In [21]:
```
import pandas as pd
import numpy as np
from nltk.tokenize import sent_tokenize, word_tokenize
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.datasets import fetch_20newsgroups
from nltk.corpus import stopwords
import string
from nltk import pos_tag
from nltk.stem import WordNetLemmatizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [22]:
```
import nltk
nltk.download('stopwords')
```

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

Out[22]: True

In [24]:
```
data = pd.read_csv('D:/prodigy/twitter_training.csv')
v_data = pd.read_csv('D:/prodigy/twitter_validation.csv')
```

In [25]:
```
data
```

Out[25]:

	2401	Borderlands	Positive	im getting on borderlands and i will murder you all ,
0	2401	Borderlands	Positive	I am coming to the borders and I will kill you...
1	2401	Borderlands	Positive	im getting on borderlands and i will kill you ...
2	2401	Borderlands	Positive	im coming on borderlands and i will murder you...
3	2401	Borderlands	Positive	im getting on borderlands 2 and i will murder ...
4	2401	Borderlands	Positive	im getting into borderlands and i can murder y...
...	...	...	...	...
74676	9200	Nvidia	Positive	Just realized that the Windows partition of my...
74677	9200	Nvidia	Positive	Just realized that my Mac window partition is ...
74678	9200	Nvidia	Positive	Just realized the windows partition of my Mac ...
74679	9200	Nvidia	Positive	Just realized between the windows partition of...
74680	9200	Nvidia	Positive	Just like the windows partition of my Mac is l...

74681 rows × 4 columns

In [26]:
```
v_data
```

Out[26]:

	3364	Facebook	Irrelevant	I mentioned on Facebook that I was struggling for motivation to go for a run the other day, which has been translated by Tom’s great auntie as ‘Hayley can’t get out of bed’ and told to his grandma, who now thinks I’m a lazy, terrible person 🤣
0	352	Amazon	Neutral	BBC News - Amazon boss Jeff Bezos rejects clai...
1	8312	Microsoft	Negative	@Microsoft Why do I pay for WORD when it funct...
2	4371	CS-GO	Negative	CSGO matchmaking is so full of closet hacking,...
3	4433	Google	Neutral	Now the President is slapping Americans in the...
4	6273	FIFA	Negative	Hi @EAHelp I’ve had Madeleine McCann in my cel...
...	...	...	...	...
994	4891	GrandTheftAuto(GTA)	Irrelevant	⭐ Toronto is the arts and culture capital of ...
995	4359	CS-GO	Irrelevant	tHIS IS ACTUALLY A GOOD MOVE TOT BRING MORE VI...
996	2652	Borderlands	Positive	Today sucked so it’s time to drink wine n play...
997	8069	Microsoft	Positive	Bought a fraction of Microsoft today. Small wins.
998	6960	johnson&johnson	Neutral	Johnson & Johnson to stop selling talc baby po...

999 rows × 4 columns

In [27]:
```
data.columns = ['id', 'game', 'sentiment', 'text']
v_data.columns = ['id', 'game', 'sentiment', 'text']
```

In [28]:
```
data
```

Out[28]:

	id	game	sentiment	text
0	2401	Borderlands	Positive	I am coming to the borders and I will kill you...
1	2401	Borderlands	Positive	im getting on borderlands and i will kill you ...
2	2401	Borderlands	Positive	im coming on borderlands and i will murder you...
3	2401	Borderlands	Positive	im getting on borderlands 2 and i will murder ...
4	2401	Borderlands	Positive	im getting into borderlands and i can murder y...
...	...	...	...	...
74676	9200	Nvidia	Positive	Just realized that the Windows partition of my...
74677	9200	Nvidia	Positive	Just realized that my Mac window partition is ...
74678	9200	Nvidia	Positive	Just realized the windows partition of my Mac ...
74679	9200	Nvidia	Positive	Just realized between the windows partition of...
74680	9200	Nvidia	Positive	Just like the windows partition of my Mac is l...

74681 rows × 4 columns

In [29]:
```
v_data
```

Out[29]:

	id	game	sentiment	text
0	352	Amazon	Neutral	BBC News - Amazon boss Jeff Bezos rejects clai...
1	8312	Microsoft	Negative	@Microsoft Why do I pay for WORD when it funct...
2	4371	CS-GO	Negative	CSGO matchmaking is so full of closet hacking,...
3	4433	Google	Neutral	Now the President is slapping Americans in the...
4	6273	FIFA	Negative	Hi @EAHelp I've had Madeleine McCann in my cel...
...	...	...	...	...
994	4891	GrandTheftAuto(GTA)	Irrelevant	⭐ Toronto is the arts and culture capital of ...
995	4359	CS-GO	Irrelevant	tHIS IS ACTUALLY A GOOD MOVE TOT BRING MORE VI...
996	2652	Borderlands	Positive	Today sucked so it's time to drink wine n play...
997	8069	Microsoft	Positive	Bought a fraction of Microsoft today. Small wins.
998	6960	johnson&johnson	Neutral	Johnson & Johnson to stop selling talc baby po...

999 rows × 4 columns

In [30]:
```
data.shape
```

Out[30]:
```
(74681, 4)
```

In [31]:
```
data.columns
```

Out[31]:
```
Index(['id', 'game', 'sentiment', 'text'], dtype='object')
```

In [32]:
```
data.describe(include='all')
```

Out[32]:

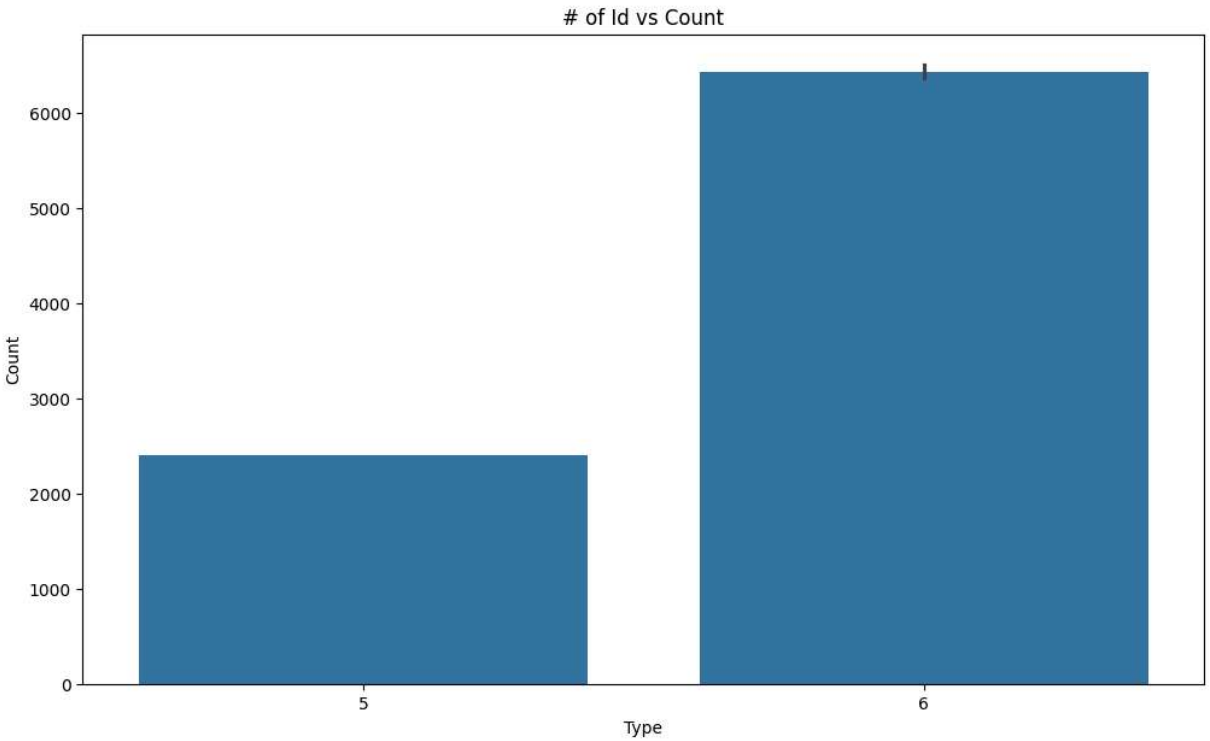
	id	game	sentiment	text
count	74681.000000	74681	74681	73995
unique	NaN	32	4	69490
top	NaN	Microsoft	Negative	It is not the first time that the EU Commissio...
freq	NaN	2400	22542	172
mean	6432.640149	NaN	NaN	NaN
std	3740.423819	NaN	NaN	NaN
min	1.000000	NaN	NaN	NaN
25%	3195.000000	NaN	NaN	NaN
50%	6422.000000	NaN	NaN	NaN
75%	9601.000000	NaN	NaN	NaN
max	13200.000000	NaN	NaN	NaN

In [33]:
```
id_types = data['id'].value_counts()
id_types
```

Out[33]:
```
id
9200    6
9199    6
2402    6
2403    6
2404    6
       ..
2435    6
2436    6
2437    6
2438    6
2401    5
Name: count, Length: 12447, dtype: int64
```

In [34]:
```
plt.figure(figsize=(12,7))
sns.barplot(y=id_types.index, x=id_types.values)
plt.xlabel('Type')
plt.ylabel('Count')
```

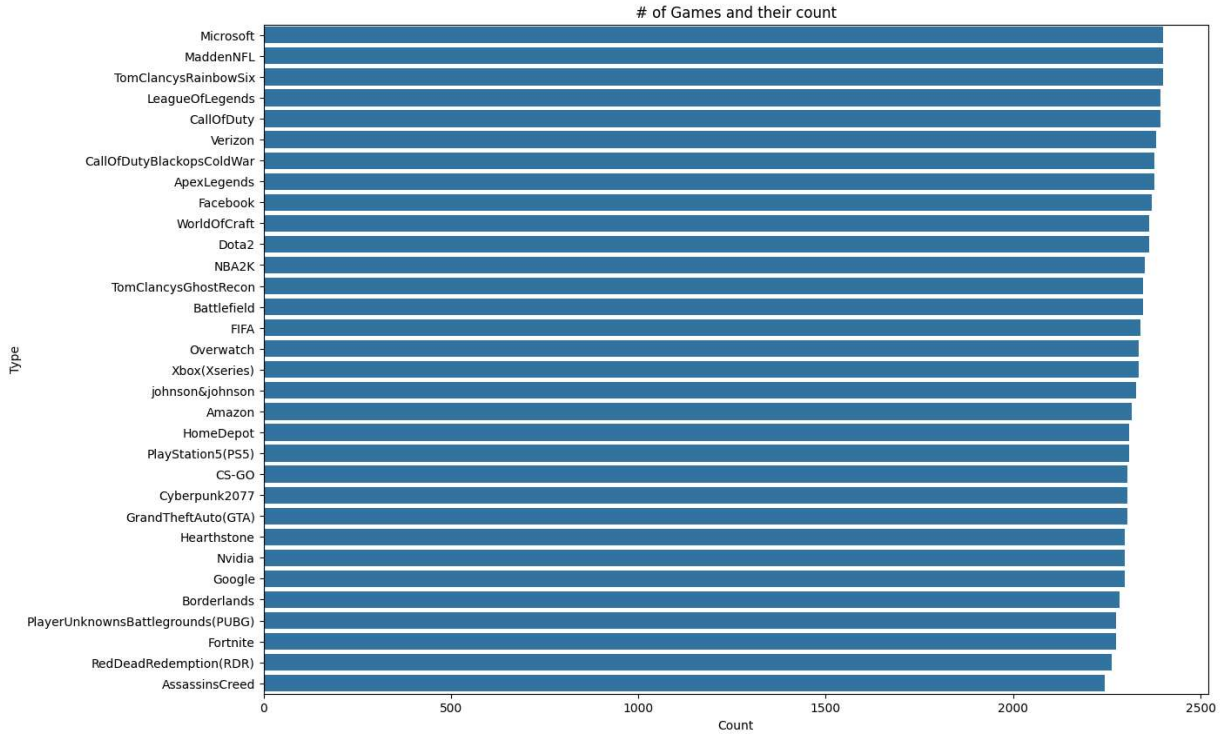
```
plt.title('# of Id vs Count')
plt.show()
```



In [35]:
```
game_types = data['game'].value_counts()
game_types
```

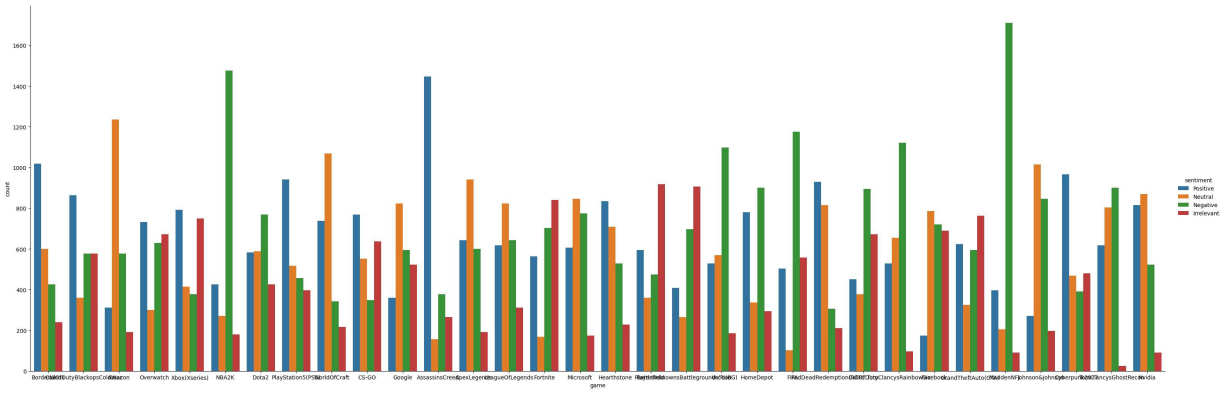
Out[35]:
```
game
Microsoft                              2400
MaddenNFL                              2400
TomClancysRainbowSix                   2400
LeagueOfLegends                        2394
CallOfDuty                             2394
Verizon                                2382
CallOfDutyBlackopsColdWar              2376
ApexLegends                            2376
Facebook                               2370
WorldOfCraft                           2364
Dota2                                  2364
NBA2K                                  2352
TomClancysGhostRecon                   2346
Battlefield                            2346
FIFA                                   2340
Overwatch                              2334
Xbox(Xseries)                          2334
johnson&johnson                        2328
Amazon                                 2316
HomeDepot                              2310
PlayStation5(PS5)                      2310
CS-GO                                  2304
Cyberpunk2077                          2304
GrandTheftAuto(GTA)                    2304
Hearthstone                            2298
Nvidia                                 2298
Google                                 2298
Borderlands                            2285
PlayerUnknownsBattlegrounds(PUBG)      2274
Fortnite                               2274
RedDeadRedemption(RDR)                 2262
AssassinsCreed                         2244
Name: count, dtype: int64
```

In [36]:
```
plt.figure(figsize=(14,10))

sns.barplot(x=game_types.values,y=game_types.index)
plt.title('# of Games and their count')
plt.ylabel('Type')
plt.xlabel('Count')

plt.show()
```



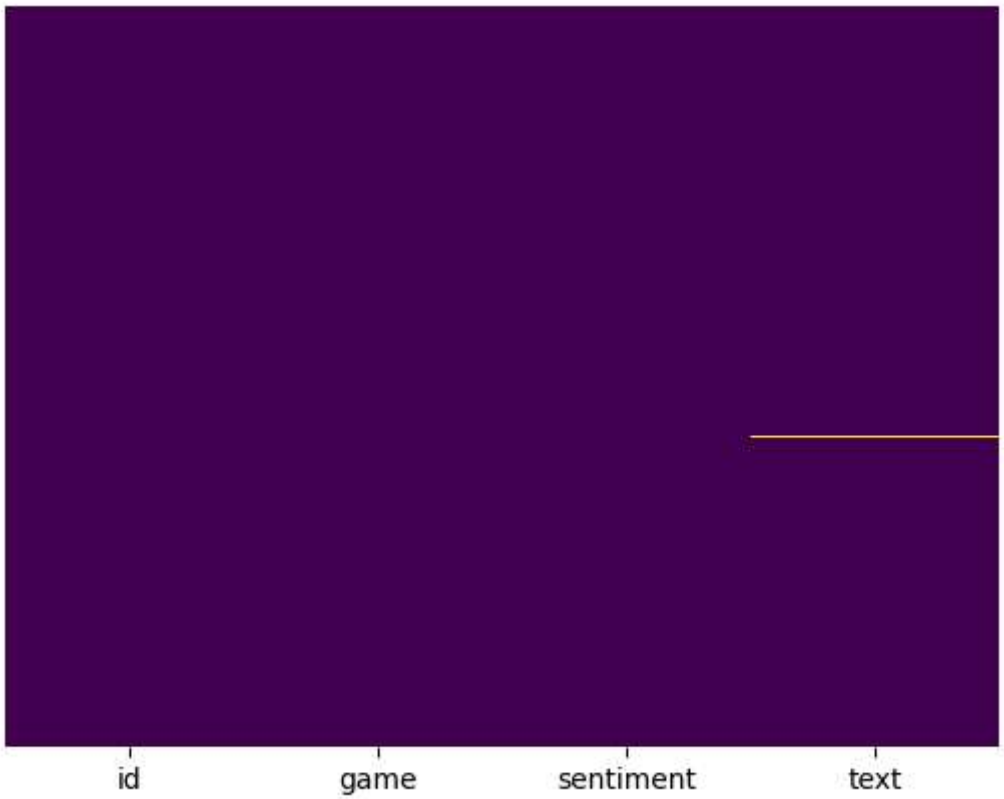
In [37]:
```
sns.catplot(x="game",hue="sentiment", kind="count",height=10,aspect=3, data=data)
```

Out[37]: <seaborn.axisgrid.FacetGrid at 0x145b6b808c0>



In [38]:
```
sns.heatmap(data.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[38]: <Axes: >



In [39]:
```
total_null=data.isnull().sum().sort_values(ascending=False)
percent = ((data.isnull().sum()/data.isnull().count())*100).sort_values(ascending =
print("Total records = ", data.shape[0])
missing_data = pd.concat([total_null,percent.round(2)],axis=1,keys=['Total Missing'
missing_data.head(10)
```

```
Total records =  74681
```

Out[39]:

	Total Missing	In Percent
text	686	0.92
id	0	0.00
game	0	0.00
sentiment	0	0.00

In [40]:
```
data.dropna(subset=['text'],inplace=True)

total_null=data.isnull().sum().sort_values(ascending=False)
percent = ((data.isnull().sum()/data.isnull().count())*100).sort_values(ascending =
print("Total records = ", data.shape[0])
missing_data = pd.concat([total_null,percent.round(2)],axis=1,keys=['Total Missing'
missing_data.head(10)
```

```
Total records =  73995
```

Out[40]:

	Total Missing	In Percent
id	0	0.0
game	0	0.0
sentiment	0	0.0
text	0	0.0

In [41]:
```
train0=data[data['sentiment']=="Negative"]
train1=data[data['sentiment']=="Positive"]
train2=data[data['sentiment']=="Irrelevant"]
train3=data[data['sentiment']=="Neutral"]
```

In [42]:
```
train0.shape, train1.shape, train2.shape, train3.shape
```

Out[42]:
```
((22358, 4), (20654, 4), (12875, 4), (18108, 4))
```

In [43]:
```
data=pd.concat([train0,train1,train2,train3],axis=0)
data
```

Out[43]:

	id	game	sentiment	text
23	2405	Borderlands	Negative	the biggest dissappoinment in my life came out...
24	2405	Borderlands	Negative	The biggest disappointment of my life came a y...
25	2405	Borderlands	Negative	The biggest disappointment of my life came a y...
26	2405	Borderlands	Negative	the biggest dissappoinment in my life coming o...
27	2405	Borderlands	Negative	For the biggest male dissappoinment in my life...
...	...	...	...	...
74658	9197	Nvidia	Neutral	Nvidia plans to release its 2017 "Crypto Craze...
74659	9197	Nvidia	Neutral	Nvidia does not want to give up its "cryptoins...
74660	9197	Nvidia	Neutral	Nvidia doesn't intend to give away its 2017 ad...
74661	9197	Nvidia	Neutral	Nvidia therefore doesn ' t want to give up its...
74662	9197	Nvidia	Neutral	is doesn't should I give up its password 'cryp...

73995 rows × 4 columns

In [44]:
```
id_types = data['id'].value_counts()
id_types
```

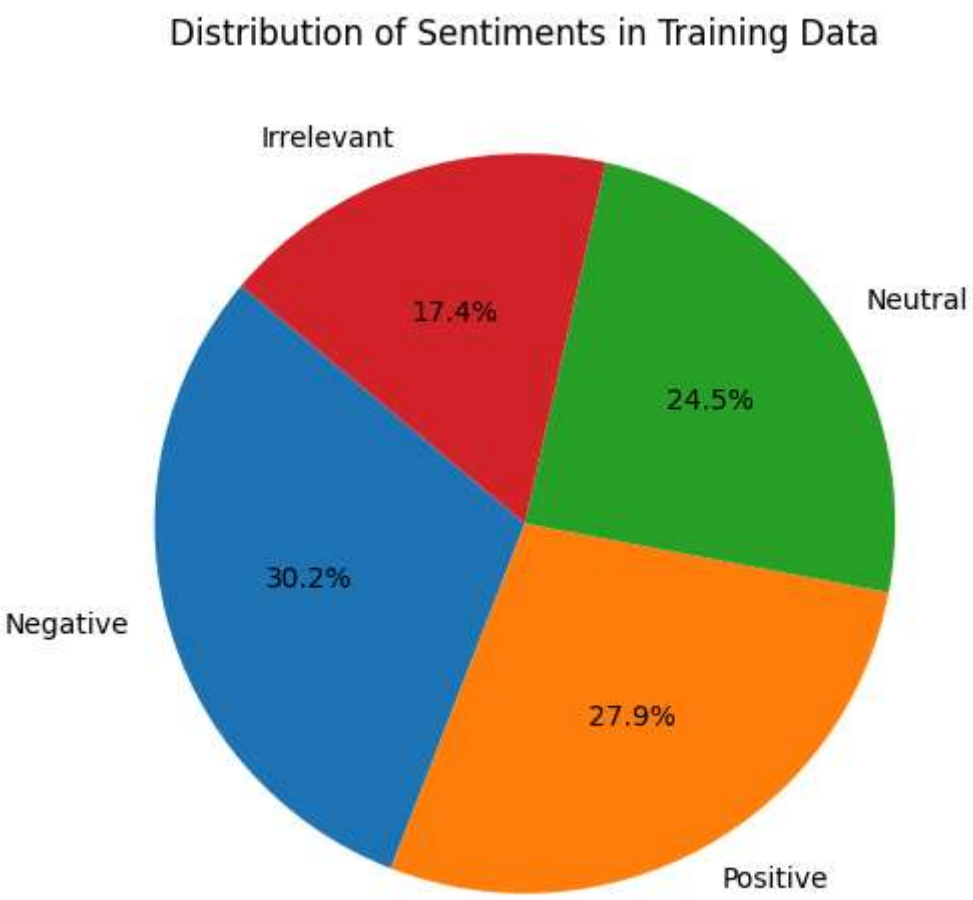
Out[44]:
```
id
9197    6
2405    6
2407    6
2410    6
2417    6
       ..
8334    3
8018    3
6534    3
7761    3
7861    3
Name: count, Length: 12447, dtype: int64
```

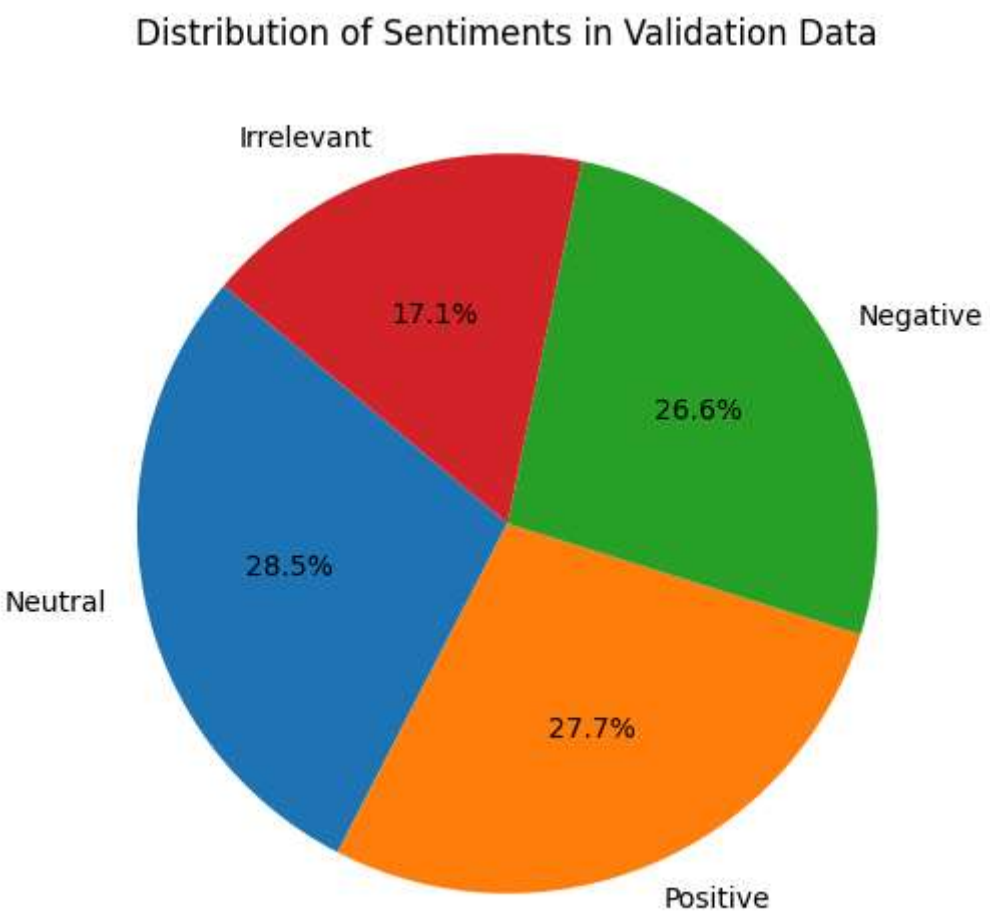
In [51]:
```
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
import string
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')

def preprocess_text(text):
    tokens = word_tokenize(text.lower())
    stop_words = set(stopwords.words('english'))
    tokens = [token for token in tokens if token not in stop_words and token not in
    lemmatizer = WordNetLemmatizer()
    tokens = [lemmatizer.lemmatize(token) for token in tokens]
    return tokens

data['tokens'] = data['text'].apply(preprocess_text)
v_data['tokens'] = v_data['text'].apply(preprocess_text)

print(data[['text', 'tokens']].head())
print(v_data[['text', 'tokens']].head())
```

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

```
                                                 text  \
23  the biggest dissappoinment in my life came out...
24  The biggest disappointment of my life came a y...
25  The biggest disappointment of my life came a y...
26  the biggest dissappoinment in my life coming o...
27  For the biggest male dissappoinment in my life...

                                               tokens
23  [biggest, dissappoinment, life, came, year, ag...
24   [biggest, disappointment, life, came, year, ago]
25   [biggest, disappointment, life, came, year, ago]
26  [biggest, dissappoinment, life, coming, year, ...
27  [biggest, male, dissappoinment, life, came, ha...
                                                 text  \
0  BBC News - Amazon boss Jeff Bezos rejects clai...
1  @Microsoft Why do I pay for WORD when it funct...
2  CSGO matchmaking is so full of closet hacking,...
3  Now the President is slapping Americans in the...
4  Hi @EAHelp I’ve had Madeleine McCann in my cel...

                                               tokens
0  [bbc, news, amazon, bos, jeff, bezos, reject, ...
1  [microsoft, pay, word, function, poorly, samsu...
2  [csgo, matchmaking, full, closet, hacking, 's,...
3  [president, slapping, american, face, really, ...
4  [hi, eahelp, ’, madeleine, mccann, cellar, pas...
```

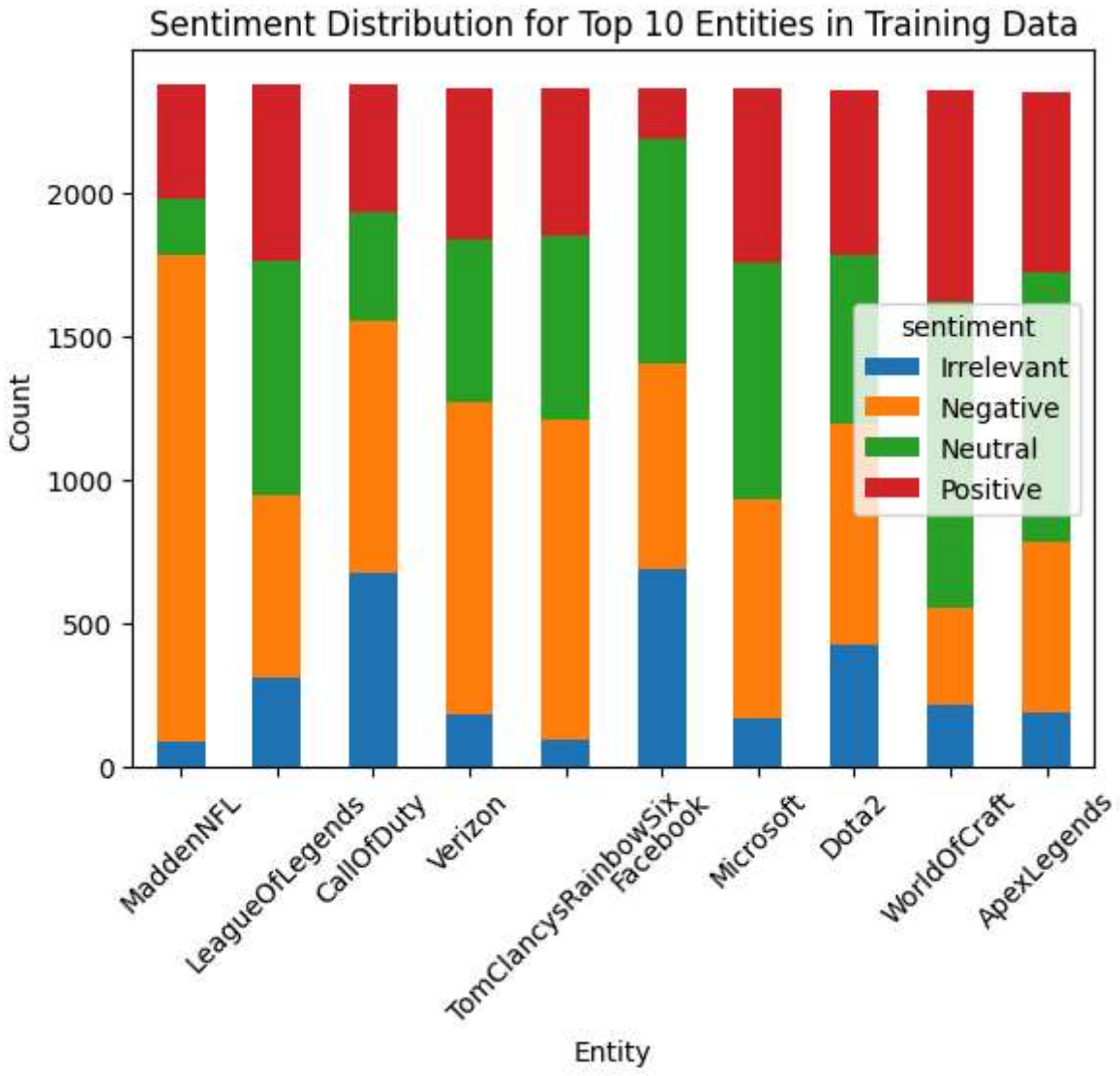
In [52]:
```
import matplotlib.pyplot as plt
train_sentiment_counts = data['sentiment'].value_counts()

plt.figure(figsize=(8, 6))
plt.pie(train_sentiment_counts, labels=train_sentiment_counts.index, autopct='%1.1f
plt.title('Distribution of Sentiments in Training Data')
plt.show()
val_sentiment_counts = v_data['sentiment'].value_counts()
plt.figure(figsize=(8, 6))
plt.pie(val_sentiment_counts, labels=val_sentiment_counts.index, autopct='%1.1f%%',
plt.title('Distribution of Sentiments in Validation Data')
plt.show()
```


Distribution of Sentiments in Training Data
Irrelevant
17.4%
Neutral
24.5%
30.2%
Negative
27.9%
Positive



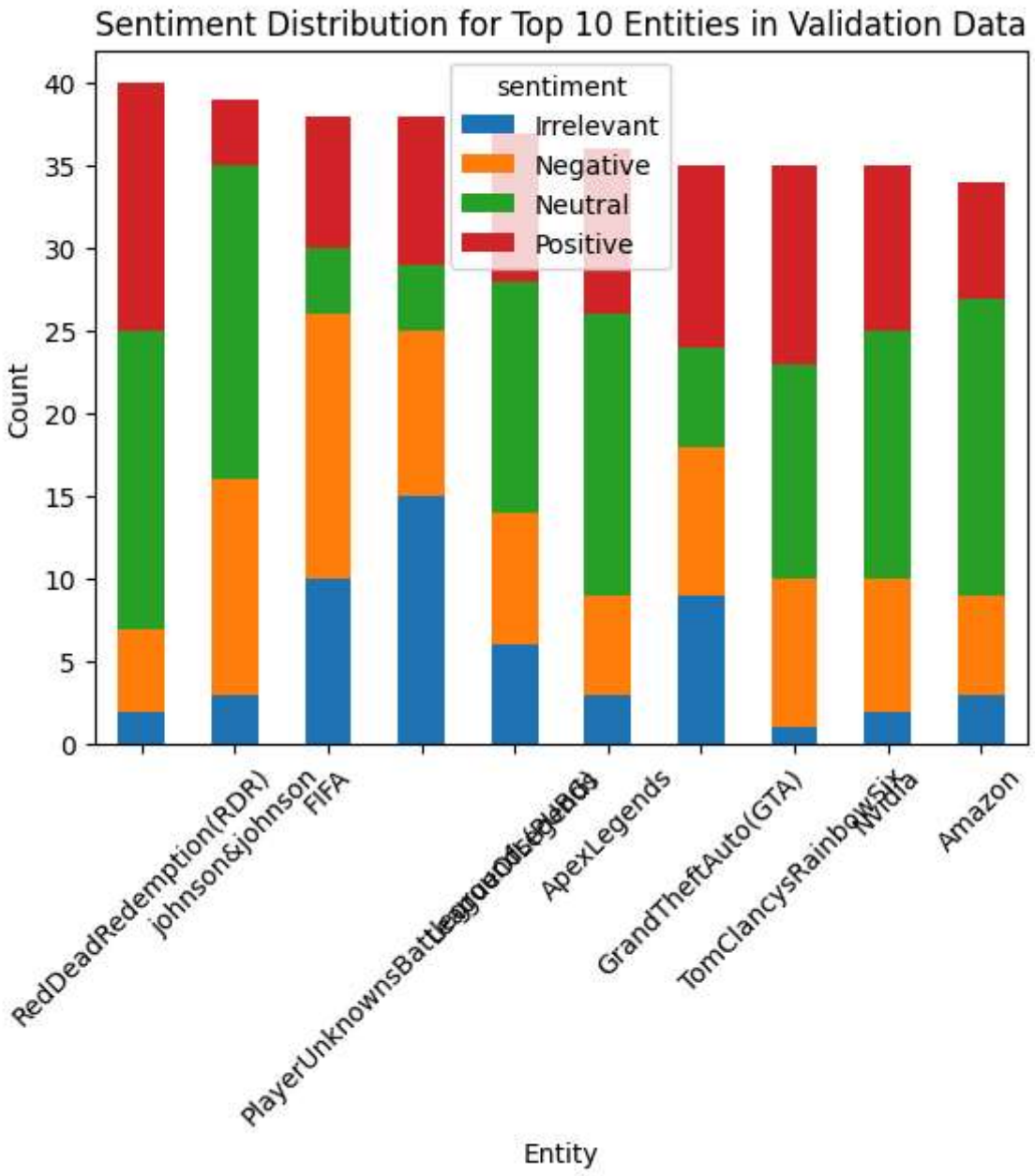
In [53]:
```
import seaborn as sns
train_entity_sentiment = data.groupby(['game', 'sentiment']).size().unstack(fill_va
top_entities = train_entity_sentiment.sum(axis=1).sort_values(ascending=False).head
train_entity_sentiment_top = train_entity_sentiment.loc[top_entities]
plt.figure(figsize=(12, 8))
train_entity_sentiment_top.plot(kind='bar', stacked=True)
plt.title('Sentiment Distribution for Top 10 Entities in Training Data')
plt.ylabel('Count')
plt.xlabel('Entity')
plt.xticks(rotation=45)
plt.show()
```

```
<Figure size 1200x800 with 0 Axes>
```



In [55]:
```
val_entity_sentiment = v_data.groupby(['game', 'sentiment']).size().unstack(fill_va
top_entities_val = val_entity_sentiment.sum(axis=1).sort_values(ascending=False).he
val_entity_sentiment_top = val_entity_sentiment.loc[top_entities_val]
plt.figure(figsize=(12, 8))
val_entity_sentiment_top.plot(kind='bar', stacked=True)
plt.title('Sentiment Distribution for Top 10 Entities in Validation Data')
plt.ylabel('Count')
plt.xlabel('Entity')
plt.xticks(rotation=45)
plt.show()
```

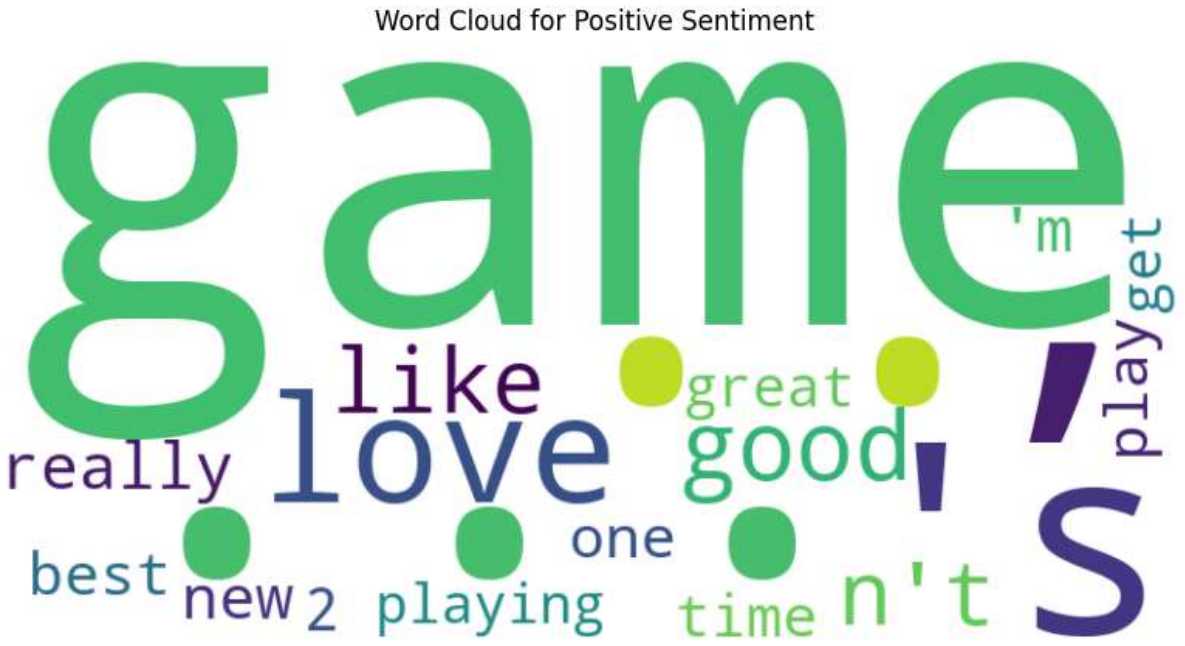
```
<Figure size 1200x800 with 0 Axes>
```

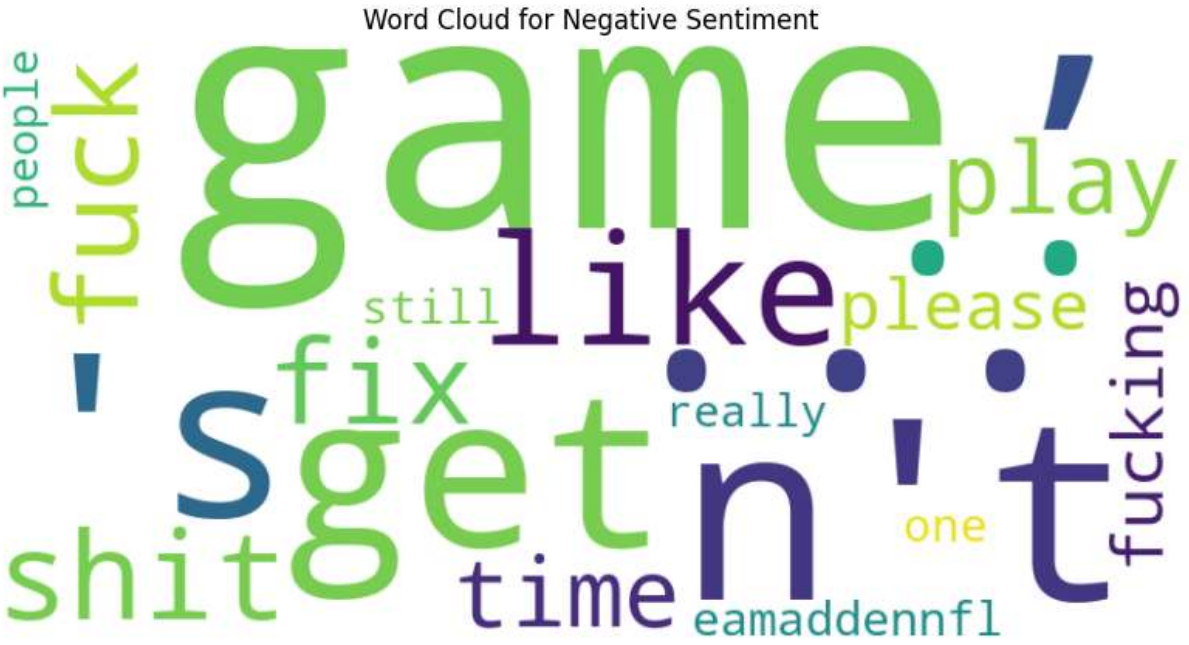


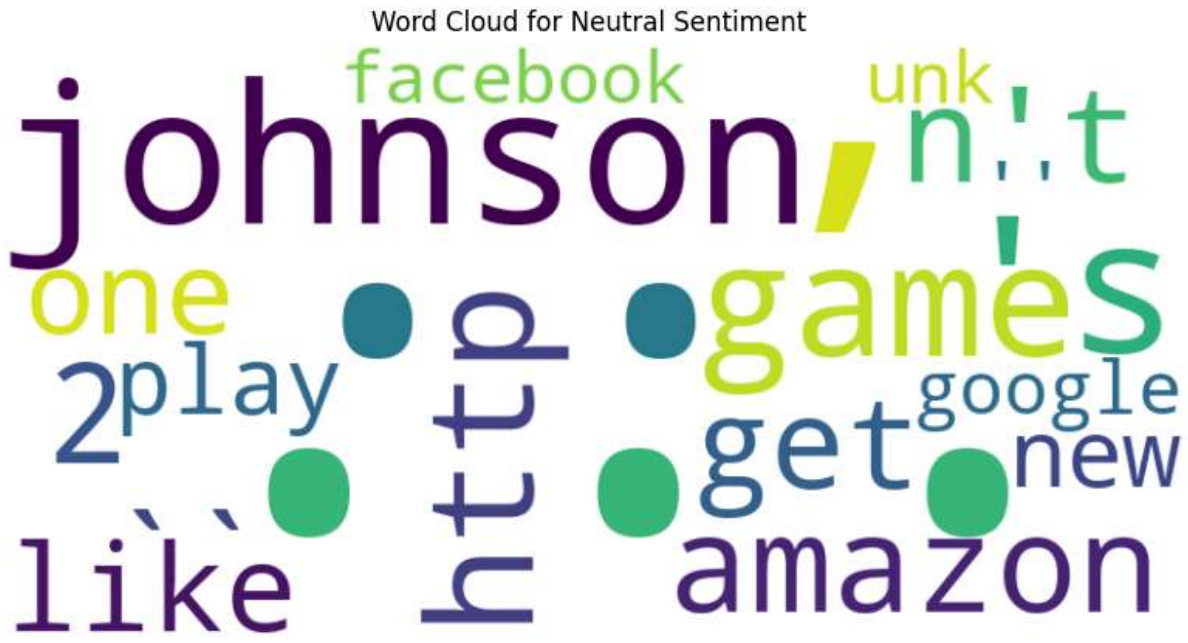
In [57]:
```
from collections import Counter
import itertools
def get_most_common_words(df, sentiment, n=20):
    tokens = data[data['sentiment'] == sentiment]['tokens']
    all_tokens = list(itertools.chain.from_iterable(tokens))
    return Counter(all_tokens).most_common(n)
positive_words = get_most_common_words(train_data, 'Positive')
negative_words = get_most_common_words(train_data, 'Negative')
neutral_words = get_most_common_words(train_data, 'Neutral')

print("Most common words in Positive sentiment:", positive_words)
print("Most common words in Negative sentiment:", negative_words)
print("Most common words in Neutral sentiment:", neutral_words)
```

```
Most common words in Positive sentiment: [('game', 3120), ('...', 2798), ('..', 247
3), ('’', 2261), ("'s", 2232), ('love', 1809), ('good', 1618), ('like', 1373),
("n't", 1332), ('really', 1281), ("'m", 1242), ('new', 1200), ('time', 1136), ('2',
1122), ('play', 1120), ('best', 1113), ('one', 1101), ('great', 989), ('get', 978),
('playing', 934)]
Most common words in Negative sentiment: [('game', 4512), ('’', 3703), ('...', 280
2), ("n't", 2524), ("'s", 2036), ('..', 1913), ('get', 1830), ('like', 1646), ('shi
t', 1530), ('fix', 1326), ('fuck', 1304), ('play', 1242), ('time', 1126), ('please',
1115), ('fucking', 1103), ('eamaddennfl', 1090), ('still', 1054), ('one', 1033), ('p
eople', 984), ('really', 983)]
Most common words in Neutral sentiment: [('...', 4413), ('..', 3147), ('’', 1841),
('johnson', 1819), ("'s", 1798), ('game', 1618), ('http', 1376), ('2', 1241), ('amaz
on', 1152), ("n't", 1105), ('``', 1098), ('get', 1008), ('like', 942), ('one', 881),
('play', 852), ('new', 850), ('google', 850), ('facebook', 845), ('unk', 828),
("''", 822)]
```

In [58]:
```
from wordcloud import WordCloud
def generate_word_cloud(words, title):
    wordcloud = WordCloud(width=800, height=400, background_color='white').generate
    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.title(title)
    plt.axis('off')
    plt.show()
generate_word_cloud(positive_words, 'Word Cloud for Positive Sentiment')
generate_word_cloud(negative_words, 'Word Cloud for Negative Sentiment')
generate_word_cloud(neutral_words, 'Word Cloud for Neutral Sentiment')
```







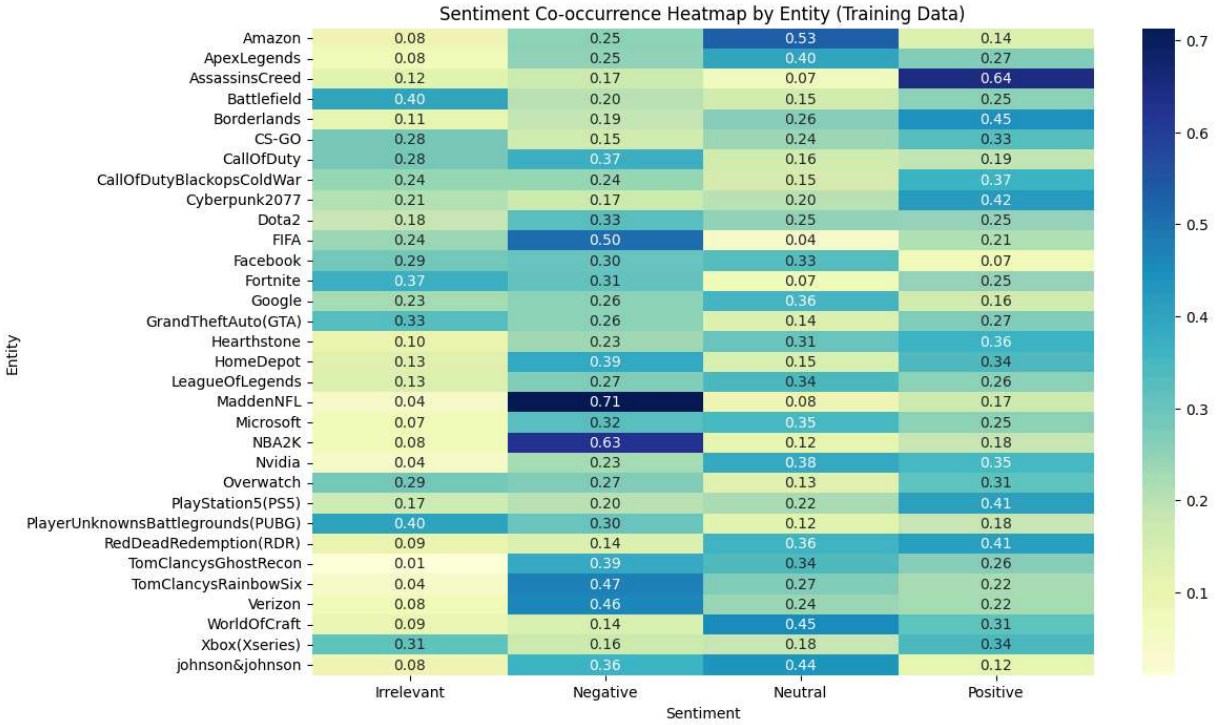
In [60]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
train_texts = data['text'].values
val_texts = v_data['text'].values
vectorizer = TfidfVectorizer(max_features=5000)
X_train = vectorizer.fit_transform(train_texts)
X_val = vectorizer.transform(val_texts)
y_train = data['sentiment']
y_val = v_data['sentiment']
```

In [61]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix
clf = LogisticRegression(max_iter=1000)
clf.fit(X_train, y_train)
y_pred = clf.predict(X_val)
print("Classification Report:")
```

```
print(classification_report(y_val, y_pred))
print("Confusion Matrix:")
print(confusion_matrix(y_val, y_pred))
```

```
Classification Report:
              precision    recall  f1-score   support

  Irrelevant       0.82      0.73      0.77       171
    Negative       0.79      0.88      0.83       266
     Neutral       0.86      0.77      0.81       285
    Positive       0.80      0.86      0.83       277

    accuracy                           0.82       999
   macro avg       0.82      0.81      0.81       999
weighted avg       0.82      0.82      0.82       999

Confusion Matrix:
[[125  16   6  24]
 [  7 233  11  15]
 [ 12  32 220  21]
 [  8  13  18 238]]
```

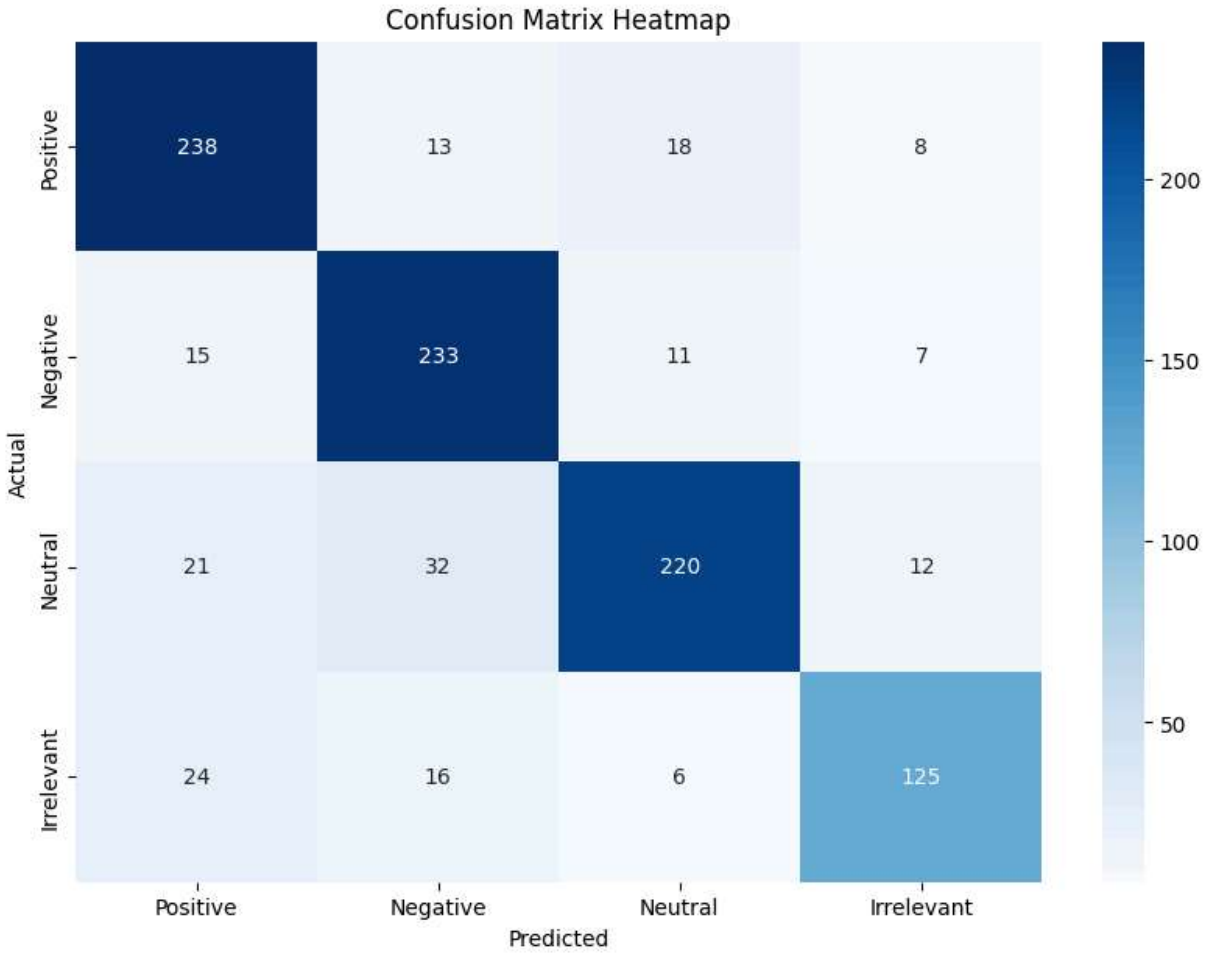
In [63]:
```
entity_sentiment_matrix = data.pivot_table(index='game', columns='sentiment', aggfu
entity_sentiment_matrix_norm = entity_sentiment_matrix.div(entity_sentiment_matrix.
plt.figure(figsize=(12, 8))
sns.heatmap(entity_sentiment_matrix_norm, cmap="YlGnBu", annot=True, fmt=".2f")
plt.title('Sentiment Co-occurrence Heatmap by Entity (Training Data)')
plt.ylabel('Entity')
plt.xlabel('Sentiment')
plt.show()
```



In [90]:
```
clf = LogisticRegression(max_iter=1000)
clf.fit(X_train, y_train)
y_pred = clf.predict(X_val)
```

```
print("Classification Report:")
print(classification_report(y_val, y_pred))
print("Confusion Matrix:")
print(confusion_matrix(y_val, y_pred))
cm = confusion_matrix(y_val, y_pred, labels=['Positive', 'Negative', 'Neutral', 'Ir
cm_df = pd.DataFrame(cm, index=['Positive', 'Negative', 'Neutral', 'Irrelevant'], c
plt.figure(figsize=(10, 7))
sns.heatmap(cm_df, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix Heatmap')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```

```
Classification Report:
              precision    recall  f1-score   support

  Irrelevant       0.82      0.73      0.77       171
    Negative       0.79      0.88      0.83       266
     Neutral       0.86      0.77      0.81       285
    Positive       0.80      0.86      0.83       277

    accuracy                           0.82       999
   macro avg       0.82      0.81      0.81       999
weighted avg       0.82      0.82      0.82       999

Confusion Matrix:
[[125  16   6  24]
 [  7 233  11  15]
 [ 12  32 220  21]
 [  8  13  18 238]]
```


Confusion Matrix Heatmap
Positive
Negative
Neutral
Irrelevant
Actual
238
13
18
8
15
233
11
7
21
32
220
12
24
16
6
125
200
150
100
50
Positive
Negative
Neutral
Irrelevant
Predicted

In [ ]:

In [ ]: