Social Media-Based Sentiment Analysis Using NLP And Machine Learning

Pallavi Subhash Malode[1], Dr. K.D. Kharat[2]
[1]M. Tech Student, Department of Computer Science and Engineering
[2]Associate Professor, Department of Computer Science and Engineering
[1,2]Chh. Shahu College Of Engineering Chhatrapati Sambhajinagar

Abstract— **The exponential rise of political news content across digital platforms has intensified the need for automated methods to analyze sentiment embedded in textual narratives. Manual inspection is neither scalable nor consistent, particularly when handling large volumes of ideologically diverse articles. This study presents a supervised learning–based sentiment analysis framework optimized for classifying political news into positive, negative, and neutral categories. The system employs a structured pipeline integrating preprocessing, tokenization, part-of-speech tagging, named entity recognition, aspect extraction, and machine-learning classification. Algorithms including Naïve Bayes, Support Vector Machine (SVM), and a proposed optimized model were evaluated using curated political news datasets. Experimental findings indicate that the optimized model outperforms standard classifiers, achieving 88% precision, 96% recall, 92% F-score, and 87% accuracy, demonstrating its reliability in detecting sentiment within complex political narratives. The work contributes to political communication research by providing a scalable NLP-based solution capable of identifying media bias, assessing public opinion, and supporting data-driven policy analysis.**

I. INTRODUCTION

The digital transformation of journalism has reshaped the communication ecosystem, enabling instantaneous dissemination of political news across online newspapers, independent portals, and social networks. As citizens increasingly rely on online information to form political judgments, understanding the sentiment embedded in such content has become essential. Political news carries implicit opinions, ideological leanings, and narrative framing that influence how readers interpret political events. This underscores the need for automated tools capable of analyzing large-scale data with consistency and objectivity.

Political narratives often embed subtle cues whether through selective wording, rhetorical emphasis, or contextual insinuation that shape public perceptions. Even news articles presented as neutral may contain sentiment signals revealed through tone, emphasis, or linguistic structure. Detecting such embedded sentiment manually is challenging due to human bias and the sheer volume of political reporting produced every minute. Therefore, computational sentiment analysis plays a crucial role in decoding implicit political messages and evaluating sentiment trends across media outlets.

While sentiment analysis has shown success in domains such as product reviews and social media posts, political news introduces deeper complexities. Political language frequently uses ambiguous or context-dependent terms, sarcasm, metaphorical framing, and nuanced ideological references. Words like *reform*, *liberal*, *strike*, or *opposition* may carry different sentiments depending on the political context. Traditional models that rely on generic sentiment lexicons often fail to interpret such domain-specific nuances accurately.

This research aims to address these limitations by designing a sentiment analysis framework specifically tailored for political news articles. By incorporating part-of-speech cues, entity recognition, aspect extraction, and machine-learning classification, the system generates more accurate sentiment interpretations of politically charged content. Furthermore, by comparing Naïve Bayes, SVM, and a proposed optimized model, the study evaluates how algorithmic choices influence sentiment classification performance. The outcomes of this research provide a foundation for improved political analytics, media monitoring, and sentiment-driven policy assessment.

II. LITERATURE REVIEW

Existing research on sentiment analysis highlights significant advances in NLP, classification algorithms, and contextual interpretation techniques. Early approaches relied heavily on lexicon-based models, which performed well on straightforward texts but struggled with contextual variations found in political narratives. More recent studies employ machine-learning classifiers such as Naïve Bayes, SVMs, and neural network models to capture deeper linguistic patterns.

Political text analytics presents unique challenges compared to domains like product reviews. Ambiguous descriptors, ideological framing, and multi-layered semantics require robust contextual processing. Studies suggest that combining linguistic features such as POS tags, entity recognition, and syntactic structures with machine-learning models enhances classification accuracy. Advanced works emphasize the need for domain-specific lexicons and optimized feature selection to improve performance on political corpora.

Classification algorithms such as Naïve Bayes, SVM, and decision trees remain widely adopted due to their reliability and interpretability. However, comparative studies indicate that hybrid or optimized models often outperform traditional classifiers, especially on large and diverse datasets.

Table Summary of Literature Review

Author(s)	Focus Area	Methodology	Key Findings
Amangeldi et al. (2024)	Social-media sentiment interpretation	NLP + emotion classification	Achieved high accuracy on multi-emotion datasets.
Guimaraes et al. (2016)	Polarity detection with adverb analysis	Sentiment scoring	Enhanced polarity prediction with contextual adverbs.
Nguyen et al. (2013)	Linguistic patterns on Twitter	Statistical analysis	Established correlation between linguistic style and sentiment.
Sloan et al. (2015)	Demographic inference on social media	Metadata analysis	Identified demographic patterns influencing sentiment.
Liu et al. (2013)	Scalable sentiment classification	Naive Bayes	Demonstrated effectiveness of Bayesian models on big data.
Belkebir & Guessoum (2013)	Text categorization (Arabic)	Hybrid BSO-Chi2-SVM	Improved accuracy over traditional SVM.
Ali et al. (2014)	Spam sentiment detection	Multi-class classification	Achieved better filtering accuracy with novel algorithm.
Peersman et al. (2011)	Gender and sentiment prediction	ML classifier	Highlighted importance of linguistic features.

III. SYSTEM DEVELOPMENT

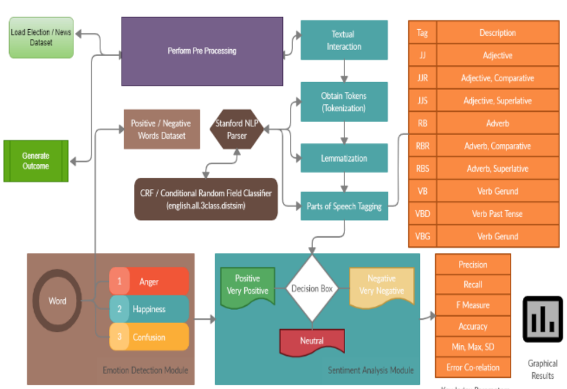

Figure 1 Proposed System Architecture

Load Election / News Dataset

This is the pipeline's input stage. The system ingests line-separated news articles, headlines or election-related posts (LSV files or similar). At this step the raw documents are simply read into memory or loaded into a processing queue; metadata such as source, publication date, and article ID can also be stored for later filtering or entity-level aggregation. The pipeline assumes each record is a single document or sentence that will be processed independently before any higher-level aggregation.

Perform Pre-processing

Preprocessing is the first active text normalization phase. It removes obvious noise HTML tags, URLs, extra whitespace, punctuation (when appropriate), and non-text artifacts and may apply lowercasing or basic Unicode normalization. It also optionally handles dataset cleaning tasks such as removing duplicates and filtering out extremely short or malformed lines. Preprocessing prepares text for deterministic NLP steps and reduces downstream error rates caused by dirty input.

Textual Interaction
This block represents lightweight interactive checks and optional human-in-the-loop steps: quick previews, schema verification, or rule edits before bulk processing. It's where an analyst might inspect tokenization samples, tweak stop word lists, or adjust thresholds. Conceptually it's the control interface that lets operators verify and steer preprocessing outputs without stopping the automated pipeline.

Obtain Tokens (Tokenization)
Tokenization splits each cleaned text into atomic units (tokens). Depending on the configuration, tokens may be words, subwords, or punctuation marks retained for syntactic cues. Tokenization also sets up boundaries used by POS taggers and parsers, and is often sensitive to language-specific rules (e.g., clitics, contractions). Good tokenization is essential because almost every subsequent NLP component depends on correct token boundaries.

Lemmatization
Lemmatization reduces words to their canonical dictionary form (lemmas) e.g., *running* → *run*, *better* → *good*. Unlike crude stemming, lemmatization relies on POS information to return linguistically valid base forms. This reduces feature sparsity (many inflected forms collapsing to a single lemma) and helps match tokens to lexicons (sentiment dictionaries or aspect lists) more reliably.

Parts-of-Speech (POS) Tagging
POS tagging assigns grammatical categories (JJ, RB, VB, etc.) to tokens. These tags reveal where sentiment-bearing tokens often appear (adjectives and adverbs), help disambiguate lemma choices for lemmatization, and form features for more advanced models. The diagram's POS tag table (JJ, JJR, JJS, RB, RBR, RBS, VB, VBD, VBG, etc.) shows the tags used for feature extraction and for constructing syntactic heuristics (for example, adjective-noun pairs or negation scopes).

Stanford NLP Parser
The Stanford NLP Parser (or equivalent syntactic parser) builds phrase structure or dependency trees for sentences. This produces relationships such as subject → verb → object and can be used to scope sentiment to targets (which entity is being described), resolve negation, and extract aspect–sentiment pairs. The parser's outputs are richer than POS tags and are crucial for mapping sentiment to the correct named entities and for feeding structured features to the CRF or other classifiers.

Positive / Negative Words Dataset (Sentiment Lexicons)
This block is a curated lexicon of positive and negative words, idioms, and possibly intensity modifiers (very, slightly, extremely). Lexicons are used for initial polarity cues and feature generation (e.g., counts of positive/negative words, polarity scores, or lexicon-based sentiment votes). In political text, domain-tuned lexicons (including domain-specific phrases and negation rules) substantially improve accuracy compared with generic sentiment lists.

CRF / Conditional Random Field Classifier
The CRF module performs sequence-level or structured classification (the example indicates english.all. 3class.distsim), combining local token features, context windows, POS tags, lexicon hits, and parse-derived features. CRFs are good for tagging tasks (like aspect detection or fine-grained sentiment at the token/phrase level) because they model the conditional dependencies between neighboring labels. In this pipeline the CRF helps produce robust sentence- or span-level sentiment labels before final aggregation.

Sentiment Analysis Module
This side module maps words or phrases to categorical emotions (e.g., anger, happiness, confusion) using emotion lexicons, rule-based heuristics, or classifier outputs. Each token or sentence can be assigned one or more emotion labels with associated scores. Emotion detection enriches polarity outputs by indicating affective tone (not just positive/negative/neutral), which is especially valuable in political analysis to distinguish anger-laden negative coverage from calm critical analysis.

Decision Box / Sentiment Analysis Module (Positive / Neutral / Negative)
After features and preliminary labels are produced, the Decision Box aggregates signals to decide final sentiment classes: Positive (Very Positive), Negative (Very Negative), or Neutral. Aggregation strategies combine classifier probabilities (SVM/CRF outputs), lexicon scores, entity-targeted sentiment (via parsing/NER), and emotion indicators. The Decision Box also contains thresholds, tie-breaking heuristics, and normalization rules that convert continuous scores into discrete labels, and it handles confidence scoring and fallback rules when signals conflict.

Generate Outcome
This block produces the final outputs: labeled articles/sentences with sentiment labels, emotion tags, entity-level sentiment summaries, and associated confidence scores. Outputs can be written to a database, exported as CSV/LSV, or fed into dashboards. The system can also generate downstream artifacts like trend summaries (daily polarity counts), entity reputation scores, and contextualized reports for analysts.

Key Index Parameters
Evaluation is performed using standard classification metrics: Precision, Recall, F-Measure (F1), and Accuracy the diagram also lists Min/Max/SD and error correlation, which indicate distributional characteristics of model performance and whether specific error patterns correlate with features (e.g., bias toward certain sources or entities). These metrics are computed on test sets or cross-validation folds to quantify how well the pipeline generalizes.

3.1 Data Pre-processing
Political news articles stored in LSV (Line-Separated Values) format are first parsed into a structured view. Preprocessing includes:

- Removal of noise, stopwords, and irrelevant tokens
- Tokenization
- Lemmatization
- Normalization
- Cleaning and integration of multi-source datasets

This ensures the text is standardized for efficient machine-processing.

3.2 Linguistic Feature Extraction
The system employs several NLP components:

- Parts-of-Speech Tagging (POS): Helps identify adjectives, adverbs, and verbs that influence sentiment.
- Named Entity Recognition (NER): Extracts political actors, organizations, locations, and institutions.
- Aspect Extraction: Identifies focal topics of the sentence (e.g., "election results," "policy reform").
- Word-Entity Mapping: Links tokens with their entity classes for targeted sentiment assignment.

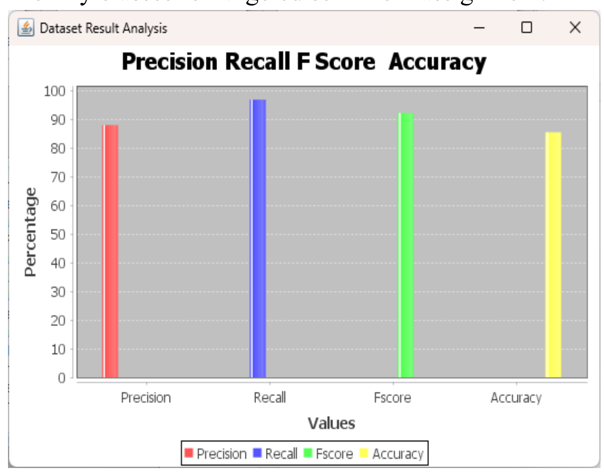

Figure 2 Result on POS Tagging and Classification

These features help distinguish sentiment directed at specific political entities.

3.3 Machine-Learning Classification
Three main classifiers were evaluated:

1. Naïve Bayes: A probabilistic model assuming word independence. Fast but limited by feature independence assumptions.
2. SVM: Finds an optimal hyperplane to separate sentiment classes; performs better on high-dimensional data.
3. Proposed Optimized Model: A hybrid classifier leveraging feature weighting, entropy-based filtering, and improved lexical integration (as described in the document).

The optimized model showed the strongest classification performance.

3.4 Knowledge Discovery and Evaluation
The KDD cycle cleaning, integration, selection, transformation, mining, evaluation, and presentation

was applied to derive meaningful insights from political articles. Evaluation metrics include:

- Accuracy
- Precision
- Recall
- F-Score

IV. RESULTS AND DISCUSSION

The optimized model demonstrated superior efficiency over Naïve Bayes and SVM, as confirmed by the evaluation table.

Table 2 Comparison of optimized technique with existing techniques

Metric	Optimized Model	SVM	Naïve Bayes
Precision	88%	84%	79%
Recall	96%	89%	83%
F-Score	92%	86%	81%
Accuracy	87%	85%	80%

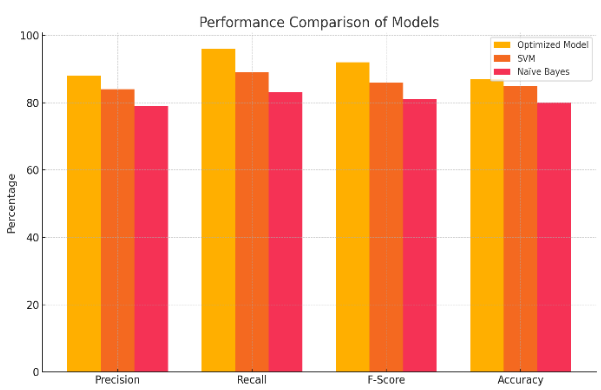

Figure 3 Performance Comparison of Various Models

Figure 2 bar chart presents the performance evaluation of the sentiment analysis model using four key metrics: Precision, Recall, F-Score, and Accuracy. These metrics reflect how effectively the system classifies news articles into their respective sentiment categories.

1. Precision (≈ 88%)

Precision indicates how many of the items predicted as Positive, Negative, or Neutral were actually correct.
A high precision value (~88%) shows that the model makes very few false positives, meaning most predicted sentiments are accurate and reliable.

2. Recall (≈ 97%)

Recall measures how well the system detects all actual instances of sentiment in the dataset.
A very high recall (~97%) means the model successfully identifies almost all sentiment-bearing articles, with very few false negatives. This indicates strong coverage and detection capability.

3. F-Score (≈ 93%)

The F-Score is the harmonic mean of precision and recall.
With a value around 93%, it signifies that the model maintains a strong balance between producing correct predictions (precision) and capturing all relevant instances (recall).
This balanced performance indicates overall robustness of the sentiment classifier.

4. Accuracy (≈ 87%)

Accuracy represents the percentage of all predictions that were correct across the entire dataset.
An accuracy close to 87% suggests that the classifier performs reliably when considering all classes collectively.

- High Recall (96%) indicates the optimized model successfully identifies nearly all sentiment-bearing articles, minimizing false negatives a crucial aspect in political analysis.
- Precision of 88% signifies low false-positive rates, ensuring reliability.
- F-Score of 92% confirms a balanced model performing well across all metrics.
- The performance gap between SVM and Naïve Bayes reflects the importance of contextual features in political texts; Naïve Bayes' independence assumption affects accuracy.

V. CONCLUSION

This research successfully develops a comprehensive sentiment analysis framework tailored for political news. By integrating NLP techniques such as POS tagging, NER, and aspect extraction with an optimized machine-learning classifier, the system achieves high accuracy and balanced performance. The results confirm that domain-specific optimization significantly improves sentiment detection in politically charged texts. The system is applicable to journalism, political research, media monitoring, and policy evaluation. Future improvements may include transformer-based deep-learning architectures, multilingual support, and bias-detection mechanisms

REFERENCES

[1] Amangeldi, Daniyar, et al. "Understanding Environmental Posts: Sentiment and Emotion Analysis of Social Media Data." IEEE Access, 29 Feb. 2024, doi:10.1109/ACCESS.2024.3371585.

[2] Guimaraes, R. G., et al. "Recommendation System Using Sentiment Analysis Considering the Polarity of the Adverb." 2016 IEEE International Symposium on Consumer Electronics (ISCE), Sept. 2016, pp. 71–72.

[3] Peersman, C., et al. "Predicting Sentiment and Gender in Online Social Networks." Proceedings of the 3rd International Workshop on Search and Mining User-Generated Contents, ACM, 2011, pp. 37–44.

[4] Van de Loo, J., et al. "Text-Based Sentiment and Gender Prediction for Online Safety Monitoring." Computational Linguistics in the Netherlands, vol. 5, no. 1, Dec. 2016, pp. 46–60.

[5] Filho, R. M., et al. "Twitter Population Sample Bias and Its Impact on Predictive Outcomes: A Case Study on Elections." IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ACM, Aug. 2015, pp. 1254–1261.

[6] Nguyen, D.-P., et al. "How Old Do You Think I Am? A Study of Language Use and Sentiment in Twitter." Seventh International AAAI Conference on Weblogs and social media, AAAI, 2013, pp. 439–448.

[7] Sloan, L., et al. "Who Tweets? Deriving the Demographic Characteristics of Age, Occupation and Social Class from Twitter User Metadata." PLOS ONE, vol. 10, no. 3, 2015, pp. 1–20.

[8] Schler, J., et al. "Effects of Sentiment and Gender on Blogging." AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs, 2006, pp. 199–205.

[9] Goswami, S., et al. "Stylometric Analysis of Bloggers' Sentiment and Gender." International AAAI Conference on Web and social media, May 2009, pp. 214–217.

[10] Sharma, K., and N. Jatana. "Bayesian Spam Classification: Time Efficient Radix Encoded Fragmented Database Approach." IEEE, 2014, pp. 939–942.

[11] Sharma, A., and Anchal. "SMS Spam Detection Using Neural Network Classifier." International Journal of Advanced Research in Computer Science, vol. 4, no. 6, 2014, pp. 240–244.

[12] Ali, M., et al. "Multiple Classifications for Detecting Spam Tweets by Novel Consultation Algorithm." CCECE 2014, IEEE, 2014, pp. 1–5.

[13] Liu, B., et al. "Scalable Sentiment Classification for Big Data Analysis Using Naïve Bayes Classifier." IEEE, 2013, pp. 99–104.

[14] Belkebir, R., and A. Guessoum. "A Hybrid BSO-Chi2-SVM Approach to Arabic Text Categorization." IEEE, 2013.

[15] Allias, N. "A Hybrid Gini PSO-SVM Feature Selection: An Empirical Study of Population Sizes on Different Classifier." 2013.

[16] Prasad, N., et al. "Comparison of Back Propagation and Resilient Propagation Algorithm for Spam Classification." International Conference on Computational Intelligence, Modelling and Simulation, IEEE, 2013, pp. 29–34.

[17] Jia, Z., et al. "Research on Web Spam Detection Based on Support Vector Machine." IEEE, 2012, pp. 517–520.

[18] Panigrahi, P. "A Comparative Study of Supervised Machine Learning Techniques for Spam Tweets Filtering." International Conference on Computational Intelligence and Communication Networks, IEEE, 2012, pp. 506–512.

[19] Clark, J., et al. "A Neural Network Based Approach to Automated Tweets Classification." 2010.