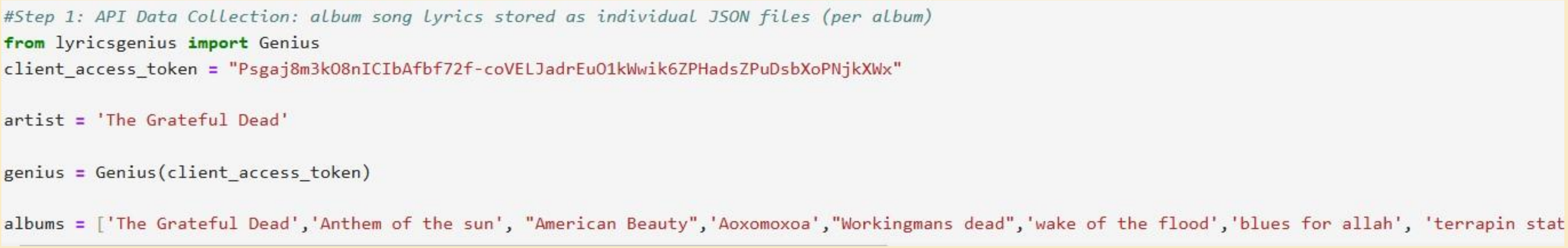
The Grateful Dead

Data Science-ified

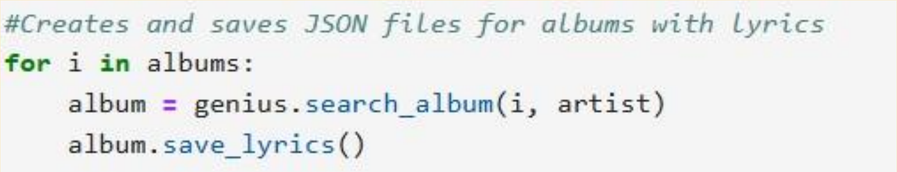
Trying to find the right words

1. Genius API to extract song information
2. GeniusLyrics Wrapper to get lyrics
3. Clean 'em up
4. Have some fun
5. Extract meaning???

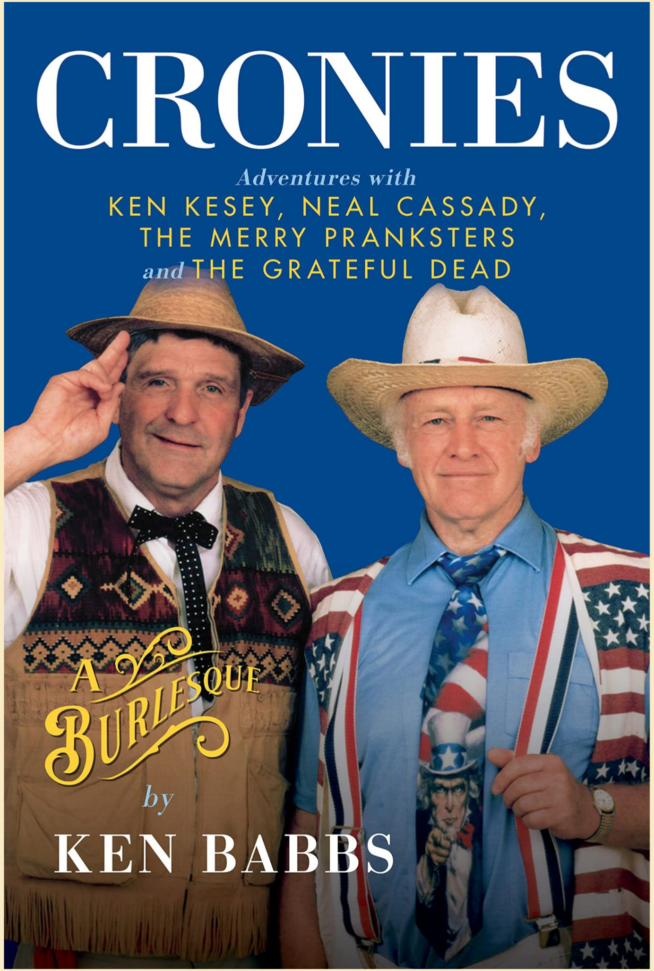
Building the DataFrame

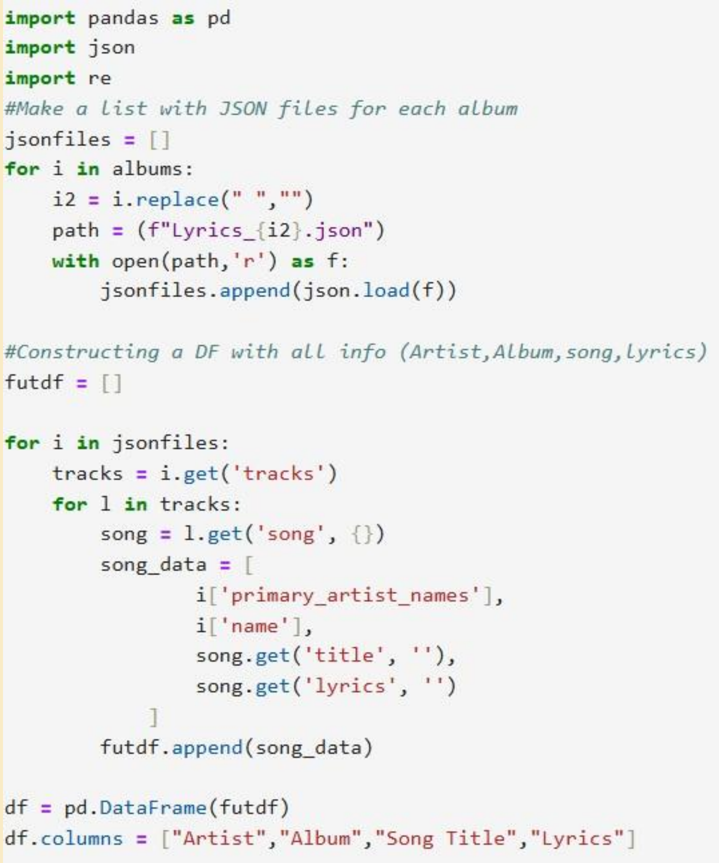


```
#Step 1: API Data Collection: album song lyrics stored as individual JSON files (per album)
from lyricsgenius import Genius
client_access_token = "Psgaj8m3kO8nICIbAfbf72f-coVELJadrEuO1kWwik6ZPHadsZPuDsbXoPNjkXWx"

artist = 'The Grateful Dead'

genius = Genius(client_access_token)

albums = ['The Grateful Dead','Anthem of the sun', "American Beauty",'Aoxomoxoa',"Workingmans dead",'wake of the flood','blues for allah', 'terrapin stat
```



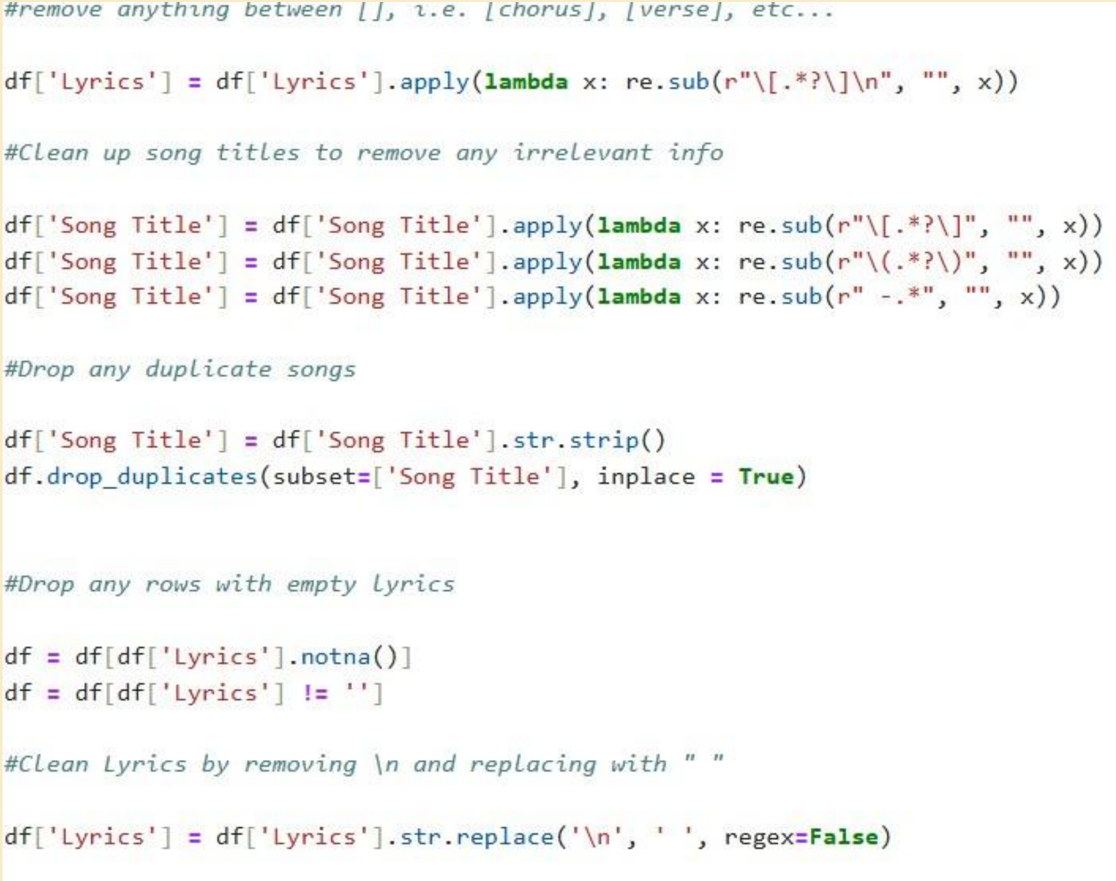
```
#Creates and saves JSON files for albums with lyrics
for i in albums:
    album = genius.search_album(i, artist)
    album.save_lyrics()
```





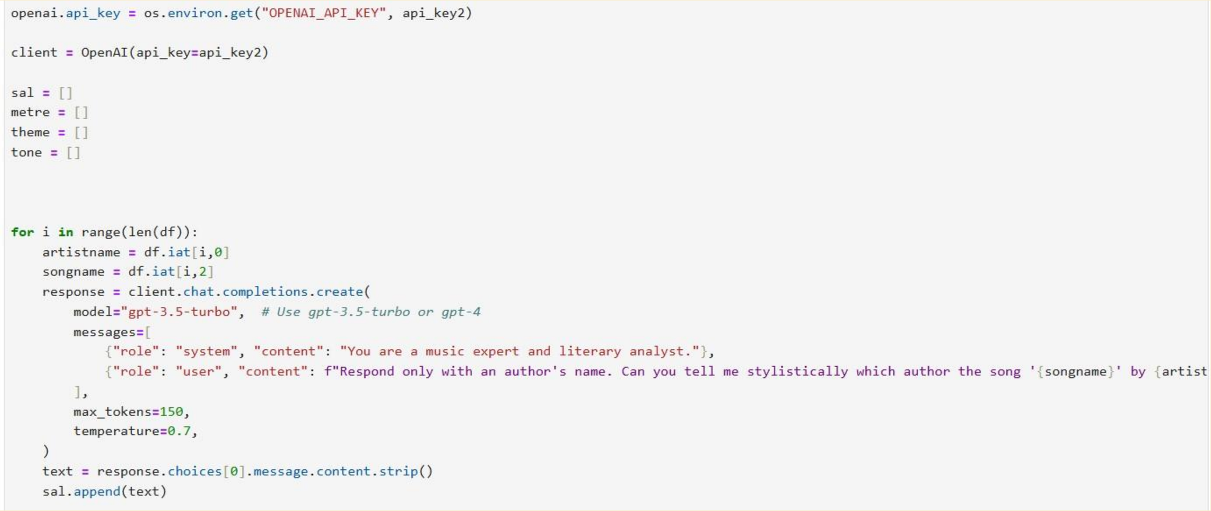
```python
import pandas as pd
import json
import re
#Make a list with JSON files for each album
jsonfiles = []
for i in albums:
    i2 = i.replace(" ","")
    path = (f"Lyrics_{i2}.json")
    with open(path,'r') as f:
        jsonfiles.append(json.load(f))

#Constructing a DF with all info (Artist,Album,song,lyrics)
futdf = []

for i in jsonfiles:
    tracks = i.get('tracks')
    for l in tracks:
        song = l.get('song', {})
        song_data = [
                    i['primary_artist_names'],
                    i['name'],
                    song.get('title', ''),
                    song.get('lyrics', '')
                ]
        futdf.append(song_data)

df = pd.DataFrame(futdf)
df.columns = ["Artist","Album","Song Title","Lyrics"]
```

Cleaning Up Our Act



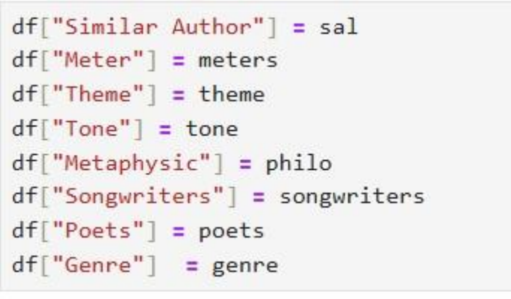
```
#remove anything between [], i.e. [chorus], [verse], etc...

df['Lyrics'] = df['Lyrics'].apply(lambda x: re.sub(r"\[.*?\]\n", "", x))

#Clean up song titles to remove any irrelevant info

df['Song Title'] = df['Song Title'].apply(lambda x: re.sub(r"\[.*?\]", "", x))
df['Song Title'] = df['Song Title'].apply(lambda x: re.sub(r"\(.*?\)", "", x))
df['Song Title'] = df['Song Title'].apply(lambda x: re.sub(r" -.*", "", x))

#Drop any duplicate songs

df['Song Title'] = df['Song Title'].str.strip()
df.drop_duplicates(subset=['Song Title'], inplace = True)


#Drop any rows with empty lyrics

df = df[df['Lyrics'].notna()]
df = df[df['Lyrics'] != '']

#Clean Lyrics by removing \n and replacing with " "

df['Lyrics'] = df['Lyrics'].str.replace('\n', ' ', regex=False)
```

Calling in Backup

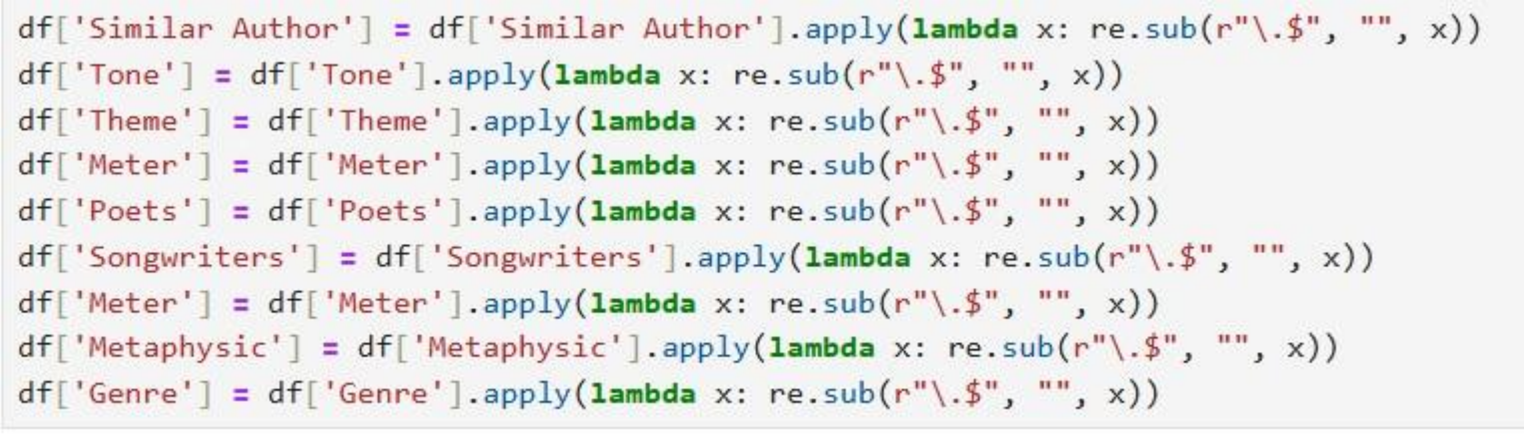


```
openai.api_key = os.environ.get("OPENAI_API_KEY", api_key2)

client = OpenAI(api_key=api_key2)

sal = []
metre = []
theme = []
tone = []



for i in range(len(df)):
    artistname = df.iat[i,0]
    songname = df.iat[i,2]
    response = client.chat.completions.create(
        model="gpt-3.5-turbo",  # Use gpt-3.5-turbo or gpt-4
        messages=[
            {"role": "system", "content": "You are a music expert and literary analyst."},
            {"role": "user", "content": f"Respond only with an author's name. Can you tell me stylistically which author the song '{songname}' by {artist
        ],
        max_tokens=150,
        temperature=0.7,
    )
    text = response.choices[0].message.content.strip()
    sal.append(text)
```

Cleaning Up Our Act..Again



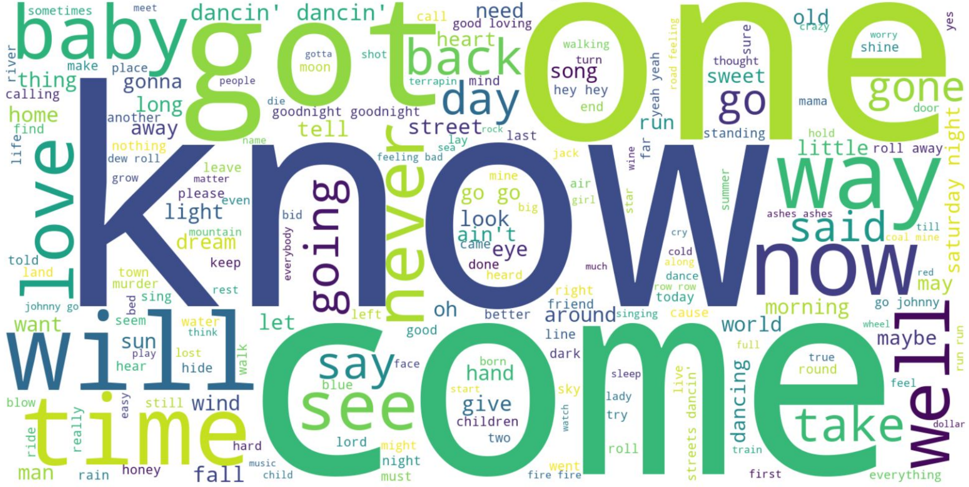
```python
df["Similar Author"] = sal
df["Meter"] = meters
df["Theme"] = theme
df["Tone"] = tone
df["Metaphysic"] = philo
df["Songwriters"] = songwriters
df["Poets"] = poets
df["Genre"]  = genre
```



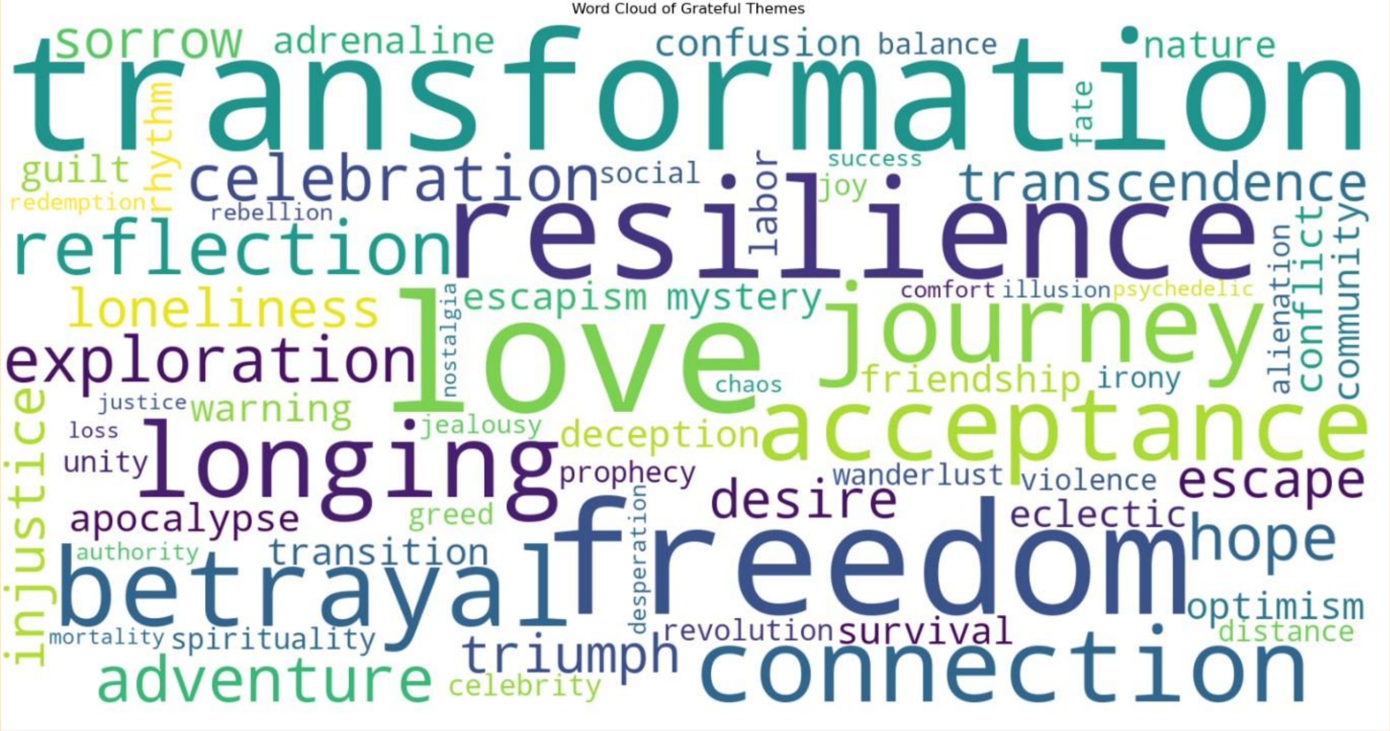
```python
df['Similar Author'] = df['Similar Author'].apply(lambda x: re.sub(r"\.$", "", x))
df['Tone'] = df['Tone'].apply(lambda x: re.sub(r"\.$", "", x))
df['Theme'] = df['Theme'].apply(lambda x: re.sub(r"\.$", "", x))
df['Meter'] = df['Meter'].apply(lambda x: re.sub(r"\.$", "", x))
df['Poets'] = df['Poets'].apply(lambda x: re.sub(r"\.$", "", x))
df['Songwriters'] = df['Songwriters'].apply(lambda x: re.sub(r"\.$", "", x))
df['Meter'] = df['Meter'].apply(lambda x: re.sub(r"\.$", "", x))
df['Metaphysic'] = df['Metaphysic'].apply(lambda x: re.sub(r"\.$", "", x))
df['Genre'] = df['Genre'].apply(lambda x: re.sub(r"\.$", "", x))
```

Lyrics Word Cloud

Word Cloud of Grateful Dead Lyrics

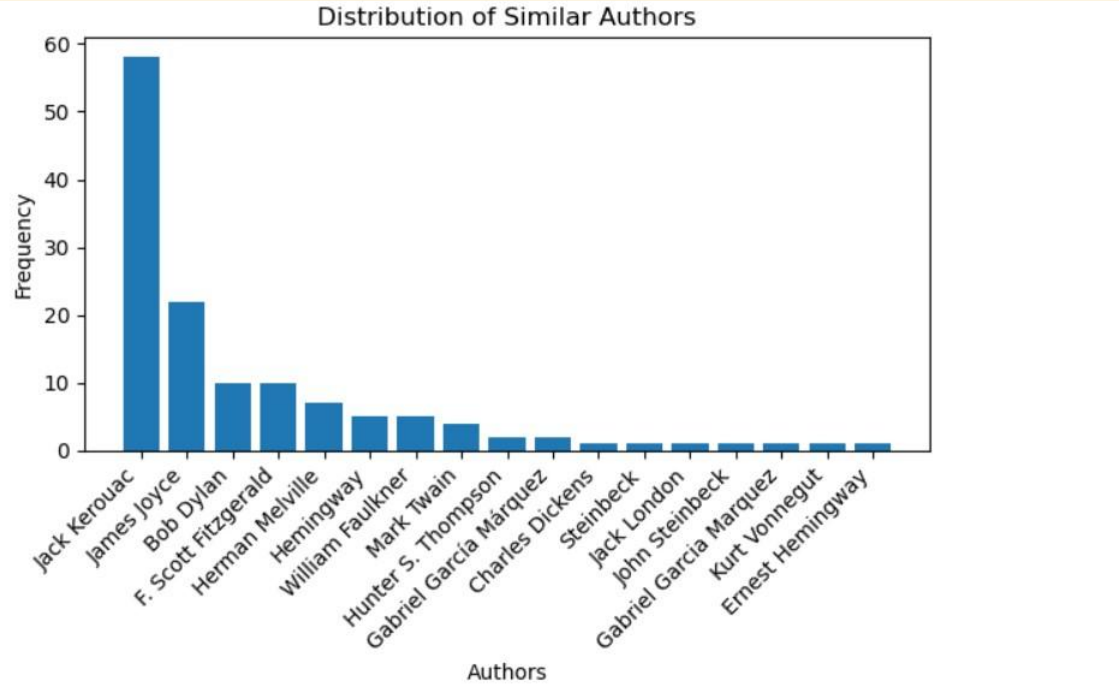


After AI

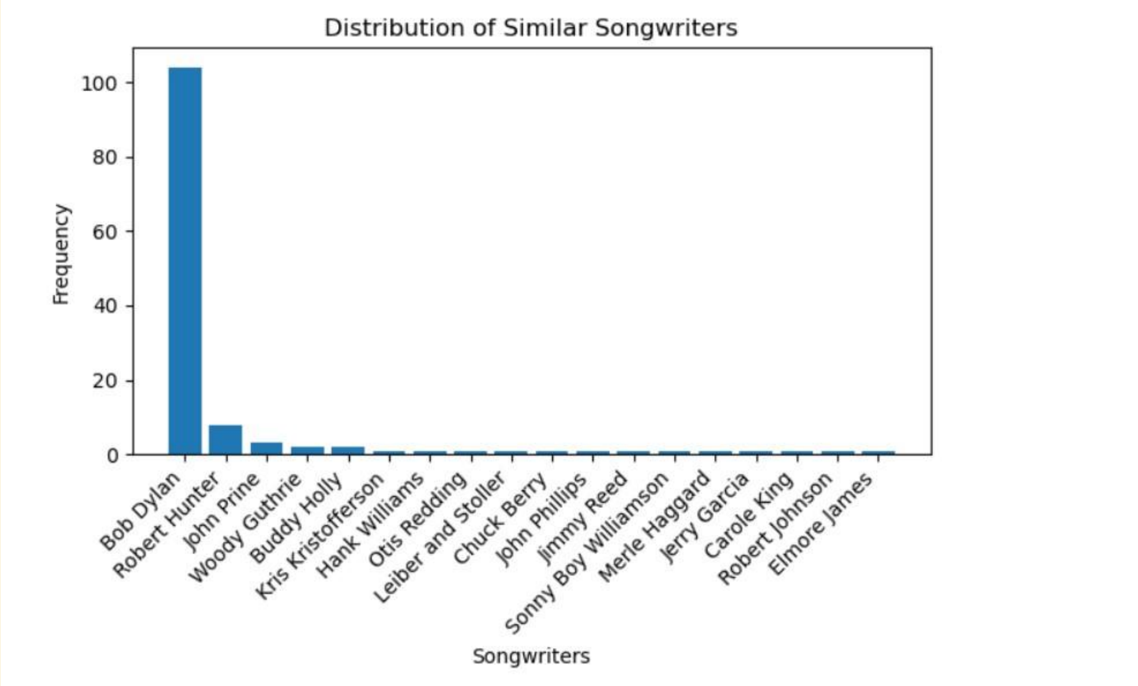


Word Cloud of Grateful Themes

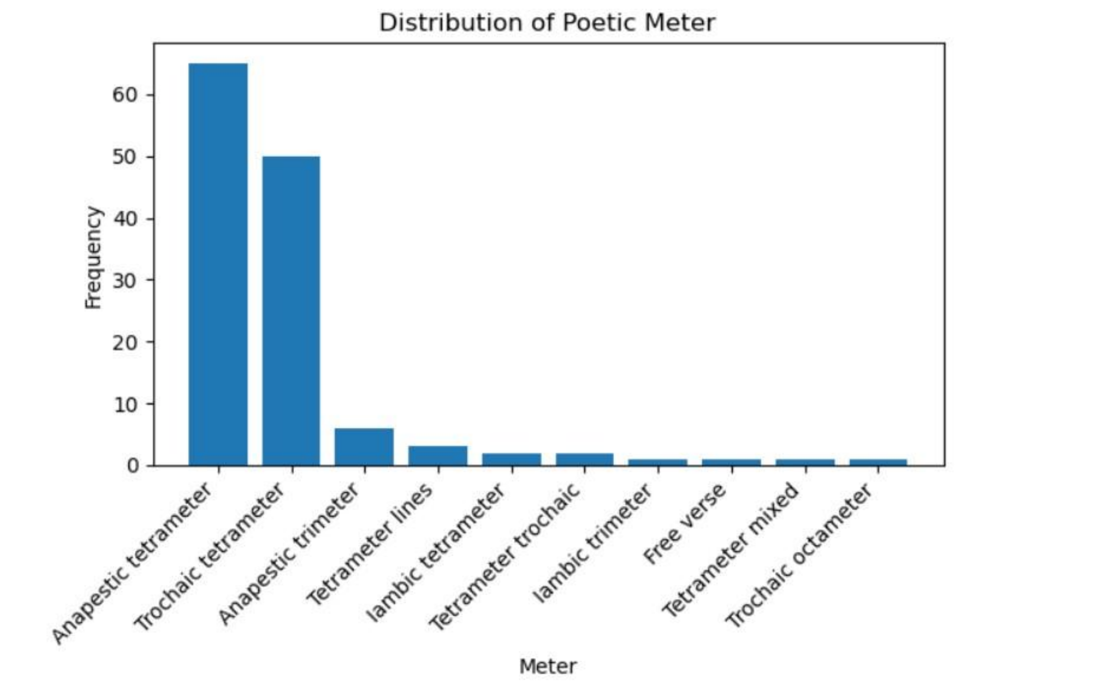
Similar Authors



Similar Songwriters



Poetic Meter



Bye Now!