In [17]:

```
import pandas as pd
import contractions
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer, WordNetLemmatizer
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
import re
```

In [18]:

```
# Download necessary NLTK data files
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
```

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

Out[18]:

```
True
```

In [19]:

```
# Load the dataset
file_path = 'amazon.csv'
df = pd.read_csv(file_path)

# Display the first few rows of the dataset
df.head()
```

Out[19]:

	reviewText	Positive
0	This is a one of the best apps acording to a b...	1
1	This is a pretty good version of the game for ...	1
2	this is a really cool game. there are a bunch ...	1
3	This is a silly game and can be frustrating, b...	1
4	This is a terrific game on any pad. Hrs of fun...	1

In [20]:

```
# Function to preprocess text
def preprocess_text(text):
    # 1. Expand Contractions
    text = contractions.fix(text)

    # 2. Remove URLs and Emails
    text = re.sub(r'http\S+|www\S+|https\S+|mailto:\S+', '', text, flags=re.MULTILINE)
    text = re.sub(r'\S+@\S+', '', text)

    # 3. Remove special characters and emojis
    text = re.sub(r'[^a-zA-Z\s]', '', text)

    # 4. Tokenization
    words = word_tokenize(text)

    # 5. Lowercasing
    words = [word.lower() for word in words]

    # 6. Removing Punctuation
    words = [word for word in words if word.isalnum()]

    # 7. Removing Stop Words
    stop_words = set(stopwords.words('english'))
    filtered_words = [word for word in words if word not in stop_words]

    # 8. Stemming
    stemmer = PorterStemmer()
    stemmed_words = [stemmer.stem(word) for word in filtered_words]

    # 9. Lemmatization
    lemmatizer = WordNetLemmatizer()
    lemmatized_words = [lemmatizer.lemmatize(word) for word in filtered_words]

    return ' '.join(lemmatized_words)

# Apply preprocessing to the dataset
df['preprocessed_text'] = df['reviewText'].apply(preprocess_text)

# Display the first few rows of the preprocessed dataset
print(df[['reviewText', 'preprocessed_text']].head())

# Perform vectorization
# Using CountVectorizer
count_vectorizer = CountVectorizer()
count_vector = count_vectorizer.fit_transform(df['preprocessed_text'])
print("Count Vectorizer - Feature Names:", count_vectorizer.get_feature_names_out())
print("Count Vectorizer - Vectorized Text:", count_vector.toarray())

# Using TfidfVectorizer
tfidf_vectorizer = TfidfVectorizer()
tfidf_vector = tfidf_vectorizer.fit_transform(df['preprocessed_text'])
print("TF-IDF Vectorizer - Feature Names:", tfidf_vectorizer.get_feature_names_out())
print("TF-IDF Vectorizer - Vectorized Text:", tfidf_vector.toarray())
```

```
                                          reviewText  \
0  This is a one of the best apps acording to a b...
1  This is a pretty good version of the game for ...
2  this is a really cool game. there are a bunch ...
3  This is a silly game and can be frustrating, b...
4  This is a terrific game on any pad. Hrs of fun...

                                   preprocessed_text
0  one best apps acording bunch people agree bomb...
1  pretty good version game free lot different le...
2  really cool game bunch level find golden egg s...
3  silly game frustrating lot fun definitely reco...
4  terrific game pad hr fun grandkids love great ...
Count Vectorizer - Feature Names: ['aa' 'aaa' 'aaaa' ... 'zzz' 'zzzz' 'zzzzz']
Count Vectorizer - Vectorized Text: [[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
TF-IDF Vectorizer - Feature Names: ['aa' 'aaa' 'aaaa' ... 'zzz' 'zzzz' 'zzzzz']
TF-IDF Vectorizer - Vectorized Text: [[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
```

```
In [12]: df['preprocessed_text']
```

Out[12]:

```
0        one best apps acording bunch people agree bomb...
1        pretty good version game free lot different le...
2        really cool game bunch level find golden egg s...
3        silly game frustrating lot fun definitely reco...
4        terrific game pad hr fun grandkids love great ...
                               ...                        
19995    app fricken stupidit froze kindle allow place ...
19996    please add need neighbor ginger thanks bunch a...
19997    love game awesome wish free stuff house cost m...
19998    love love love app side fashion story fight wo...
19999    game rip list thing make betterbull first need...
Name: preprocessed_text, Length: 20000, dtype: object
```

In [13]:

```python
processed_file_path = 'amazon_processed.csv'
df.to_csv(processed_file_path, index=False)
```