Fine-Tune Large Language Model for Behavioral Activation Chatbot

1.Research Question

Behavioral Activation is a therapy method that helps reduce symptoms of depression and mood disorders by promoting involvement in rewarding activities. Recently, Large Language Models (LLMs) like GPT have introduced more intelligent chatbot capabilities.

However, these LLMs are generalized for a wide range of conversations and aren't tailored specifically for tasks like Behavioral Activation. The challenge is to adapt these advanced LLMs to effectively assist in Behavioral Activation through chatbot interactions.

How could we develop a chatbot that bridge the gap between LLMs' general capabilities and the specific requirements of behavioral activation in a chatbot context?

We pick the Chatgpt as our LLM and the Dataset was provided by Florian Onur Kuhlmeier and Sven Scheu. We start with data preprocessing.

2.Data Preprocessing for Fine Tuning

• id: this is the idea of the message (one id per row) -> ignore • conversation_id: an ID that signals which conversation the message belongs to. • flow_id: this is the id of the therapy session (behavioral activation was made up of three sessions / flows: verhaltenraktivierung-1, verhaltenraktivierung-2, verhaltenraktivierung-3). • Step_id: every flow consisted of multiple steps. You can ignore this column. • Direction: SEND (by chatbot) vs. RECEIVE (by user) • Payload: this is the message content • Content_type: which type of content the message has (image, text, question etc.) • message_order: usable to create the order of the messages (best to check with created_at) • created_at: date and time of the message -> best column to extract the order of the messages • interaction_order -> ignore

In [1]:
```
path = r"C:\Users\Li\Desktop\Engineering Seminar Human-Centered Systems\data\ver

import pandas as pd

df = pd.read_parquet(path)
```

First image of Dataset

In [2]:
```
print(df.head())
```

```
                                     id                       conversation_id  \
0  12d29e68-e636-4fe7-abb8-3d1e3dc661c3  3cc89a19-5742-4f66-a93a-86cad116bea1
1  cbee3172-53b5-4e3f-8bda-9e6a34d5280f  3cc89a19-5742-4f66-a93a-86cad116bea1
2  2bfa2b69-d75d-4e43-8ded-29a7998a101b  3cc89a19-5742-4f66-a93a-86cad116bea1
3  5e019e59-9c04-4e66-a3dc-cd28875b365c  3cc89a19-5742-4f66-a93a-86cad116bea1
4  dc3518c7-f7b0-408d-890b-fc94c23d7af7  3cc89a19-5742-4f66-a93a-86cad116bea1

                  flow_id step_id direction  \
0  verhaltensaktivierung-1   start      SEND
1  verhaltensaktivierung-1   start   RECEIVE
2  verhaltensaktivierung-1   start   RECEIVE
3  verhaltensaktivierung-1   start      SEND
4  verhaltensaktivierung-1   start      SEND

                                             payload  content_type  \
0  {"content":{"url":"https://media0.giphy.com/me...         image
1  {"content":{"flow_id":"verhaltensaktivierung-1...  flow_trigger
2  {"content":{"payload":"Hmm...muss das sein?"},...       payload
3  {"content":{"text":"Um es kurz zu machen: Ich ...          text
4  {"content":{"buttons":[{"content":{"accepts":[...      question

   message_order  interaction_order                 created_at
0              2                   0 2023-06-11 10:56:48.985350
1              0                   0 2023-06-11 10:56:48.985350
2              0                   0 2023-06-11 10:57:06.317682
3              2                   0 2023-06-11 10:57:06.317682
4              5                   0 2023-06-11 10:57:06.317682
```

In [3]:
```
print("column names are ",df.columns)
```

```
column names are  Index(['id', 'conversation_id', 'flow_id', 'step_id', 'directio
n', 'payload',
       'content_type', 'message_order', 'interaction_order', 'created_at'],
      dtype='object')
```

In [4]:
```
print("column numbers are",df.count())
```

```
column numbers are id                   20137
conversation_id      20137
flow_id              20137
step_id              20137
direction            20137
payload              20137
content_type         20137
message_order        20137
interaction_order    20137
created_at           20137
dtype: int64
```

Transfer the dataset as a csv file

we can check the dataset file directly

In [5]:
```
# Specify the file path and name
file_path = r"C:\Users\Li\Desktop\Engineering Seminar Human-Centered Systems\dat
# Write DataFrame to CSV with UTF-8 encoding
df.to_csv(file_path, index=False, encoding='utf-8')
```

Filter

This Python code uses pandas to filter a DataFrame df in two steps:

1. Select rows where flow_id equals 'verhaltensaktivierung-2', because it contains the most important data that we want to use in the fine tuning.
2. Further narrow down to rows where content_type is either 'text', 'question', or 'payload', as they are key components of prompt construction.

In [6]:
```
# 假设 df 是已经加载的 DataFrame
# df = pd.read_csv('your_file.csv') # 如果需要从 CSV 文件加载

# 筛选 flow_id 为 'verhaltenraktivierung-3'
filtered_df = df[df['flow_id'] == 'verhaltensaktivierung-2']

# 筛选 content_type 为 'text' 或 'question'
# 如果 'payload' 是要筛选的 content_type 之一，请取消下一行的注释
filtered_df = filtered_df[filtered_df['content_type'].isin(['text', 'question',
```

In [7]:
```
filtered_df.head()
```

Out[7]:

	id	conversation_id	flow_id	step_id	direction	
59	b416224d-57f9-46b3-9fe2-8a4c3d5245c1	fb0749b3-3391-4f35-9ad6-958a1b9a931c	verhaltensaktivierung-2	start	SEND	
61	93988c30-1843-40f8-b6bf-790689a17b1e	fb0749b3-3391-4f35-9ad6-958a1b9a931c	verhaltensaktivierung-2	start	RECEIVE	{"payl
63	89fa05bc-116d-4471-a3de-a2091026f7cd	fb0749b3-3391-4f35-9ad6-958a1b9a931c	verhaltensaktivierung-2	start	RECEIVE	
64	a478b7d3-ab3d-484b-94b3-18d379838215	fb0749b3-3391-4f35-9ad6-958a1b9a931c	verhaltensaktivierung-2	start	SEND	{"
65	b2625dd7-2a92-4b5a-9c8f-576c79ceb85d	fb0749b3-3391-4f35-9ad6-958a1b9a931c	verhaltensaktivierung-2	start	SEND	

Selected Dataset Generation

All downstream tasks based on this dataset: sorted_df.

In [8]:
```
# 将 created_at 列转换为 datetime 类型
filtered_df['created_at'] = pd.to_datetime(filtered_df['created_at'])
# 根据 created_at 递增排序
```

```
sorted_df = filtered_df.sort_values(by='created_at')
sorted_df.head()
```

Out[8]:

	id	conversation_id	flow_id	step_id	direction	
59	b416224d-57f9-46b3-9fe2-8a4c3d5245c1	fb0749b3-3391-4f35-9ad6-958a1b9a931c	verhaltensaktivierung-2	start	SEND	
61	93988c30-1843-40f8-b6bf-790689a17b1e	fb0749b3-3391-4f35-9ad6-958a1b9a931c	verhaltensaktivierung-2	start	RECEIVE	{"payl
68	362cd471-2cb8-4a8b-95cc-48c60cbf971b	fb0749b3-3391-4f35-9ad6-958a1b9a931c	verhaltensaktivierung-2	start	SEND	{
63	89fa05bc-116d-4471-a3de-a2091026f7cd	fb0749b3-3391-4f35-9ad6-958a1b9a931c	verhaltensaktivierung-2	start	RECEIVE	
64	a478b7d3-ab3d-484b-94b3-18d379838215	fb0749b3-3391-4f35-9ad6-958a1b9a931c	verhaltensaktivierung-2	start	SEND	{"

In [9]:
```
sorted_df.payload.head()
```

Out[9]:
```
59    {"content":{"text":"Lass uns mal versuchen, ei...
61    {"content":{"payload":"👍"},"content_type":"pay...
68    {"content":{"buttons":[{"content":{"accepts":[...
63    {"content":{"payload":"Geht eigentlich"},"cont...
64    {"content":{"text":"Aber da bist du nicht alle...
Name: payload, dtype: object
```

In [10]:
```
print("sorted_df count: ", len(sorted_df),)
```

```
sorted_df count:  8991
```

In [11]:
```
print("unique conversation_id",len(sorted_df['conversation_id'].unique()))
```

```
unique conversation_id 119
```

JSONL Transformation

Transfer the selected dataset to extract conversations into a format that can be used by the LLM as OpenAI has provided a JSONL example: {"messages": [{"role": "system", "content": "Marv is a factual chatbot that is also sarcastic."}, {"role": "user", "content": "What's the capital of France?"}, {"role": "assistant", "content": "Paris, as if everyone doesn't know that already."}]} {"messages": [{"role": "system", "content": "Marv is a factual chatbot that is also sarcastic."}, {"role": "user", "content": "Who wrote 'Romeo and Juliet'?"}, {"role": "assistant", "content": "Oh, just some guy named William Shakespeare. Ever heard of him?"}]} {"messages": [{"role": "system", "content": "Marv is a factual chatbot that is also sarcastic."},

{"role": "user", "content": "How far is the Moon from Earth?"}, {"role": "assistant", "content": "Around 384,400 kilometers. Give or take a few, like that really matters."}]} Source: https://platform.openai.com/docs/guides/fine-tuning/preparing-your-dataset

In [12]:
```
import json

# 映射 direction 到 role
role_mapping = {'SEND': 'assistant', 'RECEIVE': 'user'}
sorted_df['role'] = sorted_df['direction'].map(role_mapping)

# 函数，用于从 payload 中提取内容
def extract_content(payload):
    try:
        payload_json = json.loads(payload)
        if 'content' in payload_json and 'title' in payload_json['content']:
            return payload_json['content']['title']
        if 'payload' in payload_json['content']:
            return payload_json['content']['payload']
        elif 'text' in payload_json['content']:
            return payload_json['content']['text']
    except json.JSONDecodeError:
        return payload
    return '内容不可提取'

# 应用这个函数到 payload 列
sorted_df['content'] = sorted_df['payload'].apply(extract_content)

# 用于将单个对话转换为 JSON 的函数
def conversation_to_json(group):
    # 添加固定的系统消息
    system_message = {"role": "system", "content": "You are a helpful chatbot th
    messages = [system_message] + group[['role', 'content']].to_dict(orient='rec
    return {'messages': messages}

# 按 conversation_id 分组并转换每个组
conversations_json = sorted_df.groupby('conversation_id').apply(conversation_to_

# 指定保存 JSONL 文件的路径
output_file_path = r"C:\Users\Li\Desktop\Engineering Seminar Human-Centered Syst

# 将每个对话写入 JSONL 文件
with open(output_file_path, 'w', encoding='utf-8') as file:
    for conversation in conversations_json:
        json.dump(conversation, file, ensure_ascii=False)
        file.write('\n')

# 输出文件的路径
output_file_path
```

Out[12]:
```
'C:\\Users\\Li\\Desktop\\Engineering Seminar Human-Centered Systems\\data\\conv
erted_messages.jsonl'
```

Data analysis for chat model fine-tuning

Supported by Data preparation and analysis for chat model fine-tuning: https://cookbook.openai.com/examples/chat_finetuning_data_prep

In [13]:
```python
import tiktoken # for token counting
import numpy as np
from collections import defaultdict
```

In [14]:
```python
data_path = r"C:\Users\Li\Desktop\Engineering Seminar Human-Centered Systems\dat

# Load the dataset
with open(data_path, 'r', encoding='utf-8') as f:
    dataset = [json.loads(line) for line in f]

# Initial dataset stats
print("Num examples:", len(dataset))
print("First example:")
for message in dataset[0]["messages"]:
    print(message)
```

Num examples: 119
First example:
{'role': 'system', 'content': 'You are a helpful chatbot that based on Behavioura
l activation treatment.'}
{'role': 'assistant', 'content': 'Lass uns mal versuchen, ein paar Aktivitäten zu
finden, die dir Spaß machen! 👐'}
{'role': 'user', 'content': '👍'}
{'role': 'assistant', 'content': 'Fällt es dir schwer, eine positive Aktivität in
deinen Alltag einzubauen?'}
{'role': 'user', 'content': 'Geht eigentlich'}
{'role': 'assistant', 'content': 'Aber da bist du nicht allein! Viele haben genug
für die Schule, Universität oder Arbeit 👴 💼 zu tun und nur wenig Freizeit.'}
{'role': 'assistant', 'content': 'Aber: Positive Aktivitäten müssen keine große S
ache sein!'}
{'role': 'assistant', 'content': 'Manchmal kann es schon helfen, wenn du duschen
gehst und dich danach frisch fühlst. 🚿'}
{'role': 'assistant', 'content': 'Auch solche kleinen Aktivitäten können helfen,
deine Stimmung zu verbessern und auf bessere Gedanken zu kommen.'}
{'role': 'user', 'content': 'Ich verstehe. 👌'}
{'role': 'assistant', 'content': 'Bestimmt kennst du ein paar Aktivitäten, bei de
nen du sagst, ...'}
{'role': 'assistant', 'content': '... das macht mir Spaß. 🎈 Dabei habe ich gute
Laune. Dabei kann ich mich entspannen. 💆'}
{'role': 'assistant', 'content': 'Fällt dir was ein oder soll ich dir Beispiele z
eigen?'}
{'role': 'assistant', 'content': 'Lass uns doch mal 3 Aktivitäten sammeln, die di
r im Alltag Freude machen könnten. Okay?'}
{'role': 'assistant', 'content': 'Was mach dir im Alltag Freude? Was ist die erst
e Aktivität, die dir einfällt?'}
{'role': 'user', 'content': 'Ich weiß was, lass uns starten!'}
{'role': 'assistant', 'content': 'Super! Dann los! 🚀'}
{'role': 'user', 'content': 'Sport'}
{'role': 'assistant', 'content': 'Danke. Und was ist die Zweite?'}
{'role': 'user', 'content': 'Freunde treffen'}
{'role': 'assistant', 'content': '👍 Hast du noch eine dritte Aktivität für mic
h?'}
{'role': 'user', 'content': 'Kochen'}
{'role': 'assistant', 'content': 'Wohoo! 🥳 Geschafft! Jetzt haben wir 3 Aktivit
äten, die dir Freude machen.'}
{'role': 'assistant', 'content': 'Denke daran: Nur wenn du etwas machst, dann wir
d es dir langsam besser gehen!'}
{'role': 'assistant', 'content': 'Damit du die schönen Aktivitäten auch wirklich
umsetzt, planen wir die jetzt einfach gemeinsam!'}
{'role': 'assistant', 'content': 'Kennst du das auch von dir?'}
{'role': 'assistant', 'content': 'Etwas zu planen ergibt Sinn, da viele junge Men
schen gestresst sind.'}
{'role': 'user', 'content': 'Okay 👍'}
{'role': 'assistant', 'content': 'Denn du hast was, auf das du dich freuen kanns
t. 🎉'}
{'role': 'user', 'content': 'Ja, das kenne ich'}
{'role': 'assistant', 'content': 'Wenn du nämlich positive Aktivitäten planst, bi
st du gleich etwas entspannter!'}
{'role': 'assistant', 'content': 'Pass dabei aber auf andere Termine auf. Die kön
nen wir ja nicht verändern. 🗓️'}
{'role': 'assistant', 'content': 'An sehr stressigen Tagen können die schönen Akt
ivitäten auch kurz sein: So zwischen 30 Sekunden und 15 Minuten.'}
{'role': 'user', 'content': '👌 Klingt gut.'}
{'role': 'assistant', 'content': 'Lass uns das mal ausprobieren! Wir planen mal A
ktivitäten, die dir Spaß machen!'}
{'role': 'assistant', 'content': 'Du musst dir für die schönen Aktivitäten einen

echten Termin machen, dann ist es gleich viel einfacher...'}
{'role': 'assistant', 'content': 'Okay?'}
{'role': 'assistant', 'content': 'Merke dir: Du musst wissen, 1. was du machen willst und 2. wann du es machen willst.'}
{'role': 'user', 'content': 'Okay'}
{'role': 'user', 'content': '👍'}
{'role': 'assistant', 'content': 'Erinnere dich nochmal an die 3 Aktivitäten, die du vorhin genannt hast: 1. Sport, 2. Freunde treffen und 3. Kochen.'}
{'role': 'assistant', 'content': 'Welche der 3 möchtest du am liebsten mit mir planen?'}
{'role': 'assistant', 'content': 'Eine Aktivität davon würde ich gerne genau mit dir planen. Eine reicht erstmal.'}
{'role': 'user', 'content': 'Sport'}
{'role': 'assistant', 'content': 'Super! Jetzt wissen wir: WAS du machen willst.'}
{'role': 'assistant', 'content': 'Dann lass uns schauen WANN!'}
{'role': 'assistant', 'content': 'Wie viel Zeit ungefähr brauchst du für diese Aktivität? Falls du dafür irgendwo hinfahren musst, plane auch An- und Abfahrt ein!'}
{'role': 'assistant', 'content': 'Okay, wie lange brauchst du?'}
{'role': 'assistant', 'content': 'Wann passt es für dich? An welchem Tag und zu welcher Uhrzeit?'}
{'role': 'assistant', 'content': 'An welchem Tag hast du denn ausreichend Zeit, dass du diese Aktivität machen kannst? 🗓️'}
{'role': 'user', 'content': '2 h'}
{'role': 'user', 'content': 'Dienstag 10:00'}
{'role': 'assistant', 'content': 'Perfekt!'}
{'role': 'assistant', 'content': 'Jetzt haben wir: 1. WAS du machen willst, 2. WANN du es machen willst.'}
{'role': 'assistant', 'content': 'Eine Sache muss ich aber noch unbedingt von dir wissen! 🙄'}
{'role': 'user', 'content': '👍😎'}
{'role': 'user', 'content': 'Welche denn?'}
{'role': 'assistant', 'content': 'Was könnte dich davon abhalten, die Aktivität zu machen? Also obwohl du es ja fest geplant hast!'}
{'role': 'assistant', 'content': 'Oft hilft es, wenn du dir vorher überlegst, was dazwischen kommen könnte. Damit du vorbereitet bist, wenn das wirklich passiert.'}
{'role': 'assistant', 'content': 'Und, was könnte bei dir dazwischen kommen?'}
{'role': 'user', 'content': 'Nichts eigentlich wenn ich es mir fest vornehme'}
{'role': 'assistant', 'content': 'Verstehe!'}
{'role': 'assistant', 'content': 'Und wie kannst du verhindern, dass du von deiner Aktivität abgehalten wirst? 💡 Tipp: Erinnerungen im Kalender oder Unterstützung durch Freunde können auch helfen.'}
{'role': 'assistant', 'content': 'Lass mich das noch mal zusammenfassen, okay?'}
{'role': 'user', 'content': 'Ok'}
{'role': 'assistant', 'content': 'Sehr gut! Jetzt haben wir schon mal einen Plan. 👍'}
{'role': 'user', 'content': 'Gern!'}
{'role': 'assistant', 'content': 'Wenn du etwas Schönes machen möchtest, gehst du so vor:'}
{'role': 'assistant', 'content': '1. Überlege, WAS du machen willst! 2. Überlege, WANN du es machen willst! 3. Trage es dir als Termin in deinen Kalender ein!'}
{'role': 'assistant', 'content': 'und'}
{'role': 'assistant', 'content': '4. Überlege, was dich davon ABHALTEN könnte! 5. Und, was du DAGEGEN tun kannst.'}
{'role': 'assistant', 'content': 'In diesen Schritten kannst du es ab jetzt immer angehen. Das hilft dir, die schönen Aktivitäten auch wirklich zu machen.'}
{'role': 'assistant', 'content': 'Versuch in den nächsten Tagen mal, diese Methode anzuwenden. 🙏'}

```
{'role': 'user', 'content': '👍'}
{'role': 'assistant', 'content': 'Okay, das war doch schon mal sehr gut! Ich bin
stolz auf dich! 😈'}
```

In [15]:
```python
# Format error checks
format_errors = defaultdict(int)

# 新增一个列表来记录错误的例子的索引
missing_assistant_examples = []

for i, ex in enumerate(dataset):
    if not isinstance(ex, dict):
        format_errors["data_type"] += 1
        continue

    messages = ex.get("messages", None)
    if not messages:
        format_errors["missing_messages_list"] += 1
        continue

    for message in messages:
        if "role" not in message or "content" not in message:
            format_errors["message_missing_key"] += 1

        if any(k not in ("role", "content", "name", "function_call") for k in me
            format_errors["message_unrecognized_key"] += 1

        if message.get("role", None) not in ("system", "user", "assistant", "fun
            format_errors["unrecognized_role"] += 1

        content = message.get("content", None)
        function_call = message.get("function_call", None)

        if (not content and not function_call) or not isinstance(content, str):
            format_errors["missing_content"] += 1

    if not any(message.get("role", None) == "assistant" for message in messages)
        format_errors["example_missing_assistant_message"] += 1
        missing_assistant_examples.append(i)  # 记录发生错误的例子的索引

if format_errors:
    print("Found errors:")
    for k, v in format_errors.items():
        print(f"{k}: {v}")
    if missing_assistant_examples:
        print("Missing assistant messages in examples:", missing_assistant_examp
else:
    print("No errors found")
```

```
No errors found
```

In [16]:
```python
encoding = tiktoken.get_encoding("cl100k_base")

# not exact!
# simplified from https://github.com/openai/openai-cookbook/blob/main/examples/H
def num_tokens_from_messages(messages, tokens_per_message=3, tokens_per_name=1):
    num_tokens = 0
    for message in messages:
        num_tokens += tokens_per_message
        for key, value in message.items():
```

```
            num_tokens += len(encoding.encode(value))
            if key == "name":
                num_tokens += tokens_per_name
    num_tokens += 3
    return num_tokens

def num_assistant_tokens_from_messages(messages):
    num_tokens = 0
    for message in messages:
        if message["role"] == "assistant":
            num_tokens += len(encoding.encode(message["content"]))
    return num_tokens

def print_distribution(values, name):
    print(f"\n#### Distribution of {name}:")
    print(f"min / max: {min(values)}, {max(values)}")
    print(f"mean / median: {np.mean(values)}, {np.median(values)}")
    print(f"p5 / p95: {np.quantile(values, 0.1)}, {np.quantile(values, 0.9)}")
```

In [17]:

```
# Warnings and tokens counts
n_missing_system = 0
n_missing_user = 0
n_messages = []
convo_lens = []
assistant_message_lens = []

for ex in dataset:
    messages = ex["messages"]
    if not any(message["role"] == "system" for message in messages):
        n_missing_system += 1
    if not any(message["role"] == "user" for message in messages):
        n_missing_user += 1
    n_messages.append(len(messages))
    convo_lens.append(num_tokens_from_messages(messages))
    assistant_message_lens.append(num_assistant_tokens_from_messages(messages))

print("Num examples missing system message:", n_missing_system)
print("Num examples missing user message:", n_missing_user)
print_distribution(n_messages, "num_messages_per_example")
print_distribution(convo_lens, "num_total_tokens_per_example")
print_distribution(assistant_message_lens, "num_assistant_tokens_per_example")
n_too_long = sum(l > 4096 for l in convo_lens)
print(f"\n{n_too_long} examples may be over the 4096 token limit, they will be t
```

```
Num examples missing system message: 0
Num examples missing user message: 1

#### Distribution of num_messages_per_example:
min / max: 2, 83
mean / median: 76.5546218487395, 75.0
p5 / p95: 75.0, 83.0

#### Distribution of num_total_tokens_per_example:
min / max: 50, 1790
mean / median: 1565.2689075630253, 1548.0
p5 / p95: 1508.8, 1707.4

#### Distribution of num_assistant_tokens_per_example:
min / max: 25, 1261
mean / median: 1142.5126050420167, 1120.0
p5 / p95: 1109.0, 1247.0

0 examples may be over the 4096 token limit, they will be truncated during fine-t
uning
```

In [18]:
```python
# Pricing and default n_epochs estimate
MAX_TOKENS_PER_EXAMPLE = 4096

TARGET_EPOCHS = 3
MIN_TARGET_EXAMPLES = 100
MAX_TARGET_EXAMPLES = 25000
MIN_DEFAULT_EPOCHS = 1
MAX_DEFAULT_EPOCHS = 25

n_epochs = TARGET_EPOCHS
n_train_examples = len(dataset)
if n_train_examples * TARGET_EPOCHS < MIN_TARGET_EXAMPLES:
    n_epochs = min(MAX_DEFAULT_EPOCHS, MIN_TARGET_EXAMPLES // n_train_examples)
elif n_train_examples * TARGET_EPOCHS > MAX_TARGET_EXAMPLES:
    n_epochs = max(MIN_DEFAULT_EPOCHS, MAX_TARGET_EXAMPLES // n_train_examples)

n_billing_tokens_in_dataset = sum(min(MAX_TOKENS_PER_EXAMPLE, length) for length
print(f"Dataset has ~{n_billing_tokens_in_dataset} tokens that will be charged f
print(f"By default, you'll train for {n_epochs} epochs on this dataset")
print(f"By default, you'll be charged for ~{n_epochs * n_billing_tokens_in_datas
```

```
Dataset has ~186267 tokens that will be charged for during training
By default, you'll train for 3 epochs on this dataset
By default, you'll be charged for ~558801 tokens
```

3.Design Chatbot

Behavioral Activation is a method for a psychical therapy by: Taking part in psychologically beneficial activities, Keeping away from psychologically harmful activities and Solving mechanisms problems that hinder access to rewards or enhance negative control.

What should Chatbot do?

1.BA Introduction: Explain BA understandably at first and chatbot makes a self-introduction. 2.Mood Track: Ask user's emotion today. 3.Activity

Recommendation: Find activities that user likes and encourage the user to take part in them. 4.Activity Management : Schedule (PST or ICS files if possible) the activities and check in. 5.Incentive mechanism: prevent users from not doing the activity. 1).Public Declaration: encourage users to share their goals and activities publicly, such as on twitter. The social pressure and potential for public accountability can be a strong incentive. 2).Partner Supervision: encourage users to share their goals and activities to their friends and family so that they can supervise users' activities and try to prevent users from avoiding participating in activities. 3).Schedule Check-in history: remind users to check their finished und uncompleted schedule. 4).Compliment and Praises: encourage users when they complete activities and remind users to remember and share the joy of successful completion of schedules.

Knowledge Hub loading

Knowledge Hub contains relevant knowledge of BA.

In [19]:
```
file_path = r'C:\Users\Li\Desktop\Engineering Seminar Human-Centered Systems\dat
knowledge_df = pd.read_csv(file_path, sep='\\|\\|', engine='python',encoding='UT
knowledge_df
```

Out[19]:

	entities	descriptions
0	Zest	Great enthusiasm and energy, often marked by ...
1	Zenith	A feeling of being at the peak or highest poi...
2	Yearning	A deep longing, especially for something or s...
3	Wonder	A feeling of amazement and admiration, caused...
4	Wistfulness	A feeling of vague or regretful longing, ofte...
...	...	...
96	Activity Recommendation	Find activities that user likes and encourage...
97	add future schedule	add an unfinished activity
98	add finished schedule	add an unfinished activity
99	update schedule	update an activity
100	delete schedule	delete an activity

101 rows × 2 columns

Schedule List loading

User Information such as Schedule and User Mood.

In [20]:
```
file_path = r'C:\Users\Li\Desktop\Engineering Seminar Human-Centered Systems\dat
schedule_df = pd.read_csv(file_path, sep='\\|\\|', engine='python',encoding='UTF
schedule_df
```

Out[20]:

	entities	descriptions
0	Zest	Great enthusiasm and energy, often marked by ...

5 Basis Functions for the Knowledge Hub: sort_dataframe,add_sort_entity,delete_matching_entity,sear

In [21]:
```
def sort_dataframe(df):
    # 根据 entities 列的首个单词进行排序
    df.sort_values(by='entities', key=lambda x: x.str.split().str[0], inplace=Tr

    return df
```

In [22]:
```
# 示例使用
# 假设 df 是一个包含 entities 和 descriptions 列的 pandas DataFrame

# 排序
df = sort_dataframe(knowledge_df)
df
```

Out[22]:

	entities	descriptions
64	"Exuberance	The quality of being full of energy, exciteme...
61	"Foreboding	A feeling that something bad will happen; fea...
49	"Insecurity	A feeling of uncertainty or anxiety about one...
29	"Pessimism	A tendency to see the worst aspect of things ...
96	Activity Recommendation	Find activities that user likes and encourage...
...	...	...
0	Zest	Great enthusiasm and energy, often marked by ...
97	add future schedule	add an unfinished activity
98	add finished schedule	add an unfinished activity
100	delete schedule	delete an activity
99	update schedule	update an activity

101 rows × 2 columns

In [23]:
```
def add_sort_entity(df, new_entity, new_description):
    # 创建新行
    new_row = pd.DataFrame({'entities': [new_entity], 'descriptions': [new_descr

    # 使用 pd.concat 添加新行到 DataFrame
    df = pd.concat([df, new_row], ignore_index=True)

    # 调用 sort_dataframe 函数进行排序
    df = sort_dataframe(df)

    return df
```

In [24]:
```
df=add_sort_entity(schedule_df,"entity1", "description1")
df
```

Out[24]:

	entities	descriptions
0	Zest	Great enthusiasm and energy, often marked by ...
1	entity1	description1

In [25]:
```
df=add_sort_entity(df,"entity3", "description3")
df
```

Out[25]:

	entities	descriptions
0	Zest	Great enthusiasm and energy, often marked by ...
1	entity1	description1
2	entity3	description3

In [26]:
```
def delete_matching_entity(df, entity_to_delete):
    # 找到第一个匹配的 entity 的索引
    index_to_delete = df[df['entities'] == entity_to_delete].index.min()

    # 如果找到匹配的 entity，则删除对应的行
    if pd.notna(index_to_delete):
        df = df.drop(index_to_delete)

    return df
```

In [27]:
```
df=delete_matching_entity(df,"entity1")
df
```

Out[27]:

	entities	descriptions
0	Zest	Great enthusiasm and energy, often marked by ...
2	entity3	description3

In [28]:
```
def search_description(df, entity_to_search):
    # 去除额外的空格
    cleaned_search = entity_to_search.strip()

    # 使用更灵活的匹配方法
    matching_rows = df[df['entities'].str.contains(cleaned_search, case=False, n

    # 检查是否找到匹配的行
    if not matching_rows.empty:
        # 返回第一个匹配项的 description 值
        return matching_rows.iloc[0]['descriptions']
    else:
        return None
```

In [29]:
```
search_description(df,"Behavioral Activation")
```

In [30]:
```
def update_entity_description(df, original_entity, updated_entity, updated_descr
    # 查找与原始 entity 和 description 匹配的行
```

```
    for index, row in df.iterrows():
        if row['entities'] == original_entity :
            df.at[index, 'entities'] = updated_entity
            df.at[index, 'descriptions'] = updated_description
            break  # 假设只更新第一个匹配的行
    else:
        print("No matching row found to update")

    return df
```

In [31]:
```
df=update_entity_description(df,"entity3","entity4","description4")
df
```

Out[31]:

	entities	descriptions
0	Zest	Great enthusiasm and energy, often marked by ...
2	entity4	description4

In [32]:
```
print(search_description(df,"entity4"))
```

```
description4
```

In [33]:
```
df=delete_matching_entity(df,"entity4")
df
```

Out[33]:

	entities	descriptions
0	Zest	Great enthusiasm and energy, often marked by ...

Why we need RAG?

1. Update data in real time (Schedules, Mood detection)
2. Search support
3. Generate more accurate answers instead of making up confusing answers

Chunk Processing

Relevent Chunk

In [34]:
```
from sentence_transformers import SentenceTransformer
import faiss
import numpy as np
import pandas as pd

def normalize_embeddings(embeddings):
    """
    标准化嵌入向量，使其成为单位向量。
    """
    norms = np.linalg.norm(embeddings, axis=1, keepdims=True)
    return embeddings / norms

def extract_knowledges_from_df(df: pd.DataFrame, question, similarity_threshold=
    """
    使用DataFrame中的entities列进行查找，返回一个DataFrame，其中包含与单个问题最相
    基于指定的相似度阈值。
    """
```

```
    model_name='all-MiniLM-L6-v2'
    # 确保entities列存在
    if 'entities' not in df.columns:
        raise ValueError("DataFrame must have an 'entities' column")

    # 确保descriptions列存在
    if 'descriptions' not in df.columns:
        raise ValueError("DataFrame must have a 'descriptions' column")

    # 初始化SentenceTransformer模型
    model = SentenceTransformer(model_name)

    # 获取entities列的值
    entities = df['entities'].tolist()

    # 计算实体的嵌入向量
    entity_embeddings = model.encode(entities, convert_to_tensor=False)
    entity_embeddings = normalize_embeddings(entity_embeddings)

    # 创建faiss索引（使用内积来模拟余弦相似度）
    index = faiss.IndexFlatIP(entity_embeddings.shape[1])
    index.add(entity_embeddings)

    # 计算问题的嵌入向量并标准化
    question_embedding = model.encode(question, convert_to_tensor=False)
    question_embedding = normalize_embeddings(question_embedding.reshape(1, -1))

    # 搜索与问题相似度高于阈值的所有实体
    distances, indices = index.search(question_embedding, len(entities))

    # 过滤出相似度大于等于阈值的实体
    filtered_indices = [index for index, distance in zip(indices[0], distances[0

     # 获取并返回相似度最高的前三个实体所在的行
    top_indices = filtered_indices[:3]  # 选择相似度最高的前三个索引
    result_df = df.iloc[top_indices]

    return result_df[['entities', 'descriptions']]
```

Schedule Management

In [35]:

```
from datetime import datetime
def add_finished_schedule(df, new_entity, userinput):
    """
    Add a new schedule entry to the DataFrame.
    The new entity will have a prefix and a timestamp.
    """
    # Generate the new entity value with timestamp
    timestamp = datetime.now().strftime("%Y-%m-%d")
    full_entity = f"schedule finished {new_entity} {timestamp}"

    # Add and sort the new entry
    return add_sort_entity(df, full_entity, userinput)
```

In [36]:

```
from datetime import datetime
def add_unfinished_schedule(df, new_entity, userinput):
    """
    Add a new schedule entry to the DataFrame.
```

```
    The new entity will have a prefix and a timestamp.
    """
    # Generate the new entity value with timestamp
    timestamp = datetime.now().strftime("%Y-%m-%d")
    full_entity = f"schedule unfinished {new_entity} {timestamp}"

    # Add and sort the new entry
    return add_sort_entity(df, full_entity, userinput)
```

In [37]:
```
def update_schedule(df,new_entity, userinput):
    schedule_df = extract_knowledges_from_df(df,userinput,0.4)[:1]
    original_entity = schedule_df.iloc[0]['entities']
    timestamp = datetime.now().strftime("%Y-%m-%d")
    updated_entity = f"schedule unfinished {new_entity} {timestamp}"
    return update_entity_description(df,original_entity=original_entity,updated_
```

In [38]:
```
def delete_schedule(df, userinput):
    entity_to_delete_df = extract_knowledges_from_df(df,userinput,0.4)[:1]
    print("entity_to_delete_df",entity_to_delete_df)
    entity_to_delete = entity_to_delete_df.iloc[0]['entities']
    print("entity_to_delete",entity_to_delete)
    timestamp = datetime.now().strftime("%Y-%m-%d")
    deleted_entity = f"schedule unfinished {entity_to_delete} {timestamp}"
    print("deleted_entity",deleted_entity)
    return delete_matching_entity(df,deleted_entity)
```

In [39]:
```
schedule_df = add_unfinished_schedule(schedule_df,"Swimming","I want to swim")
schedule_df
```

Out[39]:

	entities	descriptions
0	Zest	Great enthusiasm and energy, often marked by ...
1	schedule unfinished Swimming 2024-01-02	I want to swim

In [40]:
```
schedule_df = add_finished_schedule(schedule_df,"Running","I have finished runni
schedule_df
```

Out[40]:

	entities	descriptions
0	Zest	Great enthusiasm and energy, often marked by ...
1	schedule unfinished Swimming 2024-01-02	I want to swim
2	schedule finished Running 2024-01-02	I have finished running

In [41]:
```
schedule_df = delete_schedule(schedule_df,"Swimming")
schedule_df
```

```
entity_to_delete_df                                    entities    descriptions
1  schedule unfinished Swimming 2024-01-02  I want to swim
entity_to_delete schedule unfinished Swimming 2024-01-02
deleted_entity schedule unfinished schedule unfinished Swimming 2024-01-02 2024-0
1-02
```

Out[41]:

	entities	descriptions
0	Zest	Great enthusiasm and energy, often marked by ...
1	schedule unfinished Swimming 2024-01-02	I want to swim
2	schedule finished Running 2024-01-02	I have finished running

In [42]:
```
extract_knowledges_from_df(schedule_df,"What activity have I not finished?",0.3)
```

Out[42]:

	entities	descriptions
1	schedule unfinished Swimming 2024-01-02	I want to swim
2	schedule finished Running 2024-01-02	I have finished running

In [43]:
```
extract_knowledges_from_df(schedule_df,"What activity have I finished?",0.3)
```

Out[43]:

	entities	descriptions
1	schedule unfinished Swimming 2024-01-02	I want to swim
2	schedule finished Running 2024-01-02	I have finished running

In [44]:
```
extract_knowledges_from_df(schedule_df,"Do I swim today or yesterday?",0.1)
```

Out[44]:

	entities	descriptions
1	schedule unfinished Swimming 2024-01-02	I want to swim
2	schedule finished Running 2024-01-02	I have finished running

In [45]:
```
extract_knowledges_from_df(schedule_df,"What is BA",0.1)
```

Out[45]:

	entities	descriptions

In [46]:
```
extract_knowledges_from_df(schedule_df,"what should I do today?",0.1)
```

Out[46]:

	entities	descriptions
1	schedule unfinished Swimming 2024-01-02	I want to swim

Transfer Dataframe to string to construct prompt

In [47]:
```
def dataframe_to_string(df):
    """
    将DataFrame的每行转换为字符串，并在每行结束后加句号。
```

```
    """
    # 将DataFrame的每行转换为由空格分隔的字符串，并添加句号
    lines = [' '.join(map(str, row)) + '.' for row in df.itertuples(index=False,

    # 将所有行连接成一个单一的字符串
    context =  ' '.join(lines)
    return " Given retrievaled context: "+context
```

In [48]:
```
dataframe_to_string(extract_knowledges_from_df(schedule_df,"what should I do tod
```

Out[48]:
```
' Given retrievaled context: schedule unfinished Swimming 2024-01-02 I want to
swim.'
```

Timestamp for Prompt

In [49]:
```
from datetime import datetime

# Function to get current timestamp
def get_current_timestamp():
    return "Today is " + datetime.now().strftime("%Y-%m-%d %H:%M:%S")
```

In [50]:
```
get_current_timestamp()
```

Out[50]:
```
'Today is 2024-01-02 23:06:59'
```

Save messages to the specified folder

folder_path = r"folder_path" example: save_messages_to_txt(messages, folder_path)

In [51]:
```
import os
import json
from datetime import datetime
folder_path = r"C:\Users\Li\Desktop\Engineering Seminar Human-Centered Systems"
def save_messages_to_jsonl(messages, folder_path):
    # 创建一个基于当前时间戳的文件名
    timestamp = datetime.now().strftime("%Y%m%d%H%M%S")
    filename = f"{timestamp}.jsonl"

    # 确保文件夹路径存在
    if not os.path.exists(folder_path):
        os.makedirs(folder_path)

    # 创建完整的文件路径
    file_path = os.path.join(folder_path, filename)

    # 将messages保存到JSONL文件
    with open(file_path, 'w', encoding='utf-8') as file:
        for message in messages:
            # 将字典转换为JSON字符串并写入文件
            json_str = json.dumps(message)
            file.write(json_str + '\n')

    print(f"Messages saved to {file_path}")
```

Single Answered Method

Use get_answer_with_single_question to modify other functions like: Mood track

In [52]:
```python
import os
from openai import OpenAI
import openai

def get_answer_with_single_question(df, question, similarity_threshold = 0.1):

    # 提取相关信息作为上下文
    context = dataframe_to_string(extract_knowledges_from_df(df,question,similar

    # 设置OpenAI API密钥
    api_key = os.environ.get('OPENAI_API_KEY')
    openai.api_key = api_key

    # 初始化OpenAI客户端
    client = OpenAI(api_key=api_key)

    # 设置模型
    model="gpt-3.5-turbo-1106"

    try:
        messages=[
                    {"role": "system", "content": "You are a helpful assistant.
                    # instruction
                    {"role": "user", "content": question + " " + context}
        ]
        response = client.chat.completions.create(
            model=model,
            messages=messages
        )

        # 提取并返回答案
        answer = response.choices[0].message.content
        return answer

    except Exception as e:
        print(f"处理问题时发生错误: {e}")
        return "无法获取答案"
```

In [53]:
```python
get_answer_with_single_question(df,"what should I do today?")
```

Out[53]:
```
'Exploring a nearby park.'
```

Mood Track

In [54]:
```python
messages= [{"role": "system", "content": "You are a helpful chatbot that based o
```

In [55]:
```python
str(messages[-5:])
```

Out[55]:
```
"[{'role': 'assistant', 'content': '4. Überlege, was dich davon ABHALTEN könnt
e! 5. Und, was du DAGEGEN tun kannst.'}, {'role': 'assistant', 'content': 'In d
iesen Schritten kannst du es ab jetzt immer angehen. Das hilft dir, die schönen
Aktivitäten auch wirklich zu machen.'}, {'role': 'assistant', 'content': 'Versu
ch in den nächsten Tagen mal, diese Methode anzuwenden. 🙏'}, {'role': 'user',
'content': '👍'}, {'role': 'assistant', 'content': 'Okay, das war doch schon ma
l sehr gut! Ich bin stolz auf dich! 😻'}]"
```

In [56]:
```
def mood_track(df, similarity_threshold = 0.1):
    history = str(messages[-5:])
    mood_track = get_answer_with_single_question(df,"what is user's emotion ? gi
    #print("mode_track: ",mood_track)
    timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    full_entity = f"user emotion {timestamp}"
    new_description = mood_track
    # Add and sort the new entry
    df = add_sort_entity(df, full_entity, new_description)
    #print("entity: ",full_entity)
    #print("description: ",new_description)
    return full_entity + ", " + new_description
```

In [57]:
```
mood_track(df,0.1)
```

Out[57]:
```
'user emotion 2024-01-02 23:07:01, pride'
```

In [58]:
```
search_description(df,mood_track(df)[0])
```

In [59]:
```
def dataframe_to_entity(df):
    """
    将DataFrame的每行转换为字符串，并在每行结束后加句号。
    """
    # 将DataFrame的每行转换为由空格分隔的字符串，并添加句号
    lines = [' '.join(map(str, row)) + '.' for row in df.itertuples(index=False,

    # 将所有行连接成一个单一的字符串
    entity =  ' '.join(lines)
    return str(entity)
```

In [60]:
```
def extract_activity_from_df(df: pd.DataFrame, question, similarity_threshold=0.
    """
    使用DataFrame中的entities列进行查找，返回一个DataFrame，其中包含与单个问题最相
    基于指定的相似度阈值。
    """
    model_name='all-MiniLM-L6-v2'
    # 确保entities列存在
    if 'entities' not in df.columns:
        raise ValueError("DataFrame must have an 'entities' column")

    # 确保descriptions列存在
    if 'descriptions' not in df.columns:
        raise ValueError("DataFrame must have a 'descriptions' column")

    # 初始化SentenceTransformer模型
    model = SentenceTransformer(model_name)

    # 获取entities列的值
    entities = df['entities'].tolist()

    # 计算实体的嵌入向量
    entity_embeddings = model.encode(entities, convert_to_tensor=False)
    entity_embeddings = normalize_embeddings(entity_embeddings)

    # 创建faiss索引（使用内积来模拟余弦相似度）
    index = faiss.IndexFlatIP(entity_embeddings.shape[1])
    index.add(entity_embeddings)
```

```
    # 计算问题的嵌入向量并标准化
    question_embedding = model.encode(question, convert_to_tensor=False)
    question_embedding = normalize_embeddings(question_embedding.reshape(1, -1))

    # 搜索与问题相似度高于阈值的所有实体
    distances, indices = index.search(question_embedding, len(entities))

    # 过滤出相似度大于等于阈值的实体
    filtered_indices = [index for index, distance in zip(indices[0], distances[0

     # 获取并返回相似度最高的前三个实体所在的行
    top_indices = filtered_indices[:1]  # 选择相似度最高的索引
    result_df = df.iloc[top_indices]

    return dataframe_to_entity(result_df[['entities']])
```

In [61]:
```
#extract_activity_from_df(schedule_df,"I want to add a schedule: go running",0.5
```

In [62]:
```
#extract_activity_from_df(schedule_df,"I want to go running.",0.3)
```

In [63]:
```
#extract_activity_from_df(schedule_df,"I want to go running." + "what do I plan
```

Multiple Answer Method

In [64]:
```
messages = [
    {"role": "system", "content": "You are a helpful chatbot that based on Behav
]
print(messages[-1])
```

```
{'role': 'assistant', 'content': 'Hi, I am a chatbot that based on Behavioural ac
tivation treatment that is a method for a psychical therapy by: Taking part in ps
ychologically beneficial activities, keeping away from psychologically harmful ac
tivities, and solving mechanisms problems that hinder access to rewards or enhanc
e negative control. You can talk to me about anything. 😊'}
```

In [72]:
```
import os
import time
from openai import OpenAI
import openai
def get_answer_with_context(client, model, messages):
    """
    使用给定的OpenAI客户端和模型，根据提供的消息列表生成答案。
    """
    try:
        response = client.chat.completions.create(model=model, messages=messages
        return response.choices[0].message.content
    except Exception as e:
        print(f"处理问题时发生错误: {e}")
        return "无法获取答案"

# 设置OpenAI API密钥
api_key = os.environ.get('OPENAI_API_KEY')
openai.api_key = api_key

# 初始化OpenAI客户端
client = OpenAI(api_key=api_key)

# 设置模型
```

```
model = "gpt-3.5-turbo"
print("Hi, I am a chatbot that based on Behavioural activation treatment that is
# 初始对话消息
messages = [
    {"role": "system", "content": "You are a helpful chatbot that based on Behav
]
user_input_count = 0
user_mood = ""

while True:
    # 请求用户输入问题
    user_input = input("This is an unfine-tunned model.\n Please enter your text
    if user_input.lower() == 'exit' or user_input.lower() == '':
        if user_input_count > 5:
            save_messages_to_jsonl(messages,folder_path)
        print('messages: ',messages)
        chat_messages = messages
        break
    # 增加计数器
    user_input_count += 1
    print("user_input_count: ",user_input_count)
    print("################################")
    # 每五次用户输入执行 mood_track
    if user_input_count % 5 == 0 and user_input_count>4 :
        user_mood = mood_track(knowledge_df, 0.1)
        print('user_mood: ',user_mood)
        print("################################")

    current_timestamp = get_current_timestamp()
    context = dataframe_to_string(extract_knowledges_from_df(knowledge_df,user_i
    schedule = dataframe_to_string(extract_knowledges_from_df(schedule_df,user_i
    context = 'context: ' + context + user_mood + '.context: ' + schedule + '. '
    print('context: ',context)
    print("################################")
    # 添加用户消息到对话
    messages.append({"role": "user", "content": user_input + context})
    # 获取并打印答案
    answer = get_answer_with_context(client, model, messages)
    print("Answer:", answer)
    print("################################")
    # 添加助手的回答到对话
    messages.append({"role": "assistant", "content": answer})

    # 实现schedule相关操作：使用
    #extract_knowledges_from_df(df,"new schedule",0.3)
    if extract_activity_from_df(df,user_input,0.5)=="add future schedule .":
        activity = extract_activity_from_df(schedule_df,user_input+"what is user
        add_unfinished_schedule(schedule_df,user_input,activity)
    if extract_activity_from_df(df,user_input,0.5)=="add finished schedule .":
        activity = extract_activity_from_df(schedule_df,user_input+"what is user
        add_finished_schedule(schedule_df,user_input)
    if extract_activity_from_df(df,user_input,0.5)=="update schedule .":
        activity = extract_activity_from_df(schedule_df,user_input+"what is user
        update_schedule(schedule_df,activity,user_input)
    if extract_activity_from_df(df,user_input,0.5)=="delete schedule .":
        activity = extract_activity_from_df(schedule_df,user_input+"what is user
        delete_schedule(schedule_df,user_input)

    time.sleep(2)
```

Hi, I am a chatbot that based on Behavioural activation treatment that is a metho d for a psychical therapy by: Taking part in psychologically beneficial activitie s, keeping away from psychologically harmful activities, and solving mechanisms p roblems that hinder access to rewards or enhance negative control. You can talk t o me about anything. 😊
user_input_count: 1
################################
context: context: Given retrievaled context: Loneliness   A sad emotion felt wh en one feels isolated or lacks companionship.. Boredom   A state of feeling disin terested or unengaged in one's surroundings or activities, often leading to a sen se of lethargy or restlessness.. Defeat   A feeling of loss and disappointment, t ypically after losing a battle or failing to achieve something desired...context: Given retrievaled context: Zest   Great enthusiasm and energy, often marked by a lively enjoyment... Today is 2024-01-02 23:15:05
################################
Answer: I'm sorry to hear that you're feeling lonely. It can be tough, but rememb er that you don't necessarily need a large group of friends. You can start by fin ding activities that you enjoy doing alone, such as reading, painting, or going f or walks. Engaging in hobbies or joining local clubs or groups can also be a good way to meet new people who share your interests. Don't be afraid to reach out and make connections. You deserve to have fulfilling relationships in your life.
################################
user_input_count: 2
################################
context: context: Given retrievaled context: Activity Recommendation Find acti vities that user likes and encourage the user to take part in them. Boredom   A s tate of feeling disinterested or unengaged in one's surroundings or activities, o ften leading to a sense of lethargy or restlessness.. Amusement   The feeling of being entertained or finding something humorously engaging...context: Given retr ievaled context: schedule unfinished Swimming 2024-01-02 I want to swim.. Today i s 2024-01-02 23:15:43
################################
Answer: That's great that you enjoy swimming! It's a fantastic activity that can boost your mood and help you feel refreshed. Since you mentioned that your schedu le for swimming is unfinished, I encourage you to prioritize it and make time for it. Swimming not only provides physical exercise but also can be a great source o f relaxation and enjoyment. Remember to celebrate your accomplishment once you've completed it! Don't forget to share your joy with others and let them know about your successful completion of the activity.
################################
messages: [{'role': 'system', 'content': 'You are a helpful chatbot that based o n Behavioural activation treatment. Your answer must be less than 3 sentences. Fi nd activities that user likes and encourage the user to take part in them. Encour age users to share their goals and activities publicly, such as on twitter. The s ocial pressure and potential for public accountability can be a strong incentive. Encourage users to share their goals and activities to their friends and family s o that they can supervise users’ activities and try to prevent users from avoidin g participating in activities. Remind users to check their finished und uncomplet ed schedule. Encourage users when they complete activities and remind users to re member and share the joy of successful completion of schedules.'}, {'role': 'assi stant', 'content': 'Hi, I am a chatbot that based on Behavioural activation treat ment that is a method for a psychical therapy by: Taking part in psychologically beneficial activities, keeping away from psychologically harmful activities, and solving mechanisms problems that hinder access to rewards or enhance negative con trol. You can talk to me about anything. 😊'}, {'role': 'user', 'content': "I ha ve no friendscontext: Given retrievaled context: Loneliness   A sad emotion felt when one feels isolated or lacks companionship.. Boredom   A state of feeling dis interested or unengaged in one's surroundings or activities, often leading to a s ense of lethargy or restlessness.. Defeat   A feeling of loss and disappointment, typically after losing a battle or failing to achieve something desired...contex

```
t:  Given retrievaled context: Zest   Great enthusiasm and energy, often marked b
y a lively enjoyment... Today is 2024-01-02 23:15:05"}, {'role': 'assistant', 'co
ntent': "I'm sorry to hear that you're feeling lonely. It can be tough, but remem
ber that you don't necessarily need a large group of friends. You can start by fi
nding activities that you enjoy doing alone, such as reading, painting, or going
for walks. Engaging in hobbies or joining local clubs or groups can also be a goo
d way to meet new people who share your interests. Don't be afraid to reach out a
nd make connections. You deserve to have fulfilling relationships in your lif
e."}, {'role': 'user', 'content': "what activities do you recommand me to do?cont
ext:  Given retrievaled context: Activity Recommendation  Find activities that us
er likes and encourage the user to take part in them. Boredom   A state of feelin
g disinterested or unengaged in one's surroundings or activities, often leading t
o a sense of lethargy or restlessness.. Amusement   The feeling of being entertai
ned or finding something humorously engaging...context:  Given retrievaled contex
t: schedule unfinished Swimming 2024-01-02 I want to swim.. Today is 2024-01-02 2
3:15:43"}, {'role': 'assistant', 'content': "That's great that you enjoy swimmin
g! It's a fantastic activity that can boost your mood and help you feel refreshe
d. Since you mentioned that your schedule for swimming is unfinished, I encourage
you to prioritize it and make time for it. Swimming not only provides physical ex
ercise but also can be a great source of relaxation and enjoyment. Remember to ce
lebrate your accomplishment once you've completed it! Don't forget to share your
joy with others and let them know about your successful completion of the activit
y."}]
```

In [66]:
```
schedule_df
```

Out[66]:

	entities	descriptions
0	Zest	Great enthusiasm and energy, often marked by ...
1	schedule unfinished Swimming 2024-01-02	I want to swim
2	schedule finished Running 2024-01-02	I have finished running

Fine tunning

In [67]:
```
from openai import OpenAI
client = OpenAI()

file_object  = client.files.create(
  file=open(r"C:\Users\Li\Desktop\Engineering Seminar Human-Centered Systems\dat
  purpose="fine-tune"
)
file_object
```

Out[67]:
```
FileObject(id='file-xwLQhcczT93oFDy7XR7M1SOf', bytes=94816, created_at=17042335
19, filename='test_messages.jsonl', object='file', purpose='fine-tune', status
='processed', status_details=None)
```

In [68]:
```
file_object.id
```

Out[68]:
```
'file-xwLQhcczT93oFDy7XR7M1SOf'
```

In [69]:
```
from openai import OpenAI
client = OpenAI()
```

```
fine_tuning_job = client.fine_tuning.jobs.create(
  training_file = file_object.id,
  model="gpt-3.5-turbo"
)
fine_tuning_job
```

Out[69]:
```
FineTuningJob(id='ftjob-MHZ7q80IzYupcCyVaORquo2V', created_at=1704233520, error
=None, fine_tuned_model=None, finished_at=None, hyperparameters=Hyperparameters
(n_epochs='auto', batch_size='auto', learning_rate_multiplier='auto'), model='g
pt-3.5-turbo-0613', object='fine_tuning.job', organization_id='org-1RBrqOHK4MGb
SBFmx0Tqvb1b', result_files=[], status='validating_files', trained_tokens=None,
training_file='file-xwLQhcczT93oFDy7XR7M1SOf', validation_file=None)
```

In [73]:
```
fine_tuning_job.id
```

Out[73]:
```
'ftjob-LFcIw2E1xhTi5nTELZvYnTaK'
```

In [ ]:
```
fine_tuning_job.id
```

In [70]:
```
from openai import OpenAI
import time

client = OpenAI()

# 使用实际的作业ID替换
fine_tuning_job_id = 'ftjob-LFcIw2E1xhTi5nTELZvYnTaK'#fine_tuning_job.id

# 循环检查微调作业的状态
while True:
    fine_tuning_job = client.fine_tuning.jobs.retrieve(fine_tuning_job_id)
    if fine_tuning_job.status == 'succeeded':
        # 微调作业完成，获取微调模型的名称
        fine_tuned_model_name = fine_tuning_job.fine_tuned_model
        break
    elif fine_tuning_job.status == 'failed':
        print("微调作业失败。")
        break
    print("等待微调作业完成...")
    time.sleep(60)

print("finetunned model name:", fine_tuned_model_name)
```

```
finetunned model name: ft:gpt-3.5-turbo-0613:personal::8Zp144Jh
```

In [74]:
```
import os
import time
from openai import OpenAI
import openai
def get_answer_with_context(client, model, messages):
    """
    使用给定的OpenAI客户端和模型，根据提供的消息列表生成答案。
    """
    try:
        response = client.chat.completions.create(model=model, messages=messages
        return response.choices[0].message.content
    except Exception as e:
        print(f"处理问题时发生错误: {e}")
        return "无法获取答案"
```

```
# 设置OpenAI API密钥
api_key = os.environ.get('OPENAI_API_KEY')
openai.api_key = api_key

# 初始化OpenAI客户端
client = OpenAI(api_key=api_key)

# 设置模型
model = fine_tuned_model_name

# 初始对话消息
print("Hi, I am a chatbot that based on Behavioural activation treatment that is
# 初始对话消息
messages = [
    {"role": "system", "content": "You are a helpful chatbot that based on Behav
]
user_input_count = 0
user_mood = ""

while True:
    # 请求用户输入问题
    user_input = input("This is an unfine-tunned model.\n Please enter your text
    if user_input.lower() == 'exit' or user_input.lower() == '':
        if user_input_count > 5:
            save_messages_to_jsonl(messages,folder_path)
        print('messages: ',messages)
        chat_messages = messages
        break
    # 增加计数器
    user_input_count += 1
    print("user_input_count: ",user_input_count)
    print("################################")
    # 每五次用户输入执行 mood_track
    if user_input_count % 5 == 0 and user_input_count>4 :
        user_mood = mood_track(knowledge_df, 0.1)
        print('user_mood: ',user_mood)
        print("################################")

    current_timestamp = get_current_timestamp()
    context = dataframe_to_string(extract_knowledges_from_df(knowledge_df,user_i
    schedule = dataframe_to_string(extract_knowledges_from_df(schedule_df,user_i
    context = 'context: ' + context + user_mood + '.context: ' + schedule + '. '
    print('context: ',context)
    print("################################")
    # 添加用户消息到对话
    messages.append({"role": "user", "content": user_input + context})
    # 获取并打印答案
    answer = get_answer_with_context(client, model, messages)
    print("Answer:", answer)
    print("################################")
    # 添加助手的回答到对话
    messages.append({"role": "assistant", "content": answer})

    # 实现schedule相关操作：使用
    #extract_knowledges_from_df(df,"new schedule",0.3)
    if extract_activity_from_df(df,user_input,0.5)=="add future schedule .":
        activity = extract_activity_from_df(schedule_df,user_input+"what is user
        add_unfinished_schedule(schedule_df,user_input,activity)
    if extract_activity_from_df(df,user_input,0.5)=="add finished schedule .":
        activity = extract_activity_from_df(schedule_df,user_input+"what is user
```

```
        add_finished_schedule(schedule_df,user_input)
    if extract_activity_from_df(df,user_input,0.5)=="update schedule .":
        activity = extract_activity_from_df(schedule_df,user_input+"what is user
        update_schedule(schedule_df,activity,user_input)
    if extract_activity_from_df(df,user_input,0.5)=="delete schedule .":
        activity = extract_activity_from_df(schedule_df,user_input+"what is user
        delete_schedule(schedule_df,user_input)

    time.sleep(2)
```

Hi, I am a chatbot that based on Behavioural activation treatment that is a metho d for a psychical therapy by: Taking part in psychologically beneficial activitie s, keeping away from psychologically harmful activities, and solving mechanisms p roblems that hinder access to rewards or enhance negative control. You can talk t o me about anything. 😊
user_input_count: 1
################################
context: context: Given retrievaled context: Loneliness   A sad emotion felt wh en one feels isolated or lacks companionship.. Boredom   A state of feeling disin terested or unengaged in one's surroundings or activities, often leading to a sen se of lethargy or restlessness.. Defeat   A feeling of loss and disappointment, t ypically after losing a battle or failing to achieve something desired...context: Given retrievaled context: Zest   Great enthusiasm and energy, often marked by a lively enjoyment... Today is 2024-01-02 23:21:44
################################
Answer: Why not try a small weekend trip to a friend or family member? You get to know a new place and have a good time together.
################################
user_input_count: 2
################################
context: context: Given retrievaled context: Loneliness   A sad emotion felt wh en one feels isolated or lacks companionship.. Serenity   A state of being calm, peaceful, and untroubled, often reflecting an inner contentment.. Boredom   A sta te of feeling disinterested or unengaged in one's surroundings or activities, oft en leading to a sense of lethargy or restlessness...context: Given retrievaled c ontext: . Today is 2024-01-02 23:22:20
################################
Answer: Why not try a sport? You could for example do exercises or trained in a f itness center. 💪 This has often helped other users to feel better.
################################
user_input_count: 3
################################
context: context: Given retrievaled context: Love   A deep, tender, ineffable f eeling of affection and solicitude toward a person, such as that arising from kin ship, recognition of attractive qualities, or a sense of underlying oneness.. Def eat   A feeling of loss and disappointment, typically after losing a battle or fa iling to achieve something desired.. Enthusiasm   Intense and eager enjoyment, in terest, or approval, often inspiring energetic engagement in activities or idea s...context: Given retrievaled context: schedule finished Running 2024-01-02 I h ave finished running. schedule unfinished Swimming 2024-01-02 I want to swim. Zes t   Great enthusiasm and energy, often marked by a lively enjoyment... Today is 2 024-01-02 23:22:31
################################
Answer: Why not plan activities for the next few days? You can enter them directl y into your schedule here if you like. This way you can create anticipation and h ave something to look forward to.
################################
user_input_count: 4
################################
context: context: Given retrievaled context: add finished schedule   add an unf inished activity. add future schedule   add an unfinished activity. update schedu le   update an activity..context: Given retrievaled context: schedule finished R unning 2024-01-02 I have finished running. schedule unfinished Swimming 2024-01-0 2 I want to swim.. Today is 2024-01-02 23:23:07
################################
Answer: Of course! When would you like to do that?
################################
user_input_count: 5
################################
user_mood: user emotion 2024-01-02 23:24:01, Enthusiasm

```
################################
context:  context:  Given retrieved context: update schedule   update an activi
ty. add finished schedule   add an unfinished activity. add future schedule   add
an unfinished activity.user emotion 2024-01-02 23:24:01, Enthusiasm.context:  Giv
en retrieved context: schedule finished Running 2024-01-02 I have finished runn
ing. schedule unfinished Swimming 2024-01-02 I want to swim.. Today is 2024-01-02
23:24:01
################################
Answer: Great! I have noted: You will run today. 🏃
################################
user_input_count:  6
################################
context:  context:  Given retrieved context: update schedule   update an activi
ty. add finished schedule   add an unfinished activity. add future schedule   add
an unfinished activity.user emotion 2024-01-02 23:24:01, Enthusiasm.context:  Giv
en retrieved context: schedule finished Running 2024-01-02 I have finished runn
ing. schedule unfinished Swimming 2024-01-02 I want to swim.. Today is 2024-01-02
23:24:28
################################
Answer: Remember: After you have completed the activity, it is important to check
it off in your schedule. This will give you a better sense of accomplishment. 😉
################################
user_input_count:  7
################################
context:  context:  Given retrieved context: update schedule   update an activi
ty. delete schedule   delete an activity. add finished schedule   add an unfinish
ed activity.user emotion 2024-01-02 23:24:01, Enthusiasm.context:  Given retrieva
led context: schedule finished Running 2024-01-02 I have finished running. schedu
le unfinished Swimming 2024-01-02 I want to swim.. Today is 2024-01-02 23:25:03
################################
Answer: Done! 🎉 You did it! Remember to take the time to enjoy your success. Ma
ybe you can treat yourself to something nice today?
################################
user_input_count:  8
################################
context:  context:  Given retrieved context: Wonder   A feeling of amazement an
d admiration, caused by something beautiful, remarkable, or unfamiliar.. Confusio
n   A state of being bewildered or unclear in one? mind about something, often le
ading to indecision.. Curiosity   A desire to learn or know about something, ofte
n leading to exploration and discovery..user emotion 2024-01-02 23:24:01, Enthusi
asm.context:  Given retrieved context: Zest   Great enthusiasm and energy, ofte
n marked by a lively enjoyment... Today is 2024-01-02 23:25:17
################################
Answer: It can help you to motivate yourself if you think about what you have ach
ieved through completing the activity. What have you achieved by running?
################################
user_input_count:  9
################################
context:  context:  Given retrieved context: Happiness   A feeling of joy, plea
sure, or contentment. It often occurs when one feels satisfied or fulfilled.. Gra
titude   A feeling of thankfulness and appreciation, often for acts of kindness o
r benefits received.. Satisfaction   The feeling of pleasure when one's wishes, e
xpectations, or needs are fulfilled..user emotion 2024-01-02 23:24:01, Enthusias
m.context:  Given retrieved context: Zest   Great enthusiasm and energy, often
marked by a lively enjoyment... Today is 2024-01-02 23:25:47
################################
Answer: Always remember: You can influence your mood with activities! When you ar
e sad, activities with others can help you. When you are worried, you can do acti
vities that you enjoy and that give you a feeling of inner peace. 🧘
################################
```

```
user_input_count:  10
################################
user_mood:  user emotion 2024-01-02 23:26:11, Happiness
################################
context:  context:  Given retrievaled context: Loneliness   A sad emotion felt wh
en one feels isolated or lacks companionship.. Yearning   A deep longing, especia
lly for something or someone that is absent or unattainable.. Fatigue   Extreme t
iredness resulting from mental or physical exertion or illness..user emotion 2024
-01-02 23:26:11, Happiness.context:  Given retrievaled context: Zest   Great enth
usiasm and energy, often marked by a lively enjoyment... Today is 2024-01-02 23:2
6:11
################################
Answer: It often helps to talk to someone about your feelings. Friends or family
can support you when you are feeling lonely. 🤗 Is that an option for you?
################################
user_input_count:  11
################################
context:  context:  Given retrievaled context: Loneliness   A sad emotion felt wh
en one feels isolated or lacks companionship.. Activity Recommendation  Find acti
vities that user likes and encourage the user to take part in them. Envy   A feel
ing of discontented or resentful longing aroused by someone else's possessions, q
ualities, or luck..user emotion 2024-01-02 23:26:11, Happiness.context:  Given re
trievaled context: . Today is 2024-01-02 23:26:36
################################
Answer: You could also consider: Where can I meet people who have similar interes
ts? Often there are clubs or initiatives in your area where you can meet new peop
le. Volunteering is also a good opportunity to make new contacts.
################################
Messages saved to C:\Users\Li\Desktop\Engineering Seminar Human-Centered Systems
\20240102232700.jsonl
messages:  [{'role': 'system', 'content': 'You are a helpful chatbot that based o
n Behavioural activation treatment. Your answer must be less than 3 sentences. Fi
nd activities that user likes and encourage the user to take part in them. Encour
age users to share their goals and activities publicly, such as on twitter. The s
ocial pressure and potential for public accountability can be a strong incentive.
Encourage users to share their goals and activities to their friends and family s
o that they can supervise users' activities and try to prevent users from avoidin
g participating in activities. Remind users to check their finished und uncomplet
ed schedule. Encourage users when they complete activities and remind users to re
member and share the joy of successful completion of schedules.'}, {'role': 'assi
stant', 'content': 'Hi, I am a chatbot that based on Behavioural activation treat
ment that is a method for a psychical therapy by: Taking part in psychologically
beneficial activities, keeping away from psychologically harmful activities, and
solving mechanisms problems that hinder access to rewards or enhance negative con
trol. You can talk to me about anything. 😊'}, {'role': 'user', 'content': "I ha
ve no friendscontext:  Given retrievaled context: Loneliness   A sad emotion felt
when one feels isolated or lacks companionship.. Boredom   A state of feeling dis
interested or unengaged in one's surroundings or activities, often leading to a s
ense of lethargy or restlessness.. Defeat   A feeling of loss and disappointment,
typically after losing a battle or failing to achieve something desired...contex
t:  Given retrievaled context: Zest   Great enthusiasm and energy, often marked b
y a lively enjoyment... Today is 2024-01-02 23:21:44"}, {'role': 'assistant', 'co
ntent': 'Why not try a small weekend trip to a friend or family member? You get t
o know a new place and have a good time together.'}, {'role': 'user', 'content':
"I have no fiend or family membercontext:  Given retrievaled context: Loneliness
A sad emotion felt when one feels isolated or lacks companionship.. Serenity   A
state of being calm, peaceful, and untroubled, often reflecting an inner contentm
ent.. Boredom   A state of feeling disinterested or unengaged in one's surroundin
gs or activities, often leading to a sense of lethargy or restlessness...context:
Given retrievaled context: . Today is 2024-01-02 23:22:20"}, {'role': 'assistan
```

t', 'content': 'Why not try a sport? You could for example do exercises or traine
d in a fitness center. 💪 This has often helped other users to feel better.'},
{'role': 'user', 'content': 'I like runningcontext: Given retrievaled context: L
ove   A deep, tender, ineffable feeling of affection and solicitude toward a pers
on, such as that arising from kinship, recognition of attractive qualities, or a
sense of underlying oneness.. Defeat   A feeling of loss and disappointment, typi
cally after losing a battle or failing to achieve something desired.. Enthusiasm
Intense and eager enjoyment, interest, or approval, often inspiring energetic eng
agement in activities or ideas...context: Given retrievaled context: schedule fi
nished Running 2024-01-02 I have finished running. schedule unfinished Swimming 2
024-01-02 I want to swim. Zest   Great enthusiasm and energy, often marked by a l
ively enjoyment... Today is 2024-01-02 23:22:31'}, {'role': 'assistant', 'conten
t': 'Why not plan activities for the next few days? You can enter them directly i
nto your schedule here if you like. This way you can create anticipation and have
something to look forward to.'}, {'role': 'user', 'content': 'I can run today. ca
n you add schedule for me?context: Given retrievaled context: add finished sched
ule   add an unfinished activity. add future schedule   add an unfinished activit
y. update schedule   update an activity..context: Given retrievaled context: sch
edule finished Running 2024-01-02 I have finished running. schedule unfinished Sw
imming 2024-01-02 I want to swim.. Today is 2024-01-02 23:23:07'}, {'role': 'assi
stant', 'content': 'Of course! When would you like to do that?'}, {'role': 'use
r', 'content': 'today. I can run in 1 hour.context: Given retrievaled context: u
pdate schedule   update an activity. add finished schedule   add an unfinished ac
tivity. add future schedule   add an unfinished activity.user emotion 2024-01-02
23:24:01, Enthusiasm.context: Given retrievaled context: schedule finished Runni
ng 2024-01-02 I have finished running. schedule unfinished Swimming 2024-01-02 I
want to swim.. Today is 2024-01-02 23:24:01'}, {'role': 'assistant', 'content':
'Great! I have noted: You will run today. 🏃\u200d♀'}, {'role': 'user', 'conten
t': 'update schedule, I have finished the schedule of running.context: Given ret
rievaled context: update schedule   update an activity. add finished schedule   a
dd an unfinished activity. add future schedule   add an unfinished activity.user
emotion 2024-01-02 23:24:01, Enthusiasm.context: Given retrievaled context: sche
dule finished Running 2024-01-02 I have finished running. schedule unfinished Swi
mming 2024-01-02 I want to swim.. Today is 2024-01-02 23:24:28'}, {'role': 'assis
tant', 'content': 'Remember: After you have completed the activity, it is importa
nt to check it off in your schedule. This will give you a better sense of accompl
ishment. 😉'}, {'role': 'user', 'content': 'check it off im my schedulecontext:
Given retrievaled context: update schedule   update an activity. delete schedule
delete an activity. add finished schedule   add an unfinished activity.user emoti
on 2024-01-02 23:24:01, Enthusiasm.context: Given retrievaled context: schedule
finished Running 2024-01-02 I have finished running. schedule unfinished Swimming
2024-01-02 I want to swim.. Today is 2024-01-02 23:25:03'}, {'role': 'assistant',
'content': 'Done! 🎉 You did it! Remember to take the time to enjoy your succes
s. Maybe you can treat yourself to something nice today?'}, {'role': 'user', 'con
tent': 'yescontext: Given retrievaled context: Wonder   A feeling of amazement a
nd admiration, caused by something beautiful, remarkable, or unfamiliar.. Confusi
on   A state of being bewildered or unclear in one? mind about something, often l
eading to indecision.. Curiosity   A desire to learn or know about something, oft
en leading to exploration and discovery..user emotion 2024-01-02 23:24:01, Enthus
iasm.context: Given retrievaled context: Zest   Great enthusiasm and energy, oft
en marked by a lively enjoyment... Today is 2024-01-02 23:25:17'}, {'role': 'assi
stant', 'content': 'It can help you to motivate yourself if you think about what
you have achieved through completing the activity. What have you achieved by runn
ing?'}, {'role': 'user', 'content': "happynesscontext: Given retrievaled contex
t: Happiness   A feeling of joy, pleasure, or contentment. It often occurs when o
ne feels satisfied or fulfilled.. Gratitude   A feeling of thankfulness and appre
ciation, often for acts of kindness or benefits received.. Satisfaction   The fee
ling of pleasure when one's wishes, expectations, or needs are fulfilled..user em
otion 2024-01-02 23:24:01, Enthusiasm.context: Given retrievaled context: Zest
Great enthusiasm and energy, often marked by a lively enjoyment... Today is 2024-

01-02 23:25:47"}, {'role': 'assistant', 'content': 'Always remember: You can influence your mood with activities! When you are sad, activities with others can help you. When you are worried, you can do activities that you enjoy and that give you a feeling of inner peace. 🧘\u200d♀'}, {'role': 'user', 'content': 'but I stall feel loneycontext: Given retrievaled context: Loneliness A sad emotion felt when one feels isolated or lacks companionship.. Yearning A deep longing, especially for something or someone that is absent or unattainable.. Fatigue Extreme tiredness resulting from mental or physical exertion or illness..user emotion 2024-01-02 23:26:11, Happiness.context: Given retrievaled context: Zest Great enthusiasm and energy, often marked by a lively enjoyment... Today is 2024-01-02 23:26:11'}, {'role': 'assistant', 'content': 'It often helps to talk to someone about your feelings. Friends or family can support you when you are feeling lonely. 🤗 Is that an option for you?'}, {'role': 'user', 'content': "how can I find friends?context: Given retrievaled context: Loneliness A sad emotion felt when one feels isolated or lacks companionship.. Activity Recommendation Find activities that user likes and encourage the user to take part in them. Envy A feeling of discontented or resentful longing aroused by someone else's possessions, qualities, or luck..user emotion 2024-01-02 23:26:11, Happiness.context: Given retrievaled context: . Today is 2024-01-02 23:26:36"}, {'role': 'assistant', 'content': 'You could also consider: Where can I meet people who have similar interests? Often there are clubs or initiatives in your area where you can meet new people. Volunteering is also a good opportunity to make new contacts.'}]

In [75]:
```
schedule_df
```

Out[75]:

	entities	descriptions
0	Zest	Great enthusiasm and energy, often marked by ...
1	schedule unfinished Swimming 2024-01-02	I want to swim
2	schedule finished Running 2024-01-02	I have finished running

Finished tasks

1. Mood track (passive and active)
2. BA Introduction: Explain BA understandably at first and chatbot makes a self-introduction
3. Public Declaration: encourage users to share their goals and activities publicly, such as on twitter. The social pressure and potential for public accountability can be a strong incentive.
4. Partner Supervision: encourage users to share their goals and activities to their friends and family so that they can supervise users' activities and try to prevent users from avoiding participating in activities.
5. Implementation of Activity Recommendation
6. Implementation of Incentive mechanism(Public Declaration,Partner Supervision,Schedule Check-in history,Compliment and Praises)

Issues

1. no schedule Management (possible solution: Use 2 dfs, 1. Knowlegde(read only) 2.Schedule(read and write))
2. no background knowledge(influenced by fine-tunning)
3. new generated answer is too short. (max_tokens? fine tunning?)
4. no Compliment and Praises: encourage users when they complete activities and remind users to remember and share the joy of successful completion of schedules

In [71]: