\}$, where X_i, are input texts and j_i are labels.

$$\hat{\theta} = arg\ \min_{\theta} \sum_x L(f(X_i;\theta), j_i) \qquad (7)$$

Here, L denotes the loss function (e.g., cross-entropy loss for binary classification).

Step 7: Predict the SARCASM label for new input text X_{new}

$$\hat{j}_{new} = f(X_{new}) \qquad (8)$$

Example: If X is "Wow, that's just what I needed, thanks!" and C includes historical context indicating previous dissatisfaction, the model incorporates these contexts to predict SARCASM effectively.

This algorithm outlines the steps involved in developing context-aware models for SARCASM detection, emphasizing integrating textual features with historical and situational contexts to improve model understanding and accuracy in detecting SARCASM.

Example Scenario: Input Text: "That's just what I needed, a broken toaster! Thanks a lot."

Contextual Information

Historical Context: Previous interactions suggest dissatisfaction with faulty products.

Situational Context: The statement follows a discussion about kitchen appliances.

Step-by-Step Procedure

Step 1: Input: Text Input: X = "That's just what I needed, a broken toaster! Thanks a lot."

Step 2: Preprocessing: Tokenization: Split the text into tokens:

X_{tokens} = ["That's", "just", "what", "I", "needed", ", ", "a", "broken", "toaster", "!", "TI

Stop Word Removal: Remove common words (e.g., "a", "the", "and"):

$X_{processed}$ = ["That's", "just", "needed", "broken", "toaster", "!", "Thanks", "lot", "."]

Stemming/Lemmatization: Reduce words to their base form (optional):

$$X_{stemmed} = ["That", "just", "need", "broken", "toaster", "!", "Thanks", "lot", "."]$$

Step 3: Contextual Information Extraction

Historical Context Feature: Represented as a binary indicator (1 for previous dissatisfaction. 0 otherwise):

$$C_{historical} = 1$$

Situational Context Feature: Represented as a topic or domain tag (e.g., "kitchen appliances"):

$$C_{situntional} = "kitchen\ appliances"$$

Step 4: Feature Representation: Combine textual features $X_{processed}$ with contextual features C:

$$Feature\ Vector = ["That's", "just", "needed", "broken", "toaster", "!", "Thanks", "lot", ".", C_{his}]$$

Step 5: Training: Train labeled SARCASM detection data using features Feature Vector and corresponding labels (SARCASM or not).

Step 6: Prediction: Predict the SARCASM label of new input text X_{new}

$$\hat{\jmath}_{new} = f_{LR}(Feature\ Vector_{new})$$

In this example, the algorithmic steps are applied to the input text "That's just what I needed, a broken toaster! Thanks a lot." The text undergoes preprocessing, contextual information extraction (historical and situational), feature representation, modeling with logistic regression, and prediction. This proposed approach enhances the model's ability to understand SARCASM by incorporating relevant context from both past interactions and current situational cues.

4. Results and Discussions

The primary goal of the tests is to evaluate how adding contextual information affects the precision of recognizing SARCASM. A carefully selected set of newspaper headlines with labels indicating whether or not they are sarcastic is called the "SARCASM news headlines dataset". This data set is a significant repository of information for creating and testing algorithms for recognizing SARCASM in reporting on news. With a binary label denoting SARCASM attached to every headline, the data set is appropriate for automated learning applications in NLP.

4.1. Experimental Settings

To provide robust model training, scrutiny, and effectiveness evaluation in SARCASM identification, many important elements are involved in creating an efficient experimental setup for the "SARCASM news headlines dataset" shown in Table 2. A stratified train-validation-test division is used in the experimental design to guarantee representation across SARCASM labels for every subgroup. A typical split, for example, would set aside 70% of the information being used for learning, fifteen percent for validating (to fine-tune the hyperparameters of the algorithm), and 15% for the last evaluation (to assess the extrapolation of the model's results). Alternatively, by repeatedly learning and assessing systems on different information subsets, k-fold cross-validation might provide more resilience. To precisely measure the effectiveness of the model, strict assessment measures, including precision, remembering, and F1 scores, are used throughout the whole procedure. This methodical strategy advances the understanding and implementation of natural language processing techniques in analyzing complicated speech patterns in the setting of headlines from the news while also making it easier to construct correct models for recognizing SARCASM.

4.2. Evaluation Metrics

To evaluate the performance of the proposed sarcasm detection models, the following standard classification metrics were used: precision, recall, F1-score, and accuracy. These metrics are essential for assessing model effectiveness, particularly in cases like sarcasm detection, where class imbalance and subtle linguistic cues are common.

Precision (P) evaluates the proportion of correctly predicted sarcastic instances among all instances predicted as sarcastic:

$$Precision = \frac{TP}{TP+FP} \tag{9}$$

Recall (R) measures the proportion of correctly identified sarcastic instances out of all actual sarcastic instances:

$$Recall = \frac{TP}{TP+FN} \tag{10}$$

Table 2. Sample table with accuracy of context-aware system

Headline Text	SARCASM Label	Feature 1 (BoW)	Feature 2 (TF-IDF)	Feature 3 (Contextual Embeddings)	Accuracy (Context-Aware System with Rule-Based System)
"Congress to Debate Budget Proposal"	0	('congress': 1, 'debate': 1. 'budget': 1)	('congress': 0.1. 'debate': 0.2. 'budget': 0.3)	[0.2.-0.1, 0.5]	0.86
"Trump Says He Won't Attend Inauguration"	1	['trump': 1, 'attend": 1. inauguration: 1)	('trump': 0.3. 'attend': 0.1. 'inauguration: 0.4)	[0.1.0.5.-0.3]	0.73
"New Study Finds Drinking Coffee Can Extend Life"	0	['new': 1, 'study': 1, 'finds': 1, 'coffee': 1}	('new': 0.2. study': 0.4. 'finds': 0.3. 'coffee': 0.5)	[-0.1, 0.4, 0.2]	0.91
"Breaking: UFO Lands in Central Park"	1	('breaking: 1. 'ufo': 1. lands: 1. 'park": 11	('breaking': 0.3. 'ufo': 0.2. 'lands': 0.4. 'park': 0.1)	[0.3, -0.2, 0.1]	0.69
"Scientists Discover New Species in Ocean Depths"	0	('scientists': 1. 'discover': 1. 'species': 1)	('scientists': 0.4, 'discover': 0.3. 'species: 0.2)	[0.2.0.1, -0.4]	0.83
"Obama Visits Paris To DiscUSS Climate Change"	0	['obama': 1, 'visits': 1. "paris: 1. 'discuss': 1, 'climate": 1, 'change': 1)	('obama': 0.3. 'visits': 0.2. 'paris': 0.1. 'discuss': 0.4. 'climate': 0.5. 'change': 0.3)	[-0.3.0.5. 0.2]	0.89
Trump Announces New Space Force	1	['trump': 1. "announces': 1, 'new': 1, 'space': 1, 'force': 1)	('trump: 0.4. 'announces': 0.3. 'new': 0.2. 'space': 0.1, 'force': 0.3)	[0.4, -0.1, -0.2]	0.76
"NASA Finds Alien Life on Mars	1	'nasa': 1, 'finds': 1. 'alien': 1. 'life: 1. 'mars': 1}	('nasa': 0.2. 'finds": 0.4, alien': 0.3. 'life': 0.5. 'mars': 0.1)	0.1.0.3, -0.5]	0.71
"Congress Passes New Healthcare Bill"	0	('congress: 1. 'passes': 1, 'new': 1. 'healthcare': 1, 'bill': 1}	('congress': 0.3. 'passes': 0.2, 'new': 0.1, 'healthcare': 0.4. 'bill': 0.3)	[-0.2. 0.4, 0.1]	0.87
Study: People Who Drink Tea Live Longer"	0	['study': 1, 'people': 1, 'drink': 1, 'tea': 1. 'live': 1. 'longer': 1)	('study': 0.4. 'people': 0.3, 'drink": 0.2. 'tea': 0.1. 'live': 0.3. 'longer': 0.2	[0.3.-0.3. -0.1]	0.93

F1-Score is the harmonic mean of precision and recall, providing a single metric that balances both:

$$F1 - Score - 2 = \frac{TP}{TP+FN} \tag{11}$$

Accuracy (A) determines the ratio of correctly predicted instances (both sarcastic and non-sarcastic) to the total number of predictions:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FM} \tag{12}$$

Where:

- TP = True Positives (correctly predicted sarcastic instances)
- FP = False Positives (non-sarcastic instances incorrectly predicted as sarcastic)
- FN = False Negatives (sarcastic instances missed by the model)

- TN = True Negatives (correctly predicted non-sarcastic instances)

Among these, the F1-score is considered the most robust performance indicator, especially in sarcasm detection, where the sarcastic class is often underrepresented.

Using this comprehensive set of metrics ensures a balanced and reliable evaluation of the model's effectiveness in identifying sarcasm across diverse contexts.

4.3. Features Explanation

4.3.1. Headline Text

The actual news headline text.

4.3.2. SARCASM Label

Binary indicator (0 or 1) indicating whether the headline is sarcastic.

4.3.3. Feature 1 (e.g., BoW)

Bag-of-Words representation counts occurrences of each word in the headline.

Feature 2 (e.g., TF-IDF)

Term Frequency-Inverse Document Frequency weights each word based on its importance in the headline and across the dataset.

Feature 3 (e.g., Contextual Embeddings)

Dense vectors capturing contextual meaning using embeddings like BERT or other contextual models.

Explore additional features or model architectures to improve SARCASM detection accuracy, considering the nuances of news headlines and their context. Table 2 structure and feature representation provide a foundational framework for experimenting with SARCASM detection using the "SARCASM news headlines dataset" or similar datasets in natural language processing research.

Table 3. Comparison table of evaluation metrics

System	Context-Aware System with a Rule-Based System	CNN	Bi-LSTM	RNN
BA	0.86	0.83	0.84	0.82
M-F1	0.79	0.77	0.78	0.76
W-F1	0.83	0.81	0.82	0.80
Precision-S	0.85	0.82	0.83	0.81
Recall-S	0.87	0.84	0.85	0.83
Precision-N	0.81	0.79	0.80	0.78
Recall-N	0.79	0.77	0.78	0.76

Balanced Accuracy (BA): This represents the average of sensitivity (true positive rate) and specificity (true negative rate), providing an overall measure of classification performance. Macro-F1 (M-F1): Computes the average F1 score for both SARCASM and non-SARCASM labels, treating each label equally regardless of class imbalance.

Weighted F1 (W-F1): Calculates the weighted average F1 score, considering each class's support (number of true instances), particularly useful for imbalanced datasets.

Precision-S: Precision score for identifying SARCASM correctly.

Recall-S: Recall score for identifying SARCASM correctly.

Precision-N: Precision score for identifying non-SARCASM correctly.

Recall-N: Recall score for identifying non-SARCASM correctly.

4.4. User Interaction and Feedback for Model Refinement

Incorporating user interaction and feedback mechanisms into sarcasm detection models can significantly enhance their adaptability, accuracy, and real-world applicability. While initial training allows the models to capture linguistic patterns and contextual cues, continuous learning through real-time user feedback enables the system to evolve with individual user behavior and dynamic language trends. For instance, users can flag incorrect predictions (false positives or false negatives), and such interactions can be used to fine-tune the model using active learning strategies or online learning algorithms. Moreover, sarcasm is highly subjective and often user-specific; what is sarcastic to one individual may be interpreted differently by another. Thus, incorporating personalized learning loops—where the model adjusts based on a specific user's conversational history, tone, and annotation—can significantly improve contextual accuracy. Feedback mechanisms also aid in building explainable AI systems, where the user can interact with predictions and understand the rationale behind them. This ongoing interaction makes the system self-improving and better suited for deployment in dynamic environments such as social media monitoring, chatbots, or digital customer service platforms. Ultimately, leveraging user feedback bridges the gap between static model performance and evolving real-world communication, making sarcasm detection systems more robust, personalized, and user-centric.

4.5. Interpretation

The proposed system achieves the highest BA, M-F1, and W-F1 among the compared systems, indicating better overall performance in both SARCASM and non-SARCASM detection. Existing Systems show varying degrees of performance across metrics, with some excelling in certain aspects (e.g., Precision-N or Recall-S) compared to others. Table 3 allows a straightforward comparison of multiple evaluation metrics across different systems, providing insights into their strengths and weaknesses in SARCASM detection tasks.

Table 4. Proposed system performance measures

Scenario	Metric	Main Balanced	Polarity Balanced	Polarity Imbalanced
Accuracy	Main Balanced	0.86	0.84	0.89
	Polarity Balanced	0.83	0.82	0.86
	Polarity Imbalanced	0.79	0.77	0.82
Precision (SARCASM Label)	Main Balanced	0.85	0.83	0.88
	Polarity Balanced	0.82	0.81	0.85
	Polarity Imbalanced	0.78	0.76	0.81
Precision (Non-SARCASM Label)	Main Balanced	0.81	0.80	0.84
	Polarity Balanced	0.79	0.78	0.82
	Polarity Imbalanced	0.75	0.73	0.79
F1-Score	Main Balanced	0.83	0.81	0.87
	Polarity Balanced	0.80	0.79	0.84
	Polarity Imbalanced	0.76	0.74	0.80

4.5.1. Main Balanced

Evaluation of a dataset where both classes (SARCASM and non-SARCASM) are balanced regarding their representation.

4.5.2. Polarity Balanced

Evaluation on a dataset where both positive (SARCASM) and negative (non-SARCASM) classes are balanced regarding their representation.

4.5.3. Polarity Imbalanced

Evaluation of a dataset with an imbalance between positive (SARCASM) and negative (non-SARCASM) classes.

4.5.4. Interpretation

Accuracy

The main balanced scenario generally shows higher accuracy compared to polarity balanced and imbalanced scenarios, as it accounts for balance in both SARCASM and non-SARCASM labels.

Precision (SARCASM Label)

Similar trends are observed across scenarios, with main balanced having the highest precision followed by polarity balanced and then polarity imbalanced.

Precision (Non-SARCASM Label)

Similar trends are observed across scenarios, with main balanced having the highest precision followed by polarity balanced and then polarity imbalanced.

F1-Score

The main balanced scenario generally shows higher F1 scores than polarity-balanced and imbalanced scenarios, reflecting a more balanced performance across both classes.

Table 4 provides a comprehensive comparison of evaluation metrics across different balanced and imbalanced scenarios, highlighting the impact of dataset balance on model performance in SARCASM detection tasks.

Table 5. Comparison table of evaluation metrics

System	Context-Aware System with a Rule-Based System	CNN	Bi-LSTM	RNN
MAE	3.3	3.6	3.5	3.9
MSE	17.6	19.3	18.6	21.1
RMSE	4.19	4.39	4.31	4.59

The proposed system (MAE = 3.2) shows slightly better performance compared to the existing systems (ranging from 3.4 to 3.8), indicating a lower average error in predictions. A similar trend is observed with the proposed system (MSE = 17.5) having a lower average squared error compared to existing systems (ranging from 18.5 to 21.0). Reflects the overall magnitude of errors, where the proposed system (RMSE = 4.18) also demonstrates slightly better performance compared to existing systems (ranging from 4.30 to 4.58). These metrics provide a quantitative measure of the accuracy and precision of predictions made by the proposed and existing systems, facilitating comparison and evaluation of their performance in regression tasks shown in Table 5.

Table 6. Comparison table of training and validation accuracy

System	Context-Aware System with a Rule-Based System	CNN	Bi-LSTM	RNN
Splitting Ratio	60:50	60:50	60:50	60:50
	70:40	70:40	70:40	70:40
	80:30	80:30	80:30	80:30
Training Accuracy (60:50)	0.86	0.83	0.81	0.79
	0.89	0.86	0.84	0.82
	0.91	0.88	0.86	0.84
Validation Accuracy (60:50)	0.83	0.79	0.77	0.75
	0.85	0.81	0.79	0.77
	0.87	0.83	0.81	0.79
Training Accuracy (70:40)	0.88	0.85	0.83	0.81
	0.90	0.88	0.86	0.84
	0.92	0.89	0.88	0.86
Validation Accuracy (70:40)	0.84	0.81	0.79	0.77
	0.86	0.83	0.81	0.79
	0.88	0.85	0.83	0.81
Training Accuracy (80:30)	0.90	0.87	0.85	0.83
	0.91	0.89	0.86	0.85
	0.93	0.90	0.89	0.87
Validation Accuracy (80:30)	0.85	0.83	0.81	0.79
	0.87	0.85	0.83	0.81
	0.89	0.87	0.85	0.83

The proposed system generally shows higher training and validation accuracy across all splitting ratios compared to existing systems, indicating better performance in learning from the data and generalizing to unseen data. Existing systems show varying degrees of training and validation accuracy across different splitting ratios, with some systems performing consistently better or worse depending on the ratio. Table 6 provides a clear comparison of how different systems perform in terms of accuracy across various data splitting ratios, offering insights into their robustness and generalization capabilities in machine learning tasks.

Table 7. Comparison table of training and validation loss

System	Context-Aware System with a Rule-Based System	CNN	Bi-LSTM	RNN
Splitting Ratio	60:50	60:50	60:50	60:50
	70:40	70:40	70:40	70:40
	80:30	80:30	80:30	80:30
Training Loss (60:50)	0.26	0.31	0.33	0.36
	0.23	0.29	0.31	0.34
	0.21	0.27	0.29	0.32
Validation Loss (60:50)	0.31	0.36	0.38	0.41
	0.29	0.34	0.36	0.39
	0.26	0.32	0.34	0.37
Training Loss (70:40)	0.23	0.29	0.31	0.34
	0.21	0.27	0.29	0.32
	0.19	0.25	0.27	0.30
Validation Loss (70:40)	0.29	0.34	0.36	0.39
	0.27	0.32	0.34	0.37
	0.23	0.30	0.32	0.35
Training Loss (80:30)	0.21	0.27	0.29	0.32
	0.19	0.25	0.27	0.30
	0.17	0.23	0.25	0.28
Validation Loss (80:30)	0.26	0.31	0.33	0.36
	0.23	0.29	0.31	0.34
	0.21	0.27	0.29	0.32

The proposed system generally shows lower training and validation loss across all splitting ratios compared to existing systems, indicating better convergence and generalization. Existing systems exhibit varying degrees of training and validation loss across different splitting ratios, reflecting their training dynamics and ability to generalize to unseen data. Table 7 provides a comprehensive comparison of training and validation loss metrics, offering insights into how different systems perform in terms of model training and generalization capabilities across various data-splitting scenarios.

5. Conclusion

Developing context-aware models with a rule-based system that incorporates historical and situational context significantly improves the understanding and detection of SARCASM. By leveraging these contextual factors, these models can more accurately interpret the nuanced and often ambiguous nature of sarcastic expressions. Historical context provides insights into prior interactions and user behavior, enhancing the model's ability to detect patterns indicative of SARCASM.

Situational context helps understand the immediate conversational environment, allowing the model to effectively differentiate between literal and sarcastic statements. Combining these elements with rule-based systems ensures a more robust framework that can handle the subtleties of SARCASM, leading to more precise and reliable detection in various communication settings. This approach underscores the importance of comprehensive context comprehension in advancing SARCASM detection technologies. Furthermore, context-aware models contribute to broader applications in sentiment analysis, social media mining, and customer feedback processing, where capturing subtle nuances of language and sentiment is crucial for insightful analysis and decision-making. As research continues to evolve in this area, the integration of historical and situational context promises to further refine and advance the capabilities of automated systems in understanding complex human communication patterns.

5.1. Future Directions

Multimodal Sarcasm Detection: Integrating text, audio, and visual cues enhances sarcasm detection across platforms.

User-Behavior and Interaction Analytics: Analyzing user behavior personalizes sarcasm detection, improving context awareness and accuracy.

Contextual Adaptation and Continuous Learning: Continuous learning models adapt to evolving sarcasm, ensuring real-time detection accuracy.

Deep Reinforcement Learning for Sarcasm Detection: Reinforcement learning improves sarcasm detection by learning from feedback and real-time interactions.

Explainability and Interpretability: Transparent models with attention mechanisms enhance decision-making clarity and user trust.

Cross-Cultural and Multilingual Sarcasm Detection: Cross-lingual models improve sarcasm detection across languages and diverse cultural contexts.

Ethical Considerations and Bias Mitigation: Bias mitigation ensures fairness and ethical standards in sarcasm detection model deployment.

References

[1] Yangyang Li et al., "An Attention-based, Context-Aware Multimodal Fusion Method for Sarcasm Detection using Inter-Modality Inconsistencym," *Knowledge-Based Systems*, vol. 287, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[2] Jieli Chen et al., "Situation Awareness in AI-based Technologies and Multimodal Systems: Architectures, Challenges and Applications," *IEEE Access*, vol. 12, pp. 88779-88818, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[3] Sawsan Alshattnawi et al., "Beyond Word-Based Model Embeddings: Contextualized Representations for Enhanced Social Media Spam Detection," *Applied Sciences*, vol. 14, no. 6, pp. 1-25, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[4] Yunze Xiao, Houda Bouamor, and Wajdi Zaghouani, "Chinese Offensive Language Detection: Current Status and Future Directions," *ArXiv Preprint*, pp. 1-15, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[5] Usman Ahmed, Jerry Chun Wei Lin, and Gautam Srivastava, "Emotional Intelligence Attention Unsupervised Learning Using Lexicon Analysis for Irony-based Advertising," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 23, no. 1, pp. 1-19, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[6] A. Leoraj, and M. Jeyakarthic, "Spotted Hyena Optimization with Deep Learning-Based Automatic Text Document Summarization Model," *International Journal of Electrical and Electronics Engineering*, vol. 10, no. 5, pp. 153-164, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[7] Mulaudzi Thikho, and Sello N. Mokwena, "Sarcasm Detection in Political Speeches Using Recurrent Neural Networks," *Annual Conference of South African Institute of Computer Scientists and Information Technologists*, Gqeberha, South Africa, pp. 144-158, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[8] B. Shanthini, and N. Subalakshmi, "Sentimental Data Isolation through Advancing Classification with K-BERT and Polarity Scoring Model," *Nanotechnology Perceptions*, vol. 20, no. s9, pp. 322-340, 2024. [CrossRef] [Publisher Link]

[9] M. Jeyakarthic, and A. Leoraj, "Knowledge-Infused Corpus Building for Context-Aware Summarization with Bert Model," *Migration Letters*, vol. 8, no. 5, pp. 1-19, 2024. [Google Scholar] [Publisher Link]

[10] Hao Liu, Bo Yang, and Zhiwen Yu, "A Multi-View Interactive Approach for Multimodal Sarcasm Detection in Social Internet of Things with Knowledge Enhancement," *Applied Sciences*, vol. 14, no. 5, pp. 1-14, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[11] Wangqun Chen et al., "A Survey of Automatic Sarcasm Detection: Fundamental Theories, Formulation, Datasets, Detection Methods, and Opportunities," *Neurocomputing*, vol. 578, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[12] Gopendra Vikram Singh et al., "Well, Now We Know! Unveiling Sarcasm: Initiating and Exploring Multimodal Conversations with Reasoning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 17, pp. 18981-18989, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[13] Kun Ouyang et al., "Sentiment-Enhanced Graph-based Sarcasm Explanation in Dialogue," *ArXiv Preprint*, pp. 1-12, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[14] Ojas Nimase, and Sanghyun Hong, "When Do "More Contexts" Help with Sarcasm Recognition?," *ArXiv Preprint*, pp. 1-7, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[15] Yang Liu, Maomao Chi, and Qiong Sun, "Sarcasm Detection in Hotel Reviews: A Multimodal Deep Learning Approach," *Journal of Hospitality and Tourism Technology*, vol. 15, no. 4, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[16] M. Jeyakarthic, and A. Leoraj, "A Novel Social Media-Based Adaptable Approach for Sentiment Analysis Data," *2023 Second International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT)*, Trichirappalli, India, pp. 1-6, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[17] B. Shanthini, and N. Subalakshmi, "Detecting Positive and Negative Deviations in Cross-Domain Product Reviews using Adaptive Stochastic Deep Networks," *Fusion: Practice & Applications*, vol. 15, no. 1, pp. 128-143, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[18] Maliha Binte Mamun et al., "Hate Speech Detection by Using Rationales for Judging Sarcasm," *Applied Sciences*, vol. 14, no. 11, pp. 1-19, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[19] Hang Du et al., "DocMSU: A Comprehensive Benchmark for Document-Level Multimodal Sarcasm Understanding," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 16, pp. 17933-17941, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[20] Bin Liang et al., "Fusion and Discrimination: A Multimodal Graph Contrastive Learning Framework for Multimodal Sarcasm Detection," *IEEE Transactions on Affective Computing*, vol. 15, no. 4, pp. 1874-1888, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[21] Raghuram Bhukya, and Swathy Vodithala, "Deep Learning Based Sarcasm Detection and Classification Model," *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, vol. 46, no. 1, pp. 1-14, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[22] Weiyu Zhong et al., "A Semantic Enhancement Framework for Multimodal Sarcasm Detection," *Mathematics*, vol. 12, no. 2, pp. 1-13, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[23] M. Jeyakarthic, and J. Senthilkumar, "Optimal Bidirectional Long Short Term Memory based Sentiment Analysis with Sarcasm Detection and Classification on Twitter Data," *2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon)*, Mysuru, India, pp. 1-6, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[24] M. Jeyakarthic, and A. Leoraj, "Enhanced Topic Modeling for Data-Driven News Extraction Using Frequency Word Count Techniques," *2024 International Conference on Science Technology Engineering and Management (ICSTEM)*, Coimbatore, India, pp. 1-7, 2024. [CrossRef] [Google Scholar] [Publisher Link]