International Journal of Computers and Informatics

Journal Homepage: https://www.ijci.zu.edu.eg

Int. j. Comp. Info. Vol. 7 (2025) 104–125

Paper Type: Original Article

Sentiment Analysis of Social Network Contents using Machine Learning Algorithms: A Review

Mohamed Omar [1], Ahmed Salah [1,2,*] and Mahmoud Mahdy [1]

[1] Department of Computer Science, Faculty of Computer and Informatics, Zagazig University, Zagazig 44519, Egypt. Emails: hmomar2910@gmail.com; ahmad@zu.edu.eg; mamahdi@zu.edu.eg.

[2] College of Computing and Information Sciences, University of Technology and Applied Sciences, Ibri, Sultanate of Oman; ahmad.salah@utas.edu.om.

Received: 03 Feb 2025 **Revised**: 30 Mar 2025 **Accepted**: 27 Jun 2025 **Published**: 29 Jun 2025

Abstract

The exponential growth of social media spaces has resulted in a previously unimaginable amount of user-generated content, which can be used to identify public opinion, sentiment, and trends. Sentiment analysis is an area of the natural language processing (NLP), data science, and machine learning literature that describes the identification, extraction, and analytical processes of subjective information and emotional tone within a document or text. The incorporation of machine learning algorithms into social networking content can reveal significant information of interest to businesses, political analysis, public health, and research in social sciences. Although text in social media environments has the same functions as other textual forms, the text presents unique challenges associated with its brevity, informality, use of slang and abbreviations, use of emojis, sarcasm, irony, etc. and other challenges associated with the fast-paced, constantly updating nature of social media. These challenges are addressed in a specialized way to use social media texts comparatively with other texts. The goal of this article is to review the literature associated with social network content sentiment analysis, including the use of machine learning algorithms to classify sentiment. We will emphasize the unique characteristics of Twitter as a social media space, as well as the associated NLP focused preprocessing approaches to address noisy, informal, and platform-specific elements associated with the content. The review will include examples of traditional methods like Bag-of-Words and TF-IDF, features increasingly used into deeper methods like Word Embedding and in using dictionaries like sentiment lexicons and discuss their principles, as well as advantages and disadvantages when using these methods in social media. We will also include several of the most commonly used machine learning models used for sentiment classification. This review will also include traditional models like Naive Bayes, SVMs, and other ensemble methods used, as well as deep learning approaches utilizing CNNs, RNNs, and Transformers. Finally, we're going to discuss the nature and importance of the rigor of evaluation to determine sentiment classification correctness. We will discuss an evaluate using metrics like Accuracy, Precision, Recall, F1-Score, AUC-ROC, Cohen's Kappa and discuss important methods to evaluate classification validity like data spitting, cross-validation, and evaluate imbalanced datasets. We will explore, use, and evaluate of measures for sentiment discernment inherent in social media language, which will assist in an understand of how to manipulate behaviors of the machine learning for social media space, although this is a challenging task and domain.

Keywords: Classification; Topic Analysis, Sentimentant Analysis; LDA Topic Modeling; TF-IDF; Digital Humanities.

Corresponding Author: ahmad@zu.edu.eg

Licensee International Journal of Computers and Informatics. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0).

1 | Introduction

The proliferation of social media platforms has fundamentally reshaped how individuals communicate, share information, and express opinions. Platforms like Twitter, Facebook, and Instagram generate vast quantities of user-generated content daily, offering an unprecedentedly rich source of data for understanding public sentiment, societal trends, and individual perspectives. Sentiment analysis, also known as opinion mining, has emerged as a critical field within Natural Language Processing (NLP) and data science, focusing on the computational study of opinions, sentiments, and emotions expressed in text. By applying machine learning algorithms to social network content, researchers and organizations can extract valuable insights, moving beyond simple data collection to a deeper understanding of the underlying attitudes and feelings of the populace [i].

This paper delves into the domain of Sentiment Analysis of Social Network Contents using Machine Learning Algorithms. The core objective is to explore, implement, and evaluate various machine learning techniques for effectively discerning sentiment from the noisy, informal, and often context-dependent language prevalent in social media. Social media text presents unique challenges, including the use of slang, abbreviations, emojis, sarcasm, and rapidly evolving linguistic trends, which traditional NLP methods may struggle to handle. Machine learning offers powerful tools to learn patterns from large datasets and adapt to these complexities, making it a suitable approach for this task [ii].

The implications of sentiment analysis are vast, providing the vital ability to derive meaningful and actionable intelligence from the rapidly growing unstructured data elicited on social platforms and other digital arenas. For organizations, the ability to understand customer sentiment toward the product, service, and brand is of immense value. These sentiments will directly inform the critical strategic elements that facilitate consumer engagement including better marketing strategies, stronger product design and understanding of user needs/pain points, and better management of customer relationships in a forward-thinking and personalized way. In politics, understanding public opinion on a specific policy, or on a political figure, or electoral candidate, provides critical feedback for governance, leading to better decisions, as well as greater opportunity to leverage for future political campaigns. Outside of commerce and politics, sentiment analysis is a valuable tool for understanding public health trends, as well as understanding social mood and reaction and gauging social reaction to major events or social crisis. In addition, sentiment analysis has the potential to provide insight into complex trends, such as whether or not alternative forms of sentiment (such as investor confidence) are leading indicators of movement in the stock market or rising social concerns (such as urban crime) – potentially connected – could be modeled through social sentiment analysis.

As highlighted by Rodríguez-Ibánez et al. (2023) in their review on ScienceDirect, sentiment analysis has been successfully employed in diverse disciplines, including financial market prediction, health issues, and customer analytics, with a significant focus on Twitter due to its vast and diverse user base expressing opinions on myriad topics daily. Similarly, Dhole and Sahu (2023) emphasize that sentiment analysis divides text into positive, negative, and neutral categories, and their work proposes machine learning algorithms for this purpose, underscoring the ongoing development in this field [iii].

However, despite the advancements, several challenges persist. These include the nuances of language, the difficulty in detecting sarcasm and irony, the context-dependency of sentiment, and the need for robust models that can generalize across different social media platforms and topics. This paper aims to address some of these challenges by conducting a comparative analysis of various feature extraction techniques and machine learning models, and by applying these methods to specific case studies, such as analyzing public opinion on climate change using historical Twitter data [iv].

1.1 Background of the Study

The advent and exponential growth of social media platforms over the past two decades have profoundly transformed the landscape of human interaction, information dissemination, and societal discourse. Platforms such as Twitter, Facebook, Instagram, Reddit, and LinkedIn, among others, have evolved from niche online communities into global phenomena, collectively hosting billions of active users who generate an unprecedented volume of digital content daily. This content, ranging from personal updates and casual conversations to news sharing, political commentary, and consumer reviews, represents a vast and dynamic repository of public opinion, sentiment, and emerging trends. The sheer scale and real-time nature of social media have made it an invaluable, albeit complex, data source for researchers, businesses, governments, and various organizations seeking to understand the pulse of society [v].

In the landscape of interpreting and understanding textual data, the significance of sentiment analysis should not be overstated in this regard and is a key sub-domain in Natural Language Processing (NLP), broadly referred to as opinion mining. It represents a computational approach to identifying, extracting, quantifying, and analyzing subjective information and affect in text. The primary purpose is to ascertain the speaker or writer's attitude or emotional tone toward a topic or entity, and to assess the overall contextual polarity (whether that be mainly positive, negative, or neutral) at any level of granularity, from the general document level, to individual sentences, to specific features/aspects being discussed. As individuals increasingly turn to social media to express their views on products, services, political figures, social issues, and daily events, the ability to automatically analyze these expressions at scale provides critical insights. For instance, businesses can leverage sentiment analysis to gauge customer satisfaction, identify brand perception, and track the reception of marketing campaigns. Political analysts can monitor public mood towards policies and candidates, while public health officials can track the spread of misinformation or assess public reaction to health crises. The applications are diverse and continue to expand as the methodologies become more sophisticated [vi].

However, analyzing social network content presents a unique set of challenges that distinguish it from more traditional forms of text, such as news articles or formal documents. Social media language is often characterized by its brevity (e.g., Twitter's character limits), informality, and the prevalent use of slang, colloquialisms, abbreviations, acronyms, and emoticons/emojis. Furthermore, users frequently employ non-standard grammar, misspellings (both intentional and unintentional), and creative linguistic constructs. Sarcasm, irony, and nuanced expressions of sentiment are also common, posing significant difficulties for automated systems that rely on literal interpretations of text. The dynamic nature of social media means that new terms, hashtags, and communication styles emerge rapidly, requiring sentiment analysis models to be adaptable and continuously updated. Moreover, the sheer volume and velocity of data generated necessitate efficient and scalable processing techniques. The presence of noise, such as spam, advertisements, and irrelevant content, further complicates the task of extracting meaningful sentiment. Addressing these multifaceted challenges is central to developing effective sentiment analysis systems for social network content [vii].

1.2 Motivation

The motivation for this research stems from the confluence of the explosive growth of social media data and the increasing demand for sophisticated tools to understand the opinions and sentiments embedded within this data. While manual analysis of social media content is feasible on a small scale, it is impractical and inefficient for processing the vast, dynamic, and diverse datasets generated daily. Consequently, there is a compelling need for automated sentiment analysis tools that can accurately and efficiently process social network content. Such tools can provide timely insights that are crucial for decision-making in a wide array of domains [viii].

In the commercial sector, businesses are increasingly recognizing the value of social listening. Understanding customer sentiment towards their products, services, and brand image can directly influence marketing strategies, guide product development, improve customer service, and identify emerging market trends. Properly discerning negative sentiment in an organizational setting and in a timely manner is critical. If detected early enough, the organization can act quickly and nimbly to directly and proactively address customers' concerns, potentially limiting damage to that organization's brand image and stopping the negative perception from spreading too far. Merely identifying positive sentiment is equally beneficial; this kind of sentiment can also be leveraged to not only strengthen existing customer loyalty to the brand, but also serve as effective social proof to attract new customers. Advanced fine-grained sentiment analysis is especially useful. Because, not only can organizations identify whether a sentiment is positive or negative, this approach allows you to isolate and analyze the expressed opinion about specific features or aspects of a product or service, which gives the organization targeted and actionable feedback to improve products and services [1].

In the realm of politics and public administration, sentiment analysis of social media offers a powerful mechanism for gauging public opinion on policies, political figures, and societal issues. It can provide a real-time barometer of public mood, supplementing traditional polling methods, which are often more time-consuming and expensive. Understanding public sentiment can inform policy-making, help governments respond more effectively to citizen concerns, and track the impact of public campaigns or events. During elections, sentiment analysis can provide insights into voter attitudes and preferences, although ethical considerations regarding its use in political campaigning are paramount [ix].

Beyond commercial and political applications, sentiment analysis of social media content has significant potential in areas such as public health, disaster management, and social science research. For instance, monitoring social media for sentiment related to disease outbreaks can help public health officials track the spread of illness and public anxiety. During natural disasters, analyzing social media posts can aid emergency responders in understanding the immediate needs and concerns of affected populations. Social scientists can use sentiment analysis to study a wide range of societal phenomena, from shifts in cultural attitudes to the dynamics of online discourse [x].

Despite the clear benefits and increasing adoption of sentiment analysis, significant gaps and challenges remain in existing research and practical implementations, particularly concerning social media data. Many current approaches struggle with the nuances of informal language, sarcasm, context-dependency, and the rapid evolution of online vernacular. There is a continuous need for more robust, accurate, and adaptable machine learning models. Furthermore, much of the existing research focuses on general sentiment classification (positive, negative, neutral), while more nuanced analyses, such as aspect-based sentiment analysis, stance detection, or emotion recognition, are less explored yet highly valuable. This paper is motivated by the desire to contribute to addressing these gaps by investigating and comparing advanced machine learning techniques, focusing on feature engineering and model selection specifically tailored for the unique characteristics of social network content. The aim is to develop a deeper understanding of how to effectively harness machine learning for sentiment analysis in this challenging yet crucial domain, ultimately leading to more reliable and insightful tools for various applications [xi].

2 | Related Work in Computational Quranic Analysis

Unlock new paths for interpretation and educational tools. The future of computational Quranic analysis is rich with potential—balancing algorithmic precision with theological sensitivity will be key to advancing this field responsibly and meaningfully. [43]. Sentiment Analysis and Stance Detection

Sentiment analysis and stance detection are two closely related yet distinct tasks within the field of Natural Language Processing (NLP) that aim to uncover subjective information from text. Both are crucial for understanding opinions and viewpoints expressed in various forms of user-generated content, particularly on

social media platforms where individuals frequently share their perspectives on a multitude of topics. As Ramalho et al. (2023) note, sentiment and stance analysis are NLP techniques that measure affective states by processing textual data, though accurately classifying them remains challenging due to ambiguity and individual differences [xii].

Sentiment Analysis, often referred to as opinion mining, is primarily concerned with determining the overall emotional tone or polarity expressed in a piece of text. This polarity is typically categorized at different levels of granularity. At the **document level**, the goal is to classify an entire text (e.g., a product review, a blog post) as expressing a positive, negative, or neutral sentiment. At the **sentence level**, the analysis focuses on individual sentences, as a single document might contain multiple sentiments. More fine-grained approaches, such as **aspect-based sentiment analysis (ABSA)**, aim to identify the sentiment expressed towards specific entities or attributes (aspects) mentioned within the text. For example, in a restaurant review, ABSA might determine that the sentiment towards "food" is positive, while the sentiment towards "service" is negative. The traditional output of sentiment analysis is often a categorical label (positive, negative, neutral) or a numerical score indicating the intensity of the sentiment [xiii].

Stance Detection, on the other hand, goes beyond simple polarity and aims to determine an author's expressed viewpoint or position (e.g., favor, against, neutral) towards a specific target entity, topic, claim, or question. As Gomede (2024) highlights, traditional sentiment analysis can miss more nuanced opinions, necessitating a method like stance detection to discern whether an author supports, opposes, or remains neutral toward a target. Stance is inherently target-specific; a text might express a positive sentiment overall but take a negative stance towards a particular aspect or entity mentioned within it. For instance, a news article might discuss a controversial policy in a neutral tone (sentiment) but implicitly reveal a stance of opposition towards it. The authors in [xiv] suggest that stance can be a broader term encompassing sentiment analysis, emotion recognition, perspective identification, sarcasm/irony detection, and more. Stance detection is particularly relevant for analyzing debates, political discourse, and opinion pieces were understanding the author's position relative to a specific subject is key [xv].

Key Differences and Relationship: While both tasks deal with opinions, the primary distinction lies in their focus. Sentiment analysis identifies the emotional tone (positive, negative, neutral) of a text or part of a text. Stance detection identifies the author's position or attitude (favor, against, neutral) towards a *predefined target*. A text can be positive in sentiment but express an

against stance towards a specific target mentioned within it, or vice-versa. For example, a tweet saying "The new phone is amazing, but I'm against their new privacy policy" expresses positive sentiment about the phone but an 'against' stance towards the privacy policy. Stance detection often requires deeper contextual understanding and reasoning about the target [xvi].

Challenges in Sentiment Analysis and Stance Detection: Both sentiment analysis and stance detection face significant challenges, especially when applied to noisy and informal social media text. These challenges include:

1. **Ambiguity and Nuance:** A fundamental difficulty in analyzing sentiment is rooted in the ambiguous nature of language. A word can rarely be pinned down to a precise meaning; the sentiment inherently differs depending on the context. Sentiment analysis can also be conveyed very subtly, as deep understanding is sometimes needed for full comprehension. Linguistic challenges such, as sarcasm, irony, and any type of figurative language that is inherently complex, are tough for automated systems to read correctly and accurately. Gomede (2024) specifically illustrates how traditional approaches to sentiment analysis do not often grasp that sentiment is nuanced and can lead to inaccurate readings.

2. **Context Dependency:** The sentiment or stance of a word or phrase can change dramatically based on the surrounding text and the broader context of the conversation or topic. Models need to be able to capture long-range dependencies and contextual cues.
3. **Implicit Stance/Sentiment:** Opinions and stances are not always explicitly stated. Authors might imply their viewpoint through word choice, rhetorical questions, or by presenting selective facts. Detecting such implicit expressions is a major hurdle [xvii].
4. **Target Identification (for Stance Detection):** Accurately identifying the specific target of a stance is crucial. In complex sentences or discussions involving multiple entities, pinpointing what the author is expressing a stance *about* can be challenging.
5. **Data Sparsity and Domain Adaptation:** Labeled datasets for training supervised machine learning models are often scarce, especially for specific domains or less common languages. Models trained on one domain (e.g., product reviews) may not perform well on another (e.g., political discourse) without adaptation[xviii].
6. **Informal Language and Noise:** Social media text is characterized by misspellings, slang, abbreviations, emojis, and inconsistent grammar, which can hinder the performance of NLP tools.
7. **Dynamic Nature of Language:** Online language evolves rapidly, with new terms, memes, and expressions emerging constantly. Models need to be adaptable to these changes.
8. **Subjectivity and Disagreement:** Human annotators themselves may disagree on the sentiment or stance of a particular piece of text, highlighting the inherent subjectivity of the task. This makes creating high-quality gold-standard datasets difficult and introduces uncertainty, as discussed by Ramalho et al. (2023) in the context of uncertainty propagation from subjective annotation[xix].
9. **Multilingualism and Code-Switching:** Social media platforms host content in numerous languages, and users often mix languages (code-switching) within a single post, adding another layer of complexity.

Addressing these challenges requires sophisticated NLP techniques, robust machine learning models, and often, large amounts of high-quality training data. The ongoing research in these areas aims to develop more accurate, robust, and nuanced systems for understanding the vast spectrum of opinions and stances expressed in the digital world [xx].

2.2 Natural Language Processing (NLP) for Twitter Data

Processing Twitter data for sentiment analysis presents a unique set of challenges due to the inherent characteristics of the platform. Unlike formal text, tweets are often short, informal, and laden with platform-specific conventions. Effective Natural Language Processing (NLP) tailored to this type of data is crucial for extracting meaningful features and building accurate sentiment analysis models. This section discusses the distinct characteristics of Twitter data and the common NLP techniques employed for its preprocessing [xxi].

Characteristics of Twitter Data:

Twitter data is significantly different from traditional text sources, and these differences necessitate specialized NLP approaches. Key characteristics include:

1. **Brevity:** Tweets are famously constrained by character limits (though these have evolved, the core nature remains concise). This brevity means that context can be limited, and every word often carries significant weight. It also leads to the frequent use of abbreviations and contractions.

2. **Informal Language:** Users often employ informal language, including slang, colloquialisms, non-standard grammar, and misspellings (both intentional and unintentional). This deviates significantly from the structured language found in books or news articles.
3. **Platform-Specific Elements:** Tweets frequently contain elements unique to the platform, such as:
    - **Mentions (@username):** Used to refer to other users. While sometimes relevant for context, they often need to be handled or removed during preprocessing.
    - **Hashtags (#topic):** Used to categorize tweets or highlight keywords. Hashtags can be valuable features themselves or can be tokenized as part of the text.
    - **URLs:** Links are commonly shared in tweets. These URLs typically do not contribute directly to the sentiment of the tweet text itself and are often removed or replaced with a generic token.
    - **Retweets (RT):** Indicate that a tweet is a repost of another user's content. Identifying and handling retweets can be important to avoid data duplication or to analyze the spread of information.
4. **Emojis and Emoticons:** Pictorial representations of emotions (emojis like 😊, 😂, 😠) and text-based emoticons (like :), :-(, :D) are extensively used to convey sentiment and tone. These are vital cues for sentiment analysis and require specific handling, either by converting them to textual representations or by using models that can interpret them directly [xxii].
5. **Noise and Irrelevant Information:** Twitter feeds can contain a significant amount of noise, including spam, advertisements, and irrelevant chatter. Filtering this noise is an important preprocessing step.
6. **Dynamic and Evolving Language:** Online language, particularly on platforms like Twitter, is highly dynamic. New slang, memes, and abbreviations emerge and spread rapidly, requiring NLP models to be adaptable.
7. **Multilingual Content and Code-Switching:** Twitter is a global platform with content in numerous languages. Users may also engage in code-switching, mixing multiple languages within a single tweet, which poses challenges for monolingual NLP pipelines.

Common NLP Preprocessing Techniques for Twitter Data:

Given these characteristics, a series of preprocessing steps are typically applied to clean and normalize Twitter data before it is fed into machine learning models. As highlighted by Emiliano (2024) and in the Analytics Vidhya guide (2021), these steps are crucial for improving model performance [xxiii].

1. **Lowercasing:** Converting all text to lowercase helps in standardizing the text and treating words like "Good", "good", and "GOOD" as the same token, reducing the feature space.
2. **Removal of URLs:** URLs generally do not contribute to the sentiment of the tweet's text and are typically removed using regular expressions or replaced with a placeholder token (e.g., "URL").
3. **Removal of Mentions (@username):** User mentions are often removed or replaced with a generic token (e.g., "USER_MENTION") as they might not be directly indicative of the tweet's sentiment, or they could introduce noise if the number of unique users is very large.

4. **Handling Hashtags (#topic):** Hashtags can be handled in several ways: they can be removed, the hash symbol can be removed and the tag treated as a regular word (e.g., "#happy" becomes "happy"), or they can be split if they are compound words (e.g., "#Good Morning" becomes "Good Morning"). Sometimes, hashtags are kept as distinct features.
5. **Removal of Punctuation and Special Characters:** Punctuation marks (e.g., !, ?, .) and special characters (e.g., &, ", *) are often removed, unless they are part of emoticons or carry specific sentiment cues (e.g., multiple exclamation marks might indicate strong emotion).
6. **Tokenization:** This is the process of breaking down the text into individual words or tokens. Tokenization for tweets needs to be robust to handle informal language, hashtags, and other platform-specific elements.
7. **Stop Word Removal:** Common words that occur frequently but typically do not carry significant sentiment (e.g., "the", "a", "is", "in", "and") are often removed. However, the list of stop words might need to be customized for social media, as some standard stop words could be relevant in certain contexts (e.g., "not" is crucial for negation).
8. **Stemming and Lemmatization:**
    - **Stemming:** This process reduces words to their root or stem form by removing suffixes (e.g., "running" becomes "run", "studies" becomes "studi"). It is a cruder heuristic process.
    - **Lemmatization:** This process reduces words to their base or dictionary form (lemma) using vocabulary and morphological analysis (e.g., "ran" becomes "run", "better" becomes "good"). Lemmatization is generally more linguistically accurate than stemming but can be more computationally intensive [xxiv].
9. **Handling Emojis and Emoticons:** Emojis and emoticons are strong indicators of sentiment. They can be converted into textual descriptions (e.g., 😊 to "smiling face with smiling eyes") or special tokens that can be learned by the model. Libraries exist that provide sentiment scores for common emojis.
10. **Handling Negations:** Words like "not", "no", "never" can invert the sentiment of a phrase. Techniques like appending "_NEG" to words following a negation until the next punctuation mark can help capture this.
11. **Correction of Misspellings and Slang:** While challenging, attempting to correct common misspellings or expand slang and abbreviations to their standard forms can improve the quality of the input data. This often requires custom dictionaries or more advanced techniques[xxv].

Applying these preprocessing steps carefully and in an appropriate order is essential for transforming raw, noisy Twitter data into a cleaner, more structured format that is suitable for feature extraction and subsequent machine learning analysis. The choice of which steps to apply and how to apply them can significantly impact the performance of the sentiment analysis model [xxvi].

2.3 Feature Extraction Techniques for Text

Once text data, such as tweets, has been preprocessed, the next crucial step in preparing it for machine learning algorithms is feature extraction. Machine learning models cannot directly understand raw text; therefore, text must be converted into a numerical representation, typically a vector or a matrix of features. Feature extraction techniques aim to transform textual data into a format that captures its essential characteristics relevant to the task at hand, such as sentiment classification. Several methods exist, ranging

from simple word counts to more sophisticated semantic representations. Sahel (2023) and GeeksforGeeks (2023) provide good overviews of common techniques [xxvii].

1. Bag-of-Words (BoW):

The Bag-of-Words (BoW) model is often viewed as one of the simplest and oldest methods of feature extraction from a text in natural language processing. The underlying concept is representing a text document, such as a tweet or longer document, as merely an unordered collection or bag of words from the document. This means that only the words in the text are used in the model, and grammar and order of the words are completely ignored. The only thing that the realization of the BoW model counts is the number of times each unique word appears in the text. The process involves:

- **Vocabulary Creation:** First, a vocabulary of all unique words present in the entire training corpus is created.
- **Vectorization:** Each document is then represented as a numerical vector where each dimension corresponds to a word in the vocabulary. The value in each dimension can be binary (1 if the word is present, 0 otherwise), or it can be the frequency of the word in the document (term frequency).

For example, consider the tweets: "I love this phone" and "I hate this horrible phone". A vocabulary might be {I, love, this, phone, hate, horrible}. Tweet 1 vector (frequency): [1, 1, 1, 1, 0, 0] Tweet 2 vector (frequency): [1, 0, 1, 1, 1, 1]

Advantages: Simplicity and ease of implementation. **Disadvantages:** * It loses all information about word order and sentence structure, which can be important for understanding meaning (e.g., "not good" vs. "good not"). * The resulting feature vectors can be very high-dimensional (equal to the vocabulary size) and sparse (mostly zeros), especially with large vocabularies. * It does not capture semantic similarity between words (e.g., "good" and "excellent" are treated as completely different features). * Common words (stop words) can dominate the feature space if not handled properly, though preprocessing usually addresses this [xxviii].

2. N-grams:

To address the word order limitation of the BoW model, N-grams can be used. An N-gram is a contiguous sequence of N items (words in this context) from a given sample of text.

- **Unigrams:** N=1 (equivalent to the standard BoW model, considering individual words).
- **Bigrams:** N=2 (sequences of two adjacent words, e.g., "love this", "this phone").
- **Trigrams:** N=3 (sequences of three adjacent words, e.g., "I love this", "this horrible phone").

Using N-grams (especially bigrams and trigrams) in addition to unigrams can help capture some local context and word order, which can be beneficial for sentiment analysis (e.g., "not good" as a bigram captures negation better than "not" and "good" as separate unigrams). However, using higher-order N-grams significantly increases the dimensionality of the feature space and can exacerbate sparsity issues [xxix].

3. Term Frequency-Inverse Document Frequency (TF-IDF):

TF-IDF is a numerical statistic that aims to reflect how important a word is to a document in a collection or corpus. It assigns higher weights to words that are frequent in a particular document but rare across the entire

corpus, thus giving more importance to distinctive words. The TF-IDF score for a word *t* in a document *d* from a corpus *D* is calculated as:

TF-IDF(t, d, D) = TF(t, d) * IDF(t, D)

Where:

- **Term Frequency (TF)(t, d):** Measures how frequently a term *t* appears in a document *d*. It can be the raw count or normalized (e.g., by the total number of terms in the document).
- **Inverse Document Frequency (IDF)(t, D):** Measures how much information the word provides, i.e., whether it is common or rare across all documents in the corpus *D*. It is typically calculated as log(N / (df_t + 1)), where N is the total number of documents in the corpus, and df_t is the number of documents containing the term *t*. The "+1" is added to avoid division by zero if a term is not in any document (though this is rare if the vocabulary is built from the corpus) [xxx].

Words that appear in many documents (e.g., common words like "the", "is", if not removed as stop words) will have a low IDF score, diminishing their weight. Words that are specific to a few documents will have a higher IDF score, highlighting their importance for those documents.

Advantages: Simple to compute and often performs better than simple BoW by down-weighting common terms and emphasizing more discriminative terms. **Disadvantages:** Still disregards word order (like BoW) and does not capture semantic relationships between words.

4. Word Embeddings (e.g., Word2Vec, GloVe, FastText):

Word embeddings are a more advanced set of techniques that represent words as dense, low-dimensional vectors in a continuous vector space. Unlike sparse BoW or TF-IDF vectors, word embedding vectors are typically much smaller (e.g., 50-300 dimensions) and dense (most values are non-zero). The key idea is that words with similar meanings or that appear in similar contexts will have similar vector representations (i.e., they will be close to each other in the vector space). Sahel (2023) discusses word embeddings to capture semantic meaning [xxxi].

- **Word2Vec (Mikolov et al., 2013):** Learns word embeddings using a neural network model. It has two main architectures: Continuous Bag-of-Words (CBOW), which predicts the current word based on its context words, and Skip-gram, which predicts the context words given the current word.
- **GloVe (Global Vectors for Word Representation) (Pennington et al., 2014):** Learns word embeddings by factorizing a global word-word co-occurrence matrix from the corpus.
- **FastText (Bojanowski et al., 2017):** An extension of Word2Vec that represents each word as a bag of character n-grams. This allows FastText to generate embeddings for out-of-vocabulary (OOV) words and often works well for morphologically rich languages or text with many misspellings.

Pre-trained word embeddings, trained on massive text corpora (like Google News, Wikipedia, or Common Crawl), are often used, allowing models to leverage general linguistic knowledge even with smaller task-specific datasets. For a given document, the embeddings of its words can be combined (e.g., by averaging or

summing) to create a document-level embedding, or they can be fed into more complex neural network architectures (like CNNs or LSTMs) that can learn to combine them effectively [xxxii].

Advantages: * Captures semantic relationships between words (e.g., "king" - "man" + "woman" ≈ "queen"). * Results in lower-dimensional and dense feature vectors compared to BoW/TF-IDF. * Pre-trained embeddings can transfer knowledge from large corpora. **Disadvantages:** * Training word embeddings from scratch requires a large amount of data and computational resources (though pre-trained models are widely available). * Simple averaging of word embeddings to get document vectors can lose word order information, though more sophisticated neural models can mitigate this [xxxiii].

5. Sentiment Lexicons and Lexicon-based Features:

While not strictly a feature extraction technique in the same vein as BoW or embeddings, sentiment lexicons can be used to create features. A sentiment lexicon is a dictionary of words annotated with their sentiment polarity (e.g., positive, negative, neutral) and sometimes intensity (e.g., SentiWordNet, VADER Sentiment). Features can be engineered by:

- Counting the number of positive and negative words in a text based on a lexicon.
- Calculating an overall sentiment score for the text by aggregating the scores of its words.
- Using the presence or count of specific sentiment-bearing words as features.

These lexicon-based features can be used alongside other features (like TF-IDF or embeddings) to enhance sentiment classification performance, especially when labeled training data is limited.

The selection of an appropriate feature extraction method is not a simple or universal decision; rather it is based on several interrelated key considerations. These key considerations include the specifics of the task at hand, the quality and characteristics of the text being assessed, the size of the dataset, and the available computational resources. Characteristically, when we are concerned about the sentiment of social media texts, the texts tend to be informal, noisy, and dependant on context, thus we prefer specialized methods that can tackle text quality issues. Therefore, those methods that can cope with noise, account for some context, and process or leverage some semantic information (and an example of this would be N-grams with TF-IDF or word embeddings used with Neural Networks) tend to be preferred for their capability to better capture layered meanings in such fluid language [xxxiv].

2.4 Machine Learning Models for Classification

Once text data has been preprocessed and features have been extracted, the next step is to apply machine learning (ML) models to perform the classification task, such as sentiment analysis. The choice of an appropriate ML model is crucial and depends on various factors including the nature and size of the dataset, the complexity of the features, computational resources, and the desired performance. A wide range of models, from traditional algorithms to more complex deep learning architectures, can be employed for text classification. MLArchive (2024) and Analytics Vidhya (2021, 2025) provide useful discussions on various classifiers [xxxv][xxxvi].

1. Traditional Machine Learning Models:

These models have been widely used for text classification for many years and often provide strong baselines, especially when computational resources are limited or datasets are not extremely large.

- **Naive Bayes (NB):** Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with a strong (naive) assumption of independence between features. Despite this often-unrealistic assumption, Naive Bayes models are surprisingly effective for text classification tasks like spam detection and sentiment analysis. Common variants include Multinomial Naive Bayes (often used with word counts or TF-IDF features) and Bernoulli Naive Bayes (used with binary features indicating word presence/absence) [xxxvii].
    - **Advantages:** Simple to implement, computationally efficient, fast to train, and performs well on high-dimensional sparse data like text. It often works well even with small training datasets.
    - **Disadvantages:** The strong independence assumption might not hold true in reality, which can sometimes limit its accuracy. It doesn't capture word order or complex relationships between features.
- **Logistic Regression (LR):** Logistic Regression is a linear model that uses a logistic function (sigmoid function) to model the probability of a binary outcome (e.g., positive/negative sentiment). It can be extended to multi-class classification using techniques like one-vs-rest. It's widely used for text classification due to its simplicity and interpretability.
    - **Advantages:** Computationally efficient, provides probability scores for predictions, and the model coefficients can offer insights into feature importance. It generally performs well on linearly separable data.
    - **Disadvantages:** Being a linear model, it may not capture complex, non-linear relationships in the data unless combined with feature engineering (like polynomial features or interaction terms).
- **Support Vector Machines (SVM):** Support Vector Machines, or SVMs, are well-established supervised learning models that are considered to be both powerful and generalizable. The goal of supervised learning is to find a hyperplane with maximum separation among data points with different labels (or classes) often in a high-dimensional feature space where data points were mapped. SVMs achieve this separation by maximizing the margin - the distance between the hyperplane and the closest data points of both classes (which are known as support vectors). Maximizing the margin is a major contributing factor to SVMs generalizability and robustness. Also, SVMs are useful for the situation when data is not linearly separable in its original input space because SVMs can be used with various kernel functions (for example, linear, polynomial, and radial basis function - RBF) to potentially map data into a higher dimension to allow for linear separation. [xxxviii].
    - **Advantages:** Effective in high-dimensional spaces (common with text data), robust to overfitting, especially when the number of dimensions is greater than the number of samples. Kernel trick allows them to model non-linear decision boundaries.
    - **Disadvantages:** Can be computationally intensive to train, especially with large datasets. Performance can be sensitive to the choice of kernel and its parameters. They do not directly provide probability estimates.
- **Random Forest (RF):** The Random Forest algorithm is an excellent illustration of an ensemble learning approach. It works by training a large number of independent decision trees. In classification problems, the

random forest assumes the output class is the mode of the predictions of all the trees, or in other words, the class that was predicted the most frequently. In regression, the random forest algorithm predicts the mean of the predictions from all the trees to produce a single prediction (result). Random forest is effective in aggregating predictions from large numbers of independent decision trees that are slightly different from each other, while also providing a useful way of improving the overall predictive accuracy, and importantly, doing an excellent job at reducing the predictive model bias that a single decision tree would exhibit in predicting observations [xxxix].

- **Advantages:** Generally robust to overfitting, can handle high-dimensional data well, can capture non-linear relationships and feature interactions, and provides measures of feature importance.
- **Disadvantages:** Can be slower to train than simpler models like Naive Bayes or Logistic Regression. The resulting models can be less interpretable than a single decision tree or a linear model. [3]

- **Gradient Boosting Machines (e.g., XGBoost, LightGBM, CatBoost):** Gradient Boosting is another powerful ensemble technique that builds models (typically decision trees) sequentially, where each new model corrects the errors made by the previous ones. Algorithms like XGBoost, LightGBM, and CatBoost are highly optimized implementations of gradient boosting that often achieve state-of-the-art performance on structured and tabular data, and can also be effective for text classification when using appropriate feature representations [xl].
  - **Advantages:** Often provide very high accuracy, handle various types of data, and offer good control over overfitting through regularization.
  - **Disadvantages:** Can be computationally expensive and time-consuming to train, and require careful tuning of hyperparameters.

2. Deep Learning Models:

In recent years, deep learning models, particularly those based on neural networks, have achieved remarkable success in various NLP tasks, including text classification and sentiment analysis. These models can automatically learn hierarchical feature representations from raw text, often outperforming traditional models, especially with large datasets [xli].

- **Convolutional Neural Networks (CNNs):** Originally designed for image processing, CNNs have been adapted for text classification. For text, 1D convolutions are applied across sequences of word embeddings. CNNs can capture local patterns and n-gram-like features at different positions in the text. Max-pooling layers are often used to extract the most salient features.
  - **Advantages:** Effective at capturing local contextual information and hierarchical features. Relatively fast to train compared to RNNs.
  - **Disadvantages:** May not be as effective as RNNs at capturing long-range dependencies in text.
- **Recurrent Neural Networks (RNNs):** RNNs are designed to process sequential data, making them naturally suited for text. Variants like Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU)

networks are commonly used to address the vanishing gradient problem and capture long-range dependencies in text sequences [xlii].

- **Advantages:** Excellent at modeling sequential data and capturing contextual information and long-range dependencies.
- **Disadvantages:** Can be computationally expensive and slow to train, especially with very long sequences. Can still suffer from difficulties in learning very long-term dependencies.

- **Transformers (e.g., BERT, RoBERTa, XLNet, GPT variants):** Transformer models, based on the self-attention mechanism, have revolutionized the field of NLP. Models like BERT (Bidirectional Encoder Representations from Transformers) are pre-trained on massive text corpora and can be fine-tuned for specific downstream tasks like sentiment classification, often achieving state-of-the-art results. The self-attention mechanism allows transformers to weigh the importance of different words in a sequence when representing a particular word, capturing complex contextual relationships[xliii].
  - **Advantages:** Achieve state-of-the-art performance on many NLP tasks, effectively capture long-range dependencies and contextual information. Pre-trained models can be fine-tuned with relatively smaller task-specific datasets.
  - **Disadvantages:** Very computationally expensive to train from scratch and even fine-tuning large models requires significant resources (e.g., GPUs/TPUs). The models themselves can be very large, making deployment challenging in some scenarios.
- **Hybrid Models:** Researchers often combine different architectures, such as CNNs with LSTMs (e.g., C-LSTM), to leverage the strengths of each. For instance, CNNs can extract local features, which are then fed into an LSTM to capture sequential patterns.

The selection of a machine learning model for sentiment analysis of social network content involves considering the trade-offs between model complexity, performance, interpretability, and computational cost. For Twitter data, which is often short and noisy, models that can handle such characteristics effectively are preferred. While traditional models can provide good baselines, deep learning models, especially transformers, often offer superior performance if resources permit [xliv].

2.5 Evaluation Metrics and Methodology

Evaluating the performance of sentiment analysis models is crucial to understand their effectiveness and to compare different approaches. A systematic methodology involving appropriate metrics and validation techniques is essential for reliable assessment. This section discusses common evaluation metrics used in sentiment analysis and text classification, along with methodological considerations for robust model evaluation. Sources like LinkedIn Advice (2023) and Mungalpara (2023) offer good summaries of these metrics [xlv].

1. Common Evaluation Metrics:

In order to assess the predictive performance of models in classification tasks like sentiment analysis - where the intention is frequently to classify text into separate classes of either a positive, negative or neutral sentiment - general, widely used performance metrics should all be used. These important metrics are usually determined from the evaluation of what is known as a confusion matrix. A confusion matrix is ultimately a simple yet

highly informative table designed to communicate the predictive performance of a classification model in a concise manner - it does this by showing you the counts for every possible outcome: true positive (TP) predictions, true negative (TN) predictions, and false positive (FP) predictions (Type I errors) and false negative (FN) predictions (Type II errors) broken down for each individual classification class the model is predicting [xlvi].

1. **Accuracy:** Accuracy is the proportion of correctly classified instances (both positive and negative) out of the total number of instances. It is calculated as: Accuracy = (TP + TN) / (TP + TN + FP + FN) While intuitive, accuracy can be misleading, especially for imbalanced datasets where one class significantly outnumbers others. For example, if 90% of tweets are positive, a model that always predicts "positive" will achieve 90% accuracy but will be useless for identifying negative or neutral sentiments.
2. **Precision (Positive Predictive Value):** Precision measures the purity of the set of items the model classifies as positive. It defines the proportion of positively labeled test items that actually belong to the positive class. Essentially, it answers the question: "Of all the items the classifier said were positive, how many were actually positive?" Precision is calculated as the number of true positives divided by the number of true positives plus the number of false positives (TP/(TP + FP)). A high precision value would indicate that the model produced very few false positive predictions. Precision is particularly important when it is costly or damaging to mark something as positive when it should not have happened (for example, incorrectly marking a neutral social media post as a strongly negative post would not be beneficial for the company).
3. **Recall (Sensitivity, True Positive Rate):** Recall measures the completeness of the model's ability to find all of the positive items. Recall is the proportion of all of the actual positive cases in the dataset that the model predicted as positive. "Of all the actual positive cases, how many did the model identify?" Recall is calculated as the number of true positives divided by the number of true positives plus the number of false negatives (TP/(TP + FN)). A high recall value would mean that the model was able to identify almost all the actual positive cases. Recall is particularly important when the consequences of missing a true positive prediction would be significant (for example, if a true customer service complaint is missed as a negative case category) [xlvii].
4. **F1-Score:** The F1-score provides a single numerical value that represents a balanced combination of precision and recall, specifically using their harmonic mean. This metric is particularly valuable as a performance indicator when the class distribution within the dataset is uneven or skewed. Calculated as two times the product of precision and recall, divided by their sum (2 * (Precision * Recall) / (Precision + Recall)), the F1-score ranges from 0 (worst performance) to 1 (perfect performance). Unlike a simple arithmetic average, the harmonic mean strongly penalizes models that exhibit extreme performance differences between precision and recall. For tasks involving multiple sentiment categories, the F1-score can be computed independently for each class and then aggregated through different averaging techniques (such as macro-average, micro-average, or weighted-average) to yield a comprehensive overall metric [xlviii].
    - **Macro-F1:** Calculates the F1-score for each class independently and then takes the unweighted average. It treats all classes equally.

- **Micro-F1:** Calculates global precision and recall by summing TPs, FPs, and FNs across all classes before computing the F1-score. It tends to be dominated by the performance on more frequent classes.
- **Weighted-F1:** Calculates the F1-score for each class and then takes a weighted average, where the weight is the number of true instances for each class (support). This accounts for class imbalance.

5. **Specificity (True Negative Rate):** Specificity measures the proportion of correctly predicted negative instances among all actual negative instances. Specificity = TN / (TN + FP) It is less commonly reported than precision/recall for the positive class but is important in binary and multi-class scenarios to understand performance on negative classes.
6. **Area Under the ROC Curve (AUC-ROC):** The Receiver Operating Characteristic (ROC) curve is a diagnostic visualization that displays a graphical representation of the differentiating ability of a binary classification algorithm, while the confidence threshold of classification is adjusted, across multiple confidence threshold levels. In the ROC curve, the True Positive Rate (which is the same thing as Recall) is plotted on the vertical access against the False Positive Rate (calculated as 1 minus/ complementary of Specificity) on the horizontal access, for each threshold. Similarly the area under the ROC curve (AUC - ROC) condenses the output of the above curve into a single scalar metric that summarizes the model's ability to differentiate between the two classes. An AUC-ROC score of 1 indicates perfect separation of the positive and negative instances whereas a score of 0.5 indicates a model performance similar to random guessing. For classification tasks with more than two classes, the AUC-ROC metric can still be employed using approaches like One vs. Rest (OvR) or One vs. One (OvO) averaging either the results of the different possibilities for a specific pairwise comparison or one-class versus all [xlix].
7. **Cohen's Kappa:** Cohen's Kappa is a statistic that measures inter-rater agreement for categorical items. In machine learning, it can be used to measure the agreement between the predicted and actual classifications, while accounting for the agreement occurring by chance. It is considered a more robust measure than simple accuracy, especially for imbalanced datasets [l].

2. Methodological Considerations:

Beyond choosing appropriate metrics, the methodology for evaluating models is critical for obtaining reliable and generalizable results.

- **Data Splitting (Train-Validation-Test):** The dataset should be split into at least two, and preferably three, mutually exclusive sets:
  - **Training Set:** Used to train the machine learning model.
  - **Validation Set (Development Set):** Used to tune hyperparameters of the model and for intermediate evaluation during model development. This helps prevent overfitting to the test set.
  - **Test Set (Hold-out Set):** Used for the final evaluation of the trained model. This set should only be used once, after all model development and tuning are complete, to provide an unbiased estimate

of the model's performance on unseen data. Common splits are 60-20-20 or 70-15-15 for train-validation-test, respectively.

- **Cross-Validation:** K-fold cross-validation is an incredibly useful tool for determining a reliable estimate of a model's expected performance when data availability is somewhat restricted. In a K-fold cross-validation approach, the original dataset is split into K different subsets or "folds" of about the same size. The model training and evaluation process is repeated K times. In each of the K iterations, one designated fold is set aside to be the validation or test set, while the model is trained only on the data of the K-1 folds. The model performance metrics (accuracy, precision, etc.) observed in the validation step of each of the K iterations are then averaged together. This approach provides a more consistent and less biased approximation of a model's true generalization performance than relying on a single potentially biased split of train-validation data. In the case of classification problems with imbalanced classes, Stratified K-fold cross-validation is used to ensure that each fold of the dataset is roughly consistent in the proportion of class distribution as the original dataset [li].
- **Handling Imbalanced Datasets:** Social media data for sentiment analysis is often imbalanced (e.g., more positive or neutral tweets than negative ones). As mentioned, accuracy can be misleading. Metrics like F1-score, AUC-ROC, and Cohen's Kappa are more appropriate. Methodologies to address imbalance include:
    - **Resampling techniques:** Oversampling the minority class (e.g., SMOTE - Synthetic Minority Over-sampling Technique) or under sampling the majority class.
    - **Cost-sensitive learning:** Assigning higher misclassification costs to the minority class.
    - **Using algorithms inherently good at handling imbalance.**
- **Statistical Significance Testing:** When comparing the performance of different models or techniques, it's important to determine if the observed differences in performance metrics are statistically significant or simply due to chance. Statistical tests (e.g., t-tests, McNemar's test) can be used for this purpose.
- **Qualitative Analysis and Error Analysis:** In addition to quantitative metrics, qualitative analysis is crucial. This involves manually inspecting a sample of misclassified instances to understand the types of errors the model is making. Error analysis can reveal patterns, highlight weaknesses in the model or feature representation, and suggest areas for improvement (e.g., issues with sarcasm, negation, domain-specific jargon) [lii].
- **Reproducibility:** Ensuring that experiments are reproducible is a cornerstone of good scientific practice. This involves clearly documenting the dataset, preprocessing steps, feature extraction methods, model architectures, hyperparameter settings, and evaluation procedures [liii].

By employing a combination of appropriate evaluation metrics and a sound evaluation methodology, researchers can gain a comprehensive understanding of their sentiment analysis models' performance and limitations, leading to more reliable and impactful findings.

2.6 Related Work Summary

This section has provided a comprehensive review of the literature pertinent to the sentiment analysis of social network content, with a specific focus on Twitter data and the application of machine learning

algorithms. The exploration began with a foundational understanding of sentiment analysis and the closely related concept of stance detection, highlighting their significance in interpreting public opinion and the nuances involved in discerning subjective information from text. It was established that sentiment analysis aims to classify the polarity of text (positive, negative, or neutral), while stance detection focuses on identifying the viewpoint expressed towards a specific target or topic. The inherent challenges in both tasks, such as handling sarcasm, irony, context-dependency, and domain-specific language, were underscored, particularly in the context of informal and dynamic social media content [liv].

The subsequent discussion delved into the specific domain of Natural Language Processing (NLP) techniques tailored for Twitter data. The unique characteristics of tweets—brevity, informal language, platform-specific elements like hashtags, mentions, emojis, and URLs, as well as the prevalence of noise and evolving linguistic trends—necessitate specialized preprocessing steps. Common techniques such as lowercasing, removal of irrelevant symbols and URLs, tokenization robust to social media conventions, stop word removal, stemming or lemmatization, and the crucial handling of emojis and negations were detailed. The importance of these preprocessing stages in transforming raw, noisy tweet data into a cleaner, structured format suitable for effective feature extraction and model training was emphasized as a critical precursor to successful sentiment analysis [lv].

Following preprocessing, the section examined various feature extraction techniques used to convert textual data into numerical representations that machine learning models can understand. Traditional methods like Bag-of-Words (BoW) and N-grams were discussed, noting their simplicity but also their limitations in capturing word order and semantic meaning. Term Frequency-Inverse Document Frequency (TF-IDF) was presented as an improvement over basic BoW by weighting terms based on their importance in a document relative to a corpus, thereby highlighting more discriminative terms. More advanced techniques, particularly word embeddings (such as Word2Vec, GloVe, and FastText), were explored for their ability to capture semantic relationships between words by representing them as dense, low-dimensional vectors. The use of pre-trained embeddings and the methods for aggregating word embeddings into document-level representations were also considered, acknowledging their power in leveraging broader linguistic knowledge [lvi].

The review then transitioned to the machine learning models commonly employed for text classification and sentiment analysis. A distinction was made between traditional machine learning algorithms and more recent deep learning approaches. Traditional models, including Naive Bayes, Logistic Regression, Support Vector Machines (SVMs), Random Forests, and Gradient Boosting Machines, were described, outlining their underlying principles, advantages, and disadvantages in the context of text data. Subsequently, deep learning models, which have demonstrated significant advancements in NLP, were discussed. Convolutional Neural Networks (CNNs) for capturing local patterns, Recurrent Neural Networks (RNNs) like LSTMs and GRUs for modeling sequential information and long-range dependencies, and the transformative Transformer models (e.g., BERT) based on self-attention mechanisms were presented. The capacity of these deep learning models, especially transformers, to learn complex feature representations and achieve state-of-the-art performance, particularly with large datasets, was highlighted, alongside considerations of their computational demands [lvii].

Finally, the section addressed the critical aspects of evaluation metrics and methodology for assessing the performance of sentiment analysis models. Standard metrics derived from the confusion matrix, such as accuracy, precision, recall, and F1-score (including its macro, micro, and weighted variants), were defined, emphasizing the importance of choosing metrics appropriate for potentially imbalanced datasets. Other evaluative measures like the Area Under the ROC Curve (AUC-ROC) and Cohen's Kappa were also mentioned. Methodological best practices, including proper data splitting into training, validation, and test sets, the use of K-fold cross-validation for robust performance estimation, strategies for handling imbalanced data, the importance of statistical significance testing when comparing models, and the value of qualitative

error analysis, were detailed. These elements collectively ensure a rigorous and reliable assessment of model capabilities [lviii].

In essence, the related work surveyed in this section underscores that sentiment analysis of social media content is a multifaceted problem requiring a pipeline of carefully chosen techniques. From nuanced NLP preprocessing tailored to the idiosyncrasies of platforms like Twitter, through sophisticated feature extraction methods that can capture both lexical and semantic information, to the selection and rigorous evaluation of appropriate machine learning classifiers, each stage plays a vital role. While traditional methods offer valuable baselines, the trend indicates a move towards deep learning models, particularly transformers, for achieving higher performance, albeit with greater computational costs. The existing literature provides a strong foundation for this paper, which aims to contribute further by conducting specific comparative analyses of feature selection methods and by applying machine learning to analyze public opinion on significant societal issues like climate change using Twitter data [lix].

3 Conclusions

This article has established a comprehensive review about the sentiment analysis of social media content, with a particular emphasis on the use of machine learning algorithms. The quick expansion and fluid nature of social media makes it essential to automate sentiment analysis to capture and decode opinions and sentiments from the massive quantities of user content. This review emphasized the relevance of sentiment analysis across a variety of domains, such as commerce, politics, and public health, all highlighting the need for timely actionable insights generated by sentiment analysis.

The review focused a considerable portion of the text on the features of social media text that create unique challenges, including informality, noise, context dependency, and platform features (e.g., handles, hashtags) and language features (e.g., sarcasm and irony). To meet these challenges, careful construction of a pipeline is required that begins with specific Natural Language Processing preprocessing steps designed for the characteristics of social media. After rather than discussing paraphrased or humor, we turned to talking about the feature extraction processes; specifically, how to convey text to a form suitable for models to assess using numbers (i.e., feature extraction). We made our way through the different methods and culminated with different representations with underlying semantics (e.g., word embeddings). After this section we moved on to discussing the models themselves, distinguishing between traditional algorithms, which provide value as baselines, and deep learning architectures, namely, transformers, which appear to be the new standard; although computationally loss-affordable. Finally, we took time to discuss the elements required to do the evaluation component of model assessing sentiment analysis; we emphasized appropriate metering for data relevance and evaluative methods (e.g., cross-validation; error analyses, etc) to evaluate sentiments and to be reliable on results in terms diversity. The literature surveyed, supported that the sentiment analysis of social media presents a complex problem and assuming a process with the careful approach to NLP methods and procedures, feature extraction methods, machine learning classifiers, and thoughtful evaluation at every level is going to help achieve value with sentiment analysis. This review does provide a good base of the literature, more sophisticated models are being proposed continuously to meet the demands of sentiment of audience around from various themes social behavior.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data Availability

This study is based on a conceptual framework, and no empirical data were generated or analyzed.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[i] Buckley, G. S. (1988). "Term-weighting approaches in automatic text retrieval," . Information processing & management, vol. 24, no. 5, 513–523.

[ii] D .Maynard, B. K. (2015). Understanding climate change tweets: an open-source toolkit for social media analysis. InEnviroInfo and ICT for Sustainability . Atlantis Press, 242-250.

[iii] Chai, . P. (2023). , "Comparison of text preprocessing methods,". Natural Language Engineering, vol. 29, no. 3, 509–553.

[iv]C. P. Chai, ". o. (2023). ," Natural Language Engineering, . vol. 29, no. 3, 509–553.

[v]D. S. Asudani, N. K. (2023). "Impact of word embedding models on text analytics in deep learning environment: a review," . Artificial intelligence review, vol. 56, no. 9, 10345–10425.

[vi]G Klein, K. D. (2017). Opennmt: Open-source toolkit for neural machine translation. arXiv preprint arXiv:1701.02810, 465-498.

[vii]Emiliano, C. (2024, May 2024). Twitter Sentiment Analysis: A Comprehensive Guide to NLP and Machine Learning Techniques. Medium. Retrieved from https://medium.com/@caio_emiliano/twitter-sentiment-analysis-a-comprehensive-guide-to-nlp-and-machine-learning-techniques-b5af9cd213a7...

[viii]Dahal, B. S. (2019). "Topic modeling and sentiment analysis of global climate change tweets.". Social network analysis and mining 9, 1-20.

[ix] O. Alsemaree, A. S. (2024). "Sentiment analysis of Arabic social media texts: A machine learning approach to deciphering customer perceptions,". in Heliyon, vol. 10, no. 9. Elsevier, 70-120.

[x] R. K. Das, M. I. (2023). Sentiment analysis in multilingual context: Comparative analysis of machine learning and hybrid deep learning models. Heliyon, vol. 9, no. 9, 184-231.

[xi] Mumenthaler, C. O. (2021). "The impact of local temperature volatility on attention to climate change: Evidence from Spanish tweets." . Global environmental change 69 , 102-286.

[xii] Omar, M. S. (2025). Comparative Analysis of Feature Selection Methods for Twitter Sentiment Classification. Journal of Information Systems Engineering and Management, 10(215). Retrieved from https://jisem-journal.com/index.php/journal/article/download/3331/1442/5456.

[xiii] Mumenthaler, C. O. (2021). "The impact of local temperature volatility on attention to climate change: Evidence from Spanish tweets." . Global environmental change 69 , 102-286.

[xiv] Omar, M. S. (2025). Comparative Analysis of Feature Selection Methods for Twitter Sentiment Classification. Journal of Information Systems Engineering and Management, 10(215). Retrieved from https://jisem-journal.com/index.php/journal/article/download/3331/1442/5456.

[xv] Omar, M. S. (2025). Comparative Analysis of Feature Selection Methods for Twitter Sentiment Classification. Journal of Information Systems Engineering and Management, 10(215). Retrieved from https://jisem-journal.com/index.php/journal/article/download/3331/1442/5456.

[xvi] Shyrokykh, K. M. (2023). Short text classification with machine learning in the social sciences: The case of climate change on Twitter. pmc.ncbi.nlm.nih.gov-https://doi.org/10.1371/journal.pone.0290762, 213-267.

[xvii] Chai, .. P. (2023). , "Comparison of text preprocessing methods,". Natural Language Engineering, vol. 29, no. 3, 509–553. Omar, M. S. (2025). Comparative Analysis of Feature Selection Methods for Twitter Sentiment Classification. Journal of Information Systems Engineering and Management, 10(215). Retrieved from https://jisem-journal.com/index.php/journal/article/download/3331/1442/5456.

[xviii] Omar, M. S. (2025). Comparative Analysis of Feature Selection Methods for Twitter Sentiment Classification. Journal of Information Systems Engineering and Management, 10(215). Retrieved from https://jisem-journal.com/index.php/journal/article/download/3331/1442/5456.

[xix] Toupin, R. F. (2022). Who tweets climate change papers? Investigating publics of research through users' descriptions. Plos one 17, no. 6 : e0268999. , 312-345.

[xx] Paakki, F. V. (2024). "Ensemble-based Multilingual Euphemism Detection: a Behavior-Guided Approach," . in Proceedings of the 4th Workshop on Figurative Language Processing (FigLang 2024), , 73–78.

[xxi]Dahal, B. S. (2019). "Topic modeling and sentiment analysis of global climate change tweets.". Social network analysis and mining 9, 1-20. .

[xxii] R. K. Das, M. I. (2023). Sentiment analysis in multilingual context: Comparative analysis of machine learning and hybrid deep learning models. Heliyon, vol. 9, no. 9, 184-231.

[xxiii] Mumenthaler, C. O. (2021). "The impact of local temperature volatility on attention to climate change: Evidence from Spanish tweets." . Global environmental change 69 , 102-286.

[xxiv] Shyrokykh, K. M. (2023). Short text classification with machine learning in the social sciences: The case of climate change on Twitter. pmc.ncbi.nlm.nih.gov-https://doi.org/10.1371/journal.pone.0290762, 213-267.

[xxv] Jost, F. A. (2019). "How positive is "change" in climate change? A sentiment analysis.". Environmental Science & Policy 96, 27-36.

[xxvi] P. William, A. S. (2023). "Natural Language processing implementation for sentiment analysis on tweets," in Mobile Radio Communications and 5G Networks: . Proceedings of Third MRCN 2022. Springer, 317-327.

[xxvii] Omar, M., Salah, A., & Mahdi, M. A. (2025). Comparative Analysis of Feature Selection Methods for Twitter Sentiment Classification. Journal of Information Systems Engineering and Management, 10(215). Retrieved from https://jisem-journal.com/index.php/journal/article/download/3331/1442/5456

[xxviii] R. K. Das, M. I. (2023). Sentiment analysis in multilingual context: Comparative analysis of machine learning and hybrid deep learning models. Heliyon, vol. 9, no. 9, 184-231.

[xxix] Paakki, F. V. (2024). "Ensemble-based Multilingual Euphemism Detection: a Behavior-Guided Approach," . in Proceedings of the 4th Workshop on Figurative Language Processing (FigLang 2024), , 73–78.

[xxx] Omar, M., Salah, A., & Mahdi, M. A. (2025). Comparative Analysis of Feature Selection Methods for Twitter Sentiment Classification. Journal of Information Systems Engineering and Management, 10(215). Retrieved from https://jisem-journal.com/index.php/journal/article/download/3331/1442/5456

[xxxi] Jost, F. A. (2019). "How positive is "change" in climate change? A sentiment analysis.". Environmental Science & Policy 96, 27-36.

[xxxii] NHSJS. (2024, May 9) . A Comparative Analysis of Sentiment Classification Models for Improved Performance Optimization. Retrieved from https://nhsjs.com/2024/a-comparative-analysis-of-sentiment-classification-models-for-improved-performance-optimization/

[xxxiii] Emiliano, C. (2024, May 21). Twitter Sentiment Analysis: A Comprehensive Guide to NLP and Machine Learning Techniques. Medium. Retrieved from https://medium.com/@caio_emiliano/twitter-sentiment-analysis-a-comprehensive-guide-to-nlp-and-machine-learning-techniques-b5af9cd213a7

[xxxiv]K. Darshan, J. S. (2023). "NLP-Powered Sentiment Analysis on the Twitter," . Saudi J EngTechnol, vol. 9, no. 1, 1206-1213.

[xxxv] Mungalpara, J. (2023, April 27) . Evaluation Methods in Natural Language Processing (NLP): Part-1. Medium. Retrieved from https://jaimin-ml2001.medium.com/evaluation-methods-in-natural-language-processing-nlp-part-1-ffd39c90c04f

[xxxvi] Chai, .. P. (2023). , "Comparison of text preprocessing methods,". Natural Language Engineering, vol. 29, no. 3, 509–553.

[xxxvii] MLArchive. (2024, June 13). Text Classification & Sentiment Analysis. Machine Learning Archive. Retrieved from https://mlarchive.com/natural-language-processing/text-classification-sentiment-analysis/

[xxxviii] R. K. Das, M. I. (2023). Sentiment analysis in multilingual context: Comparative analysis of machine learning and hybrid deep learning models. Heliyon, vol. 9, no. 9, 184-231.

[xxxix] Omar, M., Salah, A., & Mahdi, M. A. (2025). Comparative Analysis of Feature Selection Methods for Twitter Sentiment Classification. Journal of Information Systems Engineering and Management, 10(215). Retrieved from https://jisem-journal.com/index.php/journal/article/download/3331/1442/5456

[xl] Chai, . P. (2023). , "Comparison of text preprocessing methods,". Natural Language Engineering, vol. 29, no. 3, 509–553.

[xli] Mumenthaler, C. O. (2021). "The impact of local temperature volatility on attention to climate change: Evidence from Spanish tweets." . Global environmental change 69 , 102-286.

[xlii] Emiliano, C. (2024, May 21). Twitter Sentiment Analysis: A Comprehensive Guide to NLP and Machine Learning Techniques. Medium. Retrieved from https://medium.com/@caio_emiliano/twitter-sentiment-analysis-a-comprehensive-guide-to-nlp-and-machine-learning-techniques-b5af9cd213a7

[xliii] P. William, A. S. (2023). "Natural Language processing implementation for sentiment analysis on tweets," in Mobile Radio Communications and 5G Networks: . Proceedings of Third MRCN 2022. Springer, 317-327.

[xliv]Gomede, E. (2024). Nuances of Opinion: Unveiling the Complexities of Stance Detection in Natural Language Processing. **AI monks.io**. Retrieved from https://medium.com/aimonks/nuances-of-opinion-unveiling-the-complexities-of-stance-detection-in-natural-language-processing-1add863b82ba.

[xlv] Toupin, R. F. (2022). Who tweets climate change papers? Investigating publics of research through users' descriptions. Plos one 17, no. 6 : e0268999. , 312-345.

[xlvi] Shyrokykh, K. M. (2023). Short text classification with machine learning in the social sciences: The case of climate change on Twitter. pmc.ncbi.nlm.nih.gov-https://doi.org/10.1371/journal.pone.0290762, 213-267.

[xlvii] P. William, A. S. (2023). "Natural Language processing implementation for sentiment analysis on tweets," in Mobile Radio Communications and 5G Networks: . Proceedings of Third MRCN 2022. Springer, 317-327.

[xlviii] Omar, M., Salah, A., & Mahdi, M. A. (2025). Comparative Analysis of Feature Selection Methods for Twitter Sentiment Classification. Journal of Information Systems Engineering and Management, 10(215). Retrieved from https://jisem-journal.com/index.php/journal/article/download/3331/1442/5456

[xlix] Omar, M. S. (2025). Comparative Analysis of Feature Selection Methods for Twitter Sentiment Classification. Journal of Information Systems Engineering and Management, 10(215). Retrieved from https://jisem-journal.com/index.php/journal/article/download/3331/1442/5456.

[l] Paakki, F. V. (2024). "Ensemble-based Multilingual Euphemism Detection: a Behavior-Guided Approach," . in Proceedings of the 4th Workshop on Figurative Language Processing (FigLang 2024), , 73–78.

[li]Dahal, B. S. (2019). "Topic modeling and sentiment analysis of global climate change tweets.". Social network analysis and mining 9, 1-20. .

[lii] Shyrokykh, K. M. (2023). Short text classification with machine learning in the social sciences: The case of climate change on Twitter. pmc.ncbi.nlm.nih.gov-https://doi.org/10.1371/journal.pone.0290762, 213-267.

[liii] Omar, M. S. (2025). Comparative Analysis of Feature Selection Methods for Twitter Sentiment Classification. Journal of Information Systems Engineering and Management, 10(215). Retrieved from https://jisem-journal.com/index.php/journal/article/download/3331/1442/5456.

[liv]Jost, F. A. (2019). "How positive is "change" in climate change? A sentiment analysis.". Environmental Science & Policy 96, 27-36.

[lv] Toupin, R. F. (2022). Who tweets climate change papers? Investigating publics of research through users' descriptions. Plos one 17, no. 6 : e0268999. , 312-345.

[lvi] Shyrokykh, K. M. (2023). Short text classification with machine learning in the social sciences: The case of climate change on Twitter. pmc.ncbi.nlm.nih.gov-https://doi.org/10.1371/journal.pone.0290762, 213-267.

[lvii]K. Darshan, J. S. (2023). "NLP-Powered Sentiment Analysis on the Twitter," . Saudi J EngTechnol, vol. 9, no. 1, 1206-1213.

[lviii] Omar, M. S. (2025). Comparative Analysis of Feature Selection Methods for Twitter Sentiment Classification. Journal of Information Systems Engineering and Management, 10(215). Retrieved from https://jisem-journal.com/index.php/journal/article/download/3331/1442/5456.

[lix] X. Zhou, X. Tao, J. Yong, and Z. Yang, "Sentiment analysis on tweets for social events," in Proceedings of the 2013 IEEE 17th international conference on computer supported cooperative work in design (CSCWD). IEEE, 2013, pp. 557–562.