

Performance Comparison of Traditional and Pre-Trained Models for News Sentiment Analysis

¹Surya Sriram Duvvuri, ²Tej Pratap, ^{3,*}Usha Moorthy

^{1,2,3} Department of Information Technology, Manipal Institute of Technology Bengaluru, Manipal Academy of Higher Education, Manipal, Karnataka, India

Abstract

This paper therefore seeks to discuss in detail Sentiment classification of live news articles using machine learning algorithms and techniques. The main focus is on using and comparing the performance of Naïve Bayes and Logistic Regression classifiers for sentiment analysis of news articles to determine their positive, negative or neutral sentiment. Also, the study will also seek to compare the performance of two pre-trained models namely VADER (Valence Aware Dictionary and Sentiment Reasoner) and RoBERTa (Robustly Optimized BERT pre-training Approach) in order to see the difference in the accuracy and analysis of the model to the traditional ML algorithmic approach. Comparing the performance of these models show that there are differences in the levels of accuracy, precision, recall and F1-score of each model, indicating that while the traditional algorithms have certain advantages and disadvantages when it comes to processing news sentiment, the pre-trained models are more effective in doing so. The results of the study are meant to give further understanding on how best to use the available machine learning algorithms in identifying sentiment from raw news texts.

1. Introduction

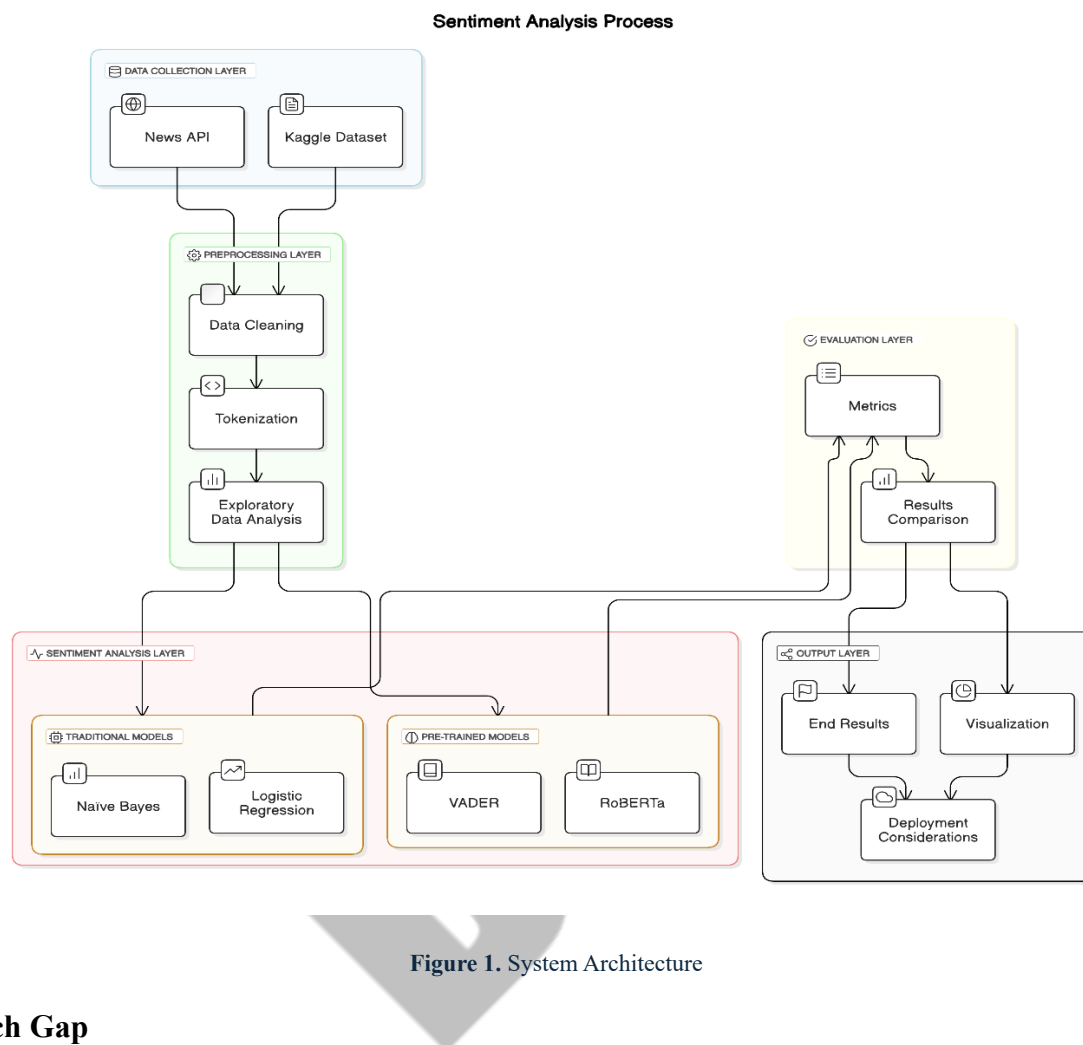
In today's world, news articles are not just means of providing information to the public. They can mould opinion, stimulate debate and even dictate action. Whenever news stories are shared around on different media platforms, it is useful to know their emotional tone as this can be useful to many different types of organization, business and policy makers. The news which is easily accessible has to represent the real world. This is where sentiment analysis comes into picture. It is a great tool that allows to determine if a particular text or title has a positive, negative or neutral sentiment.

For this study, we employ Naïve Bayes and Logistic Regression models for sentiment analysis of news articles and organising them into classes.[9] Logistic Regression is a model which is based on statistical data. It forecasts the possible probability of binary outcome by structuring the relation between dependant variable and independent variables. Logistic Regression is mostly used for classification instead of regression. It is well capable of handling the numerical datasets and features and performs better at binary classification. This model can utilize the L1 and L2 regularization that prevents overfitting making the model more accurate in terms of independent word sentiment.

Naïve bayes is based on the Bayes Theorem that works on the principle of probability of hypothesis. Naïve Bayes interprets high dimensional data well which further makes it easier to classify statements on its sentiment. It assumes that the words are independent of each other making the model more accurate in terms of sentiment. This model is immune to overfitting because it has built in smoothing that handles the words that are missing in the datasets.

The two algorithms are mostly preferred over other due to these factors that makes these two models more effective and more accurate. Considering these factors the study will help me to know how well this model compares with the more advance model such as VADER and RoBERTa. From this comparison, I want to reveal the potential and limitations of these approaches and suggest when they may be useful for identifying the sentiment of news. The detailed system architecture is depicted in Figure 1.

Sentiment analysis is a natural language processing (NLP) method used to analyze the emotional tone conveyed by a patch of text. It categorizes content as positive, neutral, or negative, offering a deeper insight into how an audience might want to see the information. [4] While traditional approaches, such as the Naïve Bayes algorithm, have been extensively used for sentiment classification due to their simplicity and efficiency, modern pre-trained models, like VADER and RoBERTa, offer more wider perspective to the interpretations. The paper utilizes an updated dataset for training and analyzing the models. This is a study that compares the two model’s efficiency, accuracy and effectiveness of the two models and exploring the Naïve bayes algorithm on the dataset.



2. Research Gap

The existing research by Parneet Kaur focuses on using machine learning algorithms for sentiment analysis of live news data collected through web scraping. While it demonstrates effective use of machine learning for sentiment classification, it is limited by relying on traditional algorithms without a detailed comparison to advanced pre-trained models, which have become increasingly influential in NLP tasks due to their enhanced contextual understanding and sentiment detection capabilities. In contrast, this study explores a broader approach by incorporating not only traditional machine learning algorithms like Naïve Bayes and Logistic Regression but also pre-trained models such as VADER and RoBERTa, comparing their effectiveness for sentiment analysis of news content. This allows for a deeper examination of how these algorithms perform across various metrics (accuracy, precision, recall, F1-score) when handling nuanced, context-rich data, such as news articles.[1]

The research gap addressed here lies in the need for a systematic analysis comparing the efficacy of traditional models against modern, pre-trained models for sentiment analysis on news datasets. This comparison aims to provide insights into selecting the most appropriate model based on data complexity and desired outcomes, an area that has been minimally explored in previous research.

3. Problem Formulation

The consumption of news in bits and pieces throughout different digital media has revolutionized how people think, act and react. The language used in writing news articles is usually complex and nuanced, which can create groups different of emotions stakeholders, in including the media readers organizations, including businesses, happiness, and sadness, governments, anger, to etc. know It the is mood crucial of for the different public towards different happenings, developments, or concerns.

Nevertheless, despite the increasing availability of news data, identifying the affective valence of these articles, that is, whether they are positive, negative or neutral, remains a challenging task. Today's systems often lack the refinement required to understand the various emotions that are expressed in the news which news hampers filtering, their sentiment utility analysis in of areas audiences like and self-organising opinion extraction.

This paper therefore seeks to solve these challenges by developing a smart framework for sentiment analysis of news articles using machine learning and pre-trained models. The specific research objectives that this study hopes to achieve are:

1. **To develop a comprehensive sentiment analysis system** that can categorize news articles as positive, negative, or neutral, based on their emotional tone.
2. **To compare the performance of traditional machine learning algorithms (Naïve Bayes, Logistic Regression)** with state-of-the-art, pre-trained models (VADER and RoBERTa), assessing their efficiency and accuracy in processing news data.

4. Research Methodology

In research paper authored by Parneet Kaur on sentiment analysis, the focus is on utilizing the Lexicon-based approach to analyze the sentiment of news articles by employing supervised machine learning algorithms and pre-trained models. Sentiment analysis is the process of determining the emotional tone behind pieces of text, which is useful for understanding the attitudes, opinions, and emotions expressed by individuals or groups.

Generally, sentiment analysis can be conducted using two main methods: supervised and unsupervised learning. Supervised learning methods work with labeled datasets where the sentiment (positive, negative, or neutral) of the text is already known. These methods train algorithms to accurately classify new, unseen data based on this labeled information. Supervised learning is divided into two categories:

1. **Classification:** Algorithms in this category sort data into specific classes or categories. Common classification algorithms include linear classifiers, support vector machines (SVMs), decision trees, and random forests. These algorithms are precise in grouping data based on learned patterns from the labeled datasets.
2. **Regression:** Regression algorithms are used for predictive analysis, where the goal is to understand the relationship between dependent and independent variables. They output continuous predicted values rather than discrete categories. Popular regression algorithms include linear regression, logistic regression, and polynomial regression.

On the other hand, unsupervised learning methods, such as the Lexicon-based approach, work with unlabeled data and do not rely on training data. This is especially useful when labeled datasets are scarce or unavailable. Unsupervised learning algorithms can be categorized into:

1. **Clustering:** Techniques like K-means clustering group data points based on similarities or differences without prior labeling. This helps in identifying inherent structures or patterns within the data.

2. **Association:** Association algorithms use rule-based approaches to find relationships between variables in a dataset. They identify patterns, correlations, or associations among data items, such as frequently co-occurring words in text analysis.
3. **Dimensionality Reduction:** These techniques simplify datasets by reducing the number of variables under consideration, extracting key features while preserving important information.

In the context of sentiment analysis, the Lexicon-based approach involves using a predefined dictionary (lexicon) of words annotated with their associated sentiment scores. Each word in the text is compared against this lexicon to determine its sentiment polarity. The overall sentiment of a document is calculated by combining the sentiment scores of individual words. This method doesn't require labeled training data because it relies on the sentiment information contained within the lexicon itself.

By integrating the Lexicon-based approach with supervised machine learning algorithms and pre-trained models, the paper aims to enhance the accuracy of sentiment classification in news articles. This combination leverages the strengths of both methods: the depth and nuance provided by lexicons and the predictive power of supervised learning models.

Understanding these methods is crucial for developing efficient sentiment analysis systems that can interpret the vast amounts of textual data available today, especially in fields like market analysis, social media monitoring, and news media.

The steps involved in this analysis are as follows:

Step 1: The identification of types of data needed is done.

Step 2: The dataset from Kaggle.com is used to analyze the model here. This dataset contains a sentiment analysis of news collected as of July 1, 2024. The news was collected using the News API, which provides access to real-time news from multiple global sources.

Step 3: Identification of packages such as the NLTK Python which includes the Sentiment Intensity Analyzer, Pandas, Matplotlib, NumPy Seaborn are done as per the data requirement.[4] The csv dataset is initialized.

```
[1] import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

data = pd.read_csv('/content/news_sentiment_analysis.csv')
data.head()
```

Figure 2. Importing Libraries

Step 4: A thorough Exploratory Data Analysis is done for the efficiency of the model. Cleaning the data and handling NaN values is necessary for successful implementation. The below is the final dataset attributes.



	Serial Number	Title	Description	Sentiment_Numeric
0	1	Pine View High teacher wins Best in State award...	ST. GEORGE — Kaitlyn Larson, a first-year teach...	1
1	2	Businesses Face Financial Strain Amid Liquidit...	Harare, Zimbabwe — Local businesses are grappl...	0
2	3	Musk donates to super pac working to elect Thu...	(marketscreener.com) Billionaire Elon Musk has...	1
3	4	US FTC issues warning to franchisors over unfu...	(marketscreener.com) A U.S. trade regulator on...	-1
4	5	Rooftop solar's dark side	4.5 million households in the U.S. have solar ...	1

Figure 3. Dataset Head

Step 5: Now for the RoBERTa pre-trained model, the following codes are initialized. These lines import the necessary tools from the Hugging Face Transformers library to work with RoBERTa. This specifies the RoBERTa model that's designed for sentiment analysis on News data[12]. The “tokenizer” variable loads the tokenizer for the model. The tokenizer breaks down text into smaller units called tokens that the model can understand since Machine Learning models don't understand String values. The “model” variable loads the actual RoBERTa model. Below is the code that is initialised.

Figure 4. Initializing RoBERTa Model and applying tokenization

```
[ ] from transformers import AutoTokenizer
    from transformers import AutoModelForSequenceClassification
    from scipy.special import softmax

MODEL = f"cardiffnlp/twitter-roberta-base-sentiment"
tokenizer = AutoTokenizer.from_pretrained(MODEL)
model = AutoModelForSequenceClassification.from_pretrained(MODEL)
```

 /usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:89: UserWarning: The secret 'HF_TOKEN' does not exist in your Colab secrets. To authenticate with the Hugging Face Hub, create a token in your settings tab (<https://huggingface.co/settings/tokens>), set it as secret in your Google Colab and You will be able to reuse this secret in all of your notebooks. Please note that authentication is recommended but still optional to access public models or datasets.

warnings.warn(

config.json: 100%  747/747 [00:00<00:00, 33.9kB/s]

vocab.json: 100%  899k/899k [00:00<00:00, 7.20MB/s]

merges.txt: 100%  456k/456k [00:00<00:00, 3.48MB/s]

special_tokens_map.json: 100%  150/150 [00:00<00:00, 5.03kB/s]

/usr/local/lib/python3.10/dist-packages/transformers/tokenization_utils_base.py:1681: FutureWarning: 'clean_up_tokenization_spaces' was not set. It will be set to

warnings.warn(

pytorch_model.bin: 100%  499M/499M [00:03<00:00, 172MB/s]

Step 6: Now applying the model to the whole dataset which gives the sentiment according to the model as shown in Fig.5.

```
[ ] res = {}
    for i, row in tqdm(data.iterrows(), total=len(data)):
        try:
            text = row['Title']
            myid = row['Serial Number']
            roberta_result = polarity_scores_roberta(text)
            rr=roberta_result
            res[myid]= rr
        except RuntimeError:
            print(f'Broke for id {myid}')
```

 100%  3500/3500 [10:22<00:00, 3.75it/s]

Figure 5. Model Implementation

Step 7: Now we get the accuracy score for this model as shown in Fig.6.

Accuracy for Negative (-1): 0.18
Accuracy for Neutral (0): 0.90
Accuracy for Positive (1): 0.13

Figure 6. Accuracy scores of RoBERTa Model.

Now for implementing the Vader Model.

Step 8: We get the following code for Vader Model initialization as shown in Fig.7. Then applying the model to the whole dataset as shown in Fig.8.

```
[ ] import nltk
    nltk.download('vader_lexicon')
    from nltk.sentiment import SentimentIntensityAnalyzer
    from tqdm.notebook import tqdm

    sia = SentimentIntensityAnalyzer()
```

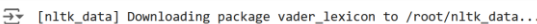
 [nltk_data] Downloading package vader_lexicon to /root/nltk_data...

Figure 7. Vader Model Initialization

```
[ ] res1 = {}
    for i, row in tqdm(data.iterrows(), total=len(data)):
        text = row['Title']
        myid = row['Serial Number']
        res1[myid] = sia.polarity_scores(text)
```

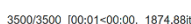
 100%  3500/3500 [00:01<00:00, 1874.88it/s]

Figure 8. Vader Model Implementation

Step 9: This model holds a fast initialisation compared to the previous model. We get the accuracy as follows as shown in Fig.9:

```
Accuracy for Negative (-1): 0.03
Accuracy for Neutral (0): 0.99
Accuracy for Positive (1): 0.02
```

Figure 9. Accuracy Scores of Vader Model.

Now for the Naive Bayes model we initialize the same dataset to get the accuracy and efficiency.

Step 10: The following code is used to initialise the Naive Bayes algorithm as shown in Fig.10.

```
[ ] import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn import naive_bayes
from sklearn.metrics import roc_auc_score

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Unzipping corpora/stopwords.zip.
```

Figure 10: Initialization of Navie Bayes.

Step 11: This code snippet sets up text data for machine learning using the CountVectorizer, which turns words into numerical counts that models can understand. It starts by initializing the vectorizer and tells it to ignore common English words (like "the" and "is") that don't contribute much to analysis. The fit_Transform function is applied to the training data X_train, learning the vocabulary and transforming the text into a matrix of word counts. Then, the transform() function is used on the test data X_test to convert it into the same word count format without learning a new vocabulary. This ensures that both training and testing data are represented in a consistent way for the model to analyze as shown in Fig.11, Fig.12 and Fig.13.

```
[ ] vectorizer = CountVectorizer(stop_words='english')
X_train_vec = vectorizer.fit_transform(X_train)
X_test_vec = vectorizer.transform(X_test)
```

Figure 11: Applying vectorization on dataset

```
model = MultinomialNB()
model.fit(X_train_vec, y_train)
```

```
▼ MultinomialNB ⓘ ?
MultinomialNB()
```

Figure 12. Implementing Test Train Method

```
Classification Report:
              precision    recall  f1-score   support

Negative (-1)    0.74      0.59      0.66       192
Neutral (0)      0.69      0.60      0.64       230
Positive (1)     0.79      0.88      0.83       628

   accuracy          0.76
  macro avg          0.74
weighted avg          0.76

Accuracy: 0.7628571428571429
```

Figure 13. Accuracy scores of Naïve Bayes.

Step 12: The Naïve Bayes algorithm gives the following accuracy shown in Fig.13:

Step 13: Now applying the Logistic Regression as shown in Fig.14 to the same dataset and analysing the result as shown in Fig.15.

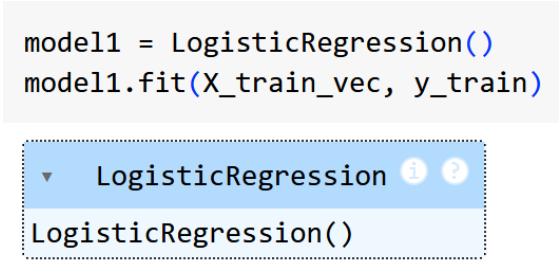


Figure 14. Initializing Logistic Regression

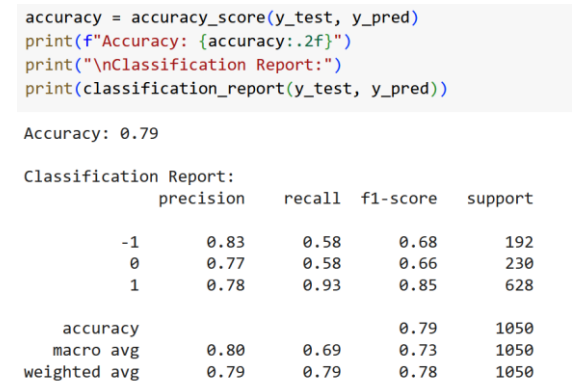


Figure 15. Accuracy Score of Logistic Regression

5. Result Analysis

Vader and RoBERTa both have their strengths and weaknesses for sentiment analysis, but they differ in performance and complexity. Vader is a rule-based tool that works well for short, informal text, offering faster results with decent accuracy for direct sentiment. However, its accuracy can be limited when dealing with nuanced language sentences. On the other hand, RoBERTa, a transformer-based model, provides a more complex analysis by understanding the context and relationships between words. This often results in higher accuracy on complex datasets but comes at the cost of a longer processing time. For news sentiment analysis, RoBERTa is far better than Vader due to its deeper comprehension of language used, but Vader may still be preferable for faster, simpler tasks (as shown in Fig.16 and Fig.17) .

The time efficiency between them is:
Vader:

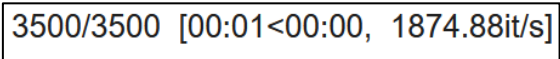


Figure 16. Vader Model's Processing Time

RoBERTa:

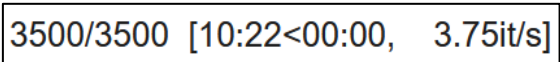


Figure 17. RoBERTA Model's Processing Time

The accuracy between them is shown in Table.1:

Table 1. Accuracy of Models

	RoBERTa	VADER
Negative	0.18	0.03
Neutral	0.90	0.99
Positive	0.13	0.02

The visual representation of the accuracy between the models are shown in Fig.18 and Fig.19:

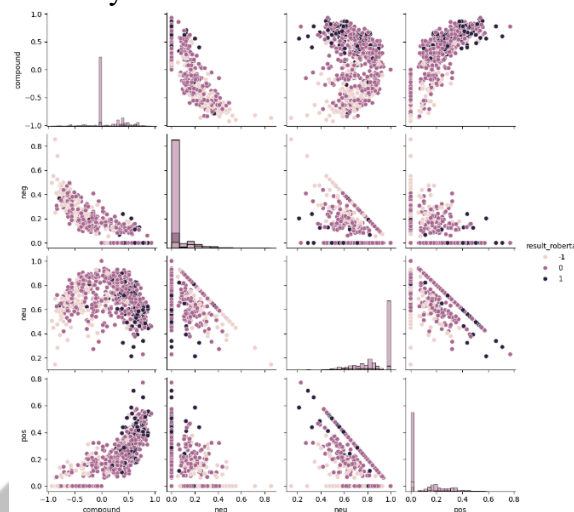


Figure 18. Pair plots showing the variation of sentiments of the models

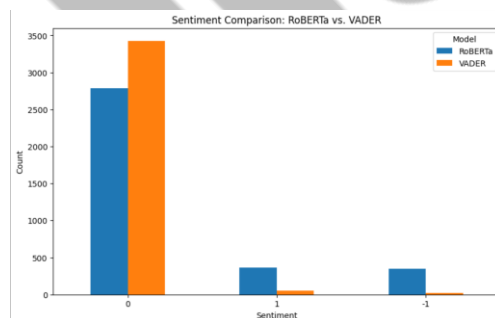


Figure 19. Bar Graph comparing accuracy of comparing the accuracy of Vader and RoBERTa models

On applying the traditional ML algorithms on the dataset we get the following accuracy using the test-train methodology as shown in Table 2 and Figure 20.

Table 2. Accuracy of Naive bayes and Logistic Regression

	Naive Bayes	Logistic Regression
Accuracy	0.76	0.79

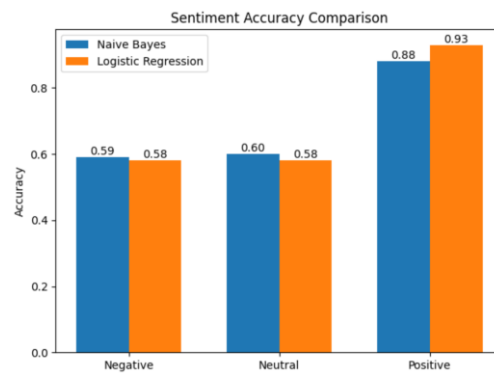


Figure 20. Bar Graph comparing the accuracy of Naive Bayes and Logistic Regression for each sentiment category

6. Conclusion:

The traditional Naïve Bayes algorithm is compared with two sophisticated pre-trained models, VADER and RoBERTa. This study's thorough examination of sentiment classification techniques for news articles and finding out how well these algorithms work in regard of accuracy and processing speed, and contextual comprehension on actual news data was the primary objective. This study provides insights for the best use of each strategy in the field of news, Sentiment analysis exposes the unique strengths and shortcomings of each method through a thorough review across several performance metrics.

One well-known supervised machine learning technique that has shown remarkable success in terms of computational efficiency and implementation simplicity is the Naïve Bayes model. Because of its ease of use, sentiment classification may be completed quickly, which is particularly advantageous for applications that need high processing speed but modest accuracy. But when it came to sophisticated grammar, irony, or sarcasm—situations where word context is essential for precise sentiment analysis—Naïve Bayes faltered.[5] Given that news articles frequently contain layered thoughts and indirect phrases, this constraint emphasizes the need for sophisticated models with greater linguistic comprehension.

On the other hand, because the VADER model is rule-based and uses a lexicon of words linked to particular sentiment scores, it works remarkably well with shorter, more straightforward texts. This model was created especially for sentiment analysis. For news data, VADER demonstrated a high level of efficiency, especially for simple sentiments. It did, however, show limitations when dealing with more complicated or ambiguous text, where comprehension of the larger context is crucial. This model works well for applications where simplicity and speed are key considerations and text inputs should have a fairly straightforward sentiment.[13]

The transformer-based model, RoBERTa, achieved the highest accuracy of all the models tested, outperforming both Naïve Bayes and VADER in handling complex text structures and capturing contextual nuances. For news sentiment analysis, which frequently uses complex language, this model was especially useful because it could identify subtle sentiment cues and comprehend the relationships between words in sentences. RoBERTa's computational demand and longer processing time, however, are trade-offs that might restrict its use for real-time sentiment analysis in settings with limited resources.[3][10]

The comparative analysis reveals that no single model serves as a universal solution for sentiment analysis in news data; rather, the choice of model depends heavily on the specific application requirements. For high-speed, lightweight applications where minor accuracy compromises are acceptable, Naïve Bayes or VADER would be advantageous. However, for tasks where high accuracy and contextual understanding are paramount, RoBERTa is the optimal choice, albeit with a need for computational resources and time management considerations.[4]

In Parneet Kaur's study, the Naïve Bayes algorithm attained a sentiment analysis accuracy of 58.66% on live news data, reflecting moderate effectiveness yet constraints in dealing with subtle language. In comparison,

our study shows a significant enhancement, as Naïve Bayes reaches an accuracy of 76.28% on identical data. This notable rise underscores the success of our strategy, indicating that through improved preprocessing and model adjustment, Naïve Bayes can produce more dependable outcomes in sentiment analysis for news articles, bridging the performance disparity observed in earlier research. In this instance, the Logistic Regression algorithms outperformed Naïve Bayes. It examined the variation in vocabulary and the tone of the speech more effectively. A significant enhancement in accuracy was noted, with the current study reaching 79%, a considerable rise from the 61% indicated in the earlier study, illustrating the effects of improved preprocessing and model adjustment.[1]

This research contrasts conventional machine learning algorithms with sophisticated pre-trained models, assisting researchers and developers in selecting the most appropriate algorithm for sentiment analysis. Subsequent studies could investigate hybrid methods that integrate conventional and transformer-based models to achieve a balance between efficiency and precision, while assessing model effectiveness on various datasets to enhance sentiment analysis for wider use.

References

- [1] Parneet Kaur. *“Sentiment analysis using web scraping for live news data with machine learning algorithms.”* (2022)
- [2] Claudio Feliciani, Alessandro Corbetta, Milad Haghani, Katsuhiro Nishinari. *“How crowd accidents are reported in the news media: Lexical and sentiment analysis”* (2024)
- [3] Amir Jabbar Lak, Reza Boostani, Farhan A. Alenizi, Amin Salih Mohammed, Seyed Mostafa Fakhrahmad. *“RoBERTa, ResNeXt and BiLSTM with self-attention: The ultimate trio for customer sentiment analysis”*(2024)
- [4] Muhammad Ittefaq, Ali Zain, Rauf Arif, Mohammad Ala-Uddin, Taufiq Ahmad, Azhar Iqbal. *“Global news media coverage of artificial intelligence: A comparative analysis of frames, sentiments, and trends across 12 countries”* (2024)
- [5] Yanshuang Li, Imran Yousaf, Shoaib Ali, Azza Bejaoui. *“Demystifying the dynamic relationship between news sentiment index and ESG stocks: Evidence from time-frequency wavelet analysis”* (2024)
- [6] Vyas, Vishal, and V. Uma. *“Approaches to Sentiment Analysis on Product Reviews.”* from the book *Sentiment Analysis and Knowledge Discovery in Contemporary Business*, edited by Dharmendra Singh Rajput, et al., IGI Global, (2019).
- [7] Manning, C. D., Prabhakar Raghavan, & Schütze, H. *“Introduction to Information Retrieval.”* from *Cambridge University Press*. (2009).
- [8] N. Godbole, M. Srinivasaiah, and S. Sekine. *“Large-scale sentiment analysis for news and blogs”*. in *International Conference on Weblogs and Social Media, Denver*, (2007).
- [9] Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. *“Thumbs up? Sentiment classification using machine learning techniques.”* (2002).
- [10] P. Chaovalit, and L. Zhou. *“Movie review mining: A comparison between supervised and unsupervised classification approaches”*. in *Proceedings 38th annual Hawaii international conference on system sciences*. (2005).
- [11] J. Reis, P. Olmo, F. Benevenuto, H. Kwak, R. Prates, and J. An. *“Breaking the news: first impressions matter on online news.”* (2015).
- [12] Noura A. Semary, Wesam Ahmed, Khalid Amin, Paweł Pławiak, and Mohamed Hammad. *“Improving sentiment classification using a RoBERTa-based hybrid model”* (2023)
- [13] C. J. Hutto, Eric Gilbert. *“VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text.”* (2015).

[14] M. Devika, C. Sunitha, A. Ganesh, “*Sentiment analysis: a comparative study on different approaches.*” (2016).

DSS