Meeting Health Dashboard: Valence-Based Visual Analytics of Emotional Dynamics in Software Team Meetings

Varshith Bala
Anuradha Vikas Sinha Department of Data Science
University of North Texas
Denton, TX, USA
Email: varshithbala@my.unt.edu

***Abstract*—In the case of software development, meetings are still the primary channel for coordinating activities, but seldom do people go further than providing their opinions while assessing the quality of these meetings. An example of such research is the fusion of automatic transcription with sentiment analysis to produce an approximation of team mood during software project meetings. We propose a Meeting Health Dashboard that takes the lead in this research, as well as the reading report we wrote earlier. The Dashboard performs analysis on the preprocessed meeting transcripts by splitting them into chunks and generates interpretable emotion and collaboration indicators throughout the process. The application requires structured CSV files with chunk-level valence, preconfigured key moments, meeting-level metrics, and topic-health summaries and it can also calculate the meeting-level health, collaboration, and tension scores directly from chunk data in situations where those metrics are not available. The web-based dashboard provides an interactive exploration of these indicators, featuring emotional flow, critical sentiment shifts, and topic-wise health. We present the data pipeline, metric definitions, and implementation details of the Dashboard in this paper and connect them to the literature on sentiment analysis and interaction analysis in software engineering.**

***Index Terms*—sentiment analysis, software engineering, meetings, emotional dynamics, collaboration analytics, valence modeling**

I. Introduction

Software development is a collaborative process at its core, and project success relies equally on the quality of communication within the team and the technical decisions made during the process. Empirical research has demonstrated that developer affect and mood are interconnected with problem-solving capability and productivity, which leads to the need for automated methods to monitor the emotional factors in collaboration [1], [2].

Most existing sentiment analysis approaches in software engineering focus on textual artifacts such as issue comments, code reviews, and Q&A posts [3], [4]. However, a substantial portion of communication occurs in synchronous meetings—planning sessions, retrospectives, design discussions, and stakeholder reviews—where emotional dynamics may strongly influence outcomes but are rarely analyzed systematically.

Herrmann et al. The application of sentiment analysis to spoken communication in software project meetings has been proven by the integration of speech recognition with a sentiment classifier, which produces polarity distributions matching human evaluations of meeting mood [5]. Their research inspires the creation of tools capable not only of classifying sentiments but also of offering more sophisticated visual analytics of emotional flow and collaboration quality.

For the course project, we created a Meeting Health Dashboard that accepts preprocessed meeting data in CSV format as input and produces meeting-level metrics and visual summaries of emotional dynamics as output. This document describes that implementation for a single submission, based on the common team project repository and the reading report prepared earlier.

II. Background and Related Work

A. Sentiment Analysis in Software Engineering

Sentiment analysis has been implemented in the software engineering domain through domain-specific lexicons, machine learning models, and evaluation standards. Senti4SD and SentiCR are names of such tools specifically designed for the communication of developers, which outperform the generic sentiment analyzers by considering the particularities of the technical language. [3], [4]. Applications include detecting frustration, triaging issues, and assessing review quality [6].

B. Sentiment Analysis for Meetings

An architecture was suggested by Herrmann et al. that accepts audio from software project meetings, transcribes it, segments it into statements, and then uses a sentiment classifier to label each segment with a polarity label, thus demonstrating that the corresponding distributions closely represent the self-reported perceptions of meeting mood by the participants [5]. The authors have concentrated their research on the discrete polarity categories (positive, negative, neutral) and aggregate statistical measures.

The implementation we are using pursues the same basic objective of identifying the mood in meetings but it has two significant differences. To begin with, we deal with valence scores which are of continuous nature and are at the chunk level rather than just polarity labels. Moreover, we calculate interpretable metrics at the meeting level and arrange the results in CSV format for interactive visualization, thereby spotlighting the derivation of the metrics and the data structures used.

C. Meeting Analytics and Interaction Analysis

The evaluation of interaction in software project meetings frequently depends on the manually coded actions like supporting one another, criticizing, and naming problems, which are associated with the performance of the team [7], [8]. Despite the fact that these coding schemes offer detailed insights, their use involves significant manual labor. On the other hand, automated sentiment- and valence-based analyses provide a less sophisticated but more scalable alternative.

III. System Overview

A. Objectives

The goal of the Meeting Health Dashboard is to provide a compact yet expressive summary of how a meeting "felt" from an emotional and collaboration perspective. Concretely, the system aims to:

- estimate an overall health score that reflects average emotional tone;
- quantify collaboration and tension using the distribution and variability of valence;
- identify key moments where the emotional tone shifts sharply; and
- relate discussion topics to their associated emotional health.

B. Data Inputs and Directory Layout

The implementation operates on precomputed CSV files stored in a project directory with the following structure: a main script, a `requirements.txt` file, a `data/` folder containing four CSV files, and accompanying documentation files such as `README.md` and `installation_guide.md` [9]. Required data files are: `meeting_metrics_df.csv`, `chunk_df.csv`, `key_moments_df.csv`, and `topic_health.csv`.[1] :contentReference[oaicite:3]index=3

Each CSV file plays a specific role:

- **Chunk-level data (`chunk_df.csv`):** contains one row per transcript chunk with columns such as `meeting_id`, `chunk_idx`, `valence`, and `chunk_text`. :contentReference[oaicite:4]index=4
- **Meeting-level metrics (`meeting_metrics_df.csv`):** stores aggregated scores per meeting: `meeting_health_score`, `collaboration_score`, and `tension_score`.
- **Key moments (`key_moments_df.csv`):** lists chunk indices and valence deltas for segments with large sentiment shifts.
- **Topic health (`topic_health.csv`):** provides topic-wise summaries relating thematic categories to average health scores.

[1] File names in the code correspond to `meeting_metrics_df.csv`, `chunk_level_df.csv`, `key_moments_df.csv`, and `topic_health.csv`.

C. Main Processing Script

The central Python script defines helper functions for loading CSV files, computing meeting-level metrics from chunk data when needed, detecting key sentiment shifts, and supporting interactive exploration of selected meetings. The script verifies that essential columns are present in the chunk-level data (`meeting_id`, `chunk_idx`, `valence`) and stops with an error message if they are missing, ensuring data integrity. :contentReference[oaicite:5]index=5

IV. Meeting Health Metrics

A. Chunk-Level Valence

The chunk-level dataset associates each text segment i with a valence value $v_i \in [-1, 1]$ that encodes the emotional tone from negative to positive. Chunks are ordered by `chunk_idx` within each meeting, representing the temporal progression of the conversation.

B. Health Score

The health score of a meeting is defined as a linear mapping of the mean valence to a 0–100 scale. For a given meeting m with chunk set C_m and valence values v_i, the health score H_m is computed as:

$$H_m = \frac{\overline{v}_m + 1}{2} \times 100,$$

where $\overline{v}_m$ is the mean valence of chunks in C_m. This mapping preserves the ordering of meetings by average valence while providing an interpretable range similar to a percentage scale. The corresponding implementation groups the chunk data by `meeting_id`, computes the mean valence, and applies the transformation. :contentReference[oaicite:6]index=6

C. Collaboration Score

Collaboration is modeled as the inverse variability of valence within a meeting. Intuitively, meetings with highly fluctuating emotional tone may indicate unstable collaboration, whereas consistent moderately positive tone corresponds to smoother interaction.

Let s_m be the standard deviation of valence values within meeting m, and let $s_{\max}$ be the maximum standard deviation across all meetings. The collaboration score C_m is defined as:

$$C_m = \left(1 - \frac{s_m}{s_{\max}}\right) \times 100,$$

clipped to the range $[0, 100]$. In the implementation, this normalization is performed using the maximum standard deviation across groups, and the resulting scores are scaled into percentages. :contentReference[oaicite:7]index=7

D. Tension Score

Tension is captured as the proportion of chunks with clearly negative valence. For each meeting, a binary indicator flags chunks with $v_i < -0.2$, and the tension score T_m is computed as the percentage of such chunks:

$$T_m = \frac{\#\{i \in C_m : v_i < -0.2\}}{|C_m|} \times 100.$$

This metric highlights meetings that contain extended segments of negative tone. The corresponding computation in the script uses a boolean column `is_neg` and averages it per meeting. :contentReference[oaicite:8]index=8

E. Key Sentiment Shift Moments

Key moments are identified by examining changes in valence between consecutive chunks within a meeting. For each chunk i with valence v_i and previous chunk $i-1$ with valence v_{i-1}, the delta is:

$$\Delta v_i = v_i - v_{i-1}.$$

The function filters for chunks where $|\Delta v_i|$ exceeds a configurable threshold (default 0.4) and returns a small set of the largest shifts, along with the associated `chunk_idx` and `chunk_text` [10]. These points often correspond to important conversational events such as disagreements, resolutions, or topic changes.

F. Topic–Health Relationship

The `topic_health.csv` file shows the health of the different topics at a higher level, and it does so by summing up the health scores given to each topic in all the meetings. For a certain topic k the health value H_k corresponds to either the average or the weighted average of the meeting health scores for the parts linked to that topic. Although topic modeling is not a part of the analysis script, the Dashboard can still show these precomputed associations as long as the file is provided.

V. Implementation Details

A. CSV Loading and Cleaning

A cached helper function loads CSV files into data frames and removes extraneous indexing columns such as `"Unnamed: 0"` when present, ensuring clean input for analysis. :contentReference[oaicite:9]index=9

B. Metric Derivation Logic

In the absence of the meeting-related metrics file, the system by itself measures health, collaboration, and tension from the chunk-level data according to the aforementioned formulas. The user is shown an information message that clearly states this action, allowing the user to know whether the scores are precomputed or derived.:contentReference[oaicite:10]index=10

C. Meeting Selection and Views

This functionality enables the user to pick out a meeting using its ID and afterwards brings back the rows from the meeting metrics and chunk-level data frames that are related to it. It shows the scores for health, collaboration and tension, afterwards presenting a series of views including valence-over-time plots, lists of key moments, topic-health tables, and the option for raw data inspection of the selected meeting.:contentReference[oaicite:11]index=11

VI. Illustrative Use Case

With the help of pre-meeting chunk-level valence data and key-moment annotations, we created meeting metrics and analyzed their emotional dynamics. In meetings where the tone was mostly positive, health scores were high and tension scores were low, with collaboration scores reflecting stable valence. On the other hand, meetings with very large negative parts showed more tension and greater key-moment deltas.

Topic-Health views showed that some topics were always associated with higher health scores than others, implying that the discussions (e.g., planning vs. conflict resolution) tend to take place in more positive or in more tense contexts. These observations are qualitatively similar to the findings of previous studies on interaction analysis in software teams [7], [8].

VII. Discussion

Our method illustrates that rather basic valence-based calculations on chunk-level transcript data can give rise to meaningful meeting-level metrics. The health, collaboration, and tension scores are easy to understand and compute, plus they are based on continuous sentiment values rather than discrete labels. Key-moment detection gives a brief overview of the most emotionally significant points during the conversation, while topic–health summaries connect emotional tone and thematic content.

On the other hand, there are drawbacks that come with the use of this approach. The quality of the metrics is very much influenced by how precise the upstream sentiment model is and how representative the chunks are. Valence by itself cannot completely encompass all factors of communication quality like sarcasm, dominance or support, therefore it should be interpreted cautiously and alongside qualitative review.

VIII. Threats to Validity

Data representativeness: The current implementation has been tested on a limited set of meetings with specific preprocessing pipelines. Results may not generalize to all software teams or domains.

Model accuracy: Errors in sentiment prediction or transcript segmentation can propagate into metric computations, potentially distorting health, collaboration, or tension scores.

Interpretation bias: Users may over-rely on quantitative scores without considering context. High tension scores may sometimes reflect productive debate rather than dysfunctional conflict.

IX. Conclusion and Future Work

We introduced the Meeting Health Dashboard's implementation that converts chunk-level valence data into understandable meeting-level metrics, main sentiment shift moments, and topic–health summaries. The system works with structured CSV inputs, and when required, it automatically calculates health, collaboration, and tension scores. These analytics pave the way for the development of more advanced tools that are capable of providing support for retrospectives, coaching, and carrying out research on emotional dynamics in software teams.

In the future, we are going to incorporate speaker information in such a way that emotional patterns could be assigned to individuals or subteams, include additional signals like engagement or dominance, and test the efficacy of the Dashboard in controlled studies with software development teams.

References

[1] D. Graziotin, X. Wang, and P. Abrahamsson, "Happy software developers solve problems better: Psychological measurements in empirical software engineering," *PeerJ*, vol. 2, p. e289, 2014.

[2] ——, "How do you feel, developer? an explanatory theory of the impact of affects on programming performance," *PeerJ Computer Science*, vol. 1, p. e18, 2015.

[3] F. Calefato, F. Lanubile, F. Maiorano, and N. Novielli, "Senti4sd: A sentiment classifier for software developers," *Empirical Software Engineering*, vol. 23, pp. 2520–2552, 2018.

[4] T. Ahmed, A. Bosu, A. Iqbal, and F. Rahman, "Senticr: A customized sentiment analysis tool for code review interactions," in *Proceedings of the 14th International Conference on Mining Software Repositories (MSR)*, 2017, pp. 103–114.

[5] M. Herrmann and J. Klünder, "From textual to verbal communication: Towards applying sentiment analysis to a software project meeting," in *2021 IEEE/ACM International Workshop on Emotion Awareness in Software Engineering (SEmotion)*, 2021, pp. 33–40.

[6] R. Al-Obaidi, S. Mahmud *et al.*, "Development and evaluation of sentiment analysis tools for software engineering: A systematic literature review," *Information and Software Technology*, vol. 135, p. 106555, 2021.

[7] K. Schneider and coauthors, "Positive and negative interactions and their relation to team performance," *Journal of Systems and Software*, 2018.

[8] J. Klünder and K. Schneider, "Meeting behavior analysis using act4teams-short: A coding scheme for interaction analysis," in *Proceedings of the International Conference on Software Engineering*, 2020, pp. 1–12.

[9] M. H. D. Team, "Installation guide for meeting health dashboard," Online documentation, 2025, project documentation file `installation_guide.md`.

[10] ——, "meeting_health_dashboard_app.py," Project analysis script, 2025, python source file implementing metric computation and key moment detection.