Task-04

Analyze and visualize sentiment patterns in social media data to understand public opinion and attitudes towards specific topics or brands.

About Dataset

Twitter Sentiment Analysis Dataset

Overview

This is an entity-level sentiment analysis dataset of twitter. Given a message and an entity, the task is to judge the sentiment of the message about the entity. There are three classes in this dataset: Positive, Negative and Neutral. We regard messages that are not relevant to the entity (i.e. Irrelevant) as Neutral.

Importing the necessary libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud
```

Reading the dataset

```
cols=['ID', 'Topic', 'Sentiment', 'Text']
train = pd.read_csv(r"C:\Users\TUFAN\Downloads\Prodigy_InfoTech\
Task_4\twitter_training.csv",names=cols)

train.head()
```

```
     ID        Topic Sentiment  \
0  2401  Borderlands  Positive
1  2401  Borderlands  Positive
2  2401  Borderlands  Positive
3  2401  Borderlands  Positive
4  2401  Borderlands  Positive

                                                Text
0  im getting on borderlands and i will murder yo...
1  I am coming to the borders and I will kill you...
2  im getting on borderlands and i will kill you ...
3  im coming on borderlands and i will murder you...
4  im getting on borderlands 2 and i will murder ...
```

Information about the dataframe

```
train.shape
```

```
(74682, 4)
```

```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 74682 entries, 0 to 74681
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   ID         74682 non-null  int64 
 1   Topic      74682 non-null  object
 2   Sentiment  74682 non-null  object
 3   Text       73996 non-null  object
dtypes: int64(1), object(3)
memory usage: 2.3+ MB
```

```
train.describe(include=object)
```

```
                       Topic Sentiment  \
count                  74682     74682   
unique                    32         4   
top     TomClancysRainbowSix  Negative   
freq                    2400     22542   

                                                     Text
count                                               73996
unique                                              69491
top     At the same time, despite the fact that there ...
freq                                                  172
```

```
train['Sentiment'].unique()
```

```
array(['Positive', 'Neutral', 'Negative', 'Irrelevant'], dtype=object)
```

Checking for null/missing values in the dataset

```
train.isnull().sum()
```

```
ID             0
Topic          0
Sentiment      0
Text         686
dtype: int64
```

```
train.dropna(inplace=True)
```

```
train.isnull().sum()
```

```
ID           0
Topic        0
Sentiment    0
Text         0
dtype: int64
```

Checking for duplicate values

```
train.duplicated().sum()
```

```
2340
```

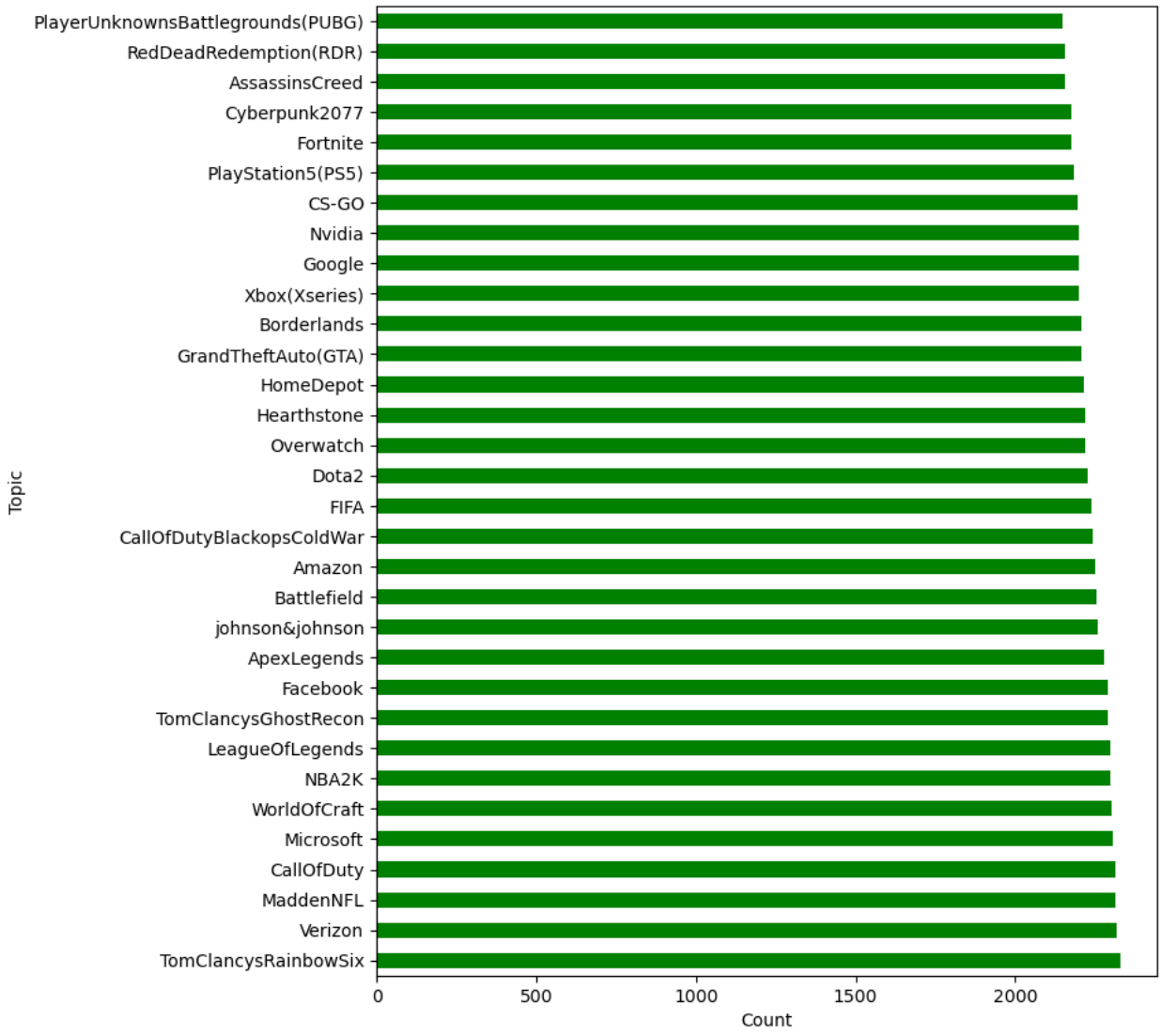
```
train.drop_duplicates(inplace=True)
```

```
train.duplicated().sum()
```

```
0
```

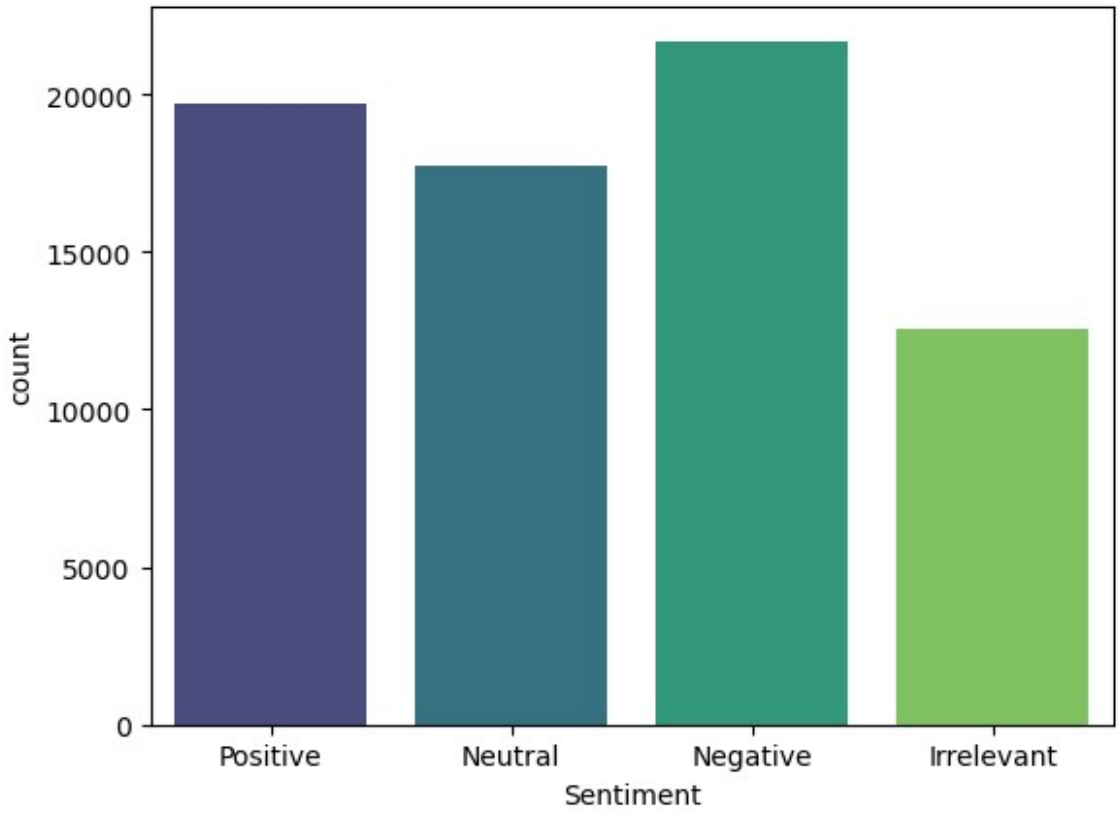
Visualization of count of different topics

```
plt.figure(figsize=(8,10))
train['Topic'].value_counts().plot(kind='barh',color='g')
plt.xlabel("Count")
plt.show()
```



Sentiment Distribution

```
sns.countplot(x = 'Sentiment',data=train,palette='viridis')
plt.show()
```

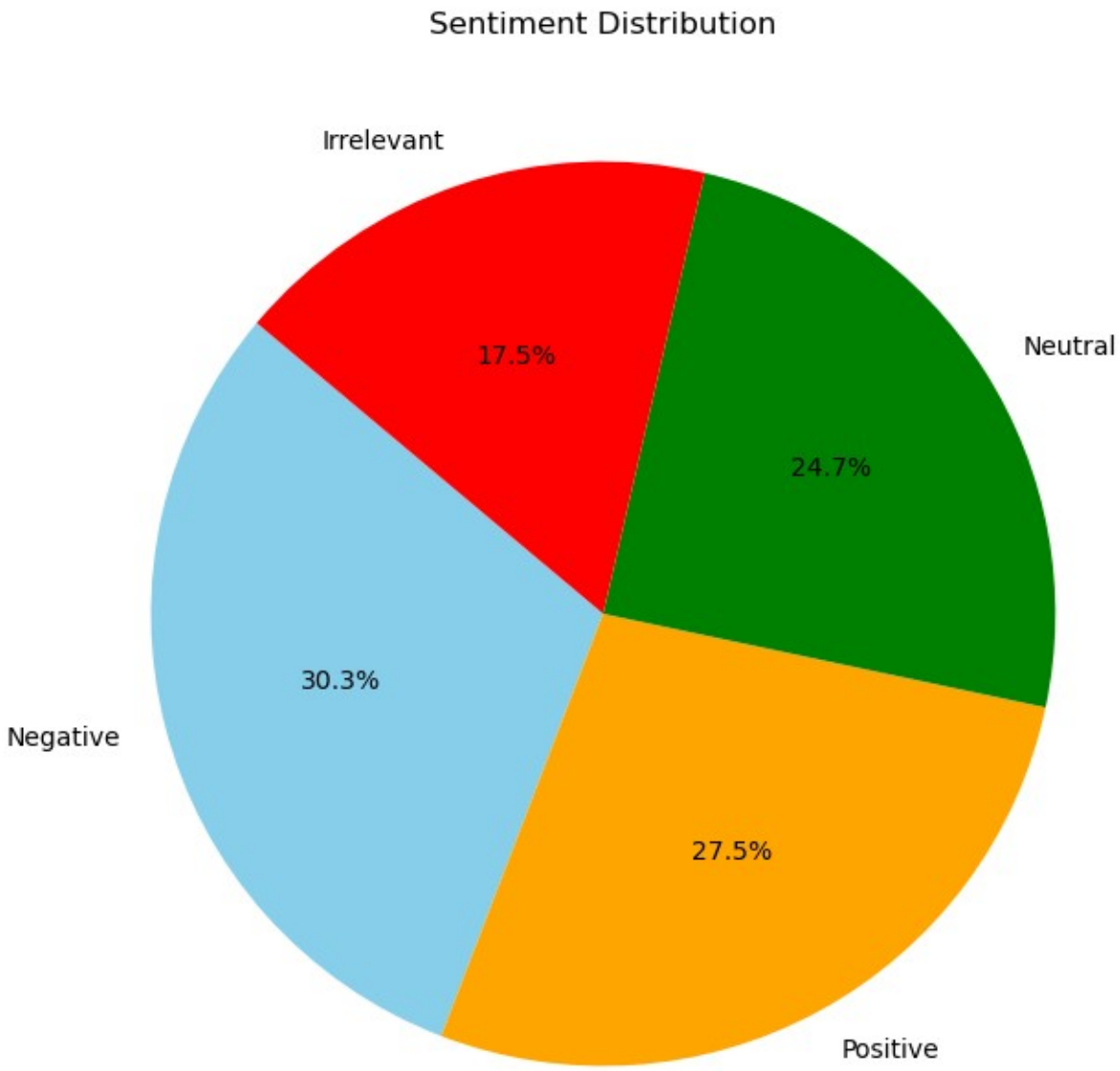



```
# Calculate the counts for each sentiment
sentiment_counts = train['Sentiment'].value_counts()

# Create the pie chart
plt.figure(figsize=(8, 8))
plt.pie(sentiment_counts, labels=sentiment_counts.index,
autopct="%1.1f%%", startangle=140, colors=['skyblue', 'orange',
'green', 'red', 'purple'])

plt.title('Sentiment Distribution')

# Show the plot
plt.show()
```



`Observation:`

- Most topic has negative sentiment

```
train

         ID        Topic Sentiment  \
0      2401  Borderlands  Positive
1      2401  Borderlands  Positive
2      2401  Borderlands  Positive
3      2401  Borderlands  Positive
4      2401  Borderlands  Positive
...     ...          ...       ...
```

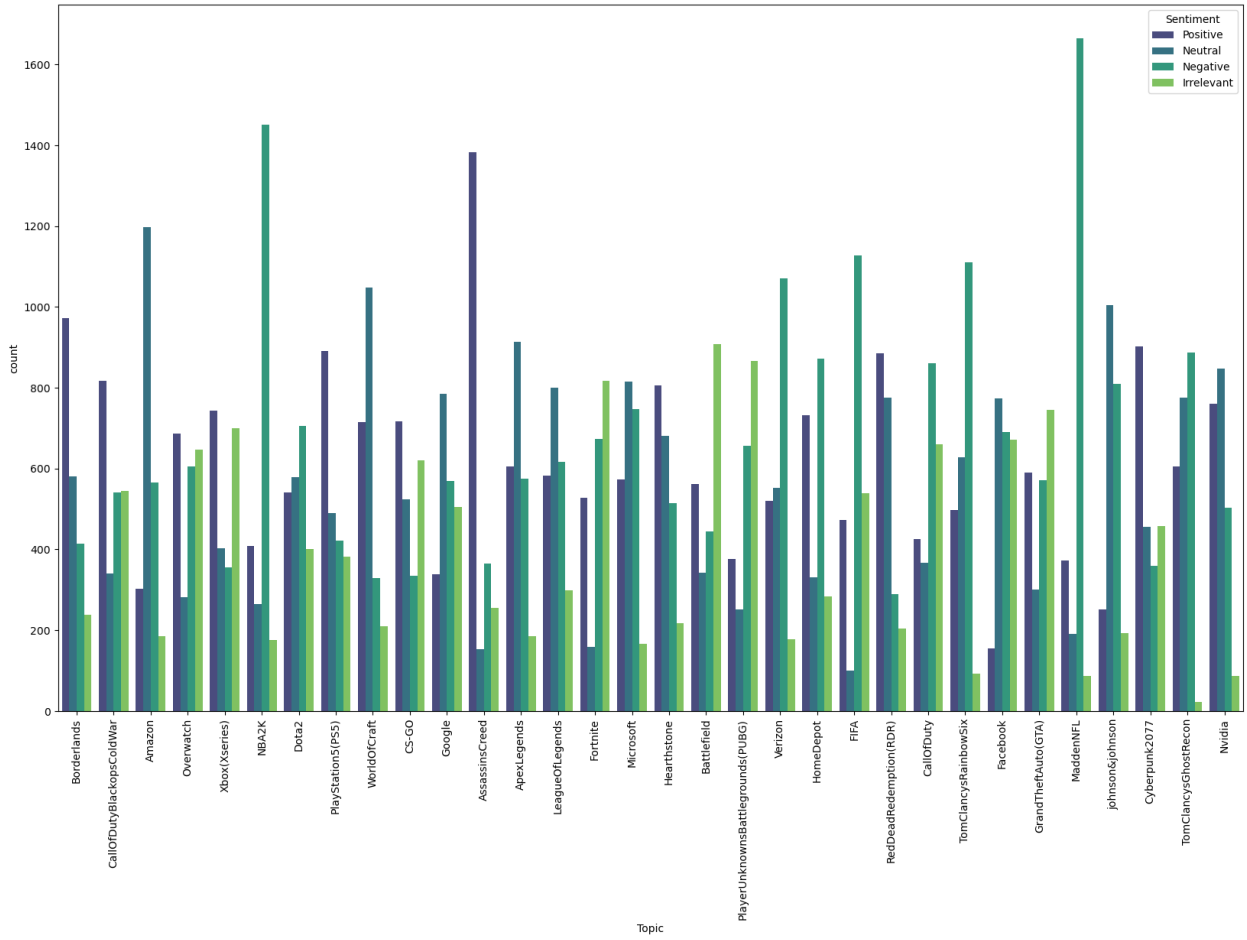
```
74677  9200       Nvidia  Positive
74678  9200       Nvidia  Positive
74679  9200       Nvidia  Positive
74680  9200       Nvidia  Positive
74681  9200       Nvidia  Positive
```

```
                                                     Text
0      im getting on borderlands and i will murder yo...
1      I am coming to the borders and I will kill you...
2      im getting on borderlands and i will kill you ...
3      im coming on borderlands and i will murder you...
4      im getting on borderlands 2 and i will murder ...
...                                                  ...
74677  Just realized that the Windows partition of my...
74678  Just realized that my Mac window partition is ...
74679  Just realized the windows partition of my Mac ...
74680  Just realized between the windows partition of...
74681  Just like the windows partition of my Mac is l...

[71656 rows x 4 columns]
```

Sentiment Distribution Topic-wise

```
plt.figure(figsize=(20,12))
sns.countplot(x='Topic',data=train,palette='viridis',hue='Sentiment')
plt.xticks(rotation=90)
plt.show()
```

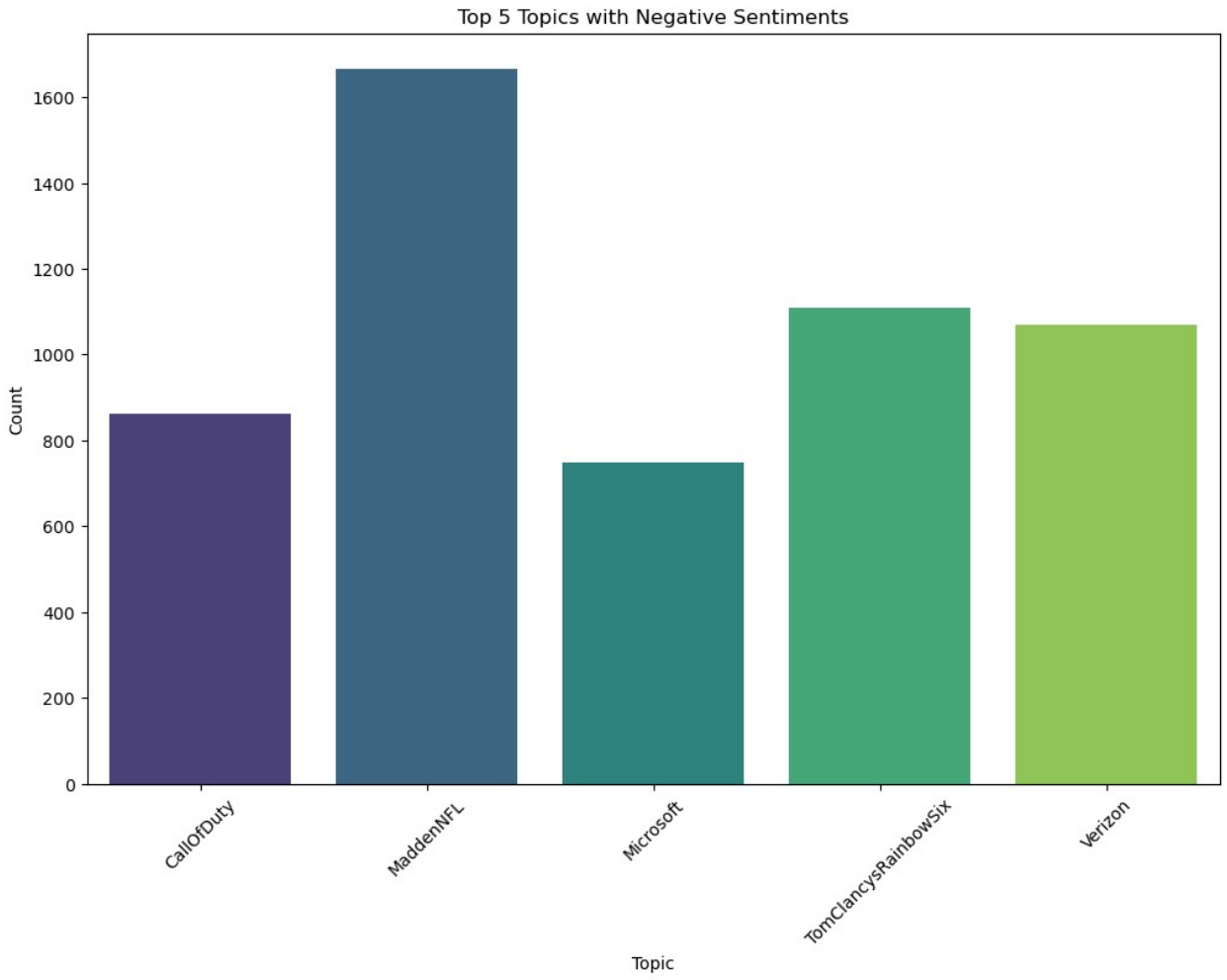



```
## Group by Topic and Sentiment
topic_wise_sentiment = train.groupby(["Topic",
"Sentiment"]).size().reset_index(name='Count')

# Step 2: Select Top 5 Topics
topic_counts = train['Topic'].value_counts().nlargest(5).index
top_topics_sentiment =
topic_wise_sentiment[topic_wise_sentiment['Topic'].isin(topic_counts)]
```

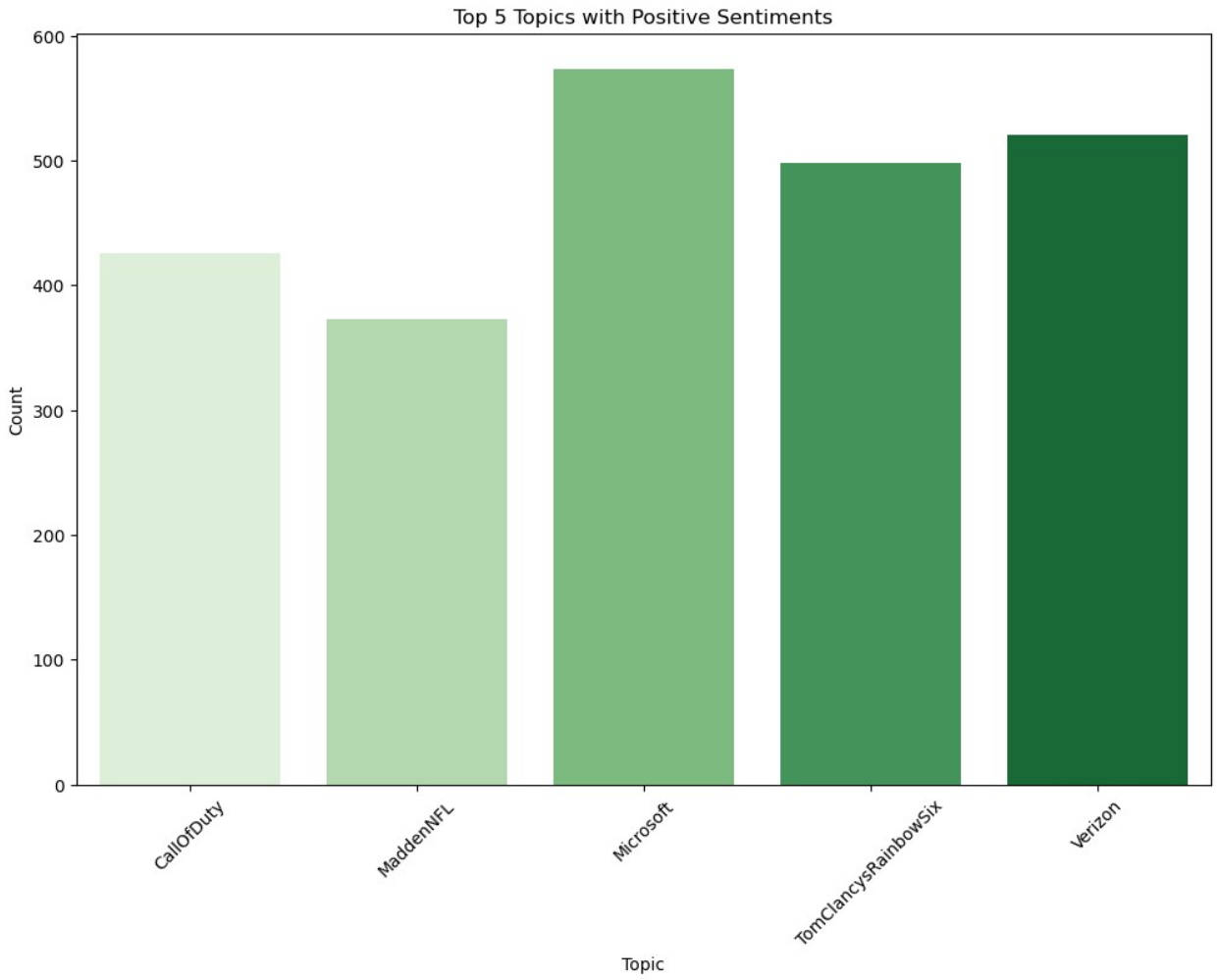
Top 5 Topics with Negative Sentiments

```
plt.figure(figsize=(12, 8))
sns.barplot(data=top_topics_sentiment[top_topics_sentiment['Sentiment'
] == 'Negative'], x='Topic', y='Count', palette='viridis')
plt.title('Top 5 Topics with Negative Sentiments')
plt.xlabel('Topic')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```



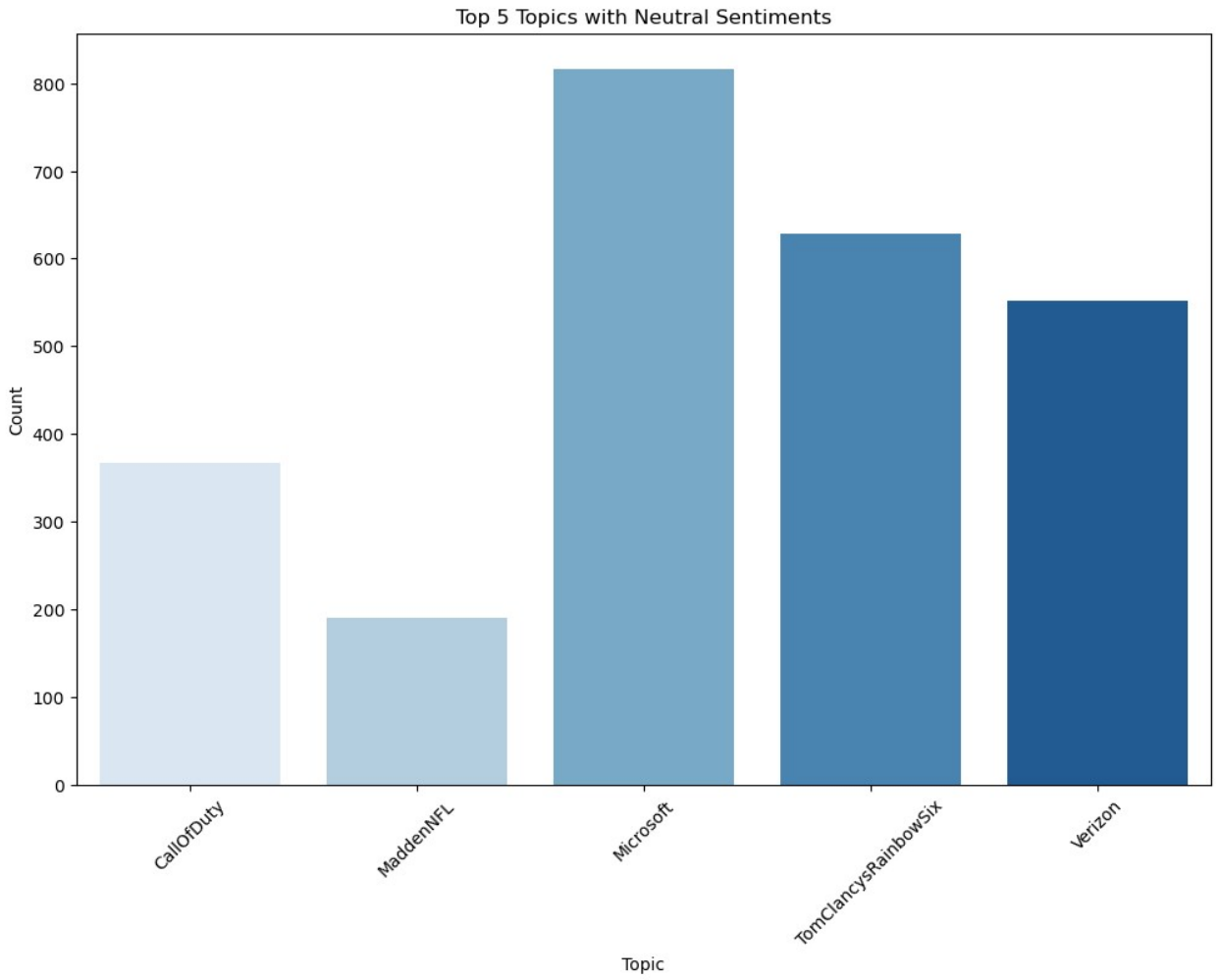
Top 5 Topics with Positive Sentiments

```
plt.figure(figsize=(12, 8))
sns.barplot(data=top_topics_sentiment[top_topics_sentiment['Sentiment'
] == 'Positive'], x='Topic', y='Count', palette='Greens')
plt.title('Top 5 Topics with Positive Sentiments')
plt.xlabel('Topic')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```



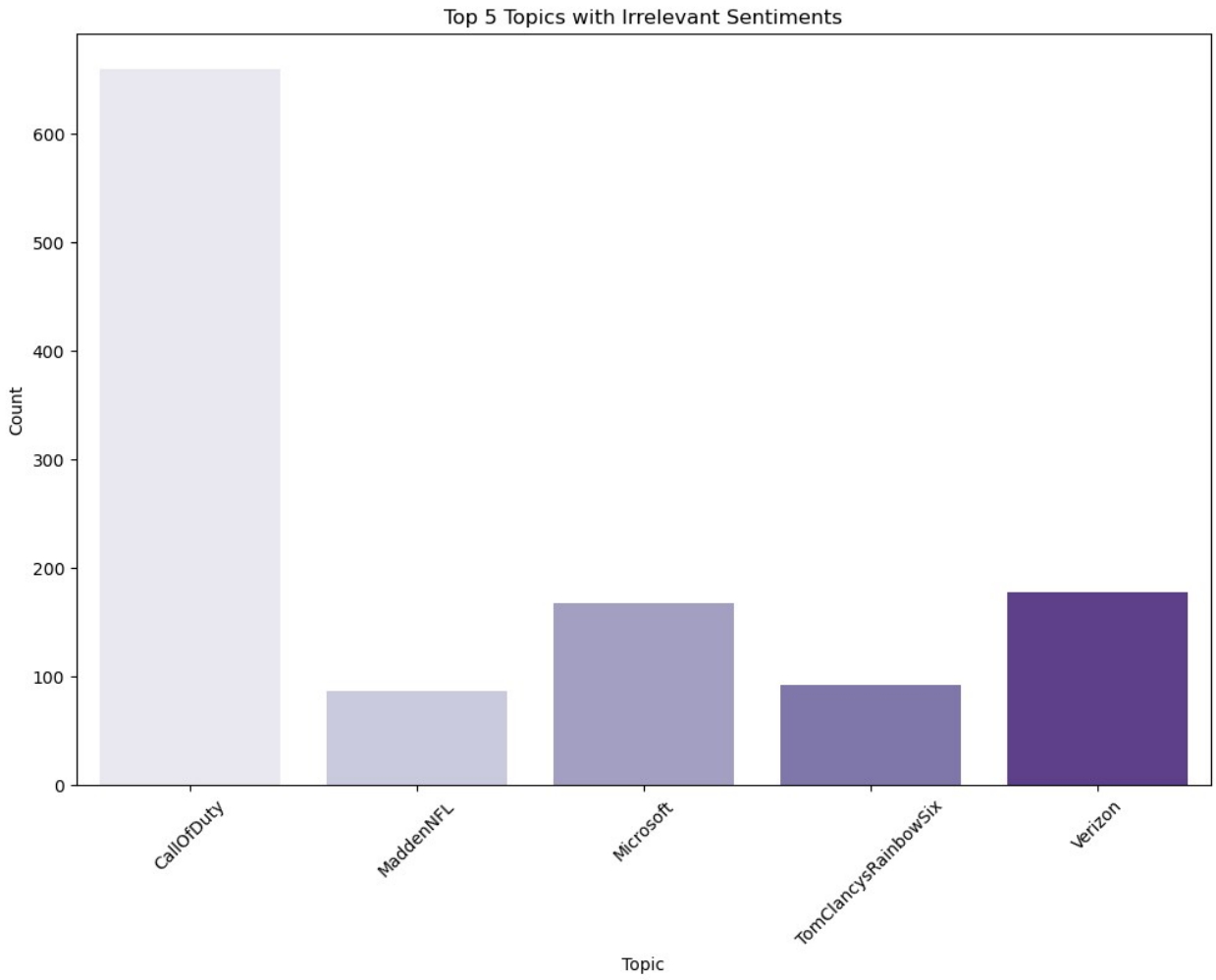
Top 5 Topics with Neutral Sentiments

```
plt.figure(figsize=(12, 8))
sns.barplot(data=top_topics_sentiment[top_topics_sentiment['Sentiment'
] == 'Neutral'], x='Topic', y='Count', palette='Blues')
plt.title('Top 5 Topics with Neutral Sentiments')
plt.xlabel('Topic')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```



Top 5 Topics with Irrelevant Sentiments

```
plt.figure(figsize=(12, 8))
sns.barplot(data=top_topics_sentiment[top_topics_sentiment['Sentiment'
] == 'Irrelevant'], x='Topic', y='Count', palette='Purples')
plt.title('Top 5 Topics with Irrelevant Sentiments')
plt.xlabel('Topic')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```

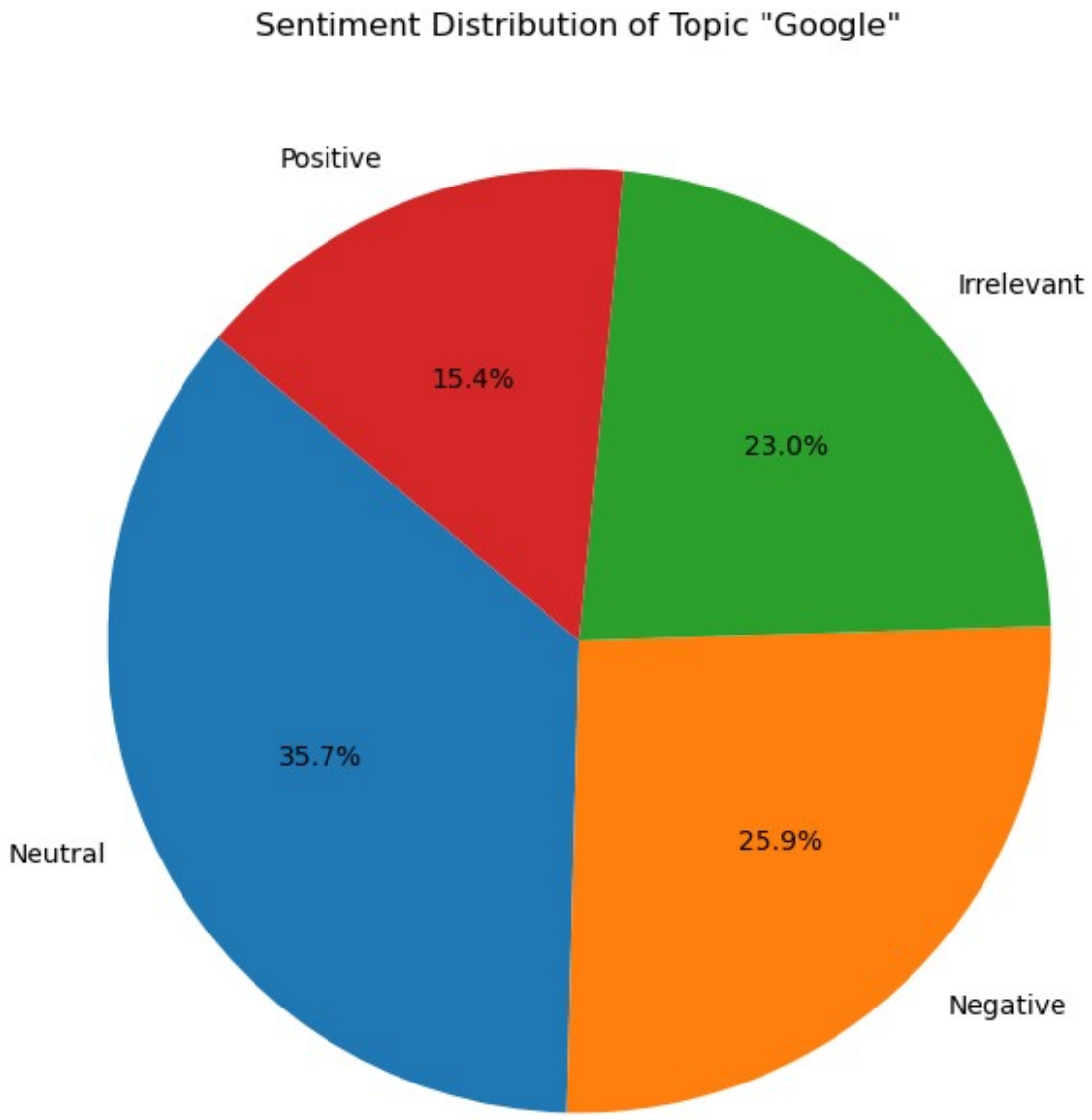


Sentiment Distribution in Google

```
# Filter the dataset to include only entries related to the topic
'Google'
google_data = train[train['Topic'] == 'Google']

# Count the occurrences of each sentiment within the filtered dataset
sentiment_counts = google_data['Sentiment'].value_counts()

# Plot the pie chart
plt.figure(figsize=(8, 8))
plt.pie(sentiment_counts, labels=sentiment_counts.index,
autopct='%1.1f%%', startangle=140)
plt.title('Sentiment Distribution of Topic "Google"')
plt.show()
```



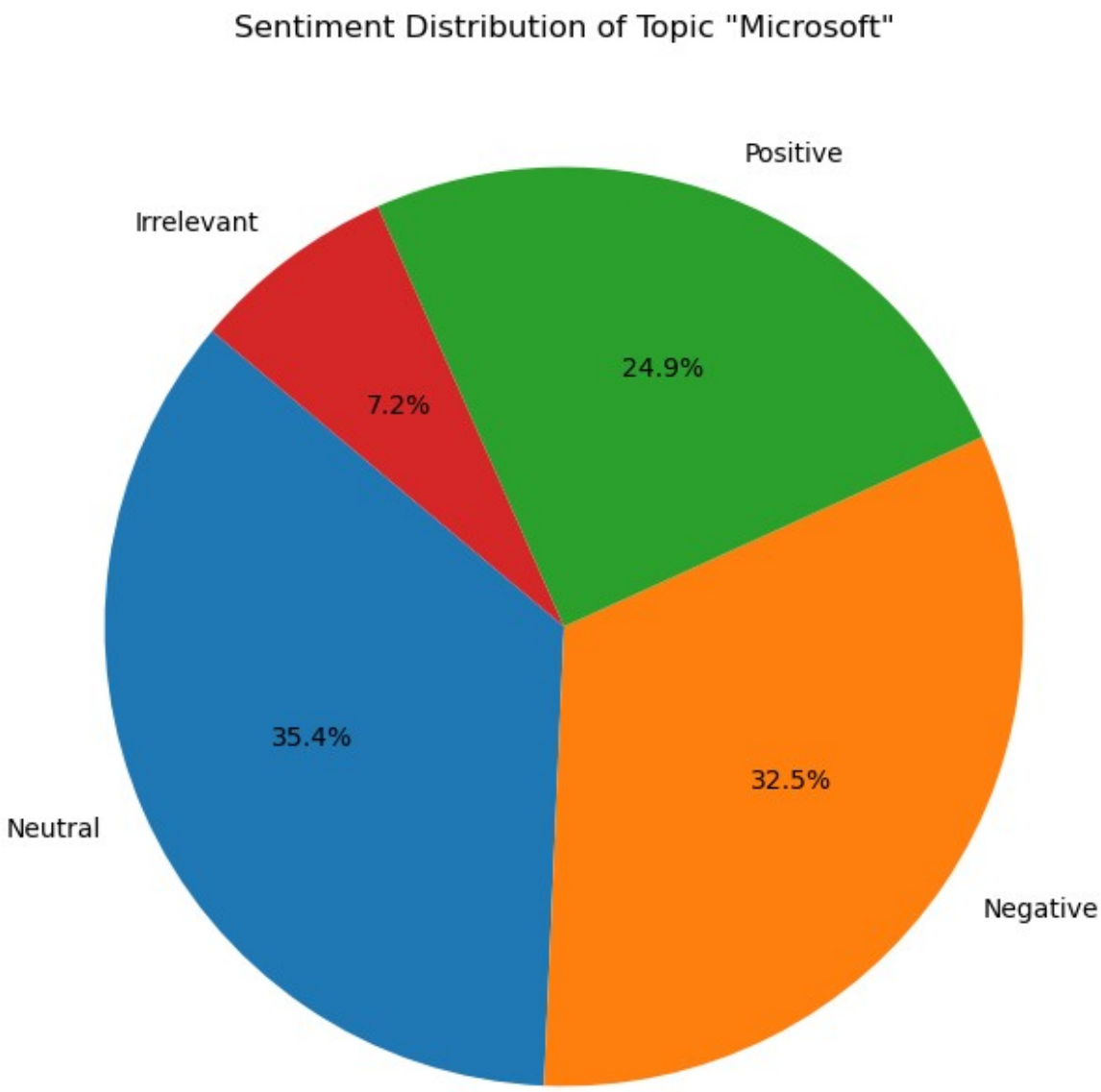
Sentiment Distribution in Microsoft

```
# Filter the dataset to include only entries related to the topic
'Microsoft'
ms_data = train[train['Topic'] == 'Microsoft']

# Count the occurrences of each sentiment within the filtered dataset
sentiment_counts = ms_data['Sentiment'].value_counts()

# Plot the pie chart
plt.figure(figsize=(8, 8))
plt.pie(sentiment_counts, labels=sentiment_counts.index,
```

```
autopct='%1.1f%%', startangle=140)
plt.title('Sentiment Distribution of Topic "Microsoft"')
plt.show()
```




```
train['msg_len'] = train['Text'].apply(len)

train
```

```
         ID        Topic Sentiment  \
0      2401  Borderlands  Positive
1      2401  Borderlands  Positive
```

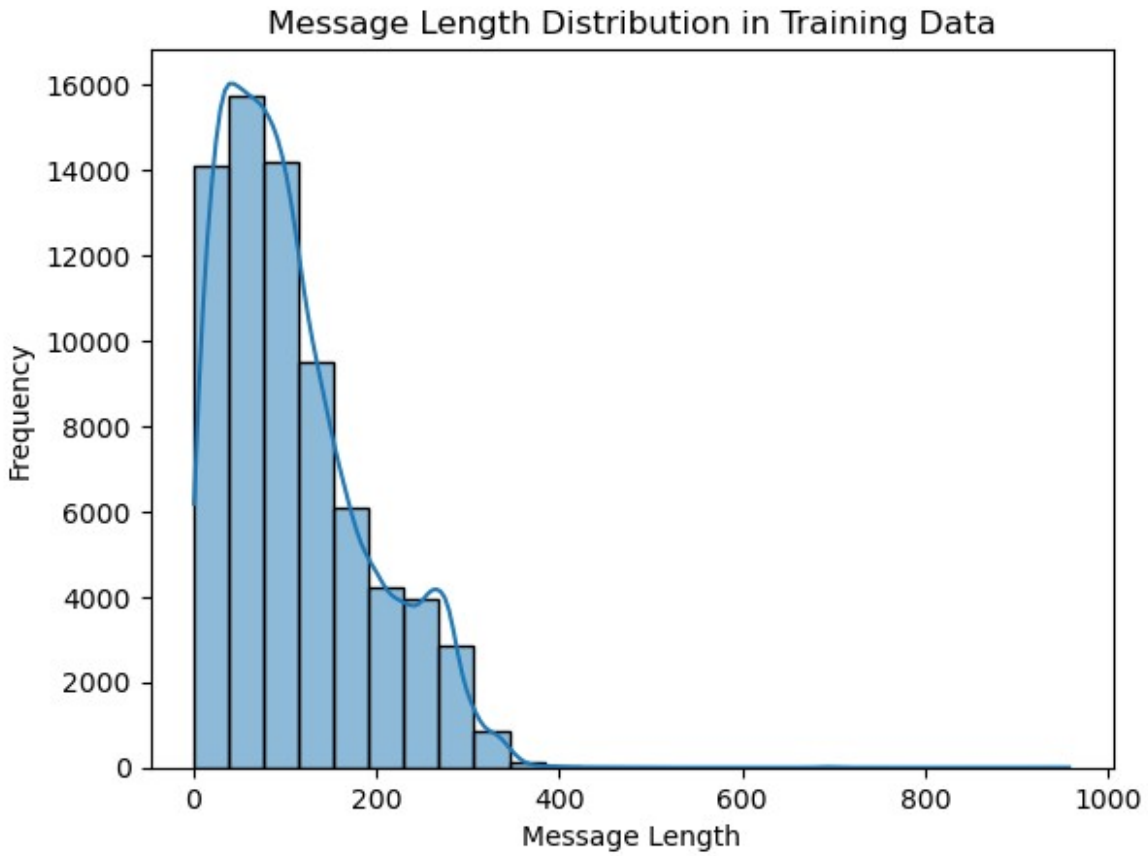
```
2      2401  Borderlands  Positive
3      2401  Borderlands  Positive
4      2401  Borderlands  Positive
...     ...          ...       ...
74677  9200       Nvidia  Positive
74678  9200       Nvidia  Positive
74679  9200       Nvidia  Positive
74680  9200       Nvidia  Positive
74681  9200       Nvidia  Positive
```

```
                                                    Text  msg_len
0      im getting on borderlands and i will murder yo...       53
1      I am coming to the borders and I will kill you...       51
2      im getting on borderlands and i will kill you ...       50
3      im coming on borderlands and i will murder you...       51
4      im getting on borderlands 2 and i will murder ...       57
...                                                  ...      ...
74677  Just realized that the Windows partition of my...      128
74678  Just realized that my Mac window partition is ...      117
74679  Just realized the windows partition of my Mac ...      125
74680  Just realized between the windows partition of...      159
74681  Just like the windows partition of my Mac is l...      119

[71656 rows x 5 columns]
```

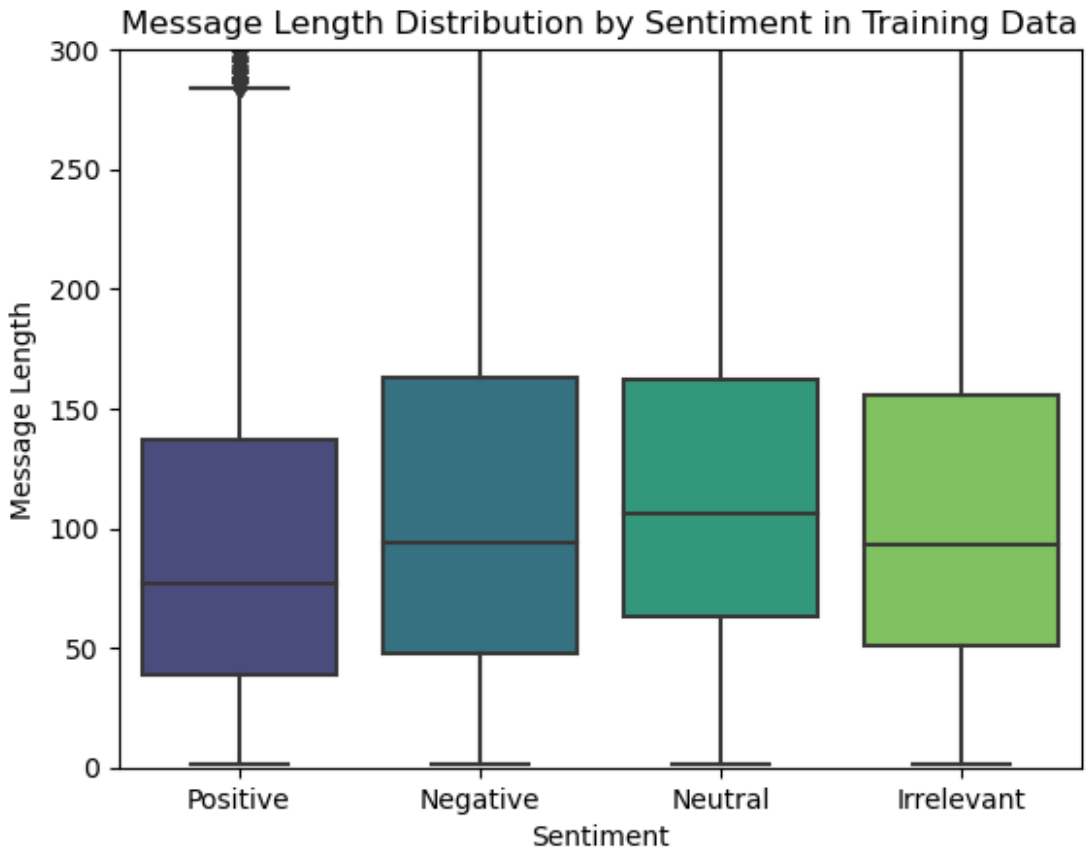
Plot of message length distribution for training data

```
sns.histplot(train['msg_len'], bins=25,kde=True)
plt.title('Message Length Distribution in Training Data')
plt.ylabel('Frequency')
plt.xlabel('Message Length')
plt.show()
```



Plot message length distribution by sentiment for training data

```
sns.boxplot(data=train, x=train['Sentiment'], y='msg_len',
palette='viridis', order=['Positive', 'Negative', 'Neutral',
'Irrelevant'])
plt.title('Message Length Distribution by Sentiment in Training Data')
plt.ylabel('Message Length')
plt.xlabel('Sentiment')
plt.ylim(0,300)
plt.show()
```

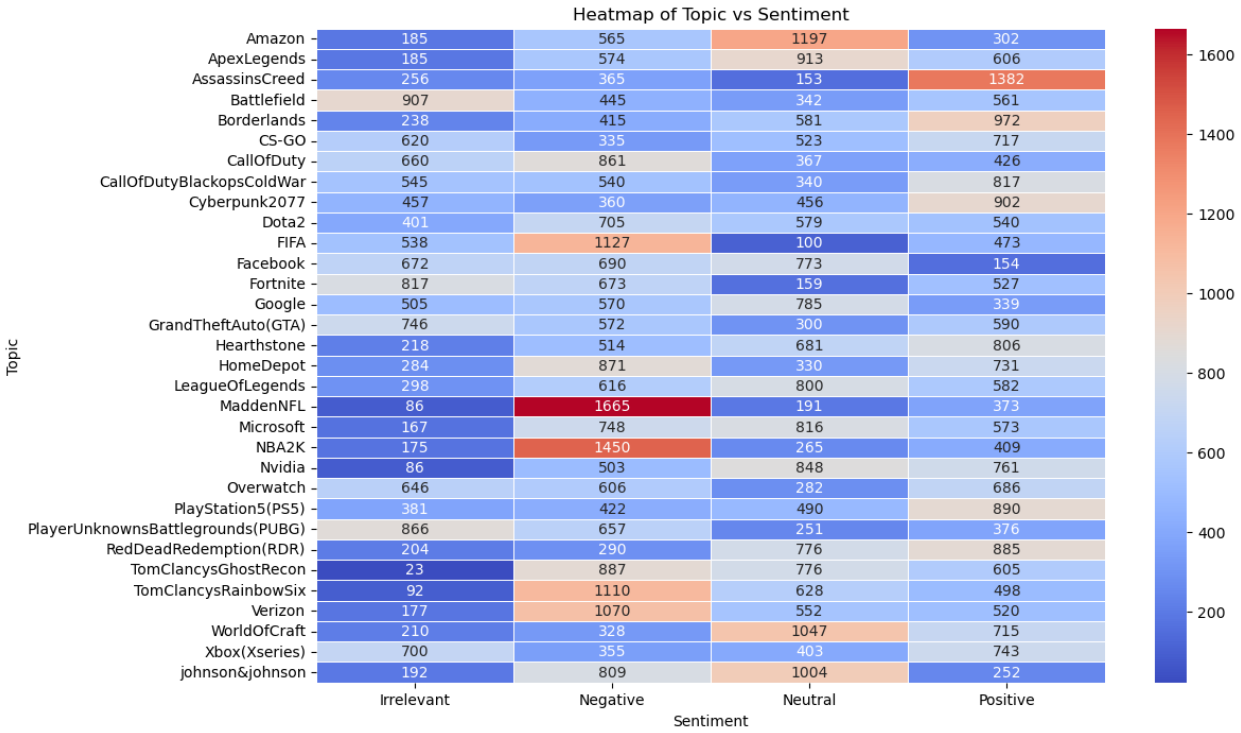



```
# Create the crosstab
crosstab = pd.crosstab(index=train['Topic'],
columns=train['Sentiment'])

# Plot the heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(crosstab, cmap='coolwarm', annot=True, fmt='d',
linewidths=.5)

# Add labels and title
plt.title('Heatmap of Topic vs Sentiment')
plt.xlabel('Sentiment')
plt.ylabel('Topic')

# Show the plot
plt.show()
```

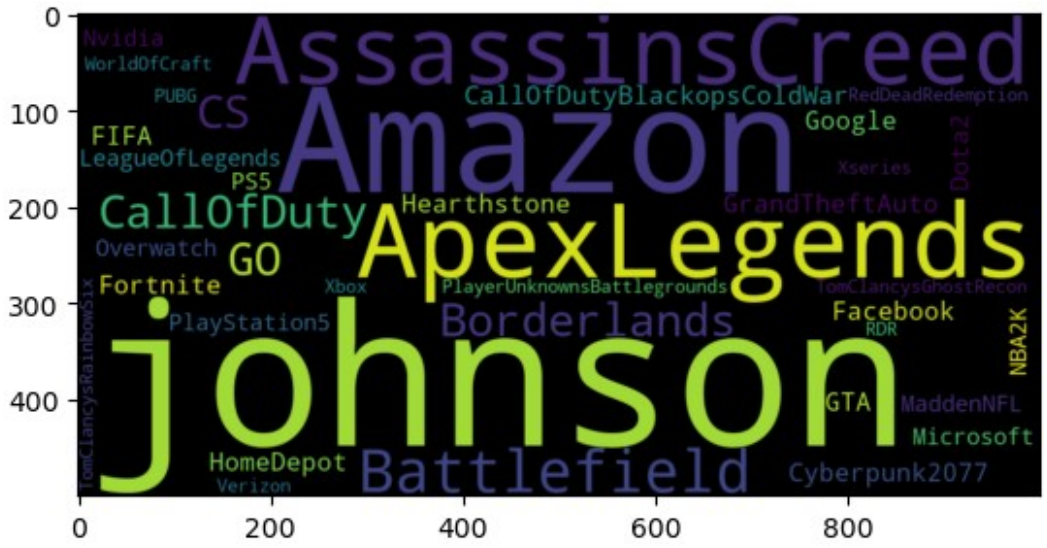



```
topic_list = ' '.join(crosstab.index)

wc = WordCloud(width=1000, height=500).generate(topic_list)

plt.imshow(wc, interpolation='bilinear')
```

```
<matplotlib.image.AxesImage at 0x211c2877a10>
```

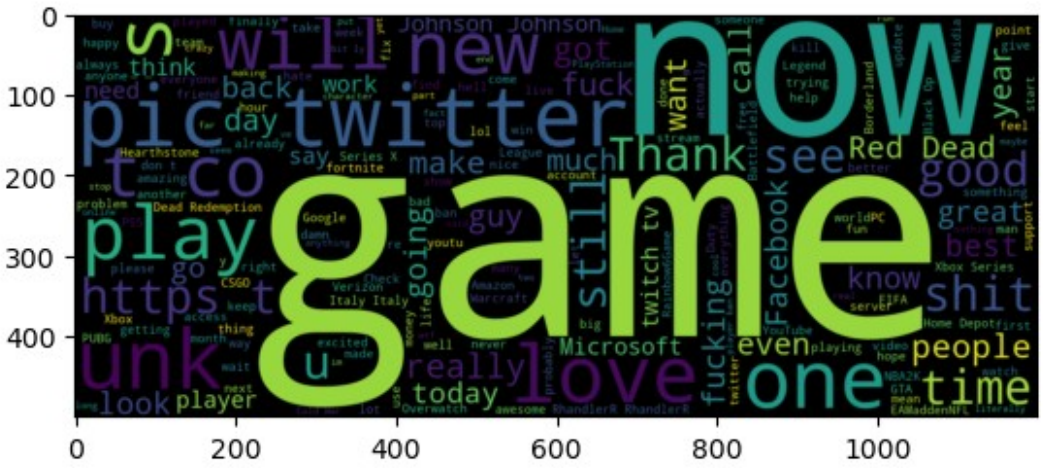


```
corpus = ' '.join(train['Text'])

wc2 = WordCloud(width=1200, height=500).generate(corpus)

plt.imshow(wc2, interpolation='bilinear')

<matplotlib.image.AxesImage at 0x211c5b358d0>
```



`Conclusion:`

Based on the observations from the Twitter sentiment analysis task, several key insights can be drawn:

1. **Most Frequent Topic**: The topic "TomClancyRainbowSix" emerges as the most frequent topic of discussion among the analyzed Twitter data. This suggests a significant level of engagement or interest in this particular topic within the Twitter community.

2. **Sentiment Distribution**: The sentiment analysis reveals that the majority of topics exhibit a negative sentiment, accounting for 30.3% of the sentiments observed. Following negative sentiment, positive sentiment is the next most prevalent, comprising 27.5% of the sentiments. Neutral sentiment closely follows at 24.7%, indicating a relatively balanced distribution between positive and neutral sentiments. Irrelevant sentiments, although less prevalent, still constitute a notable portion at 17.5%.

3. **Sentiment of Specific Topics**: Notably, topics such as "Google" and "Microsoft" predominantly exhibit a neutral sentiment. This observation suggests that discussions related to these tech giants tend to be more balanced or impartial in nature.

4. **Message Length**: Another noteworthy observation is that the majority of messages analyzed are under 400 words in length. This indicates that Twitter users tend to

convey their sentiments concisely and succinctly within the platform's character limit.

In conclusion, the sentiment analysis provides valuable insights into the prevailing attitudes and opinions within the Twitter community regarding various topics. While negative sentiments appear to be more common overall, there is a diverse range of sentiments expressed across different topics.