

Disclose: Smart Diary With Mood Tracking Capabilities

MARK MILEYEV, mmileyev@ucsc.edu
 KEVIN SERRANO, kfserran@ucsc.edu
 SHAWN CHUMBAR, schumbar@ucsc.edu
 ROBERT CHUNG, rochung@ucsc.edu

This final report will be discussing the inception, development, and purpose of "Disclose", an Android application created at the University of California Santa Cruz for CMPS 121 by the Disclose team (Shawn Chumbar, Robert Chung, Kevin Serrano, and Mark Mileyev). First, we will thoroughly cover the purpose of the application and the problem it solves. Then, we will contrast our app with other similar applications to explain why our app is far more powerful with more use cases over other existing similar applications as well as pointing out the features that are lacking in similar applications. We will follow the discussion by highlighting all the various features and components of our application. We will also give a full break-down of how the user will be using Disclose, from the opening screen to the minor subtleties that make the Disclose user interface a pleasant experience. Next we will discuss the development process as we experienced it. We will also share how the components of Disclose were implemented and the order in which we chose to implement them and why. Throughout the development, the application has taken various forms and as such we have made adjustments to the final product. We will explain and contrast between our final iteration and our original proposed application. Then we will elaborate on each individual team member's contribution to the final project with regards to each application component, interim and final presentations, and proposal and final report. As a conclusion, we will discuss the future plans we would have wanted to carry out for Disclose if time permitted.

ACM Reference format:

Mark Mileyev, Kevin Serrano, Shawn Chumbar, and Robert Chung. 2018. Disclose: Smart Diary With Mood Tracking Capabilities. 1, 1, Article 1 (December 2018), 5 pages.

1 OBJECTIVE

Disclose was designed to analyze and keep track of the user's mental health by using the IBM Watson API to dissect the journal entries and rate the overall tone of SMS message threads in the user's phone. Our purpose for creating this application is to solve the issues of: the need of a portable mental health tracker and an efficient way to quickly identify emotional states. Disclose will help our users identify which SMS messaging threads result in positive emotions like joy and which consist of mostly negative emotions, like sadness. The user can then take their own course of action depending on the results, potentially limiting communication with the people who display constant negativity and choose to frequent the more positive communication threads instead. In addition to analyzing SMS messages, Disclose's mood entry activity allows the user to create their own "diary page", which gets analyzed, stored, and displayed through an easy to read graph that helps users recognize and

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2018 Association for Computing Machinery.

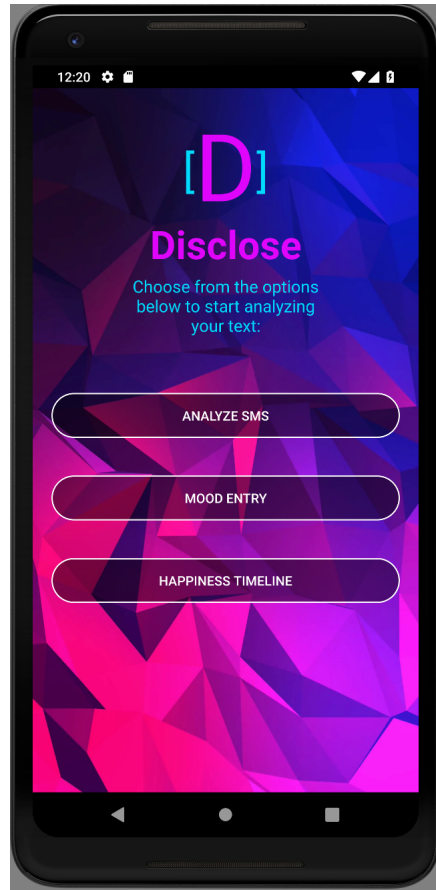
track the way they feel. There are few other similar applications that provide all the text analysis features that Disclose offers. Our most unique feature of Disclose is that it reads the SMS text messages stored on the user's phone and provides a sentimental analysis with a powerful artificial intelligence API, something that no other application is currently doing. None of them even use a central service like IBM Watson nor do they offer the option to plot the happiness scale on a graph like Disclose does. These types of apps are strictly limited to single or group messaging, as opposed to offering both the journal entry and the graph timeline. For instance, a similar app that is already garnering attention is called "Mei: SMS Messaging + AI". Mei uses artificial intelligence to optimize relationships between friends and lovers. Depending on the tone of the messages sent from another party, it would then offer advice to the user on what to say. According to the app description, it claims to be able to construct full personality profiles from the sending party and detect strange behavior. It is clear that Mei is much more involved in terms of functionalities because of how complex its built in artificial intelligence system is, although its focus is purely on relationship-building. In contrast, Disclose is far more general, as it analyzes tone from both parties regardless of what the context is. In practical terms, one could say that Disclose would be the more simpler, stepping stone counterpart to apps such as Mei. Despite the similarities of the above mentioned application, Disclose offers a unique timeline display of emotional state of your journal entries and even the emotional state of your conversation threads. This feature is lacking in most other similar applications like Mei. Furthermore, other applications are lacking our simplistic user interface that brings graphical visuals and even displaying emotional growth over time.

2 COMPONENTS

Disclose is primarily a sentiment analysis application aimed at identifying sources of negativity or positivity. When the user opens the app, he or she is greeted with a title page. The title page consists of a simple, easy to follow interface with a distinct theme. The page has three buttons: analyze SMS, mood entry, and the happiness timeline. The user can then choose between analyzing their individual feelings in the "mood entry" page, or have their text messages analyzed in the "Analyze SMS" page. If one were to choose any of the three buttons, the app is designed to use a transition animation for the next pages. In the Analyze SMS page, it allows you to choose which contacts on your phone you want to analyze. Then, there are three options: your messages, their messages, or both your and your friend's messages. Picking among the three of them will result the message entry values to be plotted in the happiness timescale page. The second component will be the Mood Entry page. This page is meant to allow the user to express their deepest thoughts, so the theme changes to a more natural one, aiming to keep the user from feeling like they are just talking to a device. It allows the user to enter a diary entry of any length, and once the user presses the "submit" button, the Watson API will be applied and it will instantly analyze all diary entries and the screen transitions to a pie chart that divides the user's overall mood based on the current and previous diary entries. In addition to the pie chart which is able to display specific moods like sadness, fear, joy, etc., the user has the option to click on "happiness timeline", which will have the timeline plotted with happiness values from when the user entered their first mood entry. Eventually, once the user has finished inputting enough diary entries into Disclose, the timeline grows into a very insightful track record of the user's mood and stands as a reminder to be more conscious of how he or she has been feeling.

3 DEVELOPMENT

Since we are students who are developing our first mobile application, we had a huge learning curve to overcome. We started off rather slowly, because we had to learn some of the basic functionality of Android Studio. However, our first major hurdle came from the IBM Watson Tone Analyzer. We had to learn a new set of tools that was poorly documented and thus, we faced two major issues. First, we had to solve how to use the API to grant us access to their servers to use the Watson Tone Analyzer. Second, using the Watson Tone Analyzer requires those requests sent to the IBM servers be implemented as background tasks; however this was never mentioned



in the documentation. Learning the API was nearly impossible from a mere trivial usage test of the API. After overcoming these two hurdles, we were finally able to develop a working command line interface that could take the user's input, and return a JSON output with various analysis results based on the input. Having gotten this far, the next main challenge was processing this dataset, figuring out how to parse through all of the data and pull the information we wanted, and then keeping it lightweight to make sure it would run on an android device. In order to do this we had to use similar techniques learned in Assignment 2, running our analysis and calling the IBM Watson API as a background task, then displaying our result on the main UI thread. We had to learn how to use the standard Java libraries to parse through the API Response, and got a rough draft working for the interim presentations. Next we focused our attention to making the individual components. We started with "Mood Entry", then we worked on the "SMS Analysis", then we added graphical visuals like the pie chart and the timeline. Lastly we added an aesthetically pleasing intuitive design. In the inception, we originally thought to make use of the RESTful Twitter API to run sentimental text analysis. However, we decided not to use Twitter for sentiment analysis since many people may display a different personality on social media compared to talking to their personal friends or loved ones via SMS. Thus, the "Mood Entry" feature was born. Rather than using social media posts to analyze mood over time, we decided the user can make their own private entries directly into our app, acting like a "Smart Diary".

In the final application, the user is able to select contacts from their phone to analyze text messages to identify sources of positivity or negative sentiment. Ultimately, our original proposed application did not change much. We kept the two main features of analyzing diary entries and reading the user's messages, which were implemented in parallel with the progression of the course (namely learning about permissions and running tasks on a separate thread). Mostly we added additional features to the original proposed application minus the Twitter implementation and the calendar view in lieu of the "Mood Entry" and "Timeline and Pie Chart". Although our proposal had the additional feature of analyzing various social media posts, implementing this would have required mastering various additional API's and further optimization to keep the app lightweight, which we quickly found out would not be possible in the span of one quarter since our team members were not familiar with web development. Also, we collectively decided as a group that reading someone's SMS conversations would give a better insight into their mood than how they chose to present themselves on social media, leading us to take this revised path in our application development.

4 CONTRIBUTION

Mark Mileyev: Implemented the API in the early stages of the application, then worked on parsing through the data returned by IBM Watson Tone Analyzer from within the android application and organizing it in an easily accessible manner. Developed the database for holding user's diary entries and relevant metadata. He planned the timeline of group work with deadlines and recognized that certain features would have to be changed to create a working product by the end of the quarter.

Kevin Serrano: Facilitated the exchange between API calls and displaying results on the user interface. Spent a lot of time fixing any issues that arose during development. Worked on customizing the user interface so as to have a uniform button design, a simple way of finding contacts, the layout of the different screens and how they interact with each other. Kevin also implemented an interface for our classes to simplify the SMS integration throughout the application.

Shawn Chumbar: Spent most of the project working on the user interface and figuring out placement of certain buttons for a design that was not only easy to use but also aesthetically pleasing. Also helped configure the pie chart that was used to display the user's emotions and tried to tackle the issue of having the mood display on a calendar. Lastly, he dealt with debugging and writing any testing code to make sure that the android application was working. He was also the morale specialist and would often resolve any conflicts that would arise among group members.

Robert Chung: Assisted with the front end code by making sure the Watson API was reading the messages a user inputted. Experimented with different designs by using a variety of ways to read the text messages and the journal entries. Also assisted with the development of the final user interface design by selecting and simulating various backgrounds.

Collectively: All group members met in person to collaborate and prepare the interim and final presentations. While most of the demo was presented by Mark Mileyev, the group planned the process and flow of the presentation and gave input on all practice presentation runs. Our Proposal and Final Report also had equal collaboration with each member working on different sections, and debating with each other on different paths to take.

5 FUTURE WORK

There were many features that we were not able to implement as a group. For example, we were not able to implement the ability to read a user's tweets and run sentiment analysis on it due to early complications with the

IBM Watson Machine Learning: Tone Analyzer. Another thing that we found difficult was the Calendar feature we were trying to implement. Our goal was to make it so that the overall mood for a person would map to a calendar and show their happiness and sadness relative to the day that they were looking at. We were then trying to make it so that clicking on a particular day opened up the information for that day. We then wanted a graph that would show the separate information for the different days, showing separate lines for happiness, joyiness, sadness, and analytical moods. The last thing we wanted was to be able to link psychiatrists' phone numbers and offices to give people a resource to visit or call when they were under mental duress. While many of these features were missing, we as a group can foresee working them into the application. Adding the mental health counseling feature into the app would help at-risk individuals get their emotions under control and help parents and teachers analyze issues with many troubled teens. However, as you can probably guess, we ran into many issues. Some of the issues stemmed from the complexity of the documentation behind the IBM Watson Machine Learning: Tone Analyzer. Another issue included finding time to meet up with our team members, as three out of the four team members were fourth years who had challenging classes that were prioritized before this class.

6 CONCLUSION

As a group, we consider the final product delivered in the time we had as a great success. We had many challenges throughout the quarter not just as developers but as university students and yet we managed to overcome these challenges and produce an amazing product. One of the big challenges that we faced was working with the IBM Watson Machine Learning: Tone Analyzer. The documentation behind the IBM Watson Tone Analyzer was extremely difficult to understand, as it was very technically tedious and robust. The difficulty was even more amplified by the fact that none of us had prior Android programming nor mobile application development experience. As a result, many of us ended up being confused along the way, and we spent many nights just trying to figure out tiny issues preventing us from delivering the product we wanted. However, we all feel that we did a good job and definitely put forth our best effort. We decided to not implement the Twitter tweets analysis, since many people may try to portray a certain type of personality online. Instead, we focused more on text messages because that allowed us to show the true emotions that people may have within their closest and personal relationships. We also wanted people to be able to enter their own text into a journal and be able to see how their emotional state is growing over time, which we were able to implement successfully although the presentation of that diary data could have been more advanced given more time to develop the application. Being able to apply what we learned in class directly to our app as we learned it was a very satisfying experience.