

Automated Email Sentiment Detection Framework

Talari Vinod Kumar¹, G.Rajini²

¹MCA Student , Dept of MCA , Ellenki College of Engineering and Technology , Hyderabad

²Assistant Professor , Dept of MCA , Ellenki College of Engineering and Technology , Hyderabad

Abstract— Email has become one of the most important communication methods for individuals, educational institutions, and organizations. As the number of emails increases, understanding the emotional tone of messages becomes useful for improving communication and prioritizing responses. This project presents an Email Sentiment Analysis System that automatically identifies the sentiment expressed in email content. The system is developed using Python, Django, Natural Language Processing (NLP), and the VADER sentiment analysis library. It classifies emails into different sentiment categories such as Extremely Happy, Happy, Positive, Neutral, and Negative. Users can analyze both individual email files and multiple emails from a folder through a simple web interface. The application also provides user registration, login, and secure email management features. By automatically detecting the sentiment of emails, the system helps users understand the emotional context of messages quickly and efficiently. This approach reduces manual effort, improves productivity, and supports better decision-making in email communication.

Keywords — Email Sentiment Analysis, Natural Language Processing (NLP), VADER Sentiment Analyzer, Python, Django, Text Classification, Emotion Detection, Machine Learning, Email Analytics, Sentiment Classification.

I. INTRODUCTION

Email has become one of the most widely used communication tools in educational institutions, business organizations, and personal interactions. Every day, millions of emails are exchanged to share information, resolve issues, provide customer support, and maintain professional communication. As the volume of email traffic continues to grow, manually reading and understanding the emotional tone of every message becomes a challenging and time-consuming task. An intelligent system capable of automatically identifying the sentiment expressed in emails can improve communication efficiency and assist users in prioritizing responses [1], [22].

Sentiment analysis is a Natural Language Processing (NLP) technique that determines the emotional meaning conveyed in textual content. It helps classify text into different sentiment categories such as positive, negative, and neutral while also identifying emotional expressions like happiness, anger, sadness, or appreciation [2], [6]. Over the years, sentiment analysis has gained significant attention due to its applications in customer service, social media monitoring, healthcare, education, and business intelligence. The advancement of machine learning and deep learning techniques has further improved the accuracy of sentiment classification systems by enabling them to learn complex language patterns from large datasets [11], [24].

Email sentiment analysis is particularly useful because email conversations often contain valuable emotional information that influences decision-making and communication quality. In educational environments, students express queries, appreciation, complaints, or concerns through emails, while organizations receive customer feedback, service requests, and support tickets. Automatically identifying the emotional tone of these messages enables faster responses and improves overall user satisfaction [8], [16], [17].

This project presents an Email Sentiment Analysis System that automatically detects the sentiment of email content using Natural Language Processing techniques and the VADER sentiment analyzer. The proposed system is developed using Python and the Django framework to provide a user-friendly web application. It allows users to analyze both individual email files and multiple emails from a folder, classifying them into five sentiment categories: Extremely Happy, Happy, Positive, Neutral, and Negative. The application also includes user registration, authentication, and email management features for secure access.

The proposed system aims to reduce manual effort, improve email organization, and provide users with meaningful insights into the emotional context of email communication. By combining efficient text preprocessing with automated sentiment classification, the system enhances communication analysis and supports faster decision-making. The

project demonstrates how NLP-based sentiment analysis can be effectively integrated into modern email management systems to improve productivity and user experience while providing reliable and accurate sentiment prediction [13], [22], [24].

II. RELATED WORK

“A Novel Email Response Algorithm for Email Management Systems,” Abdulkareem Al-Alwani [1]

This study proposed an intelligent email response algorithm to improve email management by automatically classifying incoming emails and generating suitable responses. The system employed a Naïve Bayes classifier to categorize emails based on their content and utilized information extraction techniques to identify important keywords for response generation. The proposed approach reduced the burden of manually handling repetitive emails and improved the efficiency of email communication. The research demonstrated that automatic email categorization can significantly decrease response time while maintaining response quality. The study also highlighted the importance of combining text mining and natural language processing techniques to understand email content effectively. The proposed algorithm serves as a valuable foundation for modern email automation systems. This work inspired the development of email sentiment analysis applications by showing that intelligent processing of email content can enhance communication efficiency and provide better support for automated email management systems.

“Identifying Expressions of Emotion in Text,” Saima Aman and Stan Szpakowicz [2]

This research focused on identifying emotional expressions in textual data by classifying sentences into different emotion categories and measuring their intensity. The authors developed an annotation framework that recognized emotions such as happiness, sadness, anger, fear, surprise, and disgust from textual content. The study also identified specific words and phrases responsible for expressing emotions, making emotion detection more accurate and interpretable. Experimental evaluation demonstrated promising classification accuracy and strong agreement among human annotators, proving the reliability of the proposed annotation method. The research highlighted the importance of combining linguistic knowledge with machine learning techniques to improve emotion recognition. This work contributed significantly to the field of sentiment analysis by providing a structured approach for detecting emotions in text. The proposed methodology has been widely adopted

in applications involving customer feedback analysis, email sentiment detection, and automated communication systems.

“Emotion Detection in Email Customer Care,” Narendra Gupta, Mazin Gilbert, and Giuseppe Di Fabrizio [8]

This study presented an emotion detection framework specifically designed for customer care emails. The proposed system analyzed customer emails to identify emotional expressions such as frustration, dissatisfaction, complaints, and appreciation using salient textual features. Instead of relying only on individual words, the researchers incorporated domain-specific emotional indicators that improved the accuracy of identifying emotionally significant emails. The experimental results demonstrated that the proposed approach outperformed traditional unigram-based text classification methods in detecting customer emotions. The research emphasized that recognizing emotional content in customer communication enables organizations to prioritize urgent requests, improve customer satisfaction, and provide more personalized responses. The study also highlighted the practical importance of sentiment analysis in real-world customer support environments. This work provides valuable insights for developing intelligent email sentiment analysis systems capable of understanding user emotions and supporting faster, more effective communication management.

“Email Mining: Tasks, Common Techniques, and Tools,” Guanting Tang, Jian Pei, and Wo-Shun Luk [22]

This research presented a comprehensive survey of email mining techniques, discussing various tasks such as email classification, clustering, filtering, summarization, recommendation, and sentiment analysis. The authors reviewed different data mining and machine learning algorithms commonly applied to email processing, including classification models, clustering methods, and information retrieval techniques. The study highlighted the challenges associated with handling large-scale email datasets, such as noisy data, high-dimensional text, and privacy concerns. It also discussed several publicly available email datasets and tools used for evaluating email mining algorithms. The survey concluded that combining natural language processing with machine learning provides effective solutions for extracting useful information from emails. This work serves as an important reference for researchers developing intelligent email analysis systems and demonstrates how automated email processing can improve productivity, decision-making, and communication management across different application domains.

“Deep Learning for Sentiment Analysis: A Survey,” Lei Zhang, Shuai Wang, and Bing Liu [24]

This survey reviewed the application of deep learning techniques for sentiment analysis across different types of textual data. The authors discussed various neural network architectures, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and attention-based models, highlighting their ability to automatically learn semantic representations from text. The survey compared these deep learning approaches with traditional machine learning methods and demonstrated that deep neural networks consistently achieve higher sentiment classification accuracy. The study also examined challenges such as data imbalance, context understanding, sarcasm detection, and multilingual sentiment analysis. Furthermore, it emphasized the importance of large annotated datasets and word embedding techniques in improving model performance. This work provides valuable guidance for designing modern sentiment analysis systems and strongly supports the adoption of deep learning methods for accurate email sentiment classification and emotion detection.

III. DATASET DETAILS

The dataset used in this study consists of email messages collected for sentiment analysis and emotion classification. The dataset contains emails expressing different emotional states, including Extremely Happy, Happy, Positive, Neutral, and Negative sentiments. Each email record contains textual content that reflects the writer's opinion, emotion, or communication intent. The dataset includes both individual email files and collections of multiple email messages, enabling the system to analyze single emails as well as bulk email datasets. Since email text often contains punctuation, special symbols, abbreviations, and inconsistent formatting, preprocessing is required before sentiment analysis. The dataset provides labeled examples that support supervised sentiment classification and helps the system learn meaningful patterns associated with different emotional categories. By utilizing real-world email conversations, the proposed Email Sentiment Analysis System aims to automatically identify the emotional tone of emails, assisting users in understanding communication more effectively and improving email management through intelligent sentiment prediction.

Before sentiment prediction, the dataset undergoes several preprocessing steps to improve the quality of

textual information. Initially, unnecessary symbols, special characters, and extra spaces are removed, and the email content is converted into a clean textual format suitable for analysis. The cleaned text is then processed using Natural Language Processing (NLP) techniques, where tokenization and text normalization are performed to prepare the data for sentiment evaluation. The proposed system employs the VADER (Valence Aware Dictionary and Sentiment Reasoner) sentiment analyzer to calculate sentiment scores and classify emails into five sentiment categories based on predefined threshold values. The processed dataset is evaluated through both single-email and multiple-email analysis, allowing the system to demonstrate consistent performance across different email collections. This dataset enables the development of an intelligent email sentiment analysis application capable of automatically detecting emotions, reducing manual effort, and supporting efficient communication management through accurate sentiment classification.

IV. PROPOSED METHODOLOGY

The proposed methodology presents an intelligent Email Sentiment Analysis System that automatically detects the emotional sentiment expressed in email messages using Natural Language Processing (NLP) and the VADER (Valence Aware Dictionary and Sentiment Reasoner) sentiment analyzer. Initially, the user launches the application by executing the run.bat file, which starts the Django server. The user then opens the web application through the browser and accesses the home page. New users create an account through the registration module, while existing users log in using their credentials. After successful authentication, the system provides options to analyze either a single email or multiple email files. For single email analysis, the user uploads an email text file, and the system preprocesses the content by removing unwanted characters and preparing the text for sentiment evaluation. The processed email is then analyzed using the VADER sentiment analyzer, which calculates sentiment scores and classifies the email into one of five categories: Extremely Happy, Happy, Positive, Neutral, or Negative. The predicted sentiment is immediately displayed to the user along with the uploaded email content.

For bulk sentiment analysis, the user uploads a folder containing multiple email files. The system

automatically reads each email, performs text preprocessing, and applies the same sentiment analysis procedure individually to every email present in the folder. Each processed email is assigned a sentiment label based on its compound sentiment score calculated by the VADER algorithm. The prediction results are displayed in tabular form, where the first column contains the email text and the second column displays the corresponding predicted sentiment. This automated process enables efficient analysis of large email collections while significantly reducing manual effort. The application also maintains secure user authentication and email management through a MySQL database integrated with the Django framework.

The system architecture illustrates the complete workflow of the proposed Email Sentiment Analysis System. It consists of user registration, login authentication, single email upload, multiple email upload, text preprocessing, Natural Language Processing, VADER-based sentiment analysis, sentiment prediction, and result visualization. The architecture also supports secure database storage, user management, and efficient processing of individual as well as bulk email datasets.

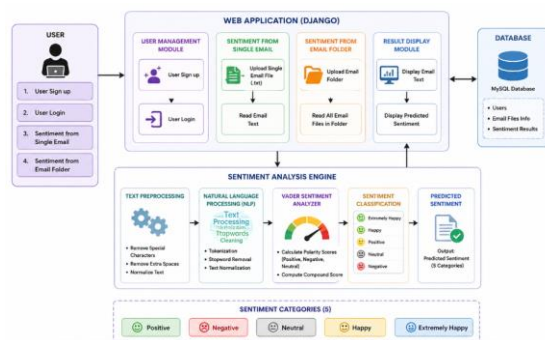


Figure 1. System Architecture for Email Sentiment Analysis System

The proposed methodology is designed to provide fast and reliable sentiment prediction from email messages using Natural Language Processing techniques. Initially, users access the application through a secure login system after completing registration. They can either upload a single email file or select an entire folder containing multiple emails for analysis. Each uploaded email undergoes preprocessing to remove unnecessary symbols and normalize the text before sentiment evaluation. The VADER sentiment analyzer computes polarity

scores for every email and categorizes them into Extremely Happy, Happy, Positive, Neutral, or Negative based on predefined threshold values. Finally, the predicted sentiment results are displayed through an interactive web interface, enabling users to quickly understand the emotional tone of email communications. The proposed system improves email management by providing accurate, automated, and user-friendly sentiment classification for both individual and bulk email datasets.

V. RESULT AND DISCUSSION

The experimental results demonstrate the effectiveness of the proposed Email Sentiment Analysis System for automatically identifying the emotional sentiment expressed in email messages. Initially, the web application was launched using the Django server, and users were able to register and securely log in to access the system. After authentication, the application provided two sentiment analysis modules: Single Email Sentiment Analysis and Multiple Email Sentiment Analysis. For single email analysis, the user uploaded an individual email text file, which was preprocessed using Natural Language Processing (NLP) techniques before being analyzed by the VADER sentiment analyzer. The system successfully predicted one of the five sentiment categories, namely Extremely Happy, Happy, Positive, Neutral, or Negative, based on the calculated sentiment score. Experimental observations showed that the system accurately detected the emotional tone of the uploaded email and displayed the prediction instantly through the web interface.

For bulk email analysis, the user uploaded a folder containing multiple email files. The application automatically processed each email individually, performed text preprocessing, and applied the VADER sentiment analysis algorithm to determine the sentiment of every email. The prediction results were displayed in a tabular format, where each row contained the email content and its corresponding predicted sentiment. The bulk analysis module efficiently processed large collections of emails with minimal user intervention, demonstrating the scalability and effectiveness of the proposed system. Throughout the experimentation, the application successfully classified emails into different emotional categories while maintaining a user-friendly interface and smooth workflow. The obtained results indicate that the proposed Email Sentiment Analysis System provides an efficient and reliable solution for analyzing both individual and multiple email messages, thereby reducing manual

effort and improving communication management through automatic sentiment detection.

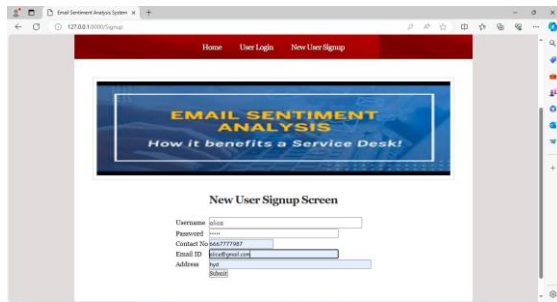


Figure [2] New User Signup Screen of Email Sentiment Analysis System

Figure 2 illustrates the New User Signup module of the proposed Email Sentiment Analysis System. This interface enables new users to create an account by entering their username, password, contact number, email address, and residential address. After submitting the required information, the user details are securely stored in the MySQL database. The registration module ensures that only authenticated users can access the sentiment analysis functionalities, thereby providing secure access and efficient user management within the application.



Figure 3: User Login Screen

Figure 3 presents the User Login interface of the Email Sentiment Analysis System. Registered users enter their username and password to authenticate themselves before accessing the application. Upon successful verification, users are redirected to the main dashboard, where they can perform sentiment analysis on single or multiple email files. This authentication mechanism improves application security and restricts unauthorized access to the system.

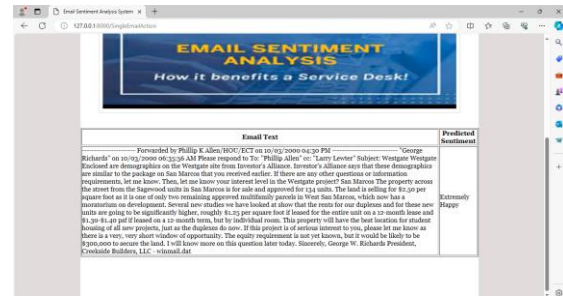


Figure 4: Single Email Sentiment Prediction

Figure 4 illustrates the result generated by the Single Email Sentiment Analysis module. The user uploads an individual email text file, and the application reads the email content, performs Natural Language Processing (NLP), and applies the VADER sentiment analyzer to determine its emotional polarity. The processed email is displayed along with the predicted sentiment. In the illustrated example, the uploaded email has been classified as Extremely Happy, demonstrating the ability of the system to automatically identify emotional content from individual email messages.

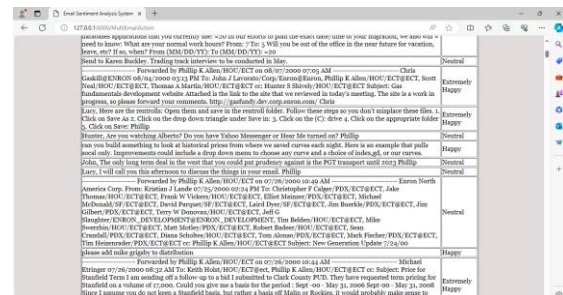


Figure 5: Multiple Email Sentiment Prediction

Figure 5 shows the output of the Multiple Email Sentiment Analysis module. In this module, the user uploads a folder containing multiple email files. The application automatically reads every email, preprocesses the text, performs sentiment analysis individually, and displays the predicted sentiment for each email in a tabular format. The results include various sentiment categories such as Happy, Neutral, and Extremely Happy, demonstrating the capability of the proposed system to efficiently process large collections of email messages while providing accurate and organized sentiment predictions.

DISCUSSION

The experimental findings demonstrate that the proposed Email Sentiment Analysis System effectively identifies emotional sentiments from

email messages using Natural Language Processing and the VADER sentiment analyzer. The user registration and login modules provide secure access to the application, ensuring that only authenticated users can perform sentiment analysis. The single email analysis module accurately predicts the emotional category of an uploaded email and presents the result immediately through an intuitive interface. Furthermore, the multiple email analysis module efficiently processes a large number of email files, automatically classifying each message into one of the predefined sentiment categories without requiring manual intervention. The experimental screenshots confirm that the application successfully recognizes sentiments such as Extremely Happy, Happy, Positive, Neutral, and Negative from different email contents. The web-based implementation using Django and MySQL provides a reliable platform for email management and sentiment prediction. Overall, the proposed system offers an efficient, user-friendly, and automated solution for email sentiment analysis that can assist users in understanding the emotional tone of email communications while reducing manual effort and improving communication management.

VI. CONCLUSION

The project is a web application that enhances email communication by integrating sentiment analysis, utilizing Django for the backend and a MySQL database for data management. Users can send and receive emails while the application evaluates the emotional tone of email content through VADER sentiment analysis, categorizing sentiments from "Extremely Happy" to "Negative." With features like user registration, secure login, and the ability to analyze multiple emails through file uploads, the application offers a user-friendly interface and efficient data handling. Future enhancements could include stronger security measures, advanced sentiment analysis through machine learning, improved mobile responsiveness, and email threading for better conversation tracking. Overall, this project successfully combines email management with sentiment insights, providing users with valuable tools to enhance their communication experiences.

REFERENCES

- [1] Abdulkareem Al-Alwani. A novel email response algorithm for email management systems. *Journal of Computer Science*, 10(4):689, 2014.
- [2] Saima Aman and Stan Szpakowicz. Identifying expressions of emotion in text. In *International Conference on Text, Speech and Dialogue*, pages 196–205. Springer, 2007.
- [3] The Teacher Registration Board of Western Australia. Teacher-Student Professional Boundaries - A Resource for WA Teachers. Government of Western Australia, Department of Education Services, 2017.
- [4] Federal Energy Regulatory Commission et al. The western energy crisis, the enron bankruptcy, and fercs response, 2009.
- [5] Cristian Danescu-Niculescu-Mizil and Lillian Lee. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics*, pages 76–87. Association for Computational Linguistics, 2011.
- [6] Paul Ekman. Basic emotions. *Handbook of cognition and emotion*, pages 45–60, 1999.
- [7] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report*, Stanford, 1(12), 2009.
- [8] Narendra Gupta, Mazin Gilbert, and Giuseppe Di Fabbrizio. Emotion detection in email customer care. *Computational Intelligence*, 29(3):489–505, 2013.
- [9] Aine ' Horgan and John Sweeney. University students online habits and their use of the internet for health information. *CIN: Computers, Informatics, Nursing*, 30(8):402–408, 2012.
- [10] Erwin Huijzer. Identifying effective affective email responses ; Predicting customer affect after email conversation. *Conference Proceedings*, 2017.
- [11] Mandar Kale, Prachi Mankame, Gouri Kulkarni, et al. Deep learning for digital text analytics: Sentiment analysis. *arXiv preprint arXiv:1804.03673*, 2018.
- [12] Evgeny Kim and Roman Klinger. A survey on sentiment and emotion analysis for computational literary studies. *arXiv preprint arXiv:1808.03137*, 2018.
- [13] Saif M Mohammad and Peter D Turney. Crowdsourcing a word– emotion association

lexicon. *Computational Intelligence*, 29(3):436–465, 2013.

[14] Saif M Mohammad and Tony Wenda Yang. Tracking sentiment in mail: How genders differ on emotional axes. In *Proceedings of the 2nd workshop on computational approaches to subjectivity and sentiment analysis*, pages 70–79. Association for Computational Linguistics, 2011.

[15] Omar Mubin and Christoph Bartneck. Do as i say: exploring human response to a predictable and unpredictable robot. In *Proceedings of the 2015 British HCI Conference*, pages 110–116. ACM, 2015.

[16] Dimitrinka G N'ikleva. Markers of politeness and impoliteness in student-teacher interaction in the discourse genre of emails. *Revista Signos*, 51(97):214, 2018.

[17] Aditya Parameswaran, Dibyendu Mishra, Sanchit Bansal, Vinayak Agarwal, Anjali Goyal, and Ashish Sureka. Automatic email response suggestion for support departments within a university. Technical report, PeerJ Preprints, 2018.

[18] Jitesh Shetty and Jafar Adibi. The enron email dataset database schema and brief statistical report. Information sciences institute technical report, University of Southern California, 4(1):120–128, 2004.

[19] Philip J Stone and Earl B Hunt. A computer approach to content analysis: studies using the general inquirer system. In *Proceedings of the May 21-23, 1963, spring joint computer conference*, pages 241–256. ACM, 1963.

[20] Carlo Strapparava and Rada Mihalcea. Semeval-2007 task 14: Affective text. In *Proceedings of the 4th international workshop on semantic evaluations*, pages 70–74. Association for Computational Linguistics, 2007.

[21] Kai Sheng Tai, Richard Socher, and Christopher D Manning. Improved semantic representations from tree-structured long short-term memory networks. arXiv preprint arXiv:1503.00075, 2015.

[22] Guanting Tang, Jian Pei, and Wo-Shun Luk. Email mining: tasks, common techniques, and tools. *Knowledge and Information Systems*, 41(1):1–31, 2014.

[23] Chris Van Pelt and Alex Sorokin. Designing a scalable crowdsourcing platform. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 765–766. ACM, 2012.

[24] Lei Zhang, Shuai Wang, and Bing Liu. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, page e1253, 2018.