Enhancing Email Communication with AI: Mitigating Misinterpretation through Sentiment and Tone Analysis

Riley Grimaud
Department of Computer Science
Tennessee Tech University
Cookeville, TN 38505
rggrimaud42@tntech.edu

Tania Perdomo Flores
Department of Computer Science
Tennessee Tech University
Cookeville, TN 38505
teperdomof42@tntech.edu

Logan Bolton
Department of Computer Science
Tennessee Tech University
Cookeville, TN 38505
lhbolton42@tntech.edu

Sharon Colson
Department of Computer Science
Tennessee Tech University
Cookeville, TN 38505
scolson21@tntech.edu

***Abstract*—Miscommunication in email exchanges often arises from the absence of non-verbal cues like tone and body language, leading to misunderstandings and unintended negativity. This project addresses these challenges by developing an AI-powered tool that evaluates email tone, sentiment, grammar, and readability using advanced Natural Language Processing (NLP) techniques. Key features include sentiment analysis via VADER, emotional tone detection with Hugging Face models, and grammar corrections through LanguageTool. The system provides real-time feedback on ambiguous phrases, hedging, and unintended negative tones, offering actionable suggestions for improvement. Evaluated through metrics like error correction rates and readability scores, this solution enhances email clarity, emotional appropriateness, and professionalism, reducing the likelihood of misinterpretation in digital communication.**

***Index Terms*—Email Communication, Natural Language Processing, Emotion Analysis, Sentiment, Tone Analysis**

I. Introduction

Email is a vital communication tool, yet its lack of non-verbal cues, such as tone and body language, often leads to misinterpretation and unintended negativity. This issue is exacerbated by the sender's mood, language proficiency, and word choice, creating emails that may appear blunt, ambiguous, or unprofessional. Miscommunication can damage relationships and productivity, particularly in professional settings, where tone and clarity are crucial.

To address this challenge, we propose an AI-powered solution that leverages Natural Language Processing (NLP) to analyze and improve email content. Key features include sentiment and emotional tone analysis, grammar and readability evaluation, and real-time suggestions for improvement. By providing actionable feedback on tone, sentiment, and clarity, the system empowers users to craft more professional, empathetic, and effective emails.

While challenges like detecting sarcasm or cultural nuance remain, this tool offers a promising step toward mitigating misunderstandings and enhancing digital communication.

II. Background Motivation

Problem Statement: Email communication often suffers from misinterpretation due to the lack of non-verbal cues like tone of voice and body language. This can lead to misunderstandings, especially when the sender's mood, stress, or language skills influence how their message is perceived. Despite advances in Natural Language Processing (NLP), no tool currently addresses emotional clarity and tone in emails, leaving room for improvement.

Objective: The goal of this project is to develop an AI-powered tool that analyzes emails in real-time, providing feedback on sentiment, tone, and potential ambiguities. It will help users write clearer, more emotionally appropriate emails, reducing miscommunication and improving the overall effectiveness of email exchanges.

Significance: Improving email communication can enhance interactions in business, education, and personal relationships. By using AI to clarify tone and sentiment, this tool can reduce misunderstandings, foster better relationships, and increase productivity, making email exchanges more effective and empathetic.

III. Overview of the Solution

The proposed solution is a Python-based tool designed to assist users in evaluating and enhancing their email content. The tool analyzes emails based on various linguistic criteria and provides actionable recommendations for improvement.

A. *Key Features*

Grammar and Spelling Correction: This feature detects spelling and grammatical errors and offers suggestions on corrections, ensuring that the email maintains professionalism while leaving the ultimate decision in the hands of the user.

Clarity and Readability Check: This feature assesses the complexity of the email's language, evaluating factors such as sentence length, vocabulary, and structure. It offers a quantitative and comparative analysis of the clarity and readability of the email in both its original and corrected state. It offers assessments based on Flesch Reading Ease, Flesch-Kincaid Grade Level, Dale-Chall Readability Score, and reading time analysis.

Sentiment Analysis: This feature evaluates the overall emotional tone of the email, categorizing it as positive, neutral, or negative. By analyzing the word choice and context, the program provides an assessment of whether the message conveys a supportive, neutral, or critical tone. This helps the sender adjust their language to avoid unintended negativity and ensure the message aligns with the desired sentiment.

Emotion Detection: This component identifies underlying emotions, such as joy, sadness, fear and worry, within the email content. By detecting emotional cues in the language, the program guides the sender on how to adjust their tone. For instance, if an email conveys unintended anger, the program may suggest changes to make the tone more neutral or empathetic, reducing the risk of misinterpretation.

B. *Tools and Libraries*

nltk (Natural Language Toolkit): Used for sentiment analysis and tokenization, NLTK is a comprehensive library that allows for parsing and analyzing text. For this project, the **VADER (Valence Aware Dictionary and sEntiment Reasoner** was implemented for sentiment analysis through the 'SentimentIntensityAnalyzer' module. This module was used to measure the polarity of text at the sentence level, providing a fast and reliable method for evaluating the sentiment of emails without requiring deep learning models.

language-tool-python: This tool integrates the LanguageTool grammar checking program, providing an automated way to detect grammar and spelling mistakes. It enhances the professionalism of emails by identifying common errors in syntax, punctuation, and spelling that could negatively impact the clarity or tone of the message.

transformers: A powerful library for emotion detection and summarization, transformers enable the use of pre-trained models for understanding emotional context in text. By leveraging models like BERT, the program can detect nuanced emotions such as sarcasm, frustration, or empathy, offering suggestions to refine the emotional tone of emails.

spaCy: A popular library for advanced natural language processing, spaCy is used for tasks such as named entity recognition, syntactic parsing, and dependency analysis. It enhances the program's ability to understand the structure and meaning of sentences, allowing for more sophisticated emotion detection and readability analysis.

textstat: This tool calculates various readability statistics, such as the Flesch-Kincaid readability score, sentence complexity, and word length. It provides insights into how complex or accessible the text is, allowing users to simplify their emails for a broader audience, ensuring effective communication.

matplotlib: Used to create a wide range of static, animated, and interactive visualizations in Python. It helps users generate detailed graphs and charts, such as line plots, scatter plots, and histograms, to better understand data trends and patterns, making it an essential tool for data analysis and presentation.

torch: This library is used for building and training neural networks. It provides flexible tools for deep learning tasks, such as computer vision, natural language processing, and reinforcement learning, with support for dynamic computation graphs and GPU acceleration, making it highly efficient for research and production applications.

These tools and libraries combine to provide a comprehensive solution for analyzing and improving email communication, helping users craft messages that are clear, emotionally balanced, grammatically correct, and professional.

IV. Methodology and Implementation

The program provides a comprehensive analysis of email content, offering real-time feedback on tone, sentiment, readability, grammar, and emotional nuance. The process is structured as follows:

Input Analysis: - Accepts email content directly from the user interface or predefined variables in the code. - Ensures seamless integration with user workflows for easy testing and iteration.

Preprocessing: - Cleans the email text by removing unnecessary spaces, line breaks, and irrelevant formatting. - Standardizes the content for consistent and accurate downstream analysis.

Sentiment and Emotion Analysis: - Utilizes models like NLTK's SentimentIntensityAnalyzer to classify the email's sentiment (positive, neutral, negative). - Leverages transformers-based emotion detection models to detect nuanced emotions (e.g., joy, anger, sadness) and assess the overall emotional tone of the email. - Provides granular feedback for tone adjustment, such as addressing fear, anger, or confusion.

Readability Evaluation: - Computes readability scores using tools like textstat, focusing on metrics such as the Flesch Reading Ease. - Highlights areas of improvement for complex or unclear sentences, encouraging the creation of accessible and professional emails.

Grammar and Tone Adjustments: - Uses language-tool-python to detect grammar, spelling, and stylistic issues. - Offers actionable corrections for problematic phrases, including hedging words, contradictions, and passive-aggressive language, ensuring clarity and professionalism.

Summarization and Suggestions: - Implements transformers-based summarization models to generate

concise summaries of the email's main points. - Combines analysis results to provide tailored suggestions for improving tone, structure, and emotional balance. - Generates a rewritten version of the email to demonstrate effective communication practices.

A. *Experimented Approaches*

Several approaches were tested but discarded due to limitations:

Large Language Models (LLMs): Discarded due to high computational demands, with transformers offering similar performance.

Rule Based Tokenization: The transformer architectures provided a more comprehensize solution.

Basic Sentiment Analysis: Too simplistic, failed to capture nuanced tone.

Emoji-Based Emotion Evaluation: Discarded due to compatibility and limited depth.

BART: Struggled with context sensitivity, misinterpreting nuanced emotions.

Website Extension: Too resource-intensive for the project scope.

B. *Technical Implementation*

The tool employs advanced natural language processing (NLP) techniques to enhance email content by addressing spelling, grammar, readability, sentiment, and emotional tone. Key components include:

Spelling and Grammar Correction: Utilizes the `language-tool-python` library to automatically detect and correct grammatical errors, spelling mistakes, and stylistic issues. The interactive correction process ensures that users can tailor their content with precision.

Readability Analysis: Implements the `textstat` library to calculate readability metrics such as the Flesch Reading Ease score. Suggestions for simplifying complex sentences are visualized with bar charts created using `matplotlib`.

Sentiment Analysis: Incorporates the VADER (Valence Aware Dictionary and sEntiment Reasoner) tool to classify the tone of the email as positive, neutral, or negative. This provides immediate insights into how the email may be perceived by recipients.

Emotion Detection: Leverages Hugging Face's `DistilRoBERTa` and `RoBERTa-large-mnli` models to classify nuanced emotions (e.g., joy, anger, sadness) and detect hedging or contradictory statements. This ensures that the emotional tone aligns with the sender's intent.

Interactive Feedback and Suggestions: Flags problematic phrases, such as hedging or passive-aggressiveness, and provides actionable options for improvement. Users can iteratively modify their email to ensure it is professional, clear, and emotionally balanced.

V. Performance Evaluation

The tool's performance was evaluated through visualizations and empirical testing, focusing on sentiment analysis, emotional tone detection, readability, and grammar correction.

Visualization: Graphs showed sentiment distribution and emotional tone frequency across email samples, making trends easy to spot and comparisons clear.

Empirical Testing: Emails with defined tones (positive, negative, neutral) were processed to test sentiment and emotional tone detection. While sentiment analysis was accurate, detecting nuanced emotions, such as sarcasm or mixed feelings, was more challenging due to the complexity of human emotion and context.

Sentiment vs. Emotion Detection: Sentiment analysis was strong, but detecting subtle emotions like passive-aggression or frustration proved difficult. This highlights the need for deeper contextual understanding in future versions.

VI. Evaluation Metrics

Accuracy: The tool was effective at detecting basic sentiment but struggled with complex emotions. Incorporating advanced models like GPT or BERT could improve emotion detection accuracy.

Readability and Grammar: Grammar and readability features performed well, with tools like language-tool-python effectively correcting errors and improving clarity.

VII. Results

Grammar and Spelling Correction: The tool integrates with LanguageTool to automatically detect and correct grammar, spelling, and style issues. Users can interactively review and apply corrections based on provided suggestions, ensuring flexibility and control. The system visualizes key insights through an error breakdown chart and a step-by-step resolution timeline.

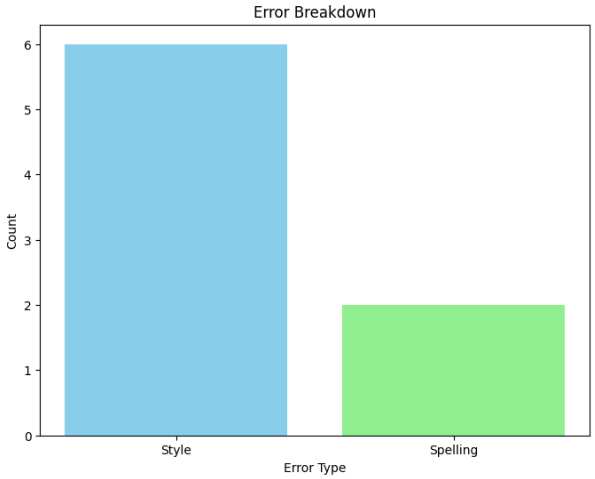

Fig. 1: Error Breakdown in Grammar and Spell Check

Readability Assessment: Utilizing TextStat, the tool analyzes email complexity with metrics like Flesch Reading Ease and Flesch-Kincaid Grade Level. It offers actionable suggestions for simplifying complex sentences and provides visual comparisons between the original and corrected text.

Sentiment Analysis: Powered by the VADER Sentiment Analyzer, the system evaluates the overall sentiment of the

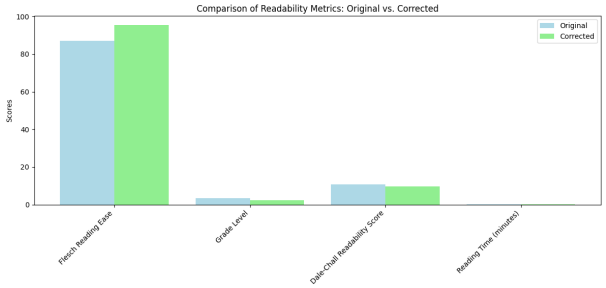

Fig. 2: Readability Scores Comparison between Original and Corrected Emails

email—classifying it as positive, neutral, or negative. Detailed sentiment scores help users understand the tone and adjust it as needed.

Emotion Detection: Hugging Face's DistilRoBERTa model identifies nuanced emotions such as joy, anger, and sadness. The tool flags problematic language, including hedging, contradictions, and passive-aggressive phrases, enabling users to refine their tone for better alignment with their communication goals.

Interactive Refinement: The system empowers users to iteratively refine their email by addressing flagged issues. Users can replace, remove, or retain content such as hedging phrases, contradictory statements, and problematic sentences, ensuring the email achieves clarity, professionalism, and emotional balance.

	Category	Details
0	Overall Emotion	sadness
1	Emotion Probabilities	{'sadness': 0.9981058835983276}
2	Hedging Phrases	None
3	Contradictions	I'm not saying its anyone's fault, but also, its kind of disappointing to see deadlines being missed over and over. <-> We agreed that the deadlines were manageable, didn't we?
4	Problematic Sentences	None
5	Suggestions	Avoid passive-aggressive phrases. Try to be more direct.

Fig. 3: Actionable Feedback for Improvement

VIII. Conclusion

The email refinement system offers a comprehensive solution for improving written communication by integrating grammar correction, readability enhancement, sentiment and emotion analysis, and interactive user refinement. This combination ensures that emails are free from linguistic errors, convey the intended emotional tone, and maintain clarity. The user-driven refinement process adds flexibility, allowing for context-sensitive adjustments that align with the sender's communication goals. Future improvements could focus on optimizing the system for longer emails, fine-tuning thresholds for different contexts, and expanding multilingual support.

While there are challenges in achieving perfect sentiment and emotion analysis, the system provides a strong foundation for automating email assessments and improving communication. By incorporating advanced NLP models and refining its features, the system's accuracy and usability can be further enhanced. Ultimately, the tool offers a powerful solution for enhancing email professionalism, emotional appropriateness, and readability.

IX. Future Work

The proposed email analysis tool has great potential for enhancing communication, but there are several key areas for improvement. Future work will focus on refining its functionality and expanding its usability:

Email Client Integration: The tool will be integrated with popular email clients like Outlook, Gmail, and Apple Mail, allowing users to analyze emails directly within their native platforms for a more seamless experience.

Tone and Style Customization: Users will be able to customize the tool's tone settings (formal, informal, friendly, professional), ensuring suggestions align with their preferred communication style and context.

Advanced Emotion Detection: Incorporating deep learning models like GPT or BERT will improve the tool's ability to detect subtle emotions, such as sarcasm or passive-aggressiveness, providing more accurate feedback.

Web Extension for Real-Time Feedback: A browser extension will offer real-time suggestions for tone, sentiment, and readability as users compose emails in web-based clients like Gmail. A prototype is available but needs further development to ensure full functionality.

In summary, these updates will expand the tool's capabilities, making it a more powerful solution for improving email communication across platforms.

Teammate Contributions

All authors named in this paper were equally involved in each aspect of the project research, production process, and technical writing for this project. Additional specific contributions include:

Logan Bolton: approach four in attempted implementations.
Riley Grimaud: Chrome Extention development.
Sharon Colson: final approach.
Tania Perdomo Flores: approaches 1-3 and 5 in attempted implementations.

Acknowledgment of Generative AI Usage

The authors acknowledge the use of ChatGPT, a language model developed by OpenAI, in the preparation of this assignment. ChatGPT assisted in generating the Python code within the Jupyter notebook, which was used to implement sentiment analysis, tone detection, and transformer models for analyzing email inputs.