

Chat Vibe Check: Project Proposal

April 11, 2025

Project Metadata

- **Project Title:** Chat Vibe Check
- **Team Members Name:**
 - Deepa Rukmini Mahalingappa (34040788, drukminimaha@umass.edu)
 - Benjamin Ibasco (28561432, bibasco@umass.edu)
 - Javin Mendiratta (33328658, jmendiratta@umass.edu)
- **Project Repository:** GitHub Repository Link

Overview and Motivation

Everyone chats. Everyone texts. But we don't all text the same- and the way we text can tell us a lot not only about ourselves, but also the people with whom we converse. Current chat visualization tools focus on the quantitative- how often you text, what times you text most often, who you text the most, and so on. But texting, and conversation in general, is inherently emotional and qualitative in nature. We wanted to design and create a platform that can analyze not only the quantitative, but also the qualitative. More specifically, our motivation is to build a website that can not only provide information on your chat/texting specifics, but also provide deeper insight into your interpersonal relationships and various sentiments across conversations. To give our analysis a focus and structure, we wanted to analyze chats based on the principles defined by modern psychological frameworks, such as attachment style and behavioral analysis.

Some more specific goals include:

- Comparing overall sentiment in a conversation (group and individual) over various time periods
- Analyzing sentiments expressed across various conversation topics to provide insight into individual/group perspectives
- Displaying response times at various points in the day and for different topics
- Estimating behavioral tendencies and attachment style of chat members using the aforementioned metrics

Related Work

The main inspiration of the project mainly came from the fact that there is growing empirical evidence that arranged marriages are scoring better on metrics like satisfaction and divorce rates compared to traditional marriages (PsychologyToday.com article). At the time, we were interested in exploring the reasons why and quickly realized this would be a challenging goal. We then explored narrowing the scope but staying on the theme of understanding relationships and was inspired by personal chat visualizations done by users analyzing trends in their relationship (Example from Reddit). From the examples we noticed the visualizations or insights were very surface level and thought that there might be use for a tool that tried to explore more meaningful behavioral patterns from the chat data. To get a better direction on how or what we would try to observe from the chat data, we decided to base our analysis around an actual psychological framework and we chose Attachment Theory for this.

Questions

The main question we're trying to answer is what behavioral patterns can we observe from chat message interactions between people to get a better understanding of their attachment styles? As we explored this idea, we focused on two key derived attributes: Sentiment and engagement. From this realization we focused on having our visualizations answer the following questions:

- How does the sentiment of my conversation evolve over time?
- Are there certain conversation topics that are intensely positive or negative?
- Are there periods of time or topics in my conversation when a person is less responsive than usual or tries to avoid or change the topic?

Data

To prototype the data pre-processing and ingestion implementations, we decided to just work with our own Whatsapp group chat data as we just needed to work with the data format for the messaging platform we wanted to support.

To prototype the visualizations, it was difficult to find a WhatsApp chat dataset in English so we settled on using a dataset from Kaggle of over one million two-person chat logs related to Ubuntu technical support: <https://www.kaggle.com/datasets/rtatman/ubuntu-dialogue-corpus/data> . At this early stage, there hasn't been a need to do any pre-processing on this dataset.

The following datasets were also considered:

- Movie Dialogue Corpus - Initially chosen as a backup in the worst case scenario wherein we couldn't find two person conversation chat data in English.
- The following WhatsApp chat datasets were considered and although there are analysis libraries that support multiple languages (e.g. multilang-sentiment) we ultimately avoided these as we wouldn't be able to compare the results of our analysis to what we thought the data was actually expressing.
 - WhatsApp Chat used for studying cyberbullying in Italy
 - WhatsApp Group Chat in Hindi

Exploratory Data Analysis

- **Download:** We'll need to make sure that our website references instructions on how to download the data from our supported platforms.
- **Cleanup:** We'll need to ingest data from multiple formats and store it such that we can at least query for the message itself and the following metadata: sender, receiver, send time, seen time (this may not be available on all platforms).
- **Quantities:** The quantities we expect to derive from the data are sentiment, response time, engagement, and conflict resolution metrics relative to topics and time.
- **Analysis:** We've decided for now that we'll keep the scope simple in the beginning and only focus on text messages and emojis. Video, audio, GIFs, and images will be ignored in the onset.

Design Evolution

The project aimed to extract and visualize emotional patterns from chat data in a way that is easy to interpret, interactive, and actionable. Our design evolved significantly from the initial proposal as we explored more effective ways to represent time-based and user-based sentiment trends.

- **Initial Visualizations Considered:**

- Bar charts to show average sentiment per user or per day.
- Line charts for sentiment flow over time.
- Pie charts to show distribution of sentiment types.
- Snakey Diagram showing the flow of conversation state transitions from neutral to conflict to resolution.

- **Final Visualizations Used:**

- **Heatmap:** Showcased sentiment variation across time periods (e.g., hourly/daily), helping identify emotional peaks.
- **Scatter Plot - Response Time vs Sentiment:** Revealed correlation between emotional intensity and reply latency.
- **Scatter Plot - Timestamp vs Sentiment:** Illustrated sentiment flow throughout the conversation, supporting detection of escalating or resolving emotional trends.
- **Chat Parser Table:** Provided a clean, searchable layout of individual messages with timestamps, user labels, and sentiment tags.

- **Deviations from Original Proposal:**

- Initially proposed line charts were replaced by scatter plots, which better captured the irregular and message-level nature of chat data.
- We're currently playing around relating time, topic and sentiment instead all in one chart rather than having a two separate ones for topic and then for time.
- The most basic implementation for showing the chat is currently just having it as a separate page but we will look into the displaying it as the user clicks on messages.

Visualizations Used

- **Heatmap: Sentiment vs Time**

- **Design Goal:** To identify how sentiment changes over time (across hours or days), highlighting peak emotional periods like high negativity during specific hours.
- **Color Mapping:**
 - * Red = Negative
 - * Green = Positive
 - * Gray = Neutral

- **Scatter Plot: Response Time vs Sentiment**

- **Design Goal:** To explore whether the speed of replies correlates with emotional tone, revealing patterns like faster responses during emotional conversations.
- **Design Enhancements:**
 - * Color-coded sentiment points for clarity.
 - * Tooltips on hover for detailed insights.
 - * Filter options by user or time range.
- **Design Principles Used:**
 - * Data-ink ratio for clean visualization.
 - * Pre-attentive processing with color and shape to distinguish sentiment.

- **Scatter Plot: Timestamp vs Sentiment**

- **Design Goal:** To display sentiment flow chronologically over the chat session, allowing trend detection like escalating negativity or emotional recovery.
- **Design Decisions:**
 - * Smooth timeline across X-axis.

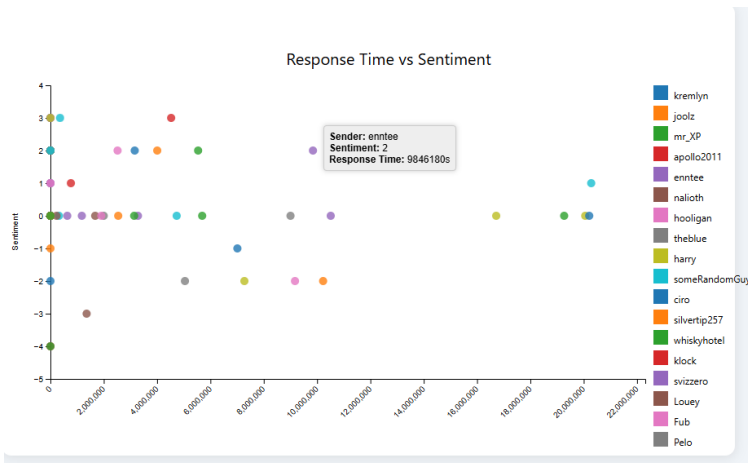
- * Hover details with sentiment classification.
- **Design Principle Used:**
 - * Temporal alignment of visual elements for easy scanning.
 - * Consistent color scale to maintain cognitive ease.
- **Chat Parser with Sentiment Tags**
 - **Design Goal:** To convert raw chat into a clean, readable, and searchable table with labeled sentiment.
 - **Features:**
 - * Each message row includes: Timestamp, User, Message, and Sentiment Score (with tag).
 - * Sentiment tags are color-coded.
 - * Optional search/filter by user or sentiment type.
 - **Design Principles Used:**
 - * Clarity and hierarchy through formatting, spacing, and alignment.
 - * Redundancy principle — sentiment shown via both text and color for accessibility.

Implementation

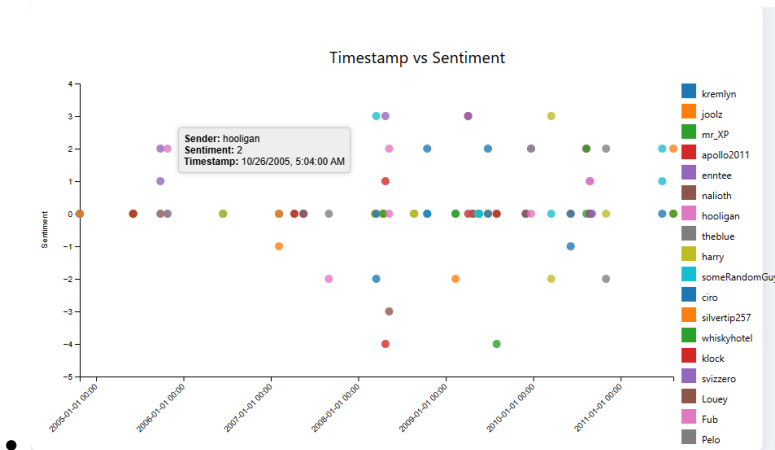
Our implementation, at the current stage, features a home page for uploading documents, and 4 main features across 3 sub-pages. We plan to add more implementations, interactivity, and design consistency before our final submission.

Before any of the views can be made, the user must upload their chat log to the website in order for it to be analyzed and parsed on the backend. As of now, our inputs must be a csv with the following format: (!). However, we plan to extend the input types to a more generic implementation.

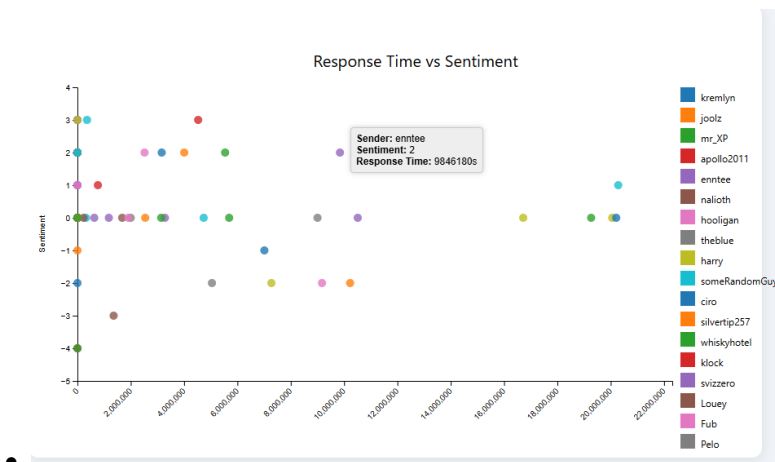
- Home page: This is the page where the user will select the file and upload it. After uploading the file, they can then choose a view from the toolbar. We will decide on and create a default view to show the user immediately upon upload to ensure a smoother and more natural experience.



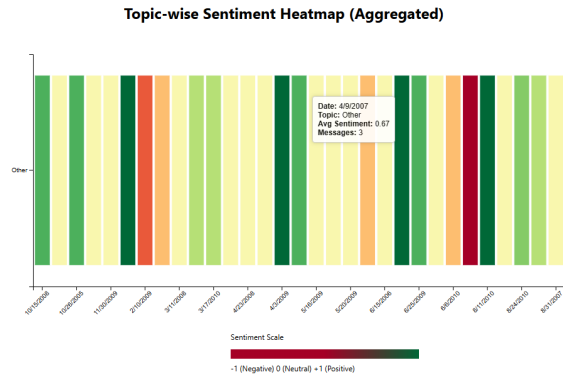
- Time (X-axis) vs Sentiment (Y-axis) Scatterplot: Each point represents a message and each point is labeled by user. This plot is meant to showcase the various sentiments shown in the conversation over time, with colors of nodes corresponding to the users and the y-axis showing sentiment (positive vs negative). Each node (message) also displays the sender, sentiment, and timestamp when hovered over. Currently, the graph looks a little cluttered and is working for group messages. We want to see how it will look for smaller chats eventually and add a feature to specify members for group chats.



- Response Time (X-axis) vs Sentiment (Y-axis) Scatter Plot: Each point represents a message and each point is labeled by user. This plot is meant to showcase the various response times shown in the conversation for varying sentiments, with colors of nodes corresponding to the users and the y-axis showing sentiment (positive vs negative). Each node (message) also displays the sender, sentiment, and response time when hovered over. Currently, the graph is a little confusing to properly interpret. We are considering switching the axis to make response time the dependent variable, so we can analyze if different sentiments influence response time. We may expand this to include topic instead of sentiment as an axis, and explore different chart styles.



- NEEDS WORK: Topic (X-axis) vs Sentiment (Y-axis) Heatmap: This graph is incorrect at the moment, as we need to update it. Currently we have the x-axis showing the timestamp of the message and the y-axis being discrete with certain topics. As such, a different chart style, where topics are more distinctly serrated may be in order. Additionally, our code that determines topics is somewhat faulty, and we need to broaden our range of topics to identity- right now every conversation is being placed in the "other" category.



-
- **Conversation Display:** This visual simply lists out the entire chat conversation in a scroll-able text box, showing the message and it's corresponding sentiment. This is to provide a familiar interface to the user, while providing a more "overall" view of the data. We will need to incorporate pages or some effective way to split up the conversation for larger datasets.

Topic-wise Sentiment Analyser	
Chat Messages & Sentiment	
[10/25/2004 2:49:00 AM] kremlyn: http://www.newsforge.com/article.pl?sid=04/10/12/1421241 😊 (0)	
[10/25/2004 2:49:00 AM] kremlyn: Article on Ubuntu. 😊 (0)	
[10/25/2004 2:50:00 AM] joolz: sloooow 😊 (0)	
[6/5/2005 2:42:00 AM] mr_XP: Hello Does Ubuntu have somekind of register to configure applications and os settings? 😊 (0)	
[6/5/2005 2:44:00 AM] mr_XP: Hello Does Ubuntu have somekind of register to configure applications and os settings? 😊 (0)	
[6/5/2005 2:45:00 AM] apollo2011: Linux doesn't have an equivalent to the Windows registry 😊 (0)	
[9/27/2005 1:48:00 AM] enntee: Can anyone recommend an NFS alternative that works transparently? 😊 (2)	
[9/27/2005 1:49:00 AM] nalieth: once you get NFS set up, it IS transparent 😊 (0)	
[9/27/2005 1:49:00 AM] enntee: yeah, i know. but it's a PITA trying to keep my uid/gid's in sync 😊 (1)	
[10/26/2005 5:04:00 AM] hooligan: Where does synaptic install the programs to? Like it aint on desktop 😊 (2)	
[10/26/2005 5:05:00 AM] hooligan: nvm 😊 (0)	
[10/26/2005 5:06:00 AM] theblue: Synaptic installs the programs to wherever they need to be installed. Most of them can be found in the Gnome menu, and if not there, in the Debian submenu. All applications on your system that are recognized by Synaptic are in your Debian submenu. 😊 (0)	
[6/15/2006 8:31:00 AM] harry: can someone giv me a link in ubuntu wiki to install sun java 😊 (0)	
[6/15/2006 8:31:00 AM] someRandomGuy: java.com 😊 (0)	
[6/15/2006 8:32:00 AM] harry: i mean running the command in terminal to get sun java. there are 2 kinds the black and the sun 😊 (0)	
[2/4/2007 4:42:00 AM] ciro: how can I install mercury messenger on ubuntu? 😊 (0)	

Optional Features

- **Message Grouping:** We'll need to decide if we'll try to group messages together as part of the same conversation.

- We can base this off two criteria: topic and time. Starting with time-based grouping and then later by topic if time permitting.
- Conflict Resolution Snakey Diagram: Depicting the flow of conversation state transitions from neutral to conflict to resolution.
- Give an attachment style assessment with a confidence score.
- Show each user's perceived attachment style over time.
- Show patterns of behavior based on sentiment and attachment style analysis relative to time and topics.

(Note: We've decided to make visualizations that try to use a derived attachment style metric optional, as we currently have no guarantee if the derived attributes conclusively determine attachment style.)

Security

- Client-side processing such that our data consists only of the messages, the timestamps and a generic username assigned like "User A" or "User B".
- Use our chosen hosting platform's (like Heroku) built-in HTTPS.
- Then we always want to delete the data once the user who has uploaded ends their session, so we're planning on using an in-memory datastore (like Redis) configured with TLS and disable any of its features that write the data to disk (like backups). This way we avoid needing to encrypt data on disk.

Tech Stack

- Front End: React, D3, Modal overlay library (react-modal?)
- Back End: Node.js + Express.js (Express for data ingestion REST APIs), Redis, some NLP library
- Deployment: Heroku, GitHub

Evaluation

- For such a project that asks users to require them to upload potentially sensitive data, the tool for sanitizing and removing that data needs to be within their control and trust. This was the main reason why we weren't able to initially get data from real people within our networks.
- We still need to fix how we're identifying topics as the current implementation is finding it hard to identify specific topics in the content and just labeling a lot of data as "Other"
- Admittedly, we may have spent too much time experimenting with applying the visualizations even in group settings and trying to group messages by topic/conversation so we still need to finetune the scatter plot visualization as well in terms of scoping the data to specific conversations to have more meaningful insights.