#Importing the necessary libraries

In [3]:

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

In [13]:

```
#Reading the dataset
cols=['ID', 'Topic', 'Sentiment', 'Text']
train = pd.read_csv(r"C:\Users\ashwa\Downloads\archive (2)\twitter_trainin
trains = pd.read_csv(r"C:\Users\ashwa\Downloads\archive (2)\twitter_valida
```

In [15]:

```
train.head()
```

Out[15]:

	ID	Topic	Sentiment	Text
0	2401	Borderlands	Positive	im getting on borderlands and i will murder yo...
1	2401	Borderlands	Positive	I am coming to the borders and I will kill you...
2	2401	Borderlands	Positive	im getting on borderlands and i will kill you ...
3	2401	Borderlands	Positive	im coming on borderlands and i will murder you...
4	2401	Borderlands	Positive	im getting on borderlands 2 and i will murder ...

In [16]:

```
trains.head()
```

Out[16]:

	ID	Topic	Sentiment	Text
0	3364	Facebook	Irrelevant	I mentioned on Facebook that I was struggling ...
1	352	Amazon	Neutral	BBC News - Amazon boss Jeff Bezos rejects clai...
2	8312	Microsoft	Negative	@Microsoft Why do I pay for WORD when it funct...
3	4371	CS-GO	Negative	CSGO matchmaking is so full of closet hacking,...
4	4433	Google	Neutral	Now the President is slapping Americans in the...

Information about the dataframe

In [6]:

```
train.shape
```

Out[6]:

```
(74682, 4)
```

In [17]:

```
trains.shape
```

Out[17]:

```
(1000, 4)
```

In [7]:
```
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
```

In [18]:
```
trains.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   ID         1000 non-null   int64 
 1   Topic      1000 non-null   object
 2   Sentiment  1000 non-null   object
 3   Text       1000 non-null   object
dtypes: int64(1), object(3)
memory usage: 31.4+ KB
```

In [8]:
```
train.describe(include=object)
```

Out[8]:

	Topic	Sentiment	Text
count	74682	74682	73996
unique	32	4	69491
top	TomClancysRainbowSix	Negative	At the same time, despite the fact that there ...
freq	2400	22542	172

In [19]:
```
trains.describe(include=object)
```

Out[19]:

	Topic	Sentiment	Text
count	1000	1000	1000
unique	32	4	999
top	RedDeadRedemption(RDR)	Neutral	Wow
freq	40	285	2

In [9]:
```
train['Sentiment'].unique()
```

Out[9]:
```
array(['Positive', 'Neutral', 'Negative', 'Irrelevant'], dtype=object)
```

```
In [20]: trains['Sentiment'].unique()
```

```
Out[20]: array(['Irrelevant', 'Neutral', 'Negative', 'Positive'], dtype=object)
```

Checking for null/missing values in the dataset

```
In [10]: train.isnull().sum()
```

```
Out[10]: ID             0
         Topic          0
         Sentiment      0
         Text         686
         dtype: int64
```

```
In [21]: trains.isnull().sum()
```

```
Out[21]: ID           0
         Topic        0
         Sentiment    0
         Text         0
         dtype: int64
```

```
In [22]: train.dropna(inplace=True)
```

```
In [23]: trains.dropna(inplace=True)
```

Checking for duplicate values

```
In [25]: train.duplicated().sum()
```

```
Out[25]: 2340
```

```
In [26]: trains.duplicated().sum()
```

```
Out[26]: 0
```

```
In [27]: train.drop_duplicates(inplace=True)
```

```
In [28]: trains.drop_duplicates(inplace=True)
```

```
In [29]: train.duplicated().sum()
```

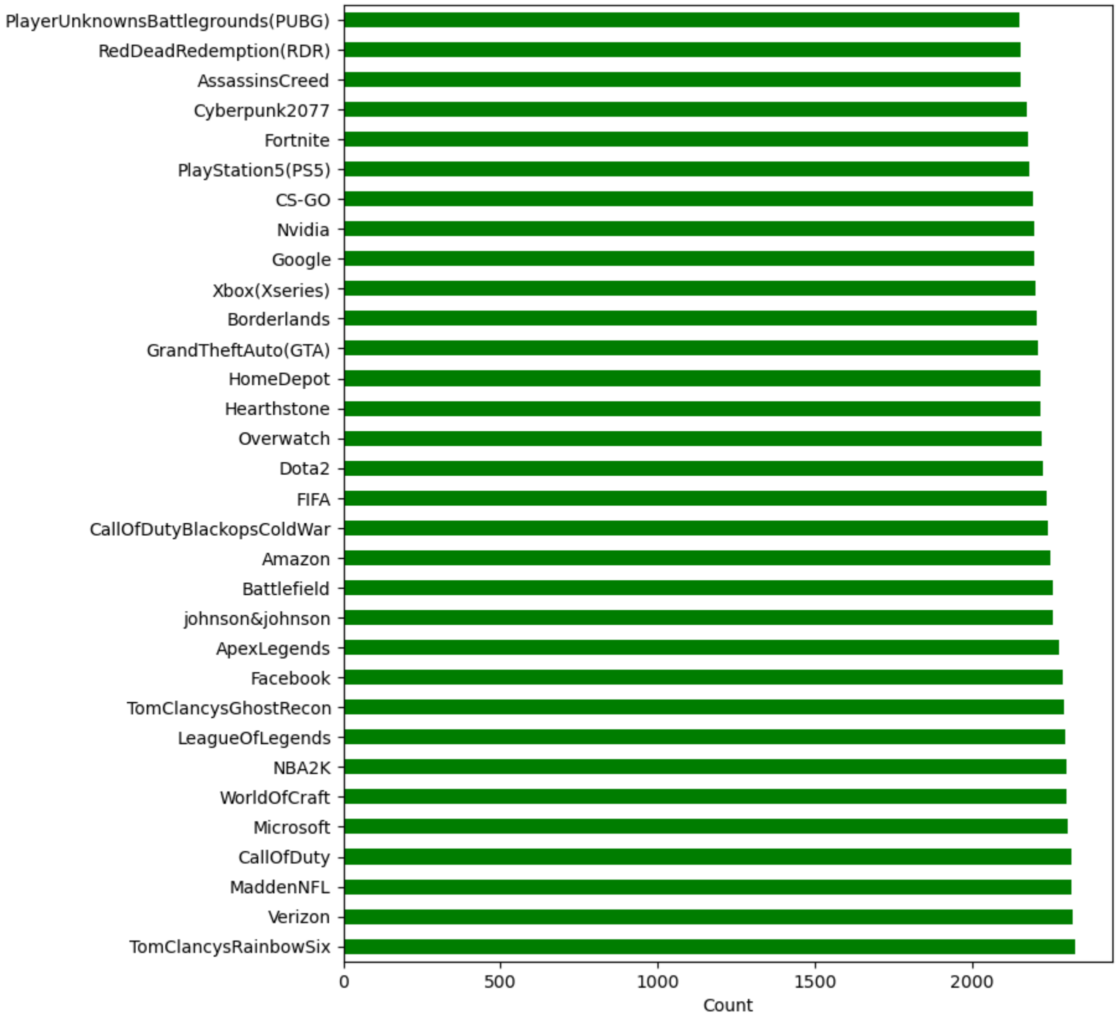
```
Out[29]: 0
```

```
In [30]: trains.duplicated().sum()
```

```
Out[30]: 0
```

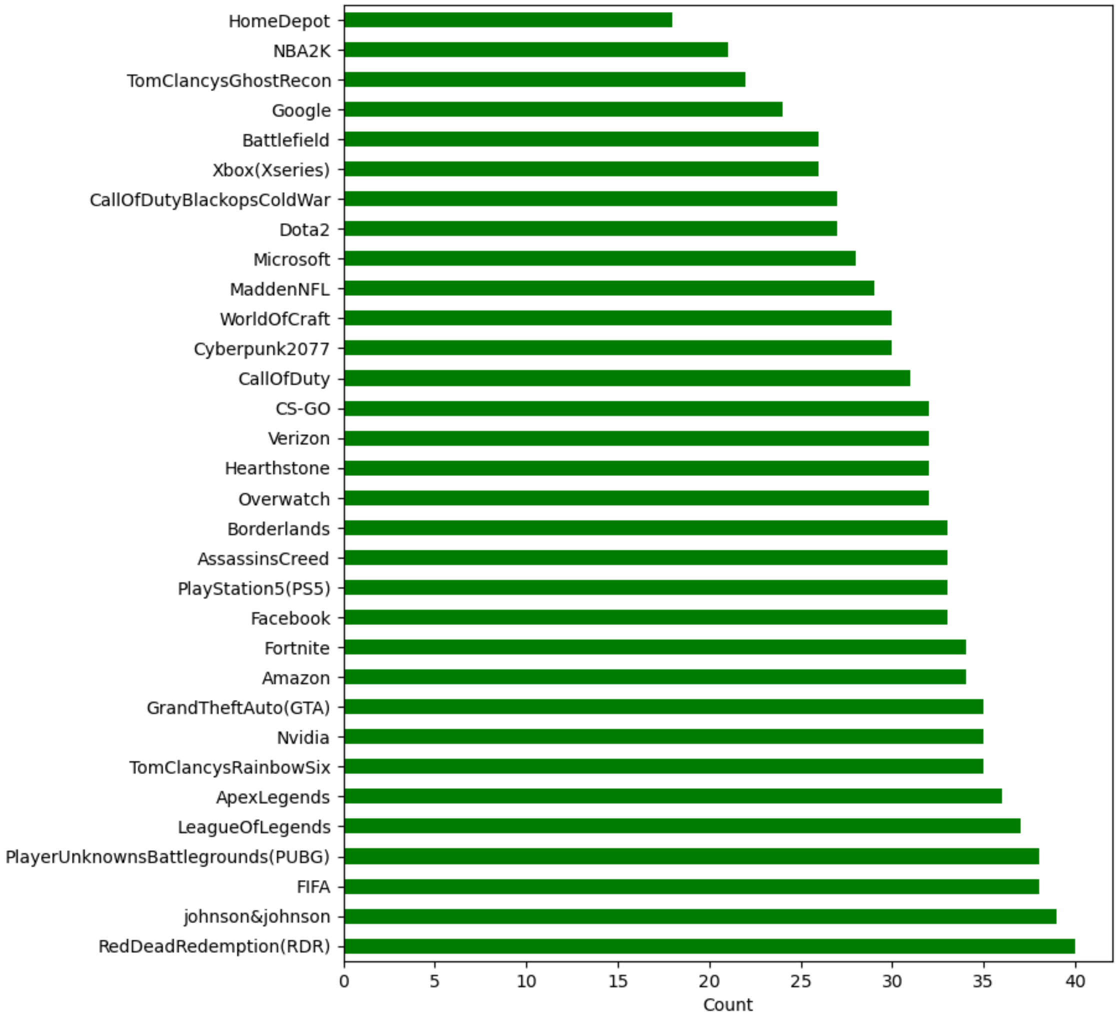
Visualization of count of different topics

```
In [31]: plt.figure(figsize=(8,10))
         train['Topic'].value_counts().plot(kind='barh',color='g')
         plt.xlabel("Count")
         plt.show()
```



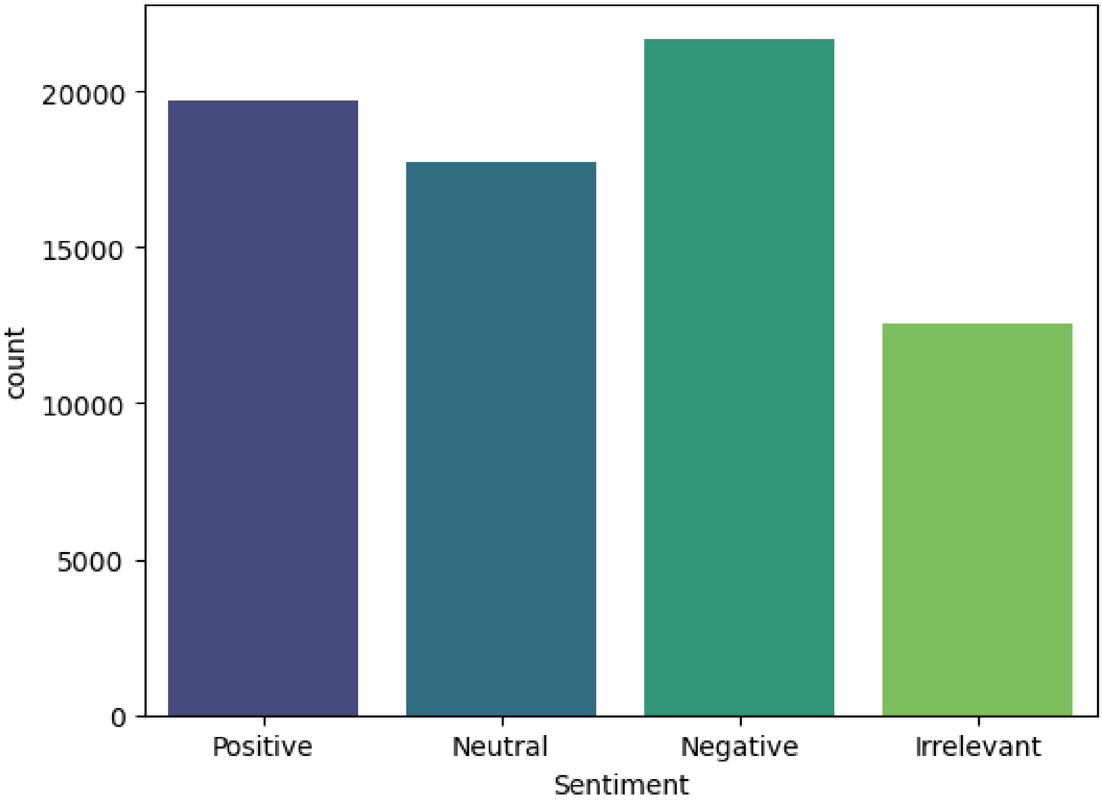
In [32]:

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

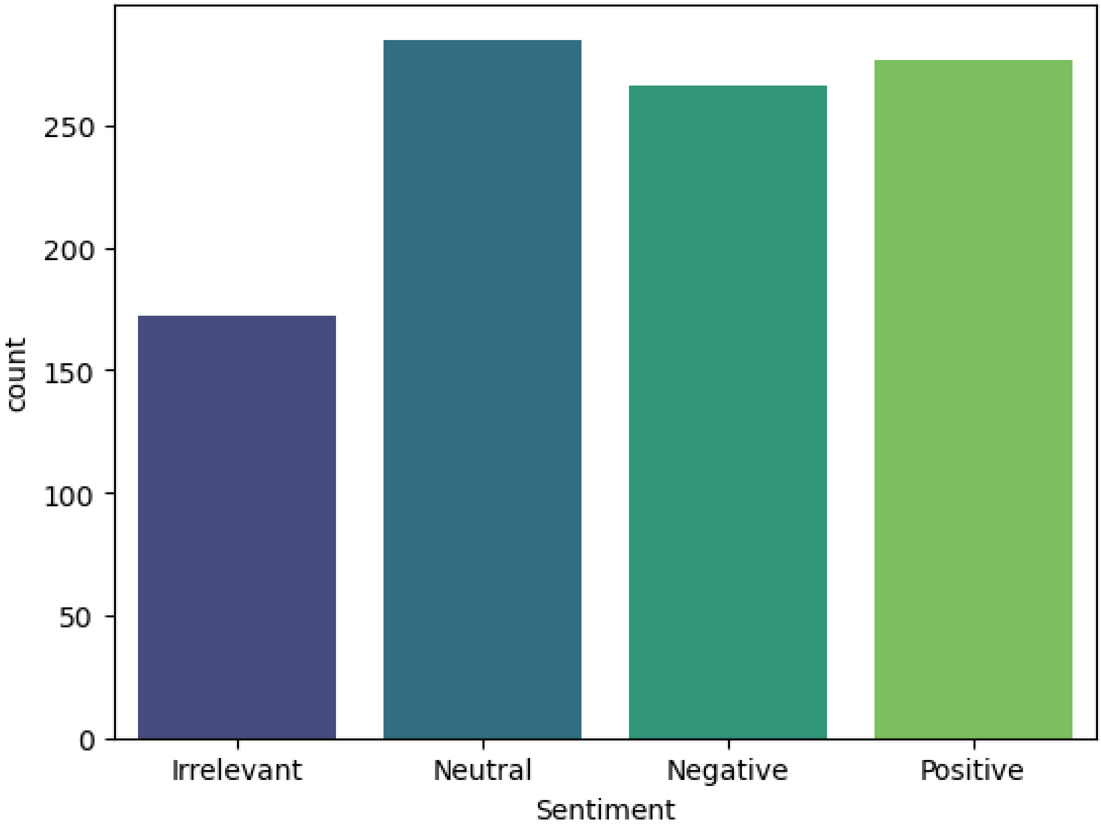


Sentiment Distribution

In [117]:
```
sns.countplot(x = 'Sentiment',data=train,palette='viridis')
plt.show()
```



In [34]:
```
sns.countplot(x = 'Sentiment',data=trains,palette='viridis')
plt.show()
```



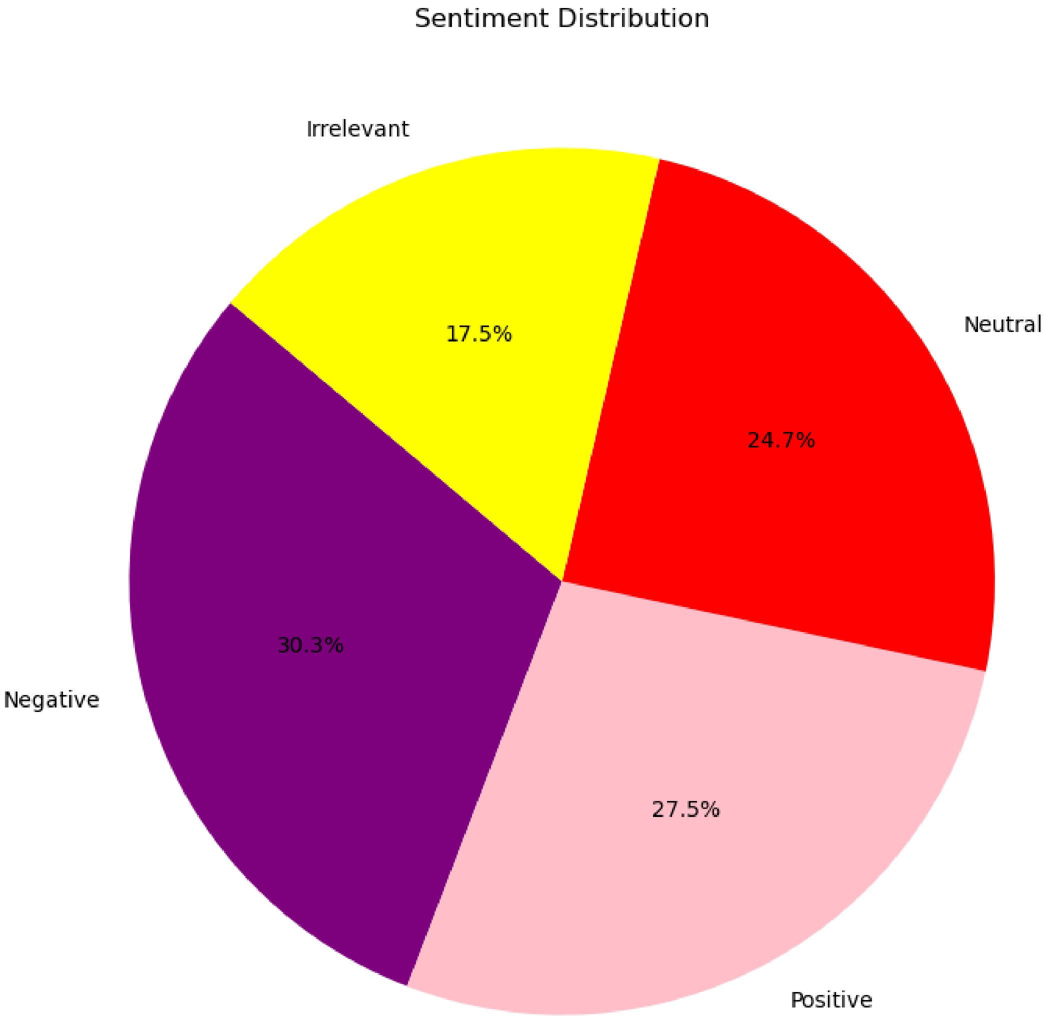
In [45]:

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

# Create the pie chart
plt.figure(figsize=(9, 9))
plt.pie(sentiment_counts, labels=sentiment_counts.index, autopct="%1.1f%%"

plt.title('Sentiment Distribution')

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

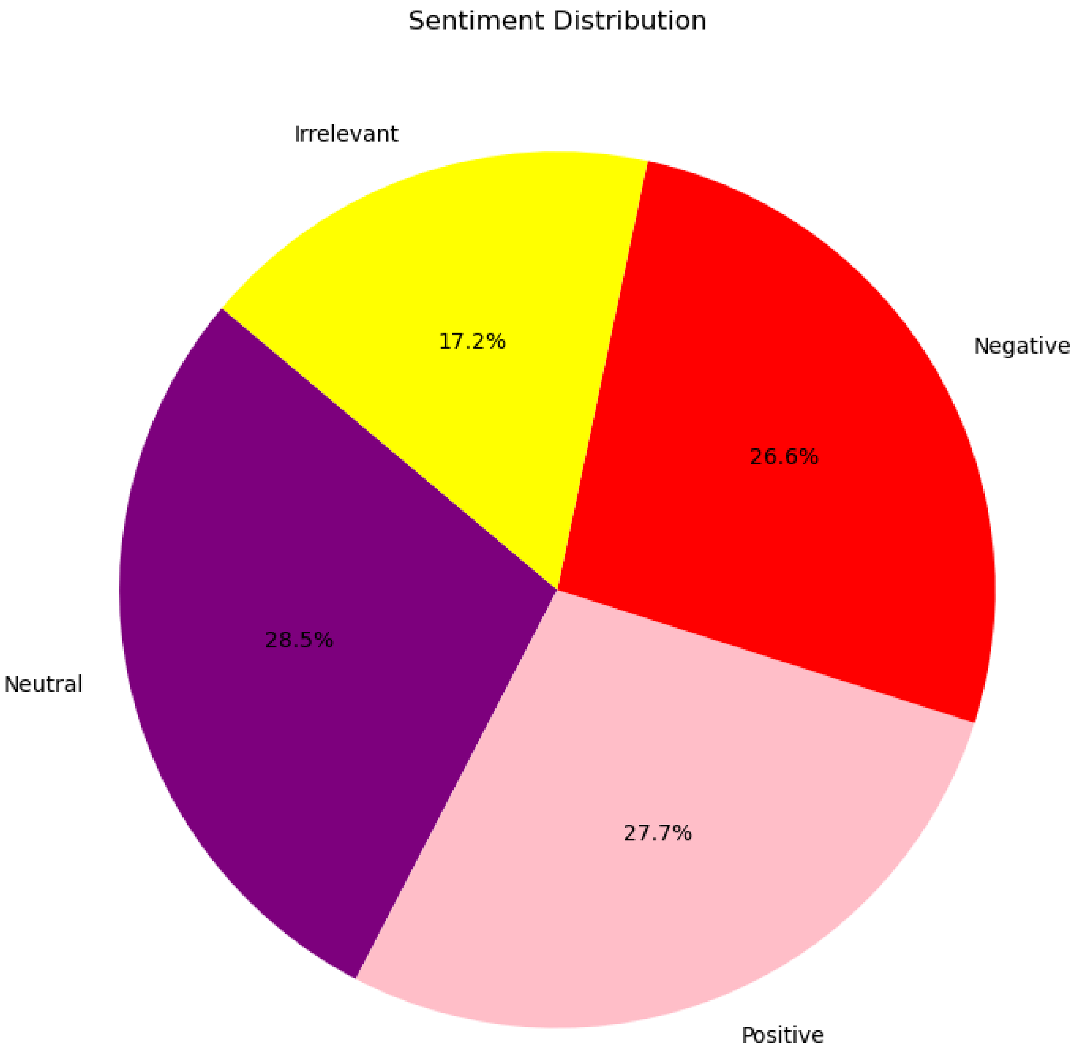


In [46]:
```
# Calculate the counts for each sentiment
sentiment_counts = trains['Sentiment'].value_counts()

# Create the pie chart
plt.figure(figsize=(9, 9))
plt.pie(sentiment_counts, labels=sentiment_counts.index, autopct="%1.1f%%"

plt.title('Sentiment Distribution')

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



train

In [42]:
```
trains
```

Out[42]:

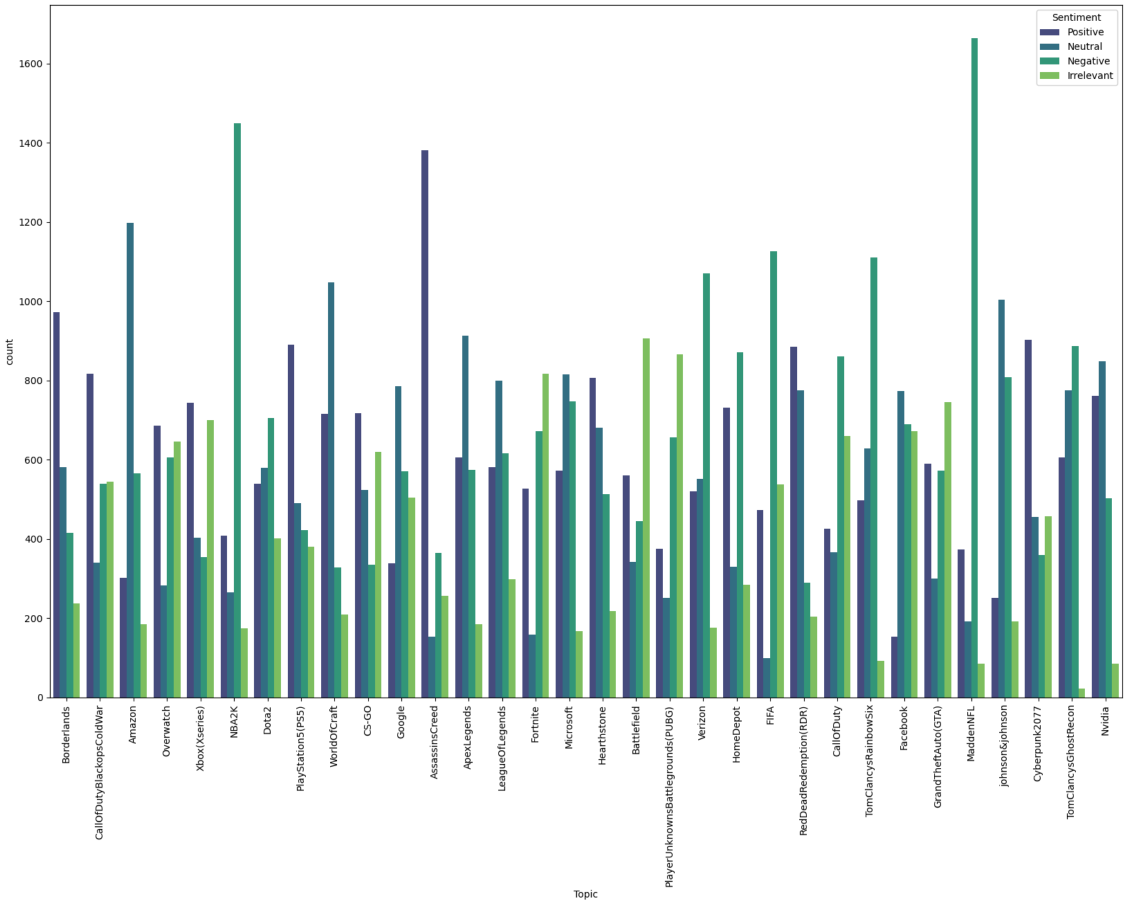
	ID	Topic	Sentiment	Text
0	3364	Facebook	Irrelevant	I mentioned on Facebook that I was struggling ...
1	352	Amazon	Neutral	BBC News - Amazon boss Jeff Bezos rejects clai...
2	8312	Microsoft	Negative	@Microsoft Why do I pay for WORD when it funct...
3	4371	CS-GO	Negative	CSGO matchmaking is so full of closet hacking,...
4	4433	Google	Neutral	Now the President is slapping Americans in the...
...	...	...	...	...
995	4891	GrandTheftAuto(GTA)	Irrelevant	⭐ Toronto is the arts and culture capital of ...
996	4359	CS-GO	Irrelevant	tHIS IS ACTUALLY A GOOD MOVE TOT BRING MORE VI...
997	2652	Borderlands	Positive	Today sucked so it’s time to drink wine n play...
998	8069	Microsoft	Positive	Bought a fraction of Microsoft today. Small wins.
999	6960	johnson&johnson	Neutral	Johnson & Johnson to stop selling talc baby po...

1000 rows × 4 columns

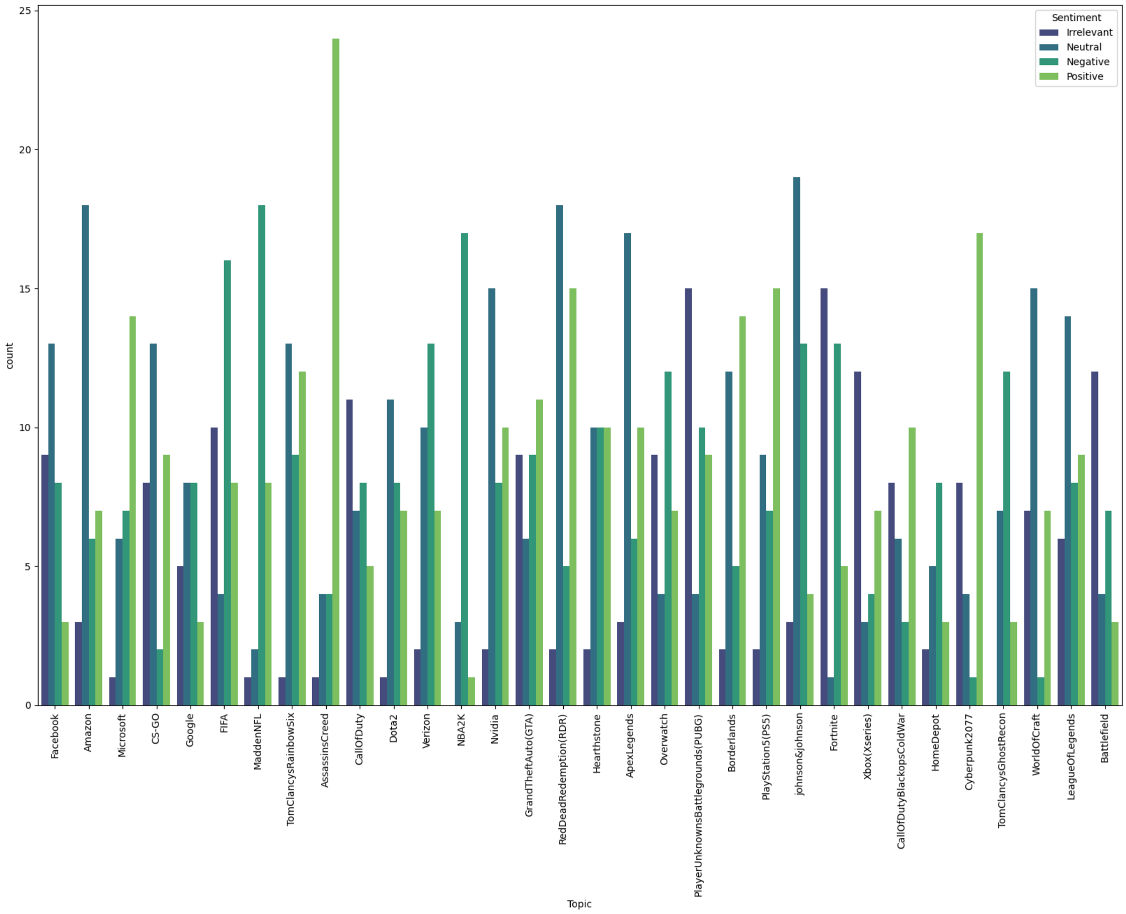
Sentiment Distribution Topic-wise

In [56]:

```
plt.figure(figsize=(20,13))
sns.countplot(x='Topic',data=train,palette='viridis',hue='Sentiment')
plt.xticks(rotation=90)
plt.show()
```



In [57]:
```
plt.figure(figsize=(20,13))
sns.countplot(x='Topic',data=trains,palette='viridis',hue='Sentiment')
plt.xticks(rotation=90)
plt.show()
```

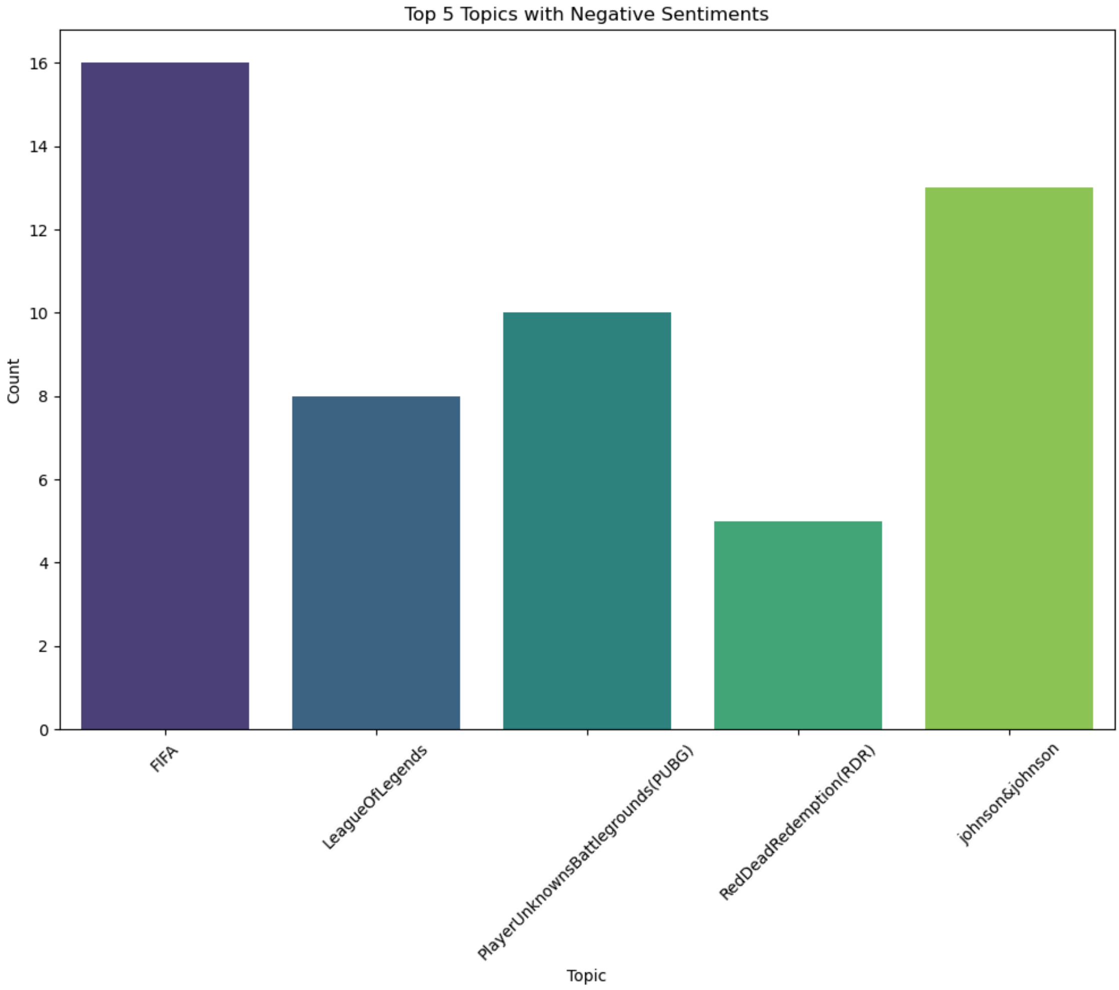


In [98]:
```
## Group by Topic and Sentiment
topic_wise_sentiment = trains.groupby(["Topic", "Sentiment"]).size().reset_

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

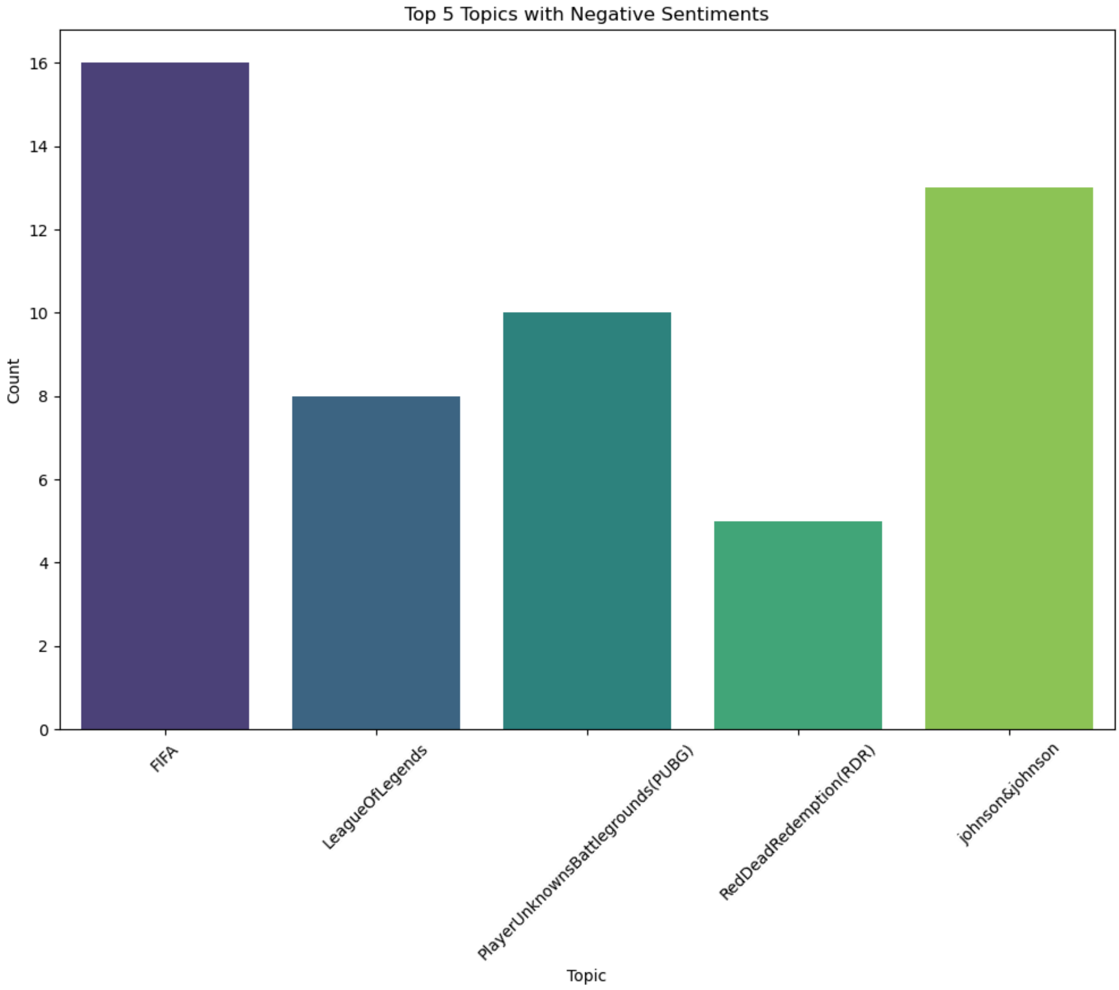
In [99]:

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

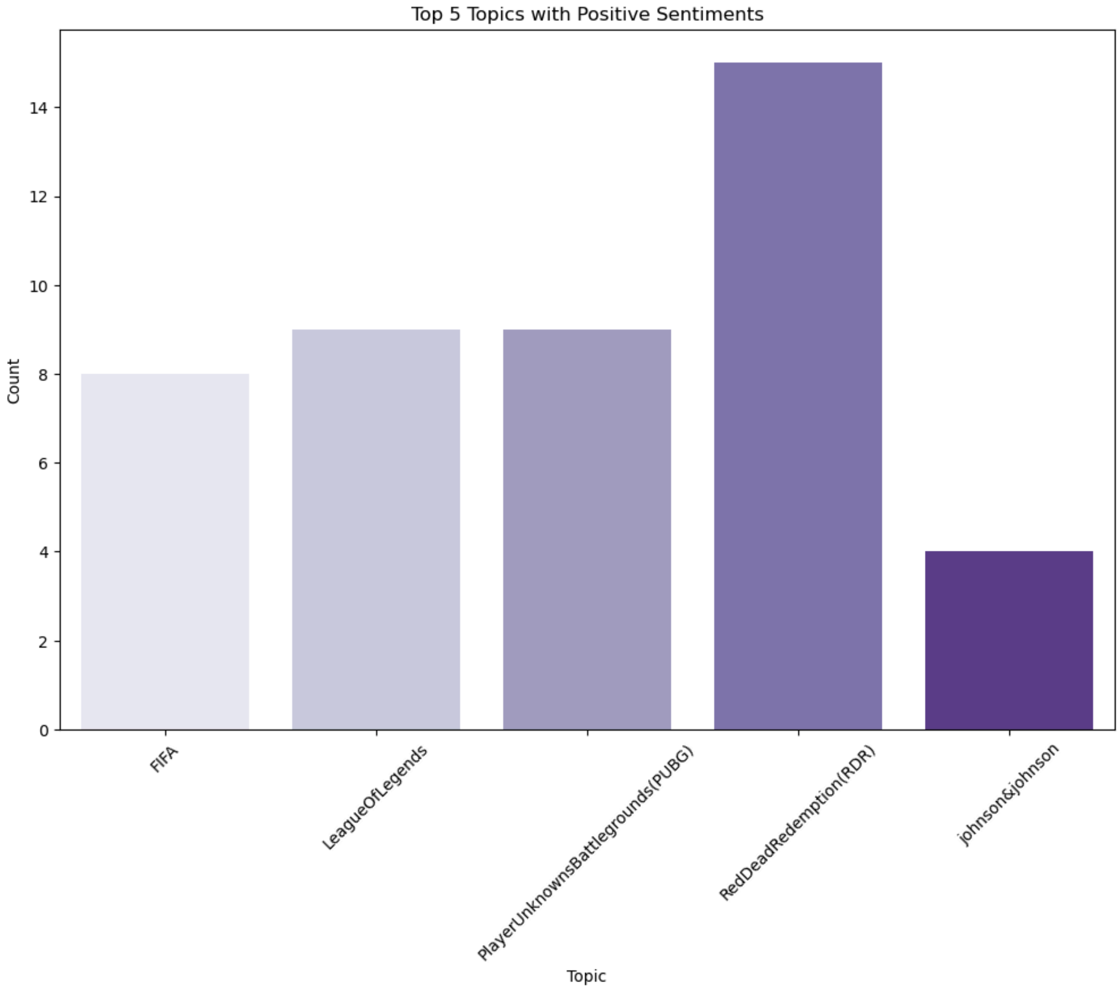


In [100]:

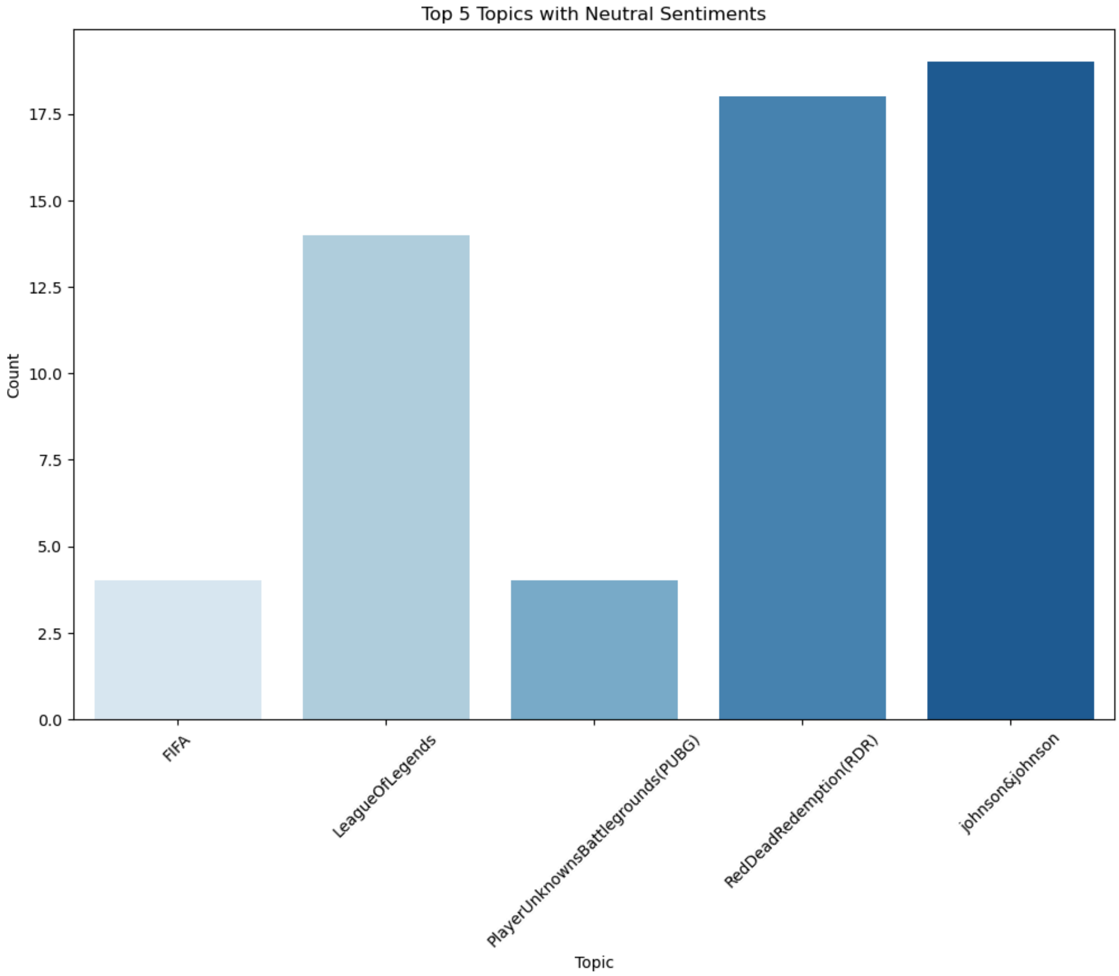
```
#Top 5 Topics with Negative Sentiments
plt.figure(figsize=(12, 8))
sns.barplot(data=top_topics_sentiment[top_topics_sentiment['Sentiment'] ==
plt.title('Top 5 Topics with Negative Sentiments')
plt.xlabel('Topic')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```



In [101]:
```
#Top 5 Topics with Positive Sentiments
plt.figure(figsize=(12, 8))
sns.barplot(data=top_topics_sentiment[top_topics_sentiment['Sentiment'] ==
plt.title('Top 5 Topics with Positive Sentiments')
plt.xlabel('Topic')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```

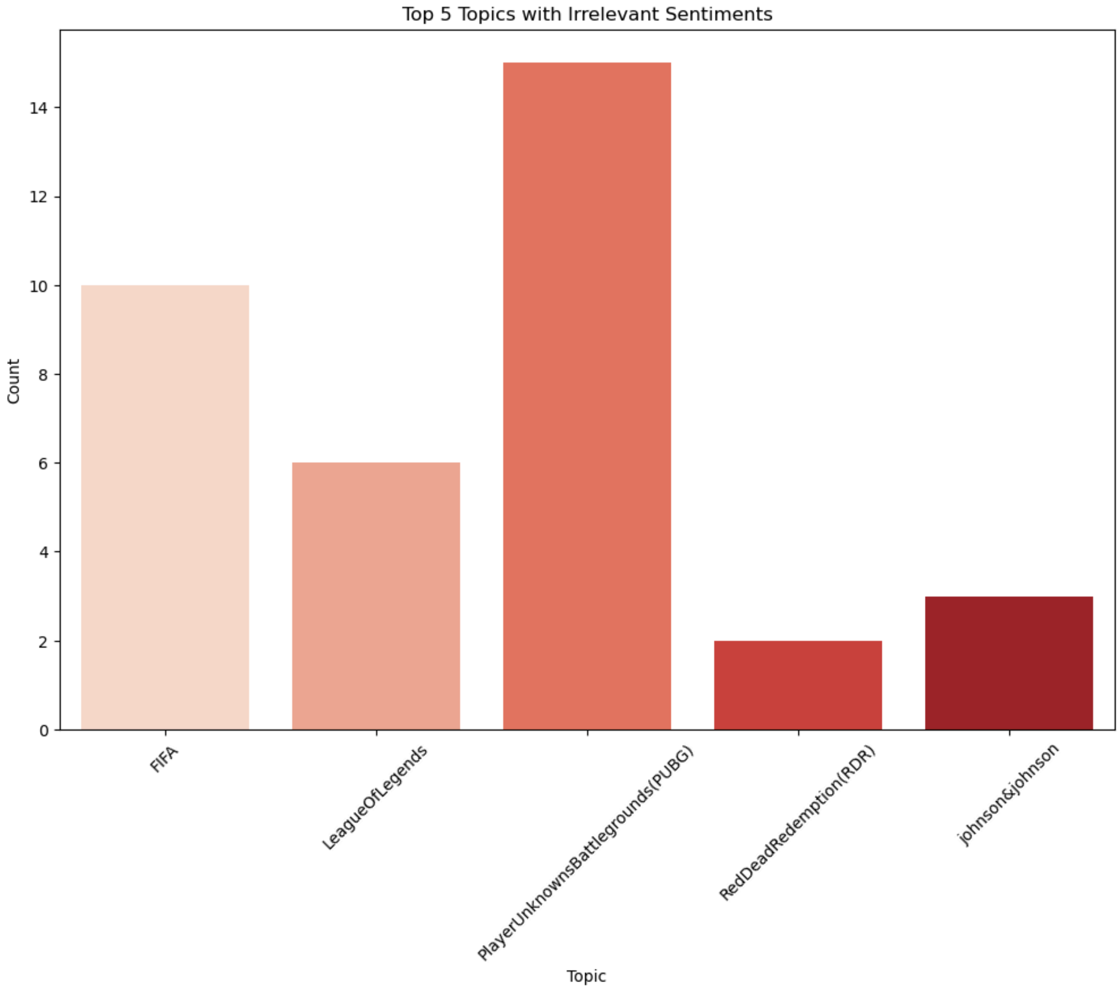


In [102]:
```
#Top 5 Topics with Neutral Sentiments
plt.figure(figsize=(12, 8))
sns.barplot(data=top_topics_sentiment[top_topics_sentiment['Sentiment'] ==
plt.title('Top 5 Topics with Neutral Sentiments')
plt.xlabel('Topic')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```



In [103]:

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

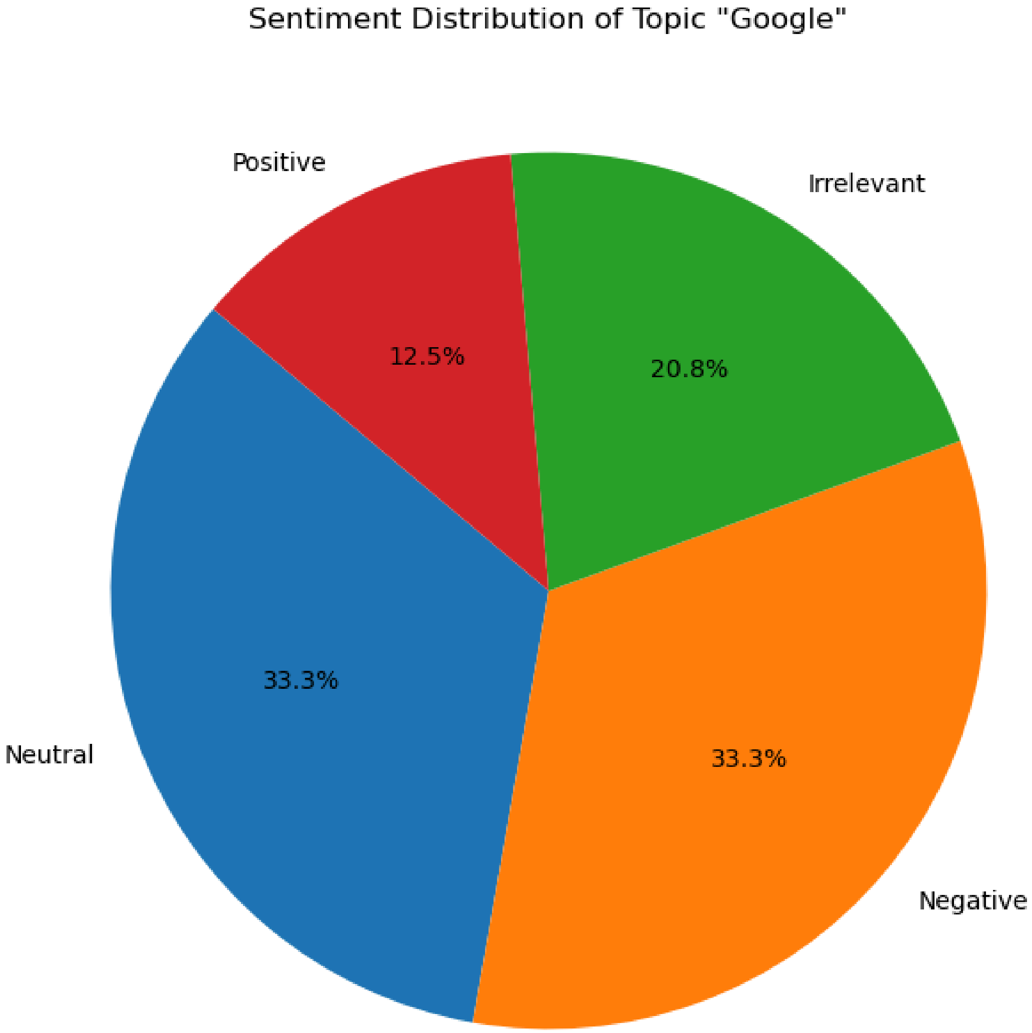


In [104]:

```
#Sentiment Distribution in Google
# Filter the dataset to include only entries related to the topic 'Google'
google_data = trains[trains['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%%'
plt.title('Sentiment Distribution of Topic "Google"')
plt.show()
```

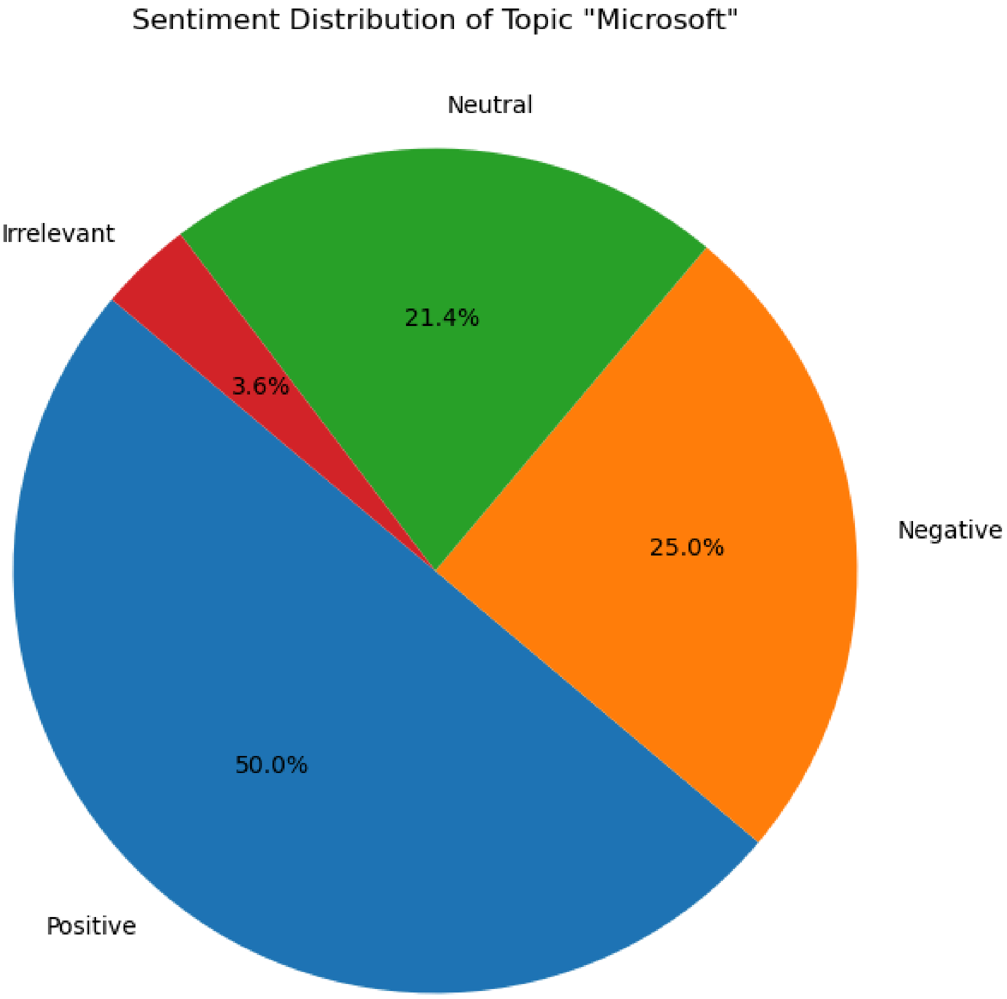


In [105]:
```
#Sentiment Distribution in Microsoft

# Filter the dataset to include only entries related to the topic 'Microso
ms_data = trains[trains['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%%'
plt.title('Sentiment Distribution of Topic "Microsoft"')
plt.show()
```



In [106]:
```
trains['msg_len'] = trains['Text'].apply(len)
```

In [107]:
```
trains
```

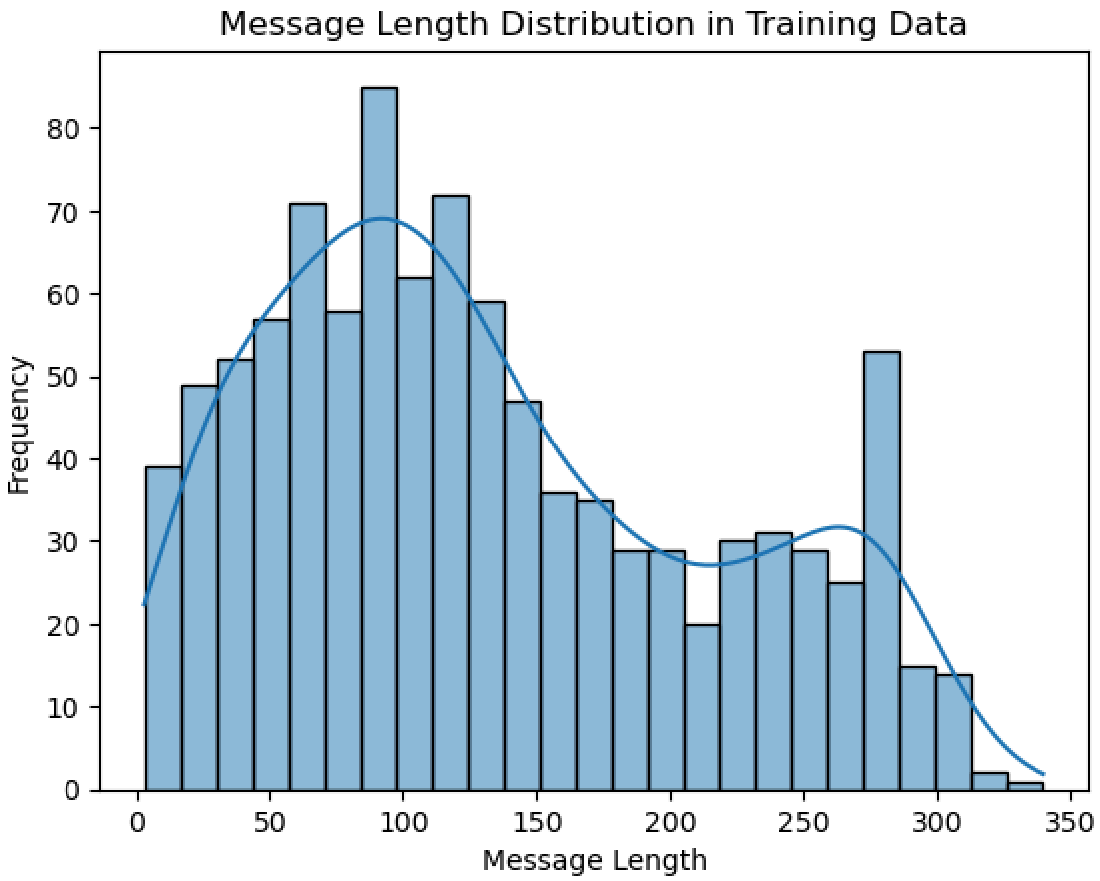
Out[107]:
```
       ID              Topic  Sentiment                                                Text  msg_len
  0  3364           Facebook  Irrelevant     I mentioned on Facebook that I was struggling ...      242
  1   352             Amazon     Neutral     BBC News - Amazon boss Jeff Bezos rejects clai...      109
  2  8312          Microsoft    Negative     @Microsoft Why do I pay for WORD when it funct...       91
  3  4371             CS-GO     Negative    CSGO matchmaking is so full of closet hacking,...       71
  4  4433             Google     Neutral    Now the President is slapping Americans in the...      170
...   ...                ...         ...                                                 ...      ...
995  4891  GrandTheftAuto(GTA)  Irrelevant  ⭐ Toronto is the arts and culture capital of ...      281
996  4359             CS-GO  Irrelevant    tHIS IS ACTUALLY A GOOD MOVE TOT BRING MORE VI...      248
997  2652        Borderlands    Positive    Today sucked so it’s time to drink wine n play...      120
998  8069          Microsoft    Positive    Bought a fraction of Microsoft today. Small wins.       49
999  6960    johnson&johnson     Neutral    Johnson & Johnson to stop selling talc baby po...      116

1000 rows × 5 columns
```

In [108]:

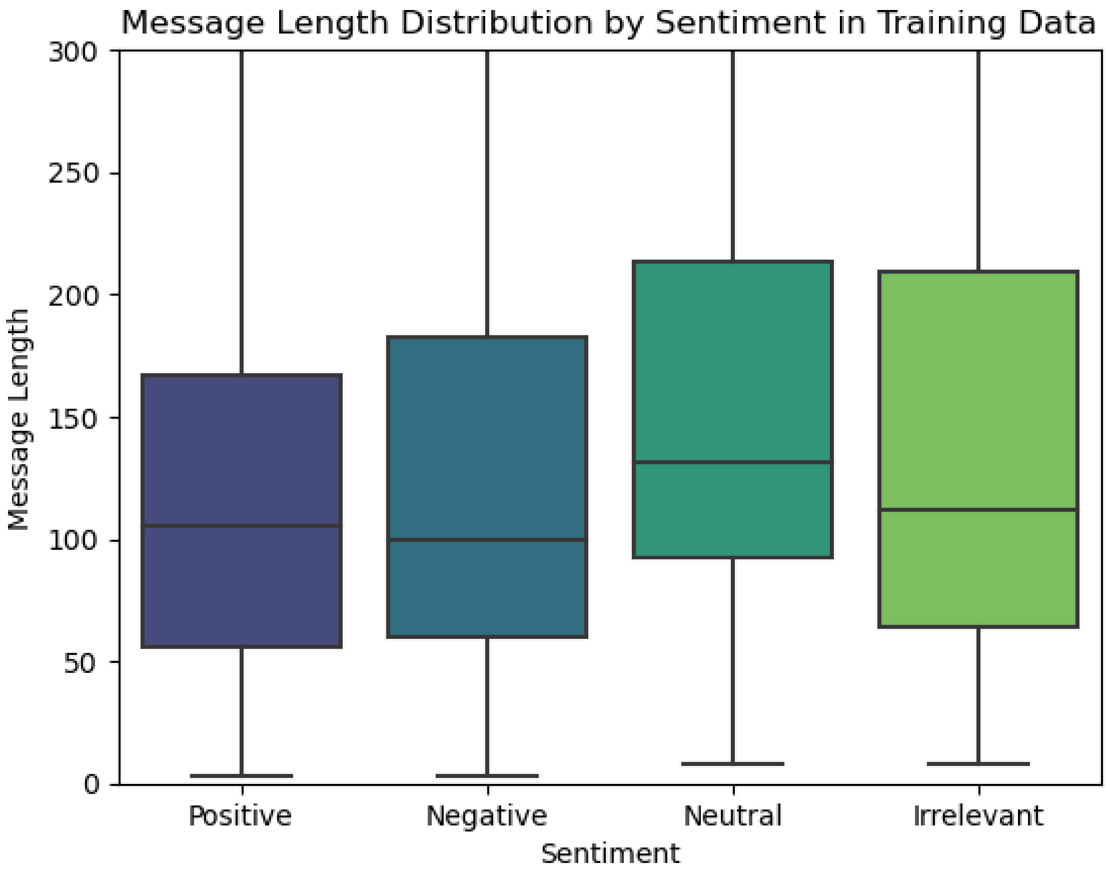
```
#Plot of message length distribution for training data

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



In [110]:
```
#Plot message length distribution by sentiment for training data


sns.boxplot(data=trains, x=trains['Sentiment'], y='msg_len', palette='viri
plt.title('Message Length Distribution by Sentiment in Training Data')
plt.ylabel('Message Length')
plt.xlabel('Sentiment')
plt.ylim(0,300)
plt.show()
```

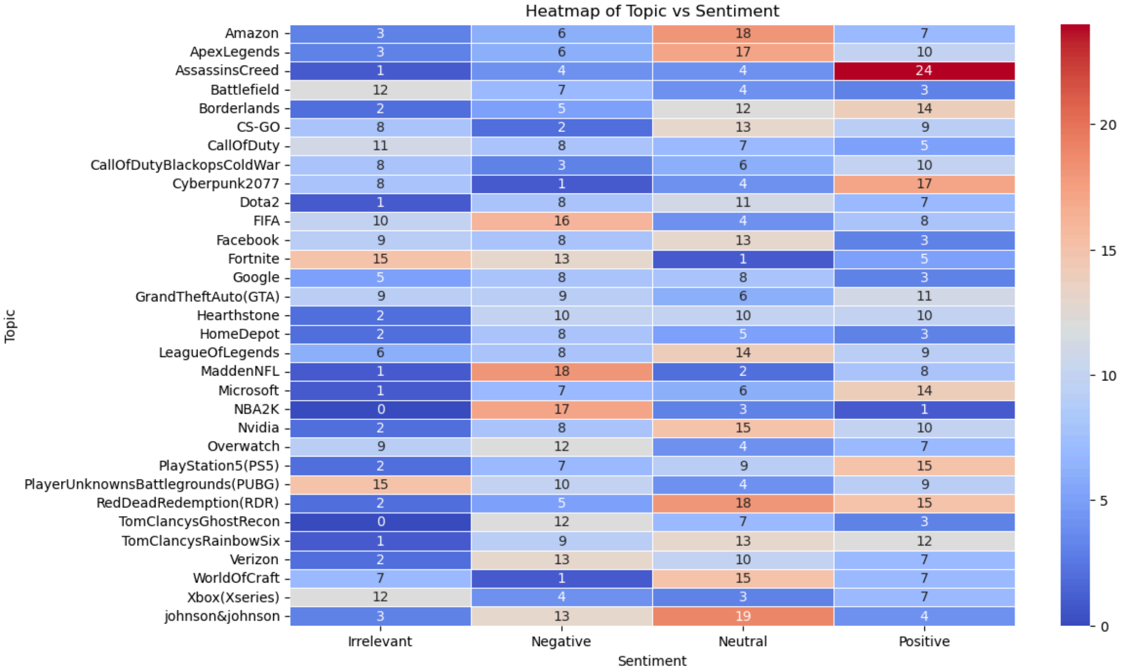


In [111]:
```
# Create the crosstab
crosstab = pd.crosstab(index=trains['Topic'], columns=trains['Sentiment'])

# 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()
```



In [114]:
```
pip install wordcloud
```

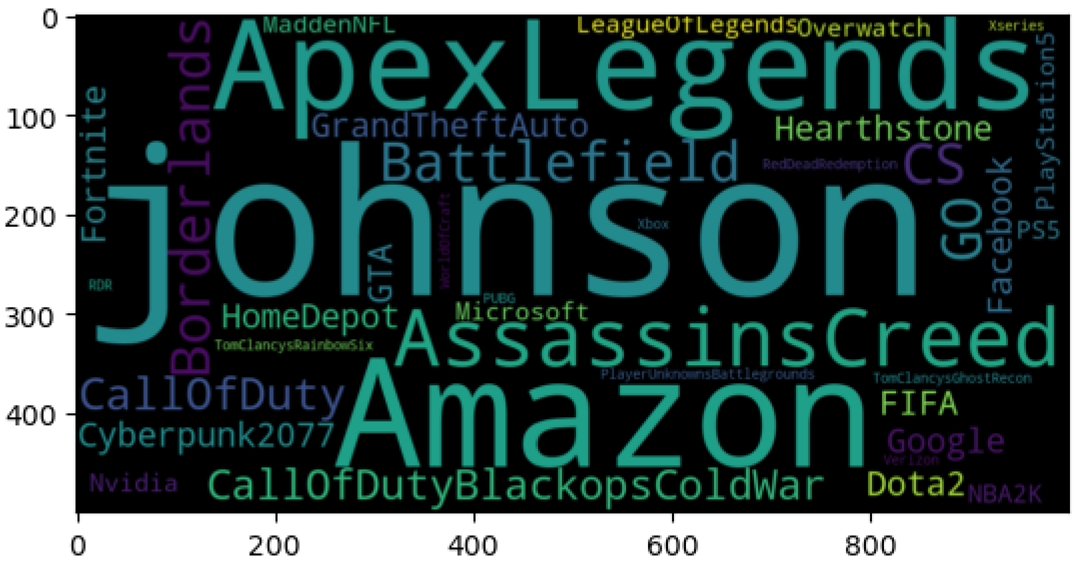
```
Collecting wordcloud
  Downloading wordcloud-1.9.3-cp310-cp310-win_amd64.whl (299 kB)
     -------------------------------------- 300.0/300.0 kB 1.2 MB/s eta
0:00:00
Requirement already satisfied: matplotlib in c:\users\ashwa\anaconda3\lib
\site-packages (from wordcloud) (3.7.0)
Requirement already satisfied: numpy>=1.6.1 in c:\users\ashwa\anaconda3\l
ib\site-packages (from wordcloud) (1.23.5)
Requirement already satisfied: pillow in c:\users\ashwa\anaconda3\lib\sit
e-packages (from wordcloud) (9.4.0)
Requirement already satisfied: packaging>=20.0 in c:\users\ashwa\anaconda
3\lib\site-packages (from matplotlib->wordcloud) (22.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\ashwa\anacond
a3\lib\site-packages (from matplotlib->wordcloud) (3.0.9)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\ashwa\anacon
da3\lib\site-packages (from matplotlib->wordcloud) (1.4.4)
Requirement already satisfied: cycler>=0.10 in c:\users\ashwa\anaconda3\l
ib\site-packages (from matplotlib->wordcloud) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\ashwa\anacon
da3\lib\site-packages (from matplotlib->wordcloud) (4.25.0)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\ashwa\ana
conda3\lib\site-packages (from matplotlib->wordcloud) (2.8.2)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\ashwa\anacond
a3\lib\site-packages (from matplotlib->wordcloud) (1.0.5)
Requirement already satisfied: six>=1.5 in c:\users\ashwa\anaconda3\lib\s
ite-packages (from python-dateutil>=2.7->matplotlib->wordcloud) (1.16.0)
Installing collected packages: wordcloud
Successfully installed wordcloud-1.9.3
Note: you may need to restart the kernel to use updated packages.
```

In [115]:
```
from wordcloud import WordCloud
topic_list = ' '.join(crosstab.index)


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

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

Out[115]:
```
<matplotlib.image.AxesImage at 0x20978779b10>
```

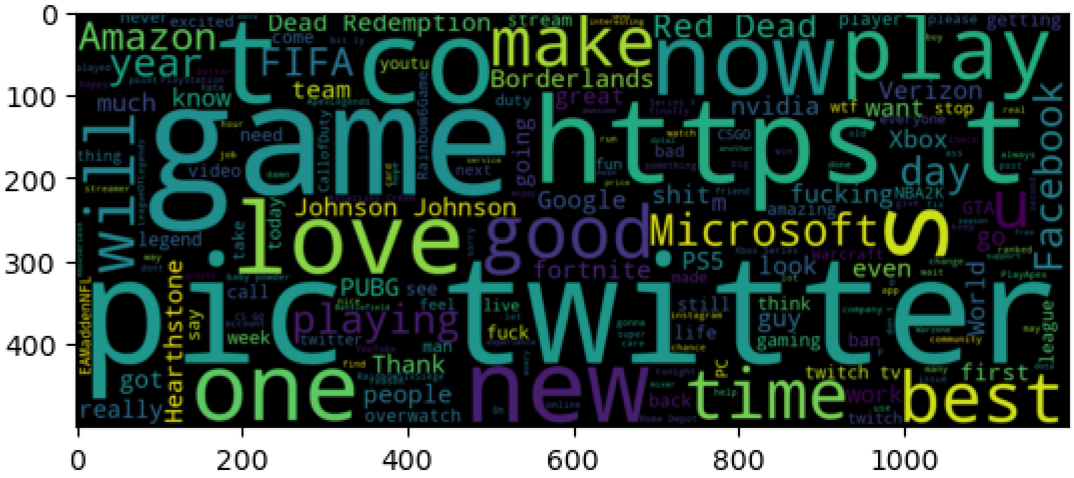


In [116]:
```
corpus = ' '.join(trains['Text'])

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

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

Out[116]:
```
<matplotlib.image.AxesImage at 0x209787e16f0>
```



In [ ]: