University Of AL-Qadisiyah

Available online at www.qu.edu.iq/journalcm

JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS

ISSN:2521-3504(online) ISSN:2074-0204(print)

Systematic review of sentiment analysis and predict sarcastic

[a]Baidaa I. Kzar, [b]Dr. Haider H. Safi *

[a]*Informatics Institute for Postgraduate Studies/ Iraqi Commission for Computers & Informatics (IIPS/ICCI), Baghdad, Iraq. Email: ms202110669@iips.icci.edu.iq*

[b]*College of Basic Education - Department of Computers, Mustansiriyah University, Baghdad, Iraq.Email: hayder.h.safi@uomustansiriyah.edu.iq*

ARTICLE INFO

Article history:
Received: 05 /05/2023
Rrevised form: 20 /06/2023
Accepted : 23 /06/2023
Available online: 30 /06/2023

Keywords:

NLP, sentiment analysis, feature extraction, machine learning, RNN.

ABSTRACT

With the rise of social media on the web, sentiment analysis has become one of the most important areas of study. Today, millions of people share their thoughts, ideas, feelings, and opinions on social media sites like Twitter and Facebook. Sentiment analysis, also called opinion mining, is mostly about classifying and predicting how people feel about a certain target. It involves putting text documents or sentences into groups based on how positive or negative they are about a certain topic. Researchers always find "natural language processing" to be one of the most interesting topics. To solve different problems and improve the accuracy of different applications, it is always helpful to know the exact meaning of what is being said in a conversation. Sentiment analysis uses natural language processing (NLP) and learning models like machine learning and deep learning algorithms to figure out how people feel about the data given. Sentiment analysis looks at sarcasm because sarcasm is a way for people to say how they feel about something without saying it directly. People means the exact opposite of what the sentence says at first glance. Sarcasm is hard to figure out because each sarcastic sentence is different. This paper will talk about what has been done in the field of sarcasm detection, the different techniques used, and the problems that still need to be solved.

MSC.

https://doi.org/10.29304/jqcm.2023.15.2.1241

1. Introduction

Twitter data analysis has become increasingly important in understanding public opinions, sentiments, and trends due to the platform's immense user base and real-time nature. With millions of tweets being generated every day, Twitter offers a wealth of information that can provide valuable insights into various domains, including marketing, politics, and social issues [1]. However, the detection of sarcasm in tweets poses a significant challenge in sentiment analysis. Sarcasm is a form of communication where the intended meaning is often opposite to the literal interpretation, making it difficult for traditional sentiment analysis methods to accurately capture the sentiment behind sarcastic statements [2]. Detecting sarcasm in tweets is crucial, as it can drastically affect the accuracy of sentiment analysis results, leading to incorrect interpretations and insights. Therefore, there is a growing need for robust and effective techniques to detect and handle sarcasm in Twitter data, enabling more accurate sentiment analysis and a deeper understanding of user sentiments on the platform [1] [3].

⁎Corresponding author

Email addresses:

Communicated by ‘sub etitor’

Much effort has gone into developing systems that can automatically recognize sarcasm, and these systems mainly rely on artificial intelligence (AI) technologies such as machine learning, deep learning, and natural language processing tools. Before, the majority of sarcasm detection algorithms depended on manually built sentiment features. Using typical machine learning models, many researchers extracted features with sentiment information to detect sarcasm. Because feature extraction requires a lot of manual labor, some studies have attempted to solve the problem using deep learning [4].Text mining is a method of extracting meaning from data by identifying patterns. Businesses improve their customer communication efficiency. A corporation can learn about public opinion regarding its products by examining client feedback. With ML algorithms, customer support tickets or reviews can be automatically classified by topic or language. Machine learning speeds up and improves the efficiency of text analysis over manual text processing. It provides for lower labor expenses and faster word processing without sacrificing quality [5]. Utilizing text mining techniques combined with the power of deep learning, models can be trained to grasp text beyond simple definitions, read for context, irony, and so on, and comprehend the writer's true attitude and feelings [2]. To fully utilize the capabilities of text analysis tools, we can include them in deep learning models. Deep learning is a branch of machine learning that uses “artificial neural networks” to process information in a manner similar to that of the human brain [6].

The systematic review provides an overview of the existing literature on sarcasm detection in Twitter data and its implications for sentiment analysis. It identifies and evaluates various approaches, techniques, and datasets used for sarcasm detection in tweets. The review highlights the strengths and limitations of different methods in the context of sentiment analysis. It offers insights into the effectiveness of sarcasm detection techniques in improving sentiment analysis accuracy. The review also identifies potential areas for future research, enabling advancements in the field and enhancing sentiment analysis in Twitter data. The structure of the paper is as follows: Section 2 covers data pre-processing, including data collection, text normalization, stop word removal, tokenization, and stemming/lemmatization. Section 3 covers feature extraction, including TF-IDF and tokenization. Naïve Bayes, SVM, Logistic Regression, CNN, and RNN are covered in Section 4. Section 5 covers feature extraction and algorithm implementation. Section 6 discusses Twitter data sarcasm detection and sentiment analysis integration. Finally, the discussion and conclusion are stated in sections 7 and 8.

2. Data pre-processing

Involves transforming raw data into an understandable format for machines through five steps: data collection, text normalization, stopword removal, text tokenization, and stemming & lemmatization .in following the details of each step:

2.1. Data collection of Twitter

Collecting data from Twitter for sentiment analysis offers numerous benefits. Twitter's real-time data flow provides access to up-to-date sentiments and opinions, enabling the analysis of current trends and immediate reactions. With its diverse user base, Twitter captures a wide range of demographics, interests, and opinions, allowing for a comprehensive analysis of sentiment across different user groups [7]. The user-generated text in tweets serves as a valuable source of data, providing insights into the underlying sentiments, attitudes, and emotions expressed by Twitter users. Hashtags and metadata associated with tweets offer contextual information, facilitating sentiment analysis by categorizing and tracking sentiment related to specific themes, events, or topics [8]. Analyzing sentiment trends over time allows researchers to understand the evolution of opinions and attitudes in response to various events or trends. However, challenges such as noise in the data, limitations of short message length, privacy concerns, and adherence to API terms and conditions should be carefully addressed during the data collection process [9].

2.2. Text normalization

Text normalization in sentiment analysis refers to the process of transforming and standardizing text data into a consistent format to improve the accuracy and effectiveness of sentiment analysis algorithms [10]. It involves applying techniques to handle variations, inconsistencies, and noise present in text data. Here are some common techniques used for text normalization in sentiment analysis [11]:

- Removing Punctuation: Getting rid of punctuation marks like periods, commas, and exclamation marks to simplify the text and reduce noise.

- Lowercasing: Converting all text to lowercase to treat uppercase and lowercase versions of words as identical, ensuring consistency.
- Removing URLs: Eliminating URLs or replacing them with a placeholder to remove irrelevant noise and focus on the textual content.
- Handling Emoticons and Emoji: Replacing emoticons and emoji with text representations or mapping them to sentiment scores to account for their impact on sentiment analysis.
- Removing Numeric Characters: Removing numbers or digits that may not carry significant sentiment information, simplifying the text, and reducing noise.
- Removing Special Characters: Eliminating symbols or non-alphanumeric characters that may not contribute much to sentiment analysis, streamlining the text, and emphasizing important words.

2.3. Stopword Removal

Stop word removal is a crucial preprocessing step in sentiment analysis that involves eliminating common words, known as stop words, from the text data. Stop words, such as "the," "is," and "and," do not carry significant sentiment information and can introduce noise to the analysis. By removing these irrelevant words, the focus is directed towards more meaningful terms that contribute to sentiment expression [12]. Stop word removal is achieved by comparing each word in the text data against a predefined list of stop words and discarding any matches. This process helps streamline the text and ensures that sentiment analysis algorithms concentrate on the most relevant and informative words for accurate sentiment classification. The resulting text data, free of stop words, enhances the quality of sentiment analysis by reducing noise and improving the extraction of sentiment-bearing words [11].

2.4. Text Tokenization

Text tokenization is a key step in sentiment analysis where the text data is split into individual tokens, such as words or phrases. This process allows for a more granular analysis of sentiment, as each token represents a discrete unit of meaning. Tokenization enables sentiment analysis algorithms to assess the sentiment associated with specific words or phrases, leading to more accurate sentiment classification. By breaking down the text into tokens, the nuances and subtleties of sentiment expressed in the text can be effectively captured [13]. Tokenization enhances the overall understanding of the sentiment conveyed, enabling deeper insights into the text's emotional tone. The resulting tokenized representation of the text data serves as a foundation for subsequent sentiment analysis tasks, contributing to improved sentiment classification outcomes [12].

2.5. Stemming and lemmatization

Stemming and lemmatization are two common techniques used in sentiment analysis to reduce words to their base or root form. Stemming involves removing suffixes from words, while lemmatization involves converting words to their base form using linguistic rules and a vocabulary. These techniques help to normalize the text data by reducing inflected forms to a common representation [13]. By reducing words to their root form, stemming and lemmatization assist in consolidating sentiment-related words and capturing their underlying sentiment more accurately. This preprocessing step enhances the effectiveness of sentiment analysis algorithms by reducing the vocabulary size and handling variations of sentiment-bearing words. By aligning similar word forms and reducing noise from word variations, stemming and lemmatization improve sentiment classification [14].

3. Feature extraction

Feature extraction is a key task in sentiment analysis as it involves the extraction of valuable information from the text data and will directly impact the performance of the model. The approach tries to extract valuable information that encapsulates the text’s most essential features. Over the years, researchers have explored various feature extraction techniques to enhance the accuracy and effectiveness of sentiment analysis algorithms. In this section, we will discuss two types of feature extraction that commonly used in previous studies.

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

TF-IDF is a widely used feature extraction technique in sentiment analysis. It calculates weights for words based on their frequency in a document and rarity across the corpus, capturing the importance of specific words in expressing sentiment. The application of TF-IDF has been effectively presented in terms of text mining. In [15] and [16] have utilized TF-IDF to represent the significance of words or phrases in sentiment classification. TF-IDF enables the identification of discriminative features that strongly contribute to sentiment analysis by considering both term frequency and inverse document frequency. It plays a crucial role in extracting informative features and enhancing sentiment classification accuracy. In [17] TF-IDF was used in text mining to handle highly correlated material and bias. They suggested TF-IDF model changes to reduce data reliance. They examined seven Facebook fan pages spanning news, finance, politics, sports, commerce, and entertainment to support their claim. Their research showed TF-IDF's potential in comment analysis and how to improve it in associated content.

3.2. Tokenizer

Tokenizer is a fundamental feature extraction technique in sentiment analysis that involves breaking down text into individual tokens, such as words or phrases. It enables a granular analysis of sentiment by identifying and examining sentiment-bearing elements. Research studies, including [18] and [19], have utilized tokenization with padding as an effective approach for feature extraction in sentiment analysis. These studies applied tokenization to segment text into meaningful units and then used padding to standardize the length of token sequences. By incorporating padding, the tokenized features can be effectively inputted into machine learning or deep learning models for sentiment classification, improving the accuracy and effectiveness of sentiment analysis.

Table (1): comparison between TF-IDF and Tokenizer

	TF-IDF	**Tokenizer**
Overview	TF-IDF assigns weights to individual terms in a document based on their frequency and rarity.	The tokenizer represents text by converting tokens into numerical vectors.
Approach	TF-IDF calculates the importance of terms by considering their frequency in a document and rarity across the corpus.	The tokenizer transforms tokens into numerical representations, typically using methods like one-hot encoding or word embeddings.
Output	TF-IDF produces a vector representation of a document where each component corresponds to the weight of a specific term.	The tokenizer generates numerical vectors for individual tokens or words in the text.
Term Weighting	TF-IDF assigns weights to terms based on their frequency and inverse document frequency.	The tokenizer does not assign weights to individual tokens; it represents tokens as discrete units.
Importance of Terms	TF-IDF emphasizes the importance of terms based on their rarity across the corpus.	The tokenizer does not inherently capture the importance of terms; all tokens are treated equally.
Contextual Information	TF-IDF does not consider the contextual information or the order of terms in the document.	The tokenizer can capture contextual information and word order by using methods like n-grams or word embeddings.
Use Case	TF-IDF is widely used in information retrieval, text mining, and text classification tasks.	The tokenizer is commonly used in natural language processing tasks like text classification or sentiment analysis.

4. Machine Learning and Deep Learning Algorithms

Machine learning and deep learning techniques have emerged as powerful approaches to sentiment analysis, enabling automated classification and interpretation of sentiment expressed in textual data [14]. This section explores the application of both machine learning and deep learning to sentiment analysis tasks. We discuss several commonly used algorithms and architectures, including traditional machine learning models such as support vector machines and random forests, as well as deep learning models such as recurrent neural networks.

4.1. Naïve Byes (NB)

Naive Bayes (NB) is a widely used classification technique in sentiment analysis for both categorization and training purposes. It is based on Bayes' theorem, which calculates the conditional probability of an event given the individual probabilities of the event and the conditional probabilities of other related events. NB assumes that the features are conditionally independent, meaning that the presence or absence of one feature does not affect the presence or absence of other features. Research studies have extensively utilized NB in sentiment analysis tasks. NB is applied when the training data size is small [20]. NB is classified as positive 10% more accurately than negative classification. This led to a decrease in average accuracy when it was taken. In [21] solved this problem using an improved version of the NB classifier. The model was tested on the restaurant review dataset.

4.2. Support Vector Machine (SVM)

The Support Vector Machine (SVM) approach is commonly employed in sentiment analysis to analyze data and establish decision boundaries. SVM is a non-probabilistic supervised learning technique widely used for classification tasks. The main goal of SVM is to find the hyperplane that effectively separates the data into different classes. The SVM algorithm aims to identify the hyperplane with the maximum possible margin, optimizing the separation between classes [22]. Several studies have utilized SVM in sentiment analysis to achieve accurate classification and decision boundary definition. In [23] used support vector machines for sentiment polarity classification. Classifying reviews based on their quality is one of the many purposes for which SVM is utilized. In [24] used two multiple class SVM based approaches. First being One-vs- all SVM and multiclass SVM to classify reviews. Second, a method was proposed to evaluate the quality of the product review dataset by considering it as a classification problem.

4.3. Logistic Regression (LR)

Logistic regression is a machine learning technique used for classification tasks, particularly in binary classification applications. It works by assigning weights to input features and multiplying them with corresponding values. By employing a probabilistic regression analysis, logistic regression learns the importance of different input properties in distinguishing between positive and negative classes. It utilizes maximum-likelihood estimation to calculate the optimal parameters [25]. Logistic regression can handle independent variables from various categories, including continuous, discrete (ordinal and nominal), making it a versatile and widely used classification algorithm. in [26] utilizes logistic regression to detect racist and sexist content in tweets, aiming to mitigate harmful content on social media. The results show moderate success in identifying such tweets, with f1 scores of 0.5303 and 0.5451 achieved using bag of words and TF-IDF features, respectively. In [27] uses logistic regression for sentiment analysis of tweets. The model is trained on a dataset of tweets from IMDb movie reviews. It applies a penalty parameter of l1 for regularization, a liblinear solver algorithm, and a maximum iteration of 10,000. The logistic function predicts categorical probabilities, which are converted into binary predictions using a threshold. Evaluation shows the logistic regression model achieves around 90% accuracy.

4.4. Convolution Neural Network (CNN)

Convolutional neural networks (CNNs) are widely used for sentiment analysis, as they can effectively extract features from textual data and classify sentiment. In a typical CNN architecture, input sequences are processed through convolutional and pooling layers to capture local patterns and down sample the data. Fully connected layers are then used for classification based on the extracted features. CNNs excel at capturing local and compositional features, enabling them to detect sentiment-related information at different scales. Training involves labeled data, where the model learns to optimize its parameters through techniques like backpropagation and gradient descent. CNNs have demonstrated high accuracy in sentiment analysis by automatically learning relevant features from the input text. In [28] compares the accuracy of CNN models and traditional machine learning methods for sentiment analysis on a public twitter dataset in Indonesia. The dataset includes neutral, positive, and negative sentiments.

Various CNN models are tested with different parameter configurations, such as the number of layers, filters, and sizes. The best-performing CNN model, CNN 12, achieves an accuracy of 81.4% using 7, 4, 3 and 256 filter sizes.

4.5. Recurrent Neural Networks (RNN)

Recurrent neural networks (RNNs) are widely used for sentiment analysis on sequential data like text reviews. They leverage a hidden state to capture temporal dependencies, with gated recurrent units (GRUs) and long short-term memory (LSTM) cells being popular variants. GRUs are computationally efficient, making them suitable for resource-constrained scenarios, while LSTMs excel at capturing long-term dependencies. RNNs consist of input, recurrent, and output layers [29]. The input layer processes sequential data, the recurrent layer updates the hidden state based on input and previous state, and the output layer performs sentiment classification. RNNs are advantageous in capturing the sequential nature and contextual information of text, which are crucial for sentiment analysis. However, challenges like vanishing or exploding gradients exist, which can impact performance. Techniques like gradient clipping and bidirectional RNNs help address these issues and improve RNNs' effectiveness in sentiment analysis [30].

Table (2): Summary analysis of machine learning and deep learning classification algorithm and its advantage and disadvantage

algorithm	Advantage	Disadvantage
Naïve Byes	- Simple and efficient. - Requires less training time and data than other approaches.	- Strong independence assumptions may not capture complex relationships between features.
Support vector machine	- Effective in high-dimensional spaces. - handles non-linear relationships.	- Computationally intensive for large datasets.
Logistic regression	- Interpretable and efficient.	- Assumes linear relationships between features and the target variable. - May not perform well with non-linear relationships.
Convolutional neural network	- Captures local and compositional features. - Effective in text classification.	- Requires large amounts of training data.
RNN	- Captures sequential dependencies and contextual information.	- Prone to vanishing/exploding gradient problems, longer training times.
LSTM	- More efficient than RNN Can map out long term dependencies. - Mitigates vanishing/exploding gradient problems.	- Requires significant computational resources. - More complex architecture and hyperparameter tuning.
GRU	-Captures long-term dependencies. - Less computationally expensive than LSTM.	- May have limitations in capturing complex relationships compared to LSTM. - May require more training data compared to LSTM.

5. Implementation of Feature Extraction and algorithms

Feature extraction techniques such as TF-IDF, tokenization, and other methods have been used in previous works in sentiment analysis and sarcastic detection to turn raw textual data into meaningful representations. They additionally utilized machine learning and deep learning algorithms to assess the retrieved characteristics, such as SVM, NB, LR, CNN, and RNN. Researchers want to increase the accuracy and efficacy of sentiment analysis and ironic detection in a variety of applications by integrating these approaches.

Hussain [31] (2020), the focus was on improving the precision of sentiment analysis for Arabic tweets. They used the multinomial naive Bayes approach on a dataset of 2000 tweets that were classified as either positive or negative to achieve their objective. The raw data was preprocessed before analysis, including tokenization with 4-grams and stemming with the Khoja stemmer. After that, the tweets were converted into TF-IDF features and categorized using a fivefold cross-validation method. The suggested technique produced encouraging results, with an accuracy rate of 87.5% on the presented dataset. In their work on enhancing Arabic sentiment analysis, the authors noted the constraints of lacking a data cleaning method and feature extraction. In order to overcome these constraints and improve the analysis, they suggested applying feature selection and reduction techniques in the future.

Alzyout et al. [32] (2021), conducted sentiment analysis on violence against women using several machine learning models such as support vector machines (SVM), k-nearest neighbors, naive Bayes, and decision trees. The dataset generated from Arabic tweets was preprocessed using tokenization, stemming, and stop-word removal. For feature extraction, the term frequency-inverse document frequency (TF-IDF) strategy was applied, and the results showed that SVM achieved the highest accuracy of 78.25% on the self-collected dataset. The study was limited by a small dataset that contained tweets from 2007 to 2019, limiting generalizability. A broader and more diverse dataset would have improved the validity of the results. For enhanced data labeling and more accurate sentiment analysis in the context of women's rights, it was suggested that a vocabulary be built and advanced deep learning techniques be explored.

S. S. Salim et al. [33] (2020), proposed using deep LSTM-RNN with word embedding techniques to detect sarcasm in tweets. To obtain equal-length inputs, they preprocessed the data using tokenization and sequence padding techniques. To avoid overfitting, the dropout layer approach was applied. The researchers discovered that their LSTM-RNN model outperformed SVM in recognizing sarcasm in tweets. Their method has the potential to detect sarcasm in online conversations. One of the study's weaknesses is that it relies on a single dataset. To address this constraint, they proposed that future research use larger and more varied datasets to improve performance and eliminate biases.

M. W. Habib and Z. N. Sultani [34] (2021), utilized a machine learning strategy to identify tweets in the Sentiment140 dataset as having positive or negative sentiment. To reduce and extract features from unstructured Twitter text data, the author examines three models, Naive Bayes, Logistic Regression, and Support Vector Machine, and employs four distinct feature extraction approaches, including BOW, TF-IDF, doc2vec, and word2vec. Logistic regression generated the most accurate results, which can be attributed to its efficacy in binary classification problems. The study's shortcomings include the use of a single dataset and the absence of model comparisons. Future research should look at larger, more varied datasets to increase performance and decrease biases. Using sophisticated feature extraction techniques such as word2vec and doc2vec with higher feature vector sizes seeks to improve classification accuracy.

Rachman and F. Hastarita [35] (2020), conducted sentiment classification of COVID-19 tweets using a dataset of 355,384 tweets. Cleaning, removal of numerals, emoticons, punctuation marks, case folding, filtering, and tokenization were all part of the preparation. The TF-IDF feature extraction approach was utilized for term weighting, taking into account both the term frequency (TF) and the inverse document frequency (IDF). The logistic regression approach was employed for sentiment categorization, with 94.71% accuracy. The research gave important insights into people's mental health throughout the epidemic, revealing a mainly neutral and positive attitude. They recognized the necessity of enhancing feature selection using n-gram approaches in order to improve classification performance in future work.

Güner et al. [18] (2019), evaluated the viability of applying sentiment analysis algorithms to Amazon.com product reviews. They examined the performance of several machine learning methods on a 40,000-review balanced dataset with an equal amount of positive and negative evaluations. TF-IDF vectorization was utilized for Multinomial Naive Bayes (MNB) and Linear Support Vector Machine (LSVM) feature extraction, while tokenization and padding were employed for the LSTM model. The LSTM model performed the best, with an accuracy of 90%. The study acknowledged the significance of classification technique and feature extraction in sentiment analysis but pointed out drawbacks such as presuming star ratings reflect sentiment and the lack of cross-validation. In future studies, they propose focusing on expanding classification to cover additional classes, improving hyperparameters, and investigating varied feature extraction and modeling methodologies.

Kamil and Setiawan [36] (2023), performed aspect-level sentiment analysis on movie reviews from Twitter using the Gated Recurrent Unit (GRU) method. The study concentrated on three aspects: narrative, acting, and director. To increase model accuracy, the researchers combined feature extraction using TF-IDF, feature expansion with GloVe, and SMOTE. The findings revealed that each test scenario raised accuracy and F1-Score values for the aspects studied. The final accuracy and F1-Score values were 70.40% (+7.62%) and 70.35% for the story aspect, 93.75% and 93.70% for the acting element, and 90.44% and 90.17% for the director aspect, respectively. Due to dataset biases and a lack of contextual awareness, sentiment analysis of movie reviews has difficulties. They proposed focusing on sophisticated deep learning models (e.g., Transformers), pre-training on bigger datasets, using transfer learning, and adding interpretability approaches to boost accuracy and generalizability.

Hossen et al. [37] (2021), analyzed the use of recurrent neural networks (RNNs) for sentiment analysis of hotel booking website customer evaluations. To clean the data, they used preprocessing techniques such as lemmatization, stemming, punctuation, and stop-word removal. Long short-term memory (LSTM) and gated recurrent unit (GRU) deep learning models were used, with the LSTM model having 30 hidden layers and the GRU model having 25 hidden layers. When tested using the dataset obtained for the study, the experimental findings showed that the LSTM and GRU models attained accuracies of 86% and 84%, respectively. They propose that in future work, they include other algorithms and bring new features to improve the system's security, acceptability, and popularity.

Jung et al. [38] (2016), used multinomial naive Bayes to analyze sentiment in tweets. They gathered characteristics from tweets and tested their method on the sentiment140 dataset, which includes 1.6 million tweets categorized as positive, negative, or neutral. The scientists discovered that when the dataset was divided into a 9:1 training-testing ratio, the multinomial naive Bayes model attained an accuracy of 85%. The study's shortcoming is that it focuses on binary sentiment analysis and does not examine sentiments other than positive and negative. They indicated that in the future, a more complete evaluation of the proposed scheme's accuracy should cover a broader spectrum of attitudes. Exploring methods to minimize test time and improve sentiment analysis performance using SparkR would also be useful.

Tyagi et al. [39] (2020), used the Sentiment140 dataset to create a hybrid model that merged a CNN with BiLSTM for sentiment analysis. Preprocessing methods included case folding, stemming, and the removal of stop words, digits, URLs, Twitter usernames, and punctuation marks from the dataset of 1.6 million positive and negative Tweets. A GloVe pretrained embedding layer, a one-dimensional convolutional layer, a BiLSTM layer, fully connected layers, dropout layers, and a classification layer were all part of the hybrid model. The model attained an accuracy of 81.20% on the Sentiment140 dataset, according to the findings. The model's shortcoming was discovered to be its reliance on a single dataset, which limited its generalizability. In the future, the model's potential might be investigated by testing its performance on a broader range of sentiment Twitter datasets. Furthermore, hybrid deep learning models might be investigated to improve the accuracy and resilience of sentiment analysis tasks.

Saad [40](2020), a comprehensive sentiment study on US airlines Six distinct machine learning models were used to analyze Twitter data, including support vector machine (SVM), logistic regression, random forest, XgBoost, naive Bayes, and decision tree. To prepare it for analysis, the data was preprocessed using techniques such as stop-word removal, punctuation removal, case folding, and stemming. The dataset was compiled via CrowdFlower and Kaggle and included 14,640 samples divided into three sentiment categories: positive, negative, and neutral. When the dataset was divided into a 70% training and 30% testing set, SVM achieved a maximum accuracy of 83.31%, followed by logistic regression with an accuracy of 81.81%. The study was limited in its generalizability due to a lack of information regarding the dataset utilized. To solve this shortcoming, future research may focus on using bigger and more varied datasets.

Table (3): Comparison between different approaches of sentiment analysis.

Ref.	database	Feature extraction	Classifier type	Accuracy

Hussain et. [31] (2020)	Arabic tweets dataset	TF-IDF	MNB	87.5%
Alzyout et. al. [32] (2021)	Self-collected dataset	TF-IDF	SVM	78.25%
S. S. Salim et. al. [33](2020)	Twitter dataset	tokenization and padding techniques	LSTM-RNN	....
M. W. Habib and Z. N. Sultani [34](2021)	Sentiment140 dataset	BOW, TF-IDF, doc2vec, and word2vec	, Naive Bayes, LR, and Support Vector Machine	------------
Rachman and F. Hastarit [35] (2020)	COVID-19 twitter dataset	TF-IDF	LR	94.71%
Güner, Levent et. al. [18] (2019)	Amazon reviews	TF-IDF, tokenization and padding techniques	MNB	
			LSVM	
			LSTM	90%
Kamil and Setiawan [36] (2023)	Movies review tweets dataset	TF-IDF, feature expansion with GloVe	GRU	Plot: 70.40%
				Actor: 93.75%
				Director: 90.44%
Hossen et al. [37] (2021)	Self-collected dataset	....	LSTM	86%
			GRU	84%
Jung et al. [38] (2016)	Sentiment140 dataset	---------	MNB	85%
Tyagi et al. [39] (2020)	Sentiment140 dataset	GloVe	CNN-BiLSTM	81.20%
Saad et. [40] (2020)	US airline Twitter dataset	Bag of Words	SVM	83.31%
			LR	81.81%

6. Detecting Internet Sarcasm in Twitter Data

Sarcasm, a unique form of expression commonly observed on Twitter, involves using positive words to convey negative sentiments or to communicate something different from the literal meaning. Detecting sarcasm in tweets is crucial for understanding the true intent and sentiment conveyed in the text. However, sarcasm detection in tweets presents challenges due to limited context and the reliance on linguistic cues. Although indicators such as tweet brevity, capital letters, emoticons, and exclamation marks can provide some clues, accurately identifying sarcasm requires a deeper understanding of linguistic and contextual nuances. Sarcasm detection plays a significant role in sentiment analysis tasks, particularly in analyzing tweets related to product reviews on platforms like Twitter. Successfully detecting sarcasm in these tweets can provide valuable insights into consumer preferences, opinions, and market behavior, leading to improved understanding and enhanced consumer experiences [41]. Throughout this paper, we explore the complexities of sarcasm detection in Twitter data and discuss strategies to enhance sentiment analysis accuracy and the integration of sentiment analysis and sarcasm detection. By addressing these challenges, we aim to achieve a comprehensive understanding of user sentiment on social media platforms and improve the effectiveness of sentiment analysis models.

6.1. features of sarcasm detection

When it comes to detecting sarcasm, there are several distinct features that can aid in identifying sarcastic statements. In this section, we discuss three key features commonly associated with sarcasm: irony, exaggeration, and juxtaposition.

- Irony: is a fundamental feature of sarcasm that involves the use of words or expressions conveying a meaning contrary to their literal interpretation. In sarcasm, irony is frequently employed to express a sentiment or viewpoint that is opposite to what is actually intended. Recognizing ironic statements is crucial in identifying sarcastic expressions. For example, saying "Oh, great!" in a sarcastic tone when something undesirable happens [42].

- Exaggeration: is another prevalent feature observed in sarcastic statements. It entails intentionally overstating or amplifying certain aspects to emphasize a sarcastic tone. By exaggerating elements, speakers aim to highlight the insincerity or mockery underlying their message. For instance, saying "I've been waiting for ages" when referring to a relatively short period of time. The presence of exaggerated statements can serve as a strong indicator of sarcasm [43].

- Juxtaposition: is a key feature in sarcasm detection that involves deliberately placing contrasting ideas, words, or concepts side by side to create a sarcastic effect. This technique often includes a contradiction between positive words and negative situations, or vice versa. By juxtaposing conflicting elements, speakers create irony and convey their sarcastic intent. For example, saying "What a lovely day for a flat tire!" Here, the positive term "lovely day" is juxtaposed with the negative situation of having a flat tire. Recognizing such contrasting elements is essential for identifying sarcasm [43].

By considering these features of sarcasm, such as irony, exaggeration, and juxtaposition, researchers and analysts can develop more effective approaches for detecting sarcasm in text, enabling a better understanding of the underlying sentiments and nuances conveyed by users.

6.2. Combination of sentiment analysis and sarcastic detection

Combining sentiment analysis with sarcasm detection enhances the understanding of sentiment in textual data by capturing the nuances of sarcasm and adjusting sentiment classification accordingly. This approach allows for a more accurate analysis of the emotions conveyed in the text by considering the context and subtleties of sarcastic remarks. Without sarcasm detection, the true sentiment behind sarcastic statements may be missed, leading to incorrect interpretations. By incorporating sarcasm detection, the system can identify instances of sarcasm and appropriately classify the sentiment, even when contradictory language is used [42]. The combination of sentiment analysis and sarcasm detection improves contextual understanding by capturing tone, context, and sarcasm-specific linguistic patterns. This integration enables a more nuanced representation of features and enhances the overall performance of machine learning and deep learning models in sentiment classification and other tasks. By accurately identifying and handling sarcastic tweets, these models can make more accurate predictions and enable more precise analysis. The incorporation of sarcasm detection in models contributes to better decision-making based on Twitter data. It allows businesses to gain a comprehensive understanding of customer sentiment, identify potential issues, track brand perception, and make informed decisions for reputation management and product improvement [44].

Previous studies have extensively explored the combination of sentiment analysis with sarcasm detection. In [45], a model combining sentiment analysis and sarcastic detection was developed to gain insights into the true sentiments of Twitter users during the Indian election, accurately capturing both genuine sentiments and sarcastic expressions. Similarly, in [46], the model was adapted to detect satire in Spanish tweets, demonstrating its versatility and effectiveness in detecting sarcasm and other forms of linguistic irony with high accuracy. Additionally, in [47], the aim was to improve sentiment analysis for Indonesian tweets by incorporating sarcasm detection. The evaluation results showed that sarcasm detection improved sentiment analysis by 5.49% through the use of relevant features related to sentiment and sarcasm.

7. Discussion

This systematic review aimed to provide an overview of the existing literature on sarcasm detection in Twitter data and its implications for sentiment analysis. The reviewed studies demonstrated the potential of artificial intelligence techniques, particularly machine learning and deep learning, in effectively detecting and handling sarcasm in tweets. We highlighted the importance of data preprocessing techniques, such as data collection, text normalization, stop word removal, text tokenization, and stemming and lemmatization, in cleaning and standardizing text data for accurate sentiment analysis. Additionally, the review highlighted the significance of feature extraction methods, such as TF-IDF and tokenizers, in improving sentiment analysis accuracy. Lastly, the review examined the use of machine learning and deep learning algorithms for sentiment analysis tasks.

The strengths of the reviewed studies include the utilization of large-scale Twitter datasets, which allowed for comprehensive analysis and generalizability of the findings. The studies also showcased the practical applications of sentiment analysis in various domains, such as social media analysis, customer reviews, and opinion mining. Rigorous evaluation measures, including accuracy, precision, recall, and F1-score, were employed to assess the effectiveness of sentiment analysis models. However, certain limitations were observed in the reviewed studies. Firstly, there was a lack of consensus on the best approach for sarcasm detection, indicating the need for further research and standardization in this area. The studies predominantly focused on English-language tweets, which limits the generalizability of the findings to other languages. The performance of sarcasm detection algorithms varied across different domains and contexts, highlighting the need for domain-specific models and datasets. Additionally, the reliance on manually labeled datasets in the reviewed studies may introduce subjectivity and biases in the evaluation process.

8. conclusion

In conclusion, this systematic review provides a comprehensive overview of sarcasm detection in Twitter data and its impact on sentiment analysis. The strengths and limitations of the reviewed studies shed light on the current state of the field. The findings suggest that deep learning methods, such as LSTM, GRU, and neural networks, demonstrate the highest accuracy and can serve as baseline learning methods for sentiment analysis and predicting sarcasm. However, it is important to note that these methods may require large amounts of labeled data and computational resources. Moving forward, future research should focus on addressing the identified limitations, such as the lack of consensus on the best approach for sarcasm detection and the need for domain-specific models and datasets. Exploring emerging methodologies, such as transfer learning and ensemble techniques, may also contribute to enhancing the accuracy and applicability of sentiment analysis models across various domains and languages. By advancing the field of sarcasm detection and sentiment analysis, a better understanding of user sentiment and opinion on social media platforms can be achieved, leading to more informed decision-making and improved user experiences.

References

[1] A. Alsaeedi and M. Z. Khan, “A Study on Sentiment Analysis Techniques of Twitter Data Efficient data dissemination protocols for IoT applications View project Sentiment Analysis of social media data View project A Study on Sentiment Analysis Techniques of Twitter Data,” Article in International Journal of Advanced Computer Science and Applications, vol. 10, no. 2, 2019, doi: 10.14569/IJACSA.2019.0100248.

[2] M. Singh, A. K. Jakhar, and S. Pandey, “Sentiment analysis on the impact of coronavirus in social life using the BERT model,” Soc Netw Anal Min, vol. 11, no. 1, Dec. 2021, doi: 10.1007/s13278-021-00737-z.

[3] M. Sykora, S. Elayan, and T. W. Jackson, “A qualitative analysis of sarcasm, irony and related #hashtags on Twitter,” Big Data Soc, vol. 7, no. 2, 2020, doi: 10.1177/2053951720972735.

[4] L. Ren, B. Xu, H. Lin, X. Liu, and L. Yang, “Sarcasm Detection with Sentiment Semantics Enhanced Multi-level Memory Network,” Neurocomputing, vol. 401, pp. 320–326, Aug. 2020, doi: 10.1016/J.NEUCOM.2020.03.081.

[5] J. Žižka, F. Dařena, and A. Svoboda, Text mining with machine learning: principles and techniques. CRC Press, 2019.

[6] L. Araujo, F. Lopez-Ostenero, J. Martinez-Romo, and L. Plaza, “Deep-Learning Approach to Educational Text Mining and Application to the Analysis of Topics’ Difficulty,” IEEE Access, vol. 8, pp. 218002–218014, 2020, doi: 10.1109/ACCESS.2020.3042099.

[7] E. Kouloumpis, T. Wilson, and J. Moore, “Twitter Sentiment Analysis: The Good the Bad and the OMG!,” Proceedings of the International AAAI Conference on Web and Social Media, vol. 5, no. 1, pp. 538–541, Aug. 2011, doi: 10.1609/ICWSM.V5I1.14185.

[8] A. Shelar and C. Y. Huang, “Sentiment analysis of twitter data,” Proceedings - 2018 International Conference on Computational Science and Computational Intelligence, CSCI 2018, pp. 1301–1302, Dec. 2018, doi: 10.1109/CSCI46756.2018.00252.

[9] V. S. Pagolu, K. N. Reddy, G. Panda, and B. Majhi, “Sentiment analysis of Twitter data for predicting stock market movements,” International Conference on Signal Processing, Communication, Power and Embedded System, SCOPES 2016 - Proceedings, pp. 1345–1350, Jun. 2017, doi: 10.1109/SCOPES.2016.7955659.

[10] H. T. Duong and T. A. Nguyen-Thi, “A review: preprocessing techniques and data augmentation for sentiment analysis,” Comput Soc Netw, vol. 8, no. 1, pp. 1–16, Dec. 2021, doi: 10.1186/S40649-020-00080-X/FIGURES/4.

[11] F. M. Javed Mehedi Shamrat et al., “Sentiment analysis on twitter tweets about COVID-19 vaccines using NLP and supervised KNN classification algorithm,” Indonesian Journal of Electrical Engineering and Computer Science, vol. 23, no. 1, pp. 463–470, 2021, doi: 10.11591/ijeecs.v23.i1.pp463-470.

[12] V. A. Fitri, R. Andreswari, and M. A. Hasibuan, “Sentiment Analysis of Social Media Twitter with Case of Anti-LGBT Campaign in Indonesia using Naïve Bayes, Decision Tree, and Random Forest Algorithm,” Procedia Comput Sci, vol. 161, pp. 765–772, Jan. 2019, doi: 10.1016/J.PROCS.2019.11.181.

[13] M. Qorib, T. Oladunni, M. Denis, E. Ososanya, and P. Cotae, “Covid-19 vaccine hesitancy: Text mining, sentiment analysis and machine learning on COVID-19 vaccination Twitter dataset,” Expert Syst Appl, vol. 212, p. 118715, Feb. 2023, doi: 10.1016/J.ESWA.2022.118715.

[14] Anees, Ansari Fatima, Arsalaan Shaikh, Arbaz Shaikh, and Sufiyan Shaikh. ‘Survey paper on sentiment analysis: Techniques and challenges.’ EasyChair2516-2314 (2020).”.

[15] R. Jamil, I. Ashraf, F. Rustam, E. Saad, A. Mehmood, and G. S. Choi, “Detecting sarcasm in multi-domain datasets using convolutional neural networks and long short term memory network model,” PeerJ Comput Sci, vol. 7, p. e645, Aug. 2021, doi: 10.7717/peerj-cs.645.

[16] S. Fransiska, R. Rianto, and A. I. Gufroni, “Sentiment Analysis Provider By.U on Google Play Store Reviews with TF-IDF and Support Vector Machine (SVM) Method,” Scientific Journal of Informatics, vol. 7, no. 2, pp. 203–212, Nov. 2020, doi: 10.15294/SJI.V7I2.25596.

[17] I. Yahav, O. Shehory, and D. Schwartz, “Comments Mining With TF-IDF: The Inherent Bias and Its Removal,” IEEE Trans Knowl Data Eng, vol. 31, no. 3, pp. 437–450, Mar. 2019, doi: 10.1109/TKDE.2018.2840127.

[18] “Güner, Levent, Emilie Coyne, and Jim Smit. ‘Sentiment analysis for Amazon. com reviews.’ Big Data in Media Technology (DM2583) KTH Royal Institute of Technology, Stockholm (2019).”.

[19] K. L. Kohsasih, B. H. Hayadi, Robet, C. Juliandy, O. Pribadi, and Andi, “Sentiment Analysis for Financial News Using RNN-LSTM Network,” 2022 4th International Conference on Cybernetics and Intelligent System, ICORIS 2022, 2022, doi: 10.1109/ICORIS56080.2022.10031595.

[20] M. Wankhade, A. C. S. Rao, and C. Kulkarni, “A survey on sentiment analysis methods, applications, and challenges,” Artif Intell Rev, vol. 55, no. 7, pp. 5731–5780, Oct. 2022, doi: 10.1007/s10462-022-10144-1.

[21] H. Kang, S. J. Yoo, and D. Han, “Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews,” Expert Syst Appl, vol. 39, no. 5, pp. 6000–6010, Apr. 2012, doi: 10.1016/J.ESWA.2011.11.107.

[22] V. Kumar and B. Subba, “A tfidfvectorizer and SVM based sentiment analysis framework for text data corpus,” 26th National Conference on Communications, NCC 2020, Feb. 2020, doi: 10.1109/NCC48643.2020.9056085.

[23] Y.-M. Li and T.-Y. Li, “Deriving market intelligence from microblogs,” Decis Support Syst, vol. 55, pp. 206–217, 2013, doi: 10.1016/j.dss.2013.01.023.

[24] C. C. Chen and Y. De Tseng, “Quality evaluation of product reviews using an information quality framework,” Decis Support Syst, vol. 50, no. 4, pp. 755–768, Mar. 2011, doi: 10.1016/J.DSS.2010.08.023.

[25] H. Hasanli and S. Rustamov, “Sentiment Analysis of Azerbaijani twits Using Logistic Regression, Naive Bayes and SVM,” 13th IEEE International Conference on Application of Information and Communication Technologies, AICT 2019 - Proceedings, Oct. 2019, doi: 10.1109/AICT47866.2019.8981793.

[26] A. Antim, Application of ML in Text Analysis: Sentiment Analysis of Twitter Data Using Logistic Regression. 2023.

[27] H. Guo, “Comparison Of Neural Network and Traditional Classifiers for Twitter Sentiment Analysis,” Highlights in Science, Engineering and Technology, vol. 38, pp. 1062–1070, Mar. 2023, doi: 10.54097/HSET.V38I.5996.

[28] Sartini, Subiyanto, and M. M. F. Alim, “A sentiment analysis study for twitter using the various model of convolutional neural network,” in Journal of Physics: Conference Series, IOP Publishing Ltd, Jun. 2021. doi: 10.1088/1742-6596/1918/4/042136.

[29] R. Ni and H. Cao, “Sentiment Analysis based on GloVe and LSTM-GRU,” Chinese Control Conference, CCC, vol. 2020-July, pp. 7492–7497, Jul. 2020, doi: 10.23919/CCC50068.2020.9188578.

[30] R. Wang, Z. Li, J. Cao, T. Chen, and L. Wang, “Convolutional Recurrent Neural Networks for Text Classification,” Proceedings of the International Joint Conference on Neural Networks, vol. 2019-July, Jul. 2019, doi: 10.1109/IJCNN.2019.8852406.

[31] AlSalman, Hussain. ‘An improved approach for sentiment analysis of arabic tweets in twitter social media.’ In 2020 3rd International conference on computer applications & information security (ICCAIS), pp. 1-4. IEEE, 2020.”.

[32] Alzyout, Moath, Emran AL Bashabsheh, Hassan Najadat, and Ahmad Alaiad. ‘Sentiment Analysis of Arabic Tweets about Violence Against Women using Machine Learning.’ In 2021 12th International Conference on Information and Communication Systems (ICICS), pp. 171-176. IEEE, 2021.”.

[33] S. S. Salim, A. Nidhi Ghanshyam, D. M. Ashok, D. Burhanuddin Mazahir, and B. S. Thakare, “Deep LSTM-RNN with Word Embedding for Sarcasm Detection on Twitter,” 2020 International Conference for Emerging Technology (INCET), Jun. 2020, doi: 10.1109/INCET49848.2020.9154162.

[34] M. W. Habib and Z. N. Sultani, “Twitter Sentiment Analysis Using Different Machine Learning and Feature Extraction Techniques,” Journal of Science ANJS, vol. 24, no. 3, pp. 50–54, 2021, doi: 10.22401/ANJS.24.3.08.

[35] “Rachman, Fika Hastarita. ‘Twitter sentiment analysis of Covid-19 using term weighting TF-IDF and logistic regresion.’ In 2020 6th Information Technology International Seminar (ITIS), pp. 238-242. IEEE, 2020.”.

[36] G. Kamil and E. B. Setiawan, “Aspect-Level Sentiment Analysis on Social Media Using Gated Recurrent Unit (GRU),” Building of Informatics, Technology and Science (BITS), vol. 4, no. 4, Mar. 2023, doi: 10.47065/BITS.V4I4.3105.

[37] M. S. Hossen, A. H. Jony, T. Tabassum, M. T. Islam, M. M. Rahman, and T. Khatun, “Hotel review analysis for the prediction of business using deep learning approach,” Proceedings - International Conference on Artificial Intelligence and Smart Systems, ICAIS 2021, pp. 1489–1494, Mar. 2021, doi: 10.1109/ICAIS50930.2021.9395757.

[38] Y. G. Jung, K. T. Kim, B. Lee, and H. Y. Youn, “Enhanced Naive Bayes Classifier for real-Time sentiment analysis with SparkR,” 2016 International Conference on Information and Communication Technology Convergence, ICTC 2016, pp. 141–146, Nov. 2016, doi: 10.1109/ICTC.2016.7763455.

[39] V. Tyagi, A. Kumar, and S. Das, “Sentiment Analysis on Twitter Data Using Deep Learning approach,” Proceedings - IEEE 2020 2nd International Conference on Advances in Computing, Communication Control and Networking, ICACCCN 2020, pp. 187–190, Dec. 2020, doi: 10.1109/ICACCCN51052.2020.9362853.

[40] A. I. Saad, “Opinion Mining on US Airline Twitter Data Using Machine Learning Techniques,” 16th International Computer Engineering Conference, ICENCO 2020, pp. 59–63, Dec. 2020, doi: 10.1109/ICENCO49778.2020.9357390.

[41] Dharmavarapu, Bala Durga, and Jayanag Bayana, ‘Sarcasm detection in Twitter using sentiment analysis.’ International Journal of Recent Technology and Engineering (IJRTE) 8, no. 1 (2018).”.

[42] “Irony and Sarcasm - Roger Kreuz - Google Books.” https://books.google.iq/books?hl=en&lr=&id=z2PMDwAAQBAJ&oi=fnd&pg=PP7&dq=features+for+sarcasm+detection+(irony,+exaggeration,+and+juxtaposition)+sentiment+analysis&ots=Sl-dct4BK0&sig=yJDUVkExgvV6n8JWTKASsLa0Bsg&redir_esc=y#v=onepage&q&f=false (accessed May 20, 2023).

[43] R. Ortega-Bueno, P. Rosso, and J. E. Medina Pagola, “Multi-view informed attention-based model for Irony and Satire detection in Spanish variants,” Knowl Based Syst, vol. 235, p. 107597, Jan. 2022, doi: 10.1016/J.KNOSYS.2021.107597.

[44] Y. Y. Tan, C. O. Chow, J. Kanesan, J. H. Chuah, and Y. L. Lim, “Sentiment Analysis and Sarcasm Detection using Deep Multi-Task Learning,” Wirel Pers Commun, vol. 129, no. 3, p. 2213, Apr. 2023, doi: 10.1007/S11277-023-10235-4.

[45] A. Khare, A. Gangwar, and S. Singh, ‘Sentiment analysis and sarcasm detection of indian general election tweets.’ arXiv preprint arXiv:2201.02127 (2022).”.

[46] E. Barbieri, F. Barbieri, F. Ronzano, and H. Saggion, “Procesamiento del Lenguaje Natural Sociedad Española para el Procesamiento del Lenguaje Natural Is This Tweet Satirical? A Computational Approach for Satire Detection in Spanish *,” pp. 135–142, 2015, [Online]. Available: http://www.redalyc.org/articulo.oa?id=515751524015

[47] Y. Yunitasari, A. Musdholifah, and A. K. Sari, “Sarcasm Detection For Sentiment Analysis in Indonesian Tweets,” IJCCS (Indonesian Journal of Computing and Cybernetics Systems), vol. 13, no. 1, p. 53, Jan. 2019, doi: 10.22146/ijccs.41136.