chat-miner: turn your chats into artwork

chat-miner provides lean parsers for every major platform transforming chats into pandas dataframes. Artistic visualizations allow you to explore your data differently and create artwork from your chats.

1. Installation

Latest release including dependencies can be installed via PyPI:

```
pip install chat-miner
```

If you're interested in contributing, running the latest source code, or just like to build everything yourself:

```
git clone https://github.com/joweich/chat-miner.git
cd chat-miner
pip install -r requirements.txt
```

2. Exporting chat logs

Have a look at the official tutorials for WhatsApp, Signal, Telegram, Facebook Messenger, or Instagram Chats to learn how to export chat logs for your platform.

3. Parsing

Following code showcases the `WhatsAppParser` module. The usage of `SignalParser`, `TelegramJsonParser`, `FacebookMessengerParser`, and `InstagramJsonParser` follows the same pattern.

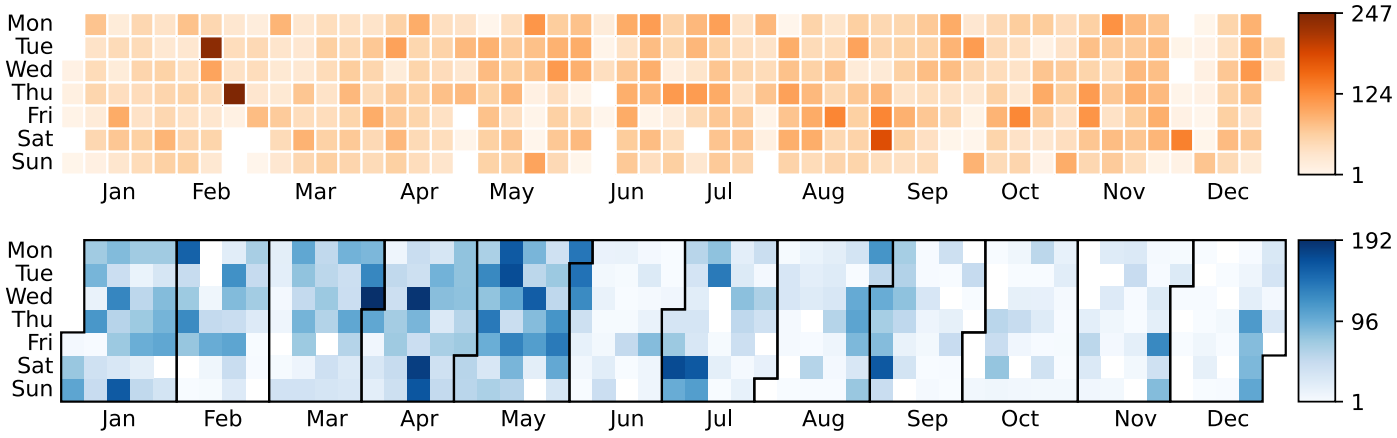
```
from chatminer.chatparsers import WhatsAppParser

parser = WhatsAppParser(FILEPATH)
parser.parse_file_into_df()
```

4. Visualizing

```
import chatminer.visualizations as vis
import matplotlib.pyplot as plt
```

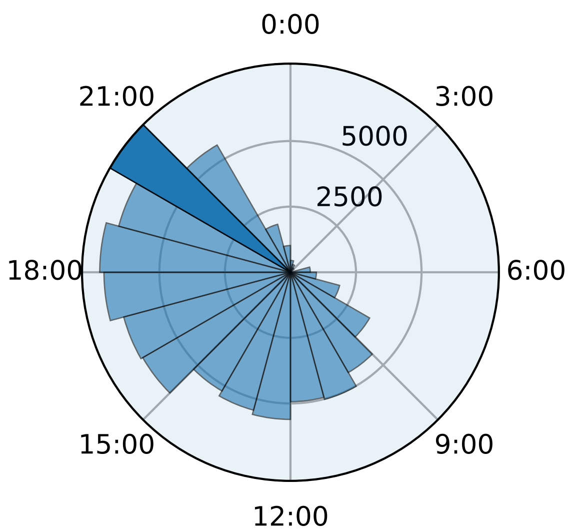
4.1 Heatmap: Message count per day

```
fig, ax = plt.subplots(2, 1, figsize=(9, 3))
ax[0] = vis.calendar_heatmap(parser.df, year=2020, cmap='Oranges', ax=ax[0])
ax[1] = vis.calendar_heatmap(parser.df, year=2021, linewidth=0, monthly_border=True, a
```



4.2 Sunburst: Message count per daytime

```
fig, ax = plt.subplots(1, 2, figsize=(7, 3), subplot_kw={'projection': 'polar'})
ax[0] = vis.sunburst(parser.df, highlight_max=True, isolines=[2500, 5000], isolines_re
ax[1] = vis.sunburst(parser.df, highlight_max=False, isolines=[0.5, 1], color='C1', ax
```

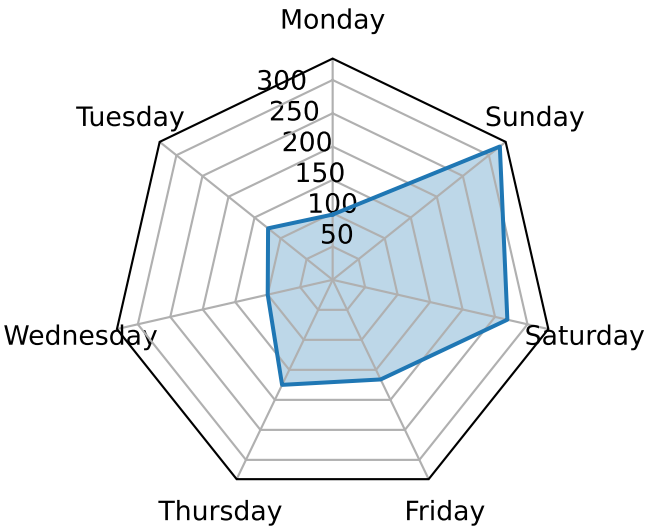


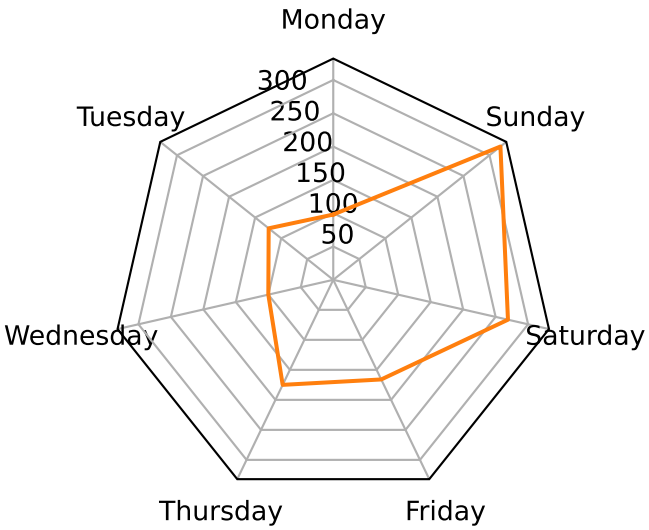
4.3 Wordcloud: Word frequencies

```
fig, ax = plt.subplots(figsize=(8, 3))
stopwords = ['these', 'are', 'stopwords']
kwargs={"background_color": "white", "width": 800, "height": 300, "max_words": 500}
ax = vis.wordcloud(parser.df, ax=ax, stopwords=stopwords, **kwargs)
```

4.4 Radarchart: Message count per weekday

```
fig, ax = plt.subplots(1, 2, figsize=(7, 3), subplot_kw={'projection': 'radar'})
ax[0] = vis.radar(parser.df, ax=ax[0])
ax[1] = vis.radar(parser.df, ax=ax[1], color='C1', alpha=0)
```





Releases 1

v0.1.0 Latest
on Nov 20, 2022

Contributors 13

+ 2 contributors

Languages

- **Python** 100.0%