Sentiment Analysis

Import Libraries and Load Dataset:

In [14]:
```
import pandas as pd
import re
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import nltk

# Download required NLTK resources
nltk.download('stopwords')
nltk.download('punkt')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\user\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\user\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

Out[14]:
```
True
```

In [13]:
```
# Load the dataset
file_path = 'twitter_training.csv'  # Replace with the actual CSV file path
column_names = ['id', 'topic', 'sentiment', 'post']
df = pd.read_csv(file_path, names=column_names, header=None)

# Print the DataFrame to verify
print(df.head())
```

```
     id        topic sentiment  \
0  2401  Borderlands  Positive
1  2401  Borderlands  Positive
2  2401  Borderlands  Positive
3  2401  Borderlands  Positive
4  2401  Borderlands  Positive

                                                post
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 ...
```

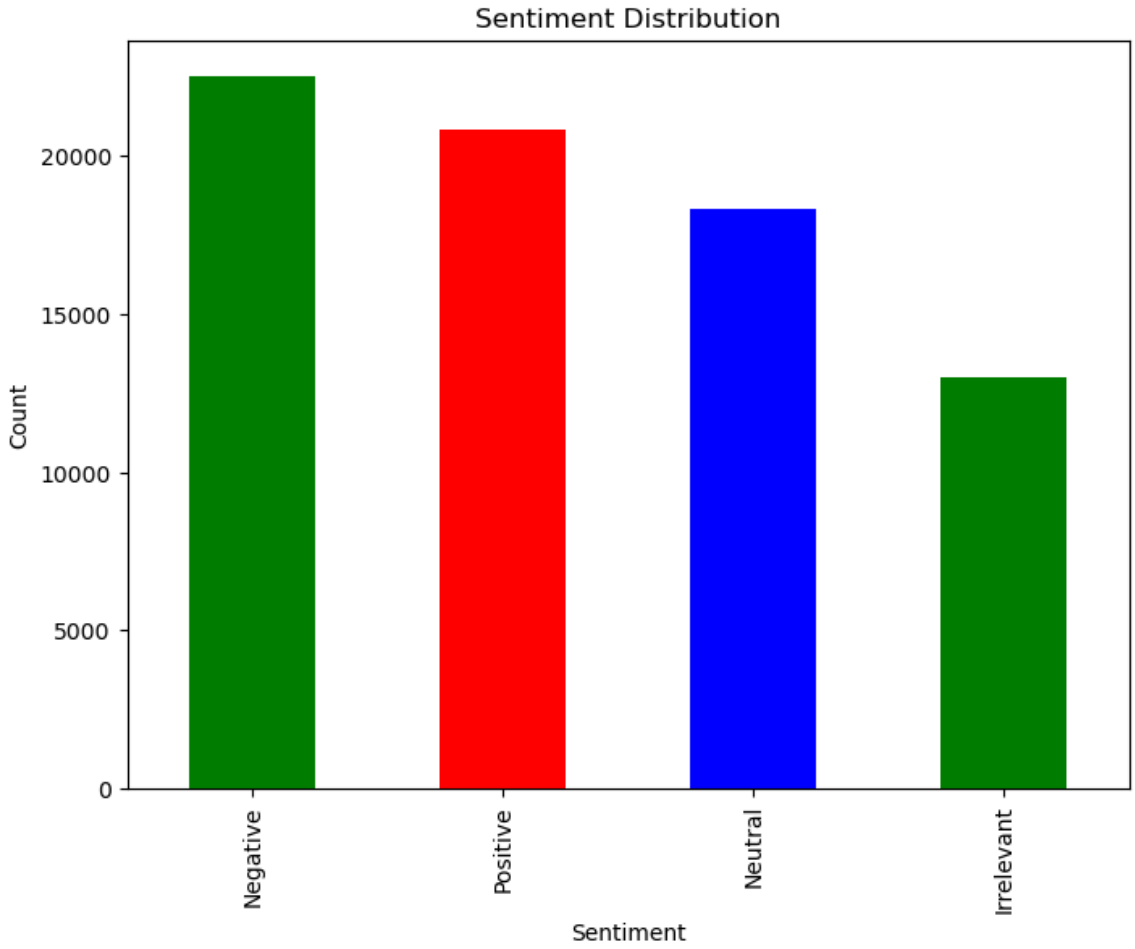
Define and Apply the Preprocessing Function:

In [12]:
```
# Data Preprocessing function
def preprocess_text(text):
    if not isinstance(text, str):
        text = ''
    text = re.sub(r'http\S+', '', text)  # Remove URLs
    text = re.sub(r'@\w+', '', text)  # Remove mentions
    text = re.sub(r'#\w+', '', text)  # Remove hashtags
    text = re.sub(r'\d+', '', text)  # Remove numbers
    text = text.lower()  # Convert to lowercase
    text = re.sub(r'[^\w\s]', '', text)  # Remove punctuation
    stop_words = set(stopwords.words('english'))
    text = ' '.join([word for word in word_tokenize(text) if word not in stc
    return text

# Apply preprocessing to the post column
df['cleaned_post'] = df['post'].apply(preprocess_text)

# Preview the cleaned data
print(df[['post', 'cleaned_post']].head())
```

```
                                                post  \
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 ...

                    cleaned_post
0  im getting borderlands murder
1            coming borders kill
2    im getting borderlands kill
3   im coming borderlands murder
4  im getting borderlands murder
```

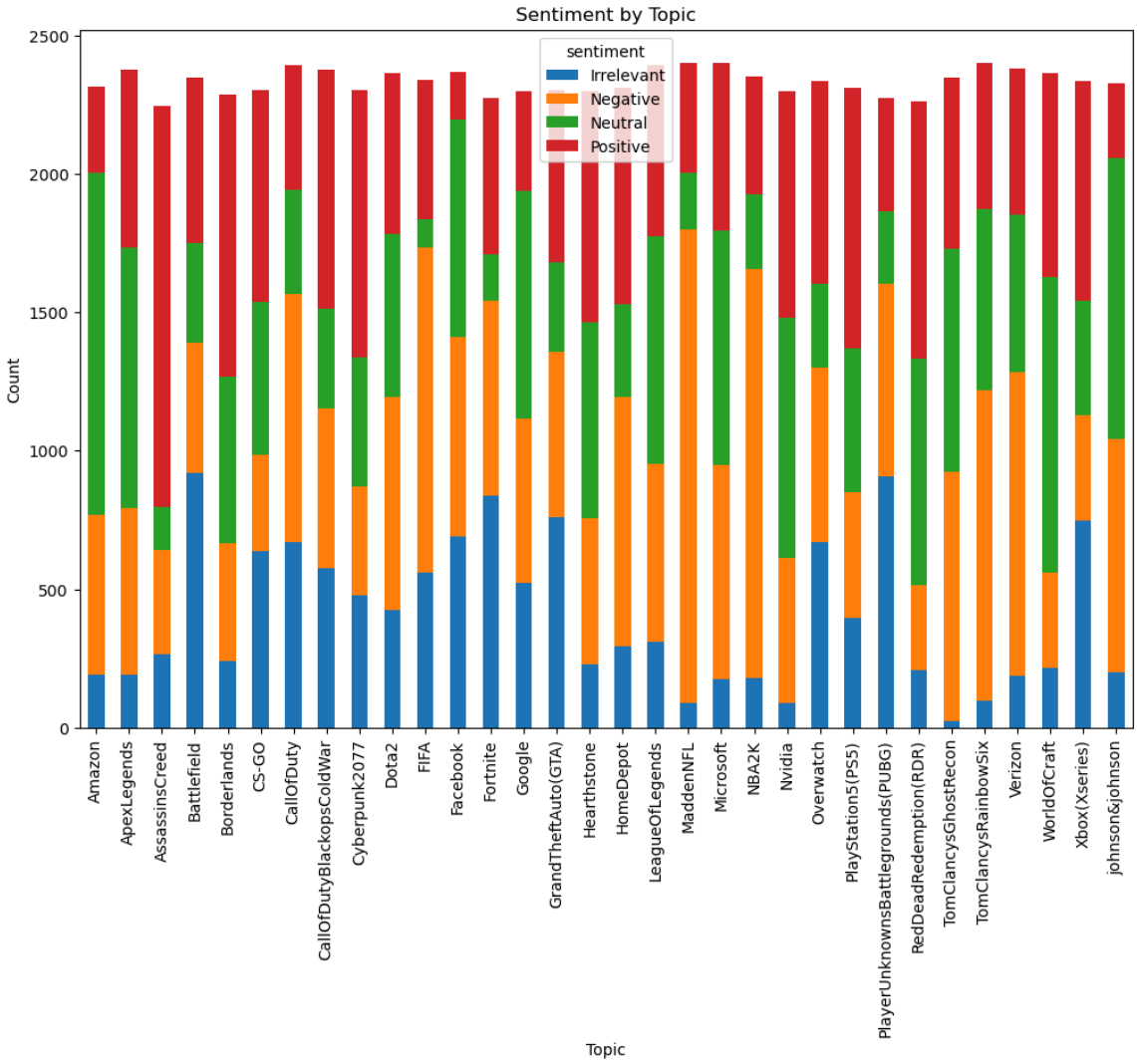
Visualize Sentiment Distribution:

In [8]:
```
# Sentiment distribution
import matplotlib.pyplot as plt
sentiment_counts = df['sentiment'].value_counts()
plt.figure(figsize=(8, 6))
sentiment_counts.plot(kind='bar', color=['green', 'red', 'blue'])
plt.title('Sentiment Distribution')
plt.xlabel('Sentiment')
plt.ylabel('Count')
plt.show()
```



Visualize Sentiment by Topic:

In [9]:
```
# Sentiment by topic
topic_sentiment = df.groupby(['topic', 'sentiment']).size().unstack().fillna
topic_sentiment.plot(kind='bar', stacked=True, figsize=(12, 8))
plt.title('Sentiment by Topic')
plt.xlabel('Topic')
plt.ylabel('Count')
plt.show()
```



In [ ]:

In [ ]: