Cloud-Based Sentiment Analysis using VADER & Amazon Comprehend

Arya Shende[1], Aditya Kakade[2], Shreyash Parve[3], Aditya Patil[4], Prof. Dr. Umang Garg*
[1,2,3,4] *Dept. of Computer Science MIT ADT University Pune, India. *Guided*

***Abstract*—With the explosion of online textual content across social media, reviews, and forums, understanding public sentiment has become essential. This project proposes a hybrid sentiment analysis approach using two tools: VADER, a rule-based model effective for short informal texts, and Amazon Comprehend, a machine learning-based cloud service. By leveraging the strengths of both models and mitigating their limitations, the system offers a more comprehensive sentiment analysis solution. The entire pipeline is deployed on AWS, utilizing services like Lambda for processing and QuickSight for real-time sentiment visualization. This hybrid model aims to provide accurate, scalable, and insightful sentiment analysis for practical applications.**

I. INTRODUCTION

In today's digital landscape, the volume of textual data generated from sources such as social media, customer reviews, and online forums has grown exponentially. Extracting meaningful insights from this unstructured text has become a critical task for organizations aiming to understand user behavior, monitor brand reputation, and make data-driven decisions. Sentiment analysis, also known as opinion mining, is a natural language processing (NLP) technique used to determine the emotional tone behind a body of text. It helps classify the sentiment expressed as positive, negative, or neutral.

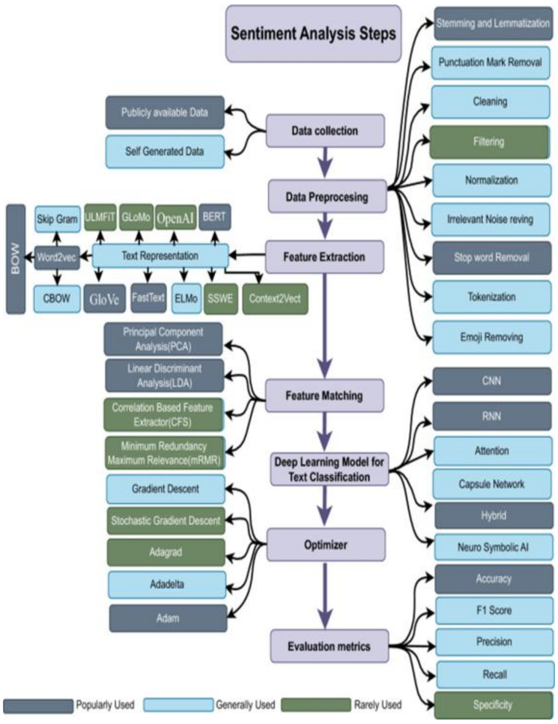
This paper presents a comparative and practical implementation of two popular sentiment analysis tools—VADER (Valence Aware Dictionary and sEntiment Reasoner) and Amazon Comprehend, a cloud-based NLP service by AWS. VADER is a rule-based model optimized for social media texts and short content, while Amazon Comprehend leverages machine learning and deep learning techniques to analyze sentiment in more complex documents.

The integration of these tools is achieved using AWS Lambda for serverless deployment, ensuring scalability and low operational overhead. The application is programmed in Python using the Boto3 SDK to interface with AWS services. For visualization, tools like Streamlit and Amazon QuickSight are employed to deliver real-time, interactive dashboards to users.

This research aims not only to demonstrate how these two sentiment analysis tools work but also to compare their performance, highlight their strengths and limitations, and deploy them in a real-time environment using cloud technologies.

II. SYSTEM DESIGN

A. Architecture Overview



Major Algorithms and Tools Used

1. VADER (Valence Aware Dictionary and sEntiment Reasoner)

VADER is a specialized sentiment analysis algorithm optimized for short text, especially social media and online content, where language can be informal and abbreviated. It uses a pre-built lexicon of words with pre-calculated sentiment scores, as well as rules that account for various language intensifiers (such as "very", "extremely"), negations (such as "not", "never"), and punctuation (e.g., exclamation marks).

- Working Mechanism:
  - Lexicon-Based Scoring: VADER uses a dictionary where each word is assigned a sentiment value (positive, negative, or neutral). It then processes the text to sum these scores.
  - Contextual Modifications: VADER also considers context-modifying factors such as punctuation and capitalization, which may alter the sentiment.
  - Sentiment Scores:
- Positive: Percentage of positive sentiment.
- Negative: Percentage of negative sentiment.
- Neutral: Percentage of neutral sentiment.
- Compound Score: A single value summarizing overall sentiment, which is bounded between -1 (negative) and +1 (positive).

2. Amazon Comprehend

Amazon Comprehend is a cloud-based NLP service that utilizes machine learning to perform a variety of text analytics tasks.

Comprehend can be used to extract sentiment, entities, language, and key phrases from unstructured text, and it is designed to scale easily for large datasets.

- Working Mechanism:
  - Sentiment Analysis: Comprehend analyzes text to classify it as positive, negative, neutral, or mixed based on context, tone, and word choice.
  - Entity Recognition: Identifies entities in the text, such as people, places, dates, and events.
  - Key Phrase Extraction: Extracts important keywords or phrases that provide insight into the document’s main points.
  - Language Detection: Automatically detects the language of the text, supporting a wide variety of languages.
  - Topic Modeling: Detects common themes across large volumes of text.

3. AWS Lambda

AWS Lambda is a serverless compute service that allows you to run code in response to events without provisioning or managing servers. Lambda is ideal for serverless architectures where you can trigger functions based on input (e.g., a new message, file upload, etc.).

- Working Mechanism:
  - Event-driven architecture: Lambda functions are triggered by specific events, such as changes in an S3 bucket, an API Gateway request, or a CloudWatch event.
  - Serverless and auto-scaling: AWS automatically provisions the compute resources needed to run the function, scaling based on demand.
  - Integration with other AWS services: Lambda works seamlessly with AWS services like S3, DynamoDB, SNS, and Comprehend, which allows you to create end-to-end data processing pipelines.

4. AWS SDK (Boto3)

Boto3 is the Python SDK for AWS, providing an interface to interact with AWS services programmatically. It allows you to manage resources (e.g., Lambda functions, Comprehend jobs) and manipulate data stored in AWS services (e.g., S3).

- Working Mechanism:
  - AWS Service Integration: Using Boto3, you can interact with AWS services directly from your Python code. It helps you to call Amazon Comprehend for text analysis, trigger Lambda functions, manage resources in S3, and much more.
  - Simplified API: Boto3 abstracts the complexity of directly working with HTTP requests, making it easier to work with AWS services.

5. Additional Tools and Libraries

- Pandas & NumPy: These Python libraries are widely used for data manipulation and analysis. Pandas simplifies handling dataframes, while NumPy assists with numerical operations. Together, they are essential for preparing and preprocessing data before passing it to machine learning models or cloud services.
- Matplotlib & Seaborn: These visualization libraries allow you to visualize sentiment analysis results. They can generate graphs like bar charts, heatmaps, and pie charts to better

represent sentiment distributions or entity relationships.

- Jupyter Notebook: A tool used for interactive coding and running Python scripts. It helps in exploring data, testing models, and visualizing results in a dynamic and iterative environment.

III. RESULTS

A . Performance of VADER on Sample Datasets

VADER demonstrated strong performance when applied to datasets composed of short, informal texts such as tweets, user comments, and product reviews. Its rule-based and lexicon-driven approach efficiently captured sentiment polarity, especially for texts with clear emotional cues. Key observations:

- High accuracy for sentiment classification in short sentences.
- Effective detection of sentiment-laden emojis, slang, and punctuation.
- Struggled with sarcasm, negations with subtle implications, and domain-specific jargon.

B. Performance of Amazon Comprehend on the Same Datasets

Amazon Comprehend offered broader contextual understanding and deeper insight through its machine learning capabilities. It performed well with both short and long texts and identified nuanced sentiments in complex sentences. Key findings:

- Handled multi-sentence reviews better by recognizing mixed sentiments within a single input.
- Capable of identifying key phrases and entities, enhancing downstream analysis.
- Higher latency compared to VADER, and its sentiment scores occasionally misclassified sarcasm or regionally specific language.

C. Comparative Analysis and Hybrid Benefits

When both tools were integrated, the hybrid framework yielded improved overall accuracy and reliability in sentiment classification. The system leveraged VADER for quick, surface-level insights and Amazon Comprehend for deeper, contextual interpretation. Highlights include:

- Reduced false positives/negatives when both tools agreed or when cross-validation was applied.
- Provided scalable, cloud-integrated analysis with storage and visualization using AWS Lambda, S3, and QuickSight.
- Enabled real-time dashboards that tracked sentiment trends over time, offering actionable insights for businesses.

IV. DISCUSSION

A. Implications of the Results

The comparative analysis of VADER and Amazon Comprehend reveals valuable insights into the applicability of different sentiment analysis tools for varying use cases:

- Real-Time Feedback: VADER’s lightweight and fast processing makes it highly suitable for real-time applications, such as live social media monitoring or chatbot responses.
- Contextual Accuracy: Amazon Comprehend demonstrates a better grasp of nuanced, mixed sentiments and contextual language, making it more reliable for analyzing longer-form content like reviews and reports.
- Business Intelligence Integration: With visualization through Streamlit or Amazon QuickSight, the sentiment results can be transformed into actionable insights, aiding decision-makers in understanding customer feedback, trends, and engagement.
- Cloud Integration and Scalability: Using AWS services like Lambda, S3, and Boto3 allows for seamless, scalable deployment, improving system responsiveness and reducing manual workloads.

B. Limitations

Despite promising results, the analysis process and tools used present several constraints:

- Rule-Based Simplicity (VADER): VADER lacks deep contextual understanding and can misinterpret sarcasm, idioms, or domain-specific terminology.
- Cost and Complexity (Amazon Comprehend): Comprehend, while more advanced, comes at a financial cost and requires deeper AWS knowledge to fully integrate and customize.
- Language Constraints: VADER supports

only English, and while Amazon Comprehend supports multiple languages, its performance across them may vary.

- Model Transparency: Amazon Comprehend operates as a black-box model; users have limited insight into how predictions are made, reducing explainability.

C. Future Work

To enhance the system's robustness and usability, the following improvements are proposed:

- Hybrid Approach: Combining VADER for real-time sentiment and Amazon Comprehend for deeper post-analysis can offer both speed and depth in understanding text.
- Improved Sentiment Models: Integration of more advanced NLP models (e.g., BERT-based transformers) can increase accuracy, especially for sarcasm and complex sentiments.
- Multilingual Sentiment Support: Adding support for additional languages using custom models or expanding the use of Amazon Comprehend's multilingual features would increase accessibility.
- Domain Customization: Training custom models with domain-specific data in Comprehend could improve classification in specialized fields like healthcare, finance, or education.
- Enhanced Visualization: Building dynamic dashboards with Amazon QuickSight or embedding Streamlit apps with real-time updates will improve decision-making capabilities.

V. CONCLUSION

This project demonstrates the capabilities of both VADER and Amazon Comprehend in sentiment analysis across different types of data. VADER offers simplicity and speed, while Amazon Comprehend excels in depth and scalability. While each has limitations, their combined use presents a powerful approach for comprehensive sentiment monitoring. Future enhancements focused on emotional depth, multilingual support, and richer visualizations will further elevate the effectiveness of the solution in real-world applications.

VI. ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to MIT ADT University for providing the necessary resources, guidance, and support throughout the development of the Sentiment Analysis using VADER and Amazon Comprehend project. Special thanks to the faculty members and research team whose invaluable insights, encouragement, and feedback were instrumental in shaping the direction of this research. We also wish to acknowledge the contributions of AWS and the broader cloud computing community for providing the tools and technologies that made this project feasible. Finally, we would like to express our appreciation to the individuals and organizations who participated in the testing phase, providing valuable feedback that helped refine the models and improve the overall performance of the sentiment analysis system. This project would not have been possible without the collaborative efforts of all those involved.

REFERENCES

[1] Hutto, C.J., & Gilbert, E. (2014). *VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text*. In Proceedings of the Eighth International Conference on Weblogs and Social Media (ICWSM-14). https://ojs.aaai.org/index.php/ICWSM/article/view/1455 0

[2] Amazon Web Services. (n.d.). *Amazon Comprehend – Natural Language Processing (NLP) Service*. Retrieved from https://aws.amazon.com/comprehend/

[3] Reddy, Y.A., Agarwal, S., Parashar, V., & Arora, A. (2025). *Real-Time Sentiment Insights from X Using VADER, DistilBERT, and Web-Scraped Data*. arXiv preprint arXiv:2504.15448. https://arxiv.org/abs/2504.15448

[4] Barrak, A., & Ksontini, E. (2025). *Scalable and Cost-Efficient ML Inference: Parallel Batch Processing with Serverless Functions*. arXiv preprint arXiv:2502.12017. https://arxiv.org/abs/2502.12017

[5] ResearchGate. (2024). *Sentiment Analysis Using Amazon Web Services and Microsoft Azure*. Retrieved from https://www.researchgate.net/publication/386026575

[6] KTH DiVA Portal. (2023). *Evaluating the Accuracy of Sentiment Analysis Models When Applied to Social Media Data*. Retrieved from https://kth.diva-portal.org/smash/get/diva2%3A1890072/FULLTEXT02.pdf

[7] Rajasekar, S., & Saminathan, S. (2023). *Analysis of Customer Reviews with an Improved VADER Lexicon*. *Journal of Big Data,* 10(1), 1-18. https://journalofbigdata.springeropen.com/articles/10.11 86/s40537-023-00861-x

[8] Python Software Foundation. (n.d.). *Boto3 Documentation*. Retrieved from https://boto3.amazonaws.com/v1/documentation/api/late st/index.html

[9] Streamlit Inc. (n.d.). *Streamlit — Turn Data Scripts into Shareable Web Apps*. Retrieved from https://streamlit.io

[10] Amazon Web Services. (n.d.). *Amazon QuickSight*. Retrieved from https://aws.amazon.com/quicksight/