A System for Storing Journal Entries and Implementing Psychotherapy Using LLMs and Support Vector Machines

Zhijun Zhang[1], Austin Amakye Ansah[2], Ang Li[3]

[1]Columbia International College, 1003 Main St W, Hamilton, ON L8S 4P3
[2]The University of Texas at Arlington, 701 S Nedderman Dr, Arlington, TX 76019
[3]California State Polytechnic University, Pomona, CA 91768

ABSTRACT

Some people keep journals to track their mental health or keep a record of things happening in their lives [9]. While journaling can help identify patterns in mood and tackle stress and anxiety at the root cause, only 8 percent of the population do it. With the 8.58 billion mobile subscriptions in use worldwide, we take advantage of mobile applications to deliver a quick on-the-go solution for mobile journaling that can be used free of charge [10]. This application utilizes NLP to determine user mood and provide tasks and goals that the user can work towards to improve their well-being. The NLP model developed in this paper has an accuracy of about 91% in correctly classifying moods on a sequence of words in a validation dataset. The model is good enough for most text-based emotional sentiment analysis needs but could be re-trained to attain better results.

KEYWORDS

Journal, AI, Mental Health, Chat

1. Introduction

Some people keep journals to track their mental health or keep a record of things happening in their lives. Journaling can help identify patterns in mood and what is causing stress or anxiety [5][6]. There are more than 8.58 billion mobile subscriptions in use worldwide, but only 8 percent of the population currently keeps a journal [7]. Journaling has a lot of benefits. It can help manage stress and process emotions and improves learning, memory, and performance [8]. According to Habbitbetter's research, approximately 22 percent of the population has kept a journal or diary in the past [4]. With so many people owning mobile phones today, it would be easier to journal on them. Mobile journaling and mental health apps can also be used on the go, and there's no need to deal with waitlists, journaling, or insurance [5].

We propose an app that lets users not only receive AI-generated mental health assistance but also keep a daily journal log of their mental health state. The journal logs are used to create unique tasks and activities for the user, which can be tallied and checked off a list. The app also features a number of chat bots with predefined personas to communicate with the user entertainingly. The app is mostly for journal logging and check-ins but has some additional features like a lifetime

David C. Wyld et al. (Eds): SE, SAIM, SIPM, CoNeCo, ICITE, ACSIT, CMIT, FCST, SNLP – 2024
pp. 175-184, 2024. CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.140817

page which displays the user's life as a battery percentage, as well as a coin system when the user completes tasks. The coins are more of an incentive to get tasks done. The user can look for logs taken on certain dates by using a calendar as a filter. After clicking on a date, journal entries taken within that 24-hour time span are displayed on a swipeable carousel and journal logs can be edited.

The NLP server uses the last few journals logs to determine the user's mood as one of three classes, angry, joyful, or sad [13]. The output from the server determines what activities the user receives for that day. The user can choose to redo activities for more coins and swipe activities off of the task list.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Finding the Right Place to Host the Flask Server

One challenge encountered in the development of this project was finding the right place to host the Flask server. We first tried repl.it to see if we could host a Flask server globally and access it anywhere on the web, but the server could only be accessed temporarily from within the website. The service we agreed on using was render.com, a free-to-use web hosting platform much like Vercel or Heroku.

2.2. The Journal Component not Reloading

The second issue encountered involved the journal component not reloading and showing zero journal entries. The issue was caused by an invalid filter in the Firebase query. Initially, notes were being fetched after the selected date, which meant that if the user had journal logs on February 2nd for instance, the app would try to load notes from February 3rd onward. A good fix was to load notes created greater than or equal to the current date but limit the query to logs created only a day before the next day. This workaround correctly loaded the selected date's notes.

2.3. The Life Timer Widget

The life timer widget was also a bit of work to implement fully. For the time to be valid, we needed to fetch the birthday of the user at run time. Some users do not have a birthday set when they start using the app, so one has to be created without overwriting one that might already exist. For this, conditional visibility is used on the whole page to display a colorful user interface when the user is logged in, otherwise, a "set birthday" button is displayed only.

3. SOLUTION

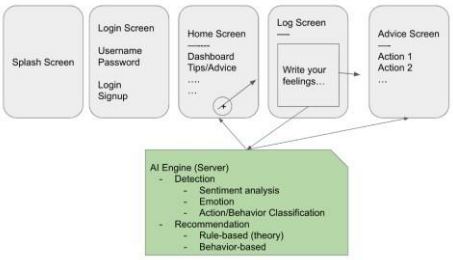

Figure 1. App design flowchart

When the user first opens the app, they are greeted with doors to different sub-applications within the main program. The first of the doors is a contacts app. This is where the user can create chat agents to converse with. Each AI agent has a different behavior as prompted using the GPT API [14].

Journal Screen

The journal screen is used to log user emotions like a diary. The user is greeted with an empty screen at first, and once they have created a journal entry, they can view it from the same page. The screen features a calendar filter feature that lets the user choose which date they want to view entries from. Each entry is given a title with the time that the entry was created/updated. The screen features a card carousel which displays a slight rotation animation for aesthetics.

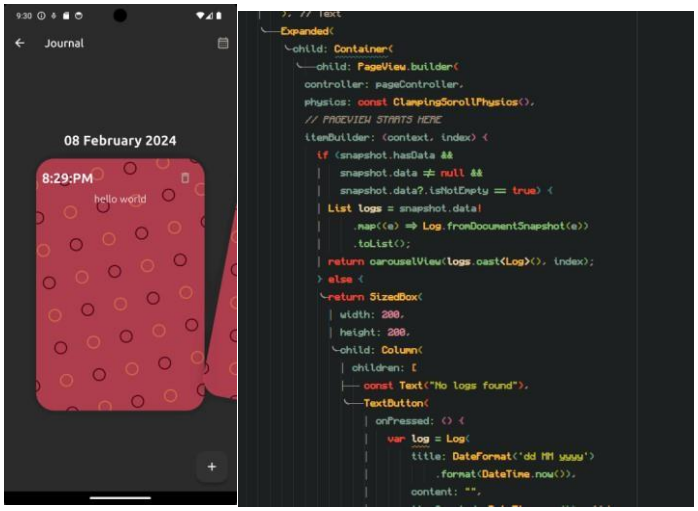

Figure 2. Journal Carousel

The journal page widget creates a swipeable pageview of note cards. Each card contains a Log object/reference that can be used to edit a note. The data is converted into a DocumentSnapshot so that the underlying data can be both edited and viewed. When the user is done editing a journal log and goes to the previous page, the ID from the document snapshot is used to overwrite the old journal entry. Journal logs are only displayed if, for the selected date, the firebase JSON response has data and the data is not empty, otherwise, a centered text saying, “No logs found” is rendered to the screen [15].

Bot personnel view

The contacts page features a list of custom natural language agents with predefined response personas. The user has the option to choose from a list of different persona patterns for bot creation. As shown in Figure 3.1, each persona pattern is described in a key-value dictionary. This context is passed to the GPT wrapper so that after a message is sent, the GPT API can reply in a way that conforms to the persona pattern that was passed in.

Figure 3. Screenshot of code 1

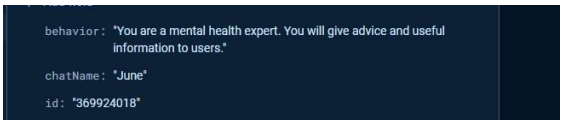

Figure 4. Firebase data view

As shown in Figure 4, the behavior is stored in firebase as a string that is referenced by the code in Figure 5. The “Role” of the bot, or system is defined as the behavior. Everything else, like the name and description of the bot is purely cosmetic.

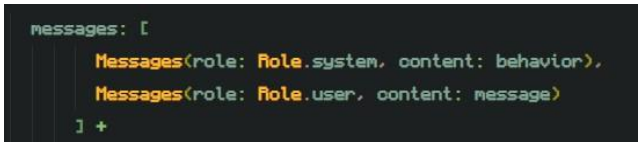

Figure 5. Screenshot of code 2

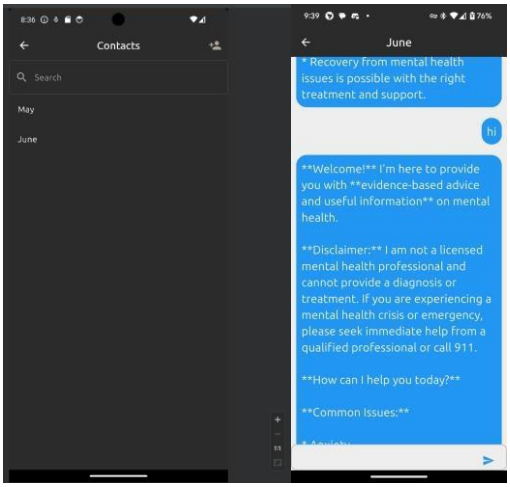

Figure 6. Contacts page

Figure 7. Screenshot of code 3

The unique ID for each chat is a unique Version 4 UUID. Initially, we used a hashcode of the currently logged-in userID plus a random number between 0 and 100 inclusive, but that would create conflicts if there were more than 100 chats for a single user. When the add button at the top right of the screen is pressed, a bottom sheet is opened which displays the persona of the bot to be chosen by the user. When a person pattern is chosen by the user, they are sent to a chat screen with the name of the persona pattern they chose.

Task Component

This component details a system for generating interesting tasks for users to accomplish while using the app. Using the user's last journal entry, a mood estimate is created by sending a request to a custom flasks server. The mood received is then used to fetch a random activity from a mapping of moods to activities.

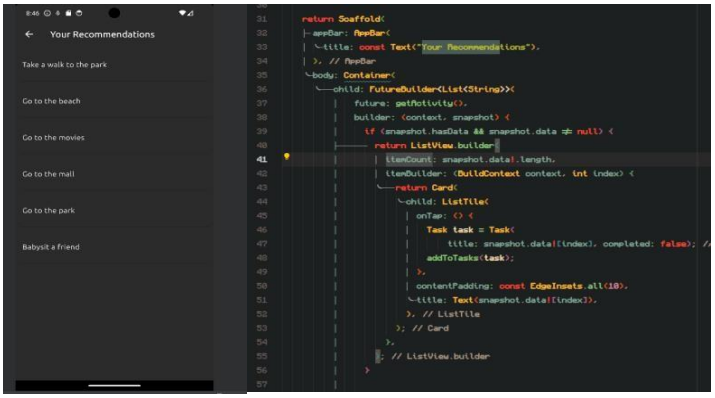

Figure 8. Task List

The above piece of code builds a component that renders a list of activities from a pre-written list of activities. Each activity is determined by the mood of the user from their last journal entry. The activity components are rendered using the listview widget and are wrapped in a swipeable widget from the Flutter standard library on the task list page. This allows the user to check an activity off of the list. The swipeable widget also provides a function callback to add coins to the user's account after a task is completed.

4. EXPERIMENT

4.1. Experiment 1

To test the accuracy of the model, we ran it against a validation dataset with 2000 different sentences. The goal is to see if the model's accuracy can be further improved and evaluate the model's weak points.

To test the model, we first preprocess all of the validation sentences into a Pandas data frame. We create a new column of pre-processed data with punctuations and non-alpha characters removed.

```
val_data = pd.read_csv('dataset/val.txt', sep=';', names=['sentence', 'emotion'])
val_data['preprocessed_sentence'] = val_data['sentence'].apply(preprocess_text)
X_val = vectorizer.transform(val_data['preprocessed_sentence'])
```

Figure 9. Figure of experiment 1

We create a table of validation labels using the emotion column and train the support vector machine classifier on our input data. The model gives predictions for each input value, and these are compared to the real values in a classification report.

Data Visualization

	precision	recall	f1-score	support
anger	0.92	0.91	0.91	275
fear	0.88	0.82	0.85	212
joy	0.92	0.94	0.93	704
love	0.87	0.83	0.85	178
sadness	0.91	0.94	0.92	550
surprise	0.84	0.79	0.82	81
accuracy			0.91	2000
macro avg	0.89	0.87	0.88	2000
weighted avg	0.91	0.91	0.91	2000

Figure 10. Classification Report

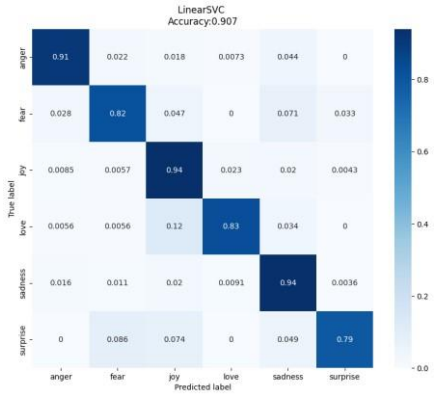

Figure 11. Confusion Matrix

Overall, the base model has about 91% (0.907) accuracy on all six emotional classes with the validation dataset. From the confusion matrix, joy and sadness are the classes that are more frequently correctly predicted with values of 0.94 and 0.94. Surprise has the least number of correctly predicted sequences out of the other classes but still has a good score of 0.79 when

finding true positives. Fear has a score of 0.82 and love has a score of 0.83. The model may have gotten certain inputs in the validation set wrong because some of the emotions from the training data may have not been properly cataloged.

To conclude, the classifier trained on the dataset is efficient enough for use in any application requiring sentiment analysis, but a score of 95% accuracy and up is more desired.

4.2. Experiment 2

The goal of this experiment is to train a Support Vector machine with a different dataset to see if the model is better at classifying text sentiment.

The same preprocessing steps as in experiment 1 were repeated with the dataset. The entire csv file is loaded into a Pandas DataFrame and split with the train_test_split function. Eighty percent of the dataset was used for training, and the remaining twenty for testing. Each entry is stripped of urls and blank data and passed into the Tfidf vectorizer of the sci-kit library. The preprocessed samples are then transformed in the vectorizer, and a Linear Support Vector Machine is used to fit the training data to its respective labels.

	precision	recall	f1-score	support
anger	0.90	0.87	0.89	607
fear	0.88	0.85	0.86	506
happy	0.90	0.93	0.91	1420
love	0.82	0.78	0.80	332
sadness	0.91	0.95	0.93	1257
surprise	0.88	0.71	0.79	170
accuracy			0.90	4292
macro avg	0.88	0.85	0.86	4292
weighted avg	0.89	0.90	0.89	4292

Figure 12. Classification report

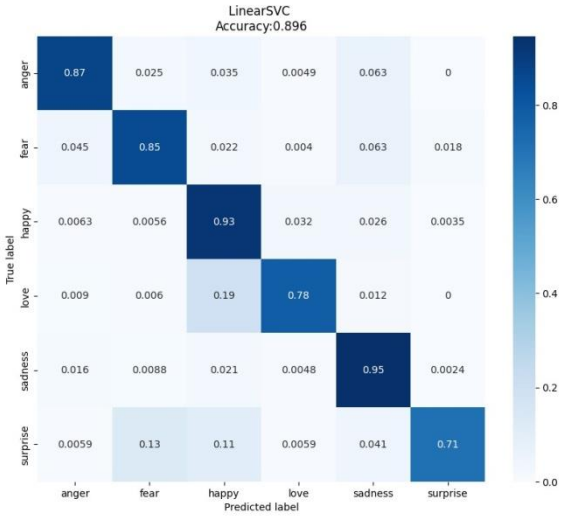

Figure 13. Confusion Matrix

Overall, the model was 90% accurate at correctly predicting the 6 label classes for the test sentences. In summary, 87% of the samples were correctly predicted as anger, 85% as fear, and

93% as happy. 78% of the samples were correctly predicted as love, 95% for sadness, and 71% for surprise. The model performed slightly worse on the new dataset than the last, and it can be speculated that this is caused by the data being overfitted to the training data.

A plausible solution is tuning the hyperparameters of the SVC to better fit the training data.

5. RELATED WORK

Throughout the design of this solution, there was a search for similar implementations of the same idea. Three papers were reviewed and compared to our solution.

Source A

The first source is titled: Developing mental health mobile apps: Exploring adolescents' perspectives. It was written by Rachel Kenny, Barbara Dooley, and Amanda Fitzgerald [1].

The paper talks about the requirements that users demand from a mental health mobile application. Although no app functionality is described, the paper details an app prototype called CopeSmart, a mental health application that mostly asks the user to explain their mood for the current day. Compared to our solution, CopeSmart is limited in the activities that it allows its users to do. Our solution not only has a journal log, but an AI assistant that can help the user cope with their problems.

Source B

The second source we researched is called, A Mobile App–Based Intervention for Depression: End-User and Expert Usability Testing Study [2]. The authors are, Fuller-Tyszkiewicz M, Richardson B, Klein B, Skouteris H, Christensen H, Austin D, Castle D, Mihalopoulos C, O'Donnell R, Arulkadacham L, Shatte A, Ware A

The paper presents a mobile application named BlueWatch. Its main set of features is a daily check-in system with alarms and reminders. Each check-in allows the app to recommend to the user a set of tasks and activities to enhance their well-being. Our solution has a similar feature that uses the journal entries as a reference in providing the most appropriate task for the mood of the user.

Source C

We analyzed a third source, A Guided Online and Mobile Self-Help Program for Individuals With Eating Disorders: An Iterative Engagement and Usability Study [3]. The research article, written by the authors: Nitsch M, Dimopoulos CN, Flaschberger E, Saffran K, Kruger JF, Garlock L, Wilfley DE, Taylor CB, and Jones M, presents a mobile app called SB-ED that aims to help individuals cope with their eating disorders through motivational CBT (Cognitive Behavioral Therapy.) The app features a check-in system to track eating habits. Lastly, it features a page to receive coaching from therapists using a one-on-one chat. Our solution does not have a check-in feature, and while it could benefit from that, it does include an AI-based chat system to receive expert coaching from a well-trained model. Though, human to human interaction is forfeited when using AI.

Limitations and Improvements

With the release of google's gemini large language model, openAI's GPT api has become redundant for text generation. Gemini is free and has a limit of 60 requests back-to-back every minute in its free plan, whereas openAI requires billing for all of its plans. If given a bit more time to work on bulk features in the app, a daily quote feature and better animation transitions would be a suitable addition.

6. CONCLUSIONS

In the first experiment, the goal was to determine the accuracy of the model on a validation dataset. The validation sentences cannot be processed by the model in text form, so they need to be preprocessed by tokenizing the words into numbers using the NLTK standard tokenizer. The tokens are then vectorized using the TF-IDF vectorizer. This vectorizer uses term or token frequency to determine which words come after the other in a sequence. The vectorizer is then fittransformed to the data to create a document-term matrix. This can be used to classify emotions based on the words present in the input sentence. The validation dataset had an accuracy of 0.907 in classifying emotions from new sentences. There is a small margin of error due to improper text classification or text that may have been assigned an emotion that it does not represent.

In the second experiment, the goal was to improve the accuracy of the model by training it on a different dataset. The model performed slightly worse than the original with an average accuracy of 0.89 and it can be speculated that overfitting caused this. A solution to this is to tune the hyperparameters of the model to better fit the data to the training set.

There are different mental health applications available in the mobile space. The first that this paper looks at is CopeSmart, a minimal yet effective app for logging use of mental health daily. Copesmart's pros was that it displayed notifications, helpful information, and encouraged emotional self-monitoring, however, some participants in the CopeSmart study reported that the app did not make them feel better, had interface issues, and engagement issues [12].

We also took a look at BlueWatch, an app that focuses on setting alarms, reminders and doing periodic check-ins and performing activities to treat depression and promote well-being. Our solution has journal logging and AI communicative features so BlueWatch is behind in some aspects.

Lastly, we analyzed a platform called SB-ED, an app where users can have one-on-one sessions with therapists in order to improve their mental health. The app emphasizes daily check-ins for users with eating disorders and utilizes CBT to improve the well-being of its users.

Overall, using 3 emotional classes is too minimal. Expanding the UI to include more AI generated content would add to the interactivity of the app [11]. Mini-games and more interactive activities would also drive-up user engagement.

REFERENCES

[1] Kenny, Rachel, Barbara Dooley, and Amanda Fitzgerald. "Developing mental health mobile apps: Exploring adolescents' perspectives." Health informatics journal 22.2 (2016): 265-275.

[2] Fuller-Tyszkiewicz, Matthew, et al. "A mobile app–based intervention for depression: End-user and expert usability testing study." JMIR mental health 5.3 (2018): e9445.

[3] Nitsch, Martina, et al. "A guided online and mobile self-help program for individuals with eating disorders: an iterative engagement and usability study." Journal of medical Internet research 18.1 (2016): e4972.
[4] Bucci, Sandra, Matthias Schwannauer, and Natalie Berry. "The digital revolution and its impact on mental health care." Psychology and Psychotherapy: Theory, Research and Practice 92.2 (2019): 277-297.
[5] Alqahtani, Felwah, Andrea Winn, and Rita Orji. "Co-designing a mobile app to improve mental health and well-being: focus group study." JMIR formative research 5.2 (2021): e18172.
[6] Sperry, Len, ed. Mental Health and Mental Disorders [3 volumes]: An Encyclopedia of Conditions, Treatments, and Well-Being [3 volumes]. Bloomsbury Publishing USA, 2015.
[7] Smith, Matthew L., Randy Spence, and Ahmed T. Rashid. "Mobile phones and expanding human capabilities." Information Technologies & International Development 7.3 (2011): pp-77.
[8] Dougherty, Barbara J. "The write way: A look at journal writing in first-year algebra." The Mathematics Teacher 89.7 (1996): 556-560.
[9] Keyes, Corey LM, and Shane J. Lopez. "Toward a science of mental health." Oxford handbook of positive psychology 2 (2009): 89-95.
[10] Copper, Jenna. "A Case Study of Mobile Journaling in the Secondary Classroom." Journal of Interactive Learning Research 29.3 (2018): 359-375.
[11] Cao, Yihan, et al. "A comprehensive survey of ai-generated content (aigc): A history of generative ai from gan to chatgpt." arXiv preprint arXiv:2303.04226 (2023).
[12] Kenny, Rachel, Barbara Dooley, and Amanda Fitzgerald. "Feasibility of" CopeSmart": a telemental health app for adolescents." JMIR mental health 2.3 (2015): e4370.
[13] Liu, Ming, et al. "Federated learning meets natural language processing: A survey." arXiv preprint arXiv:2107.12603 (2021).
[14] Tingiris, Steve, and Bret Kinsella. Exploring GPT-3: An unofficial first look at the general-purpose language processing API from OpenAI. Packt Publishing Ltd, 2021.
[15] Khawas, Chunnu, and Pritam Shah. "Application of firebase in android app development-a study." International Journal of Computer Applications 179.46 (2018): 49-53.