In [4]:

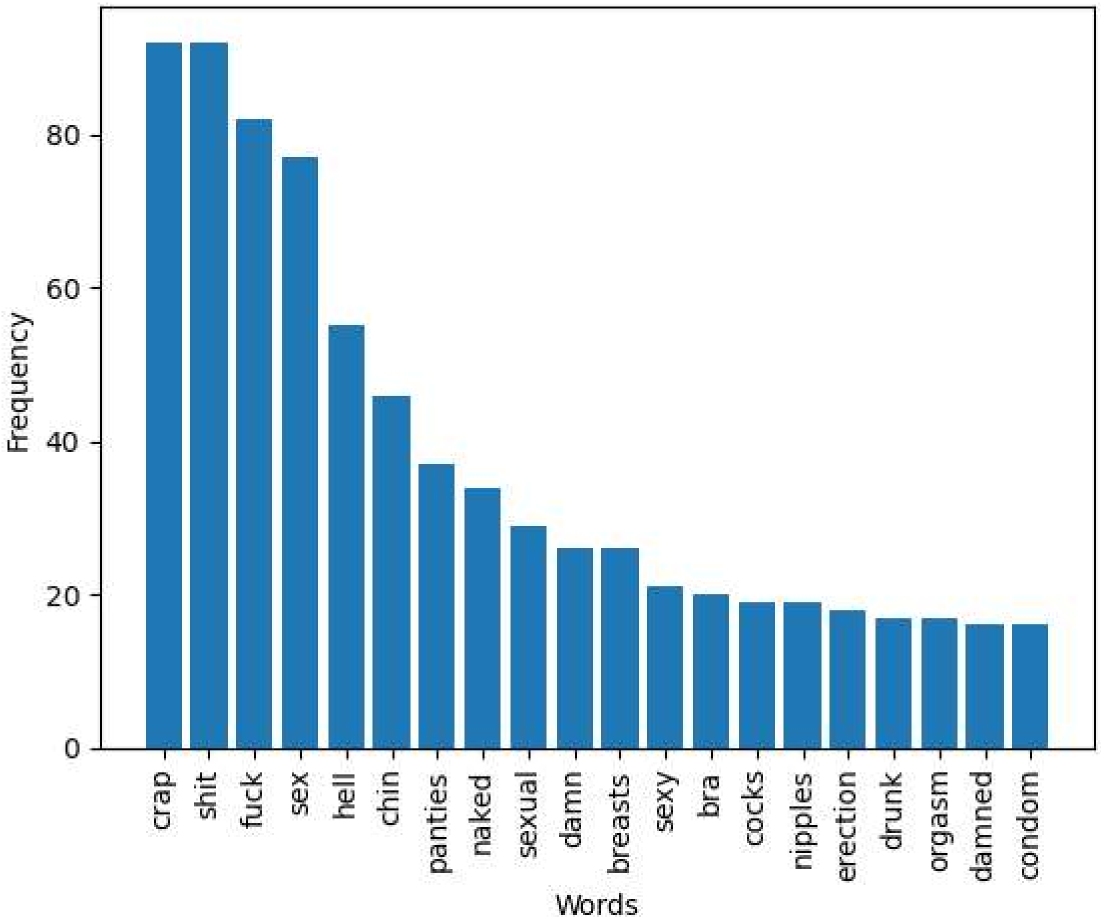
```
with open(file_name, "r") as file:
        # Read the entire contents of the file
        contents = file.read()

    # Create a list of the words in the file
    darker_list = contents.split()
    darker_list = contents.lower().translate(str.maketrans("", "", string.punctua

    # Replace any alert words with asterisks
    new_list = [word if word not in alert_words else "****" for word in darker_li

    # Count the frequency of each asterisk-replaced word
    word_counts = {}
    for word in darker_list:
        if word in alert_words:
            if word in word_counts:
                word_counts[word] += 1
            else:
                word_counts[word] = 1

    # Plot a bar graph of the word frequencies
    plt.bar(word_counts.keys(), word_counts.values())
    plt.xticks(rotation=90)
    plt.xlabel("Words")
    plt.ylabel("Frequency")
    plt.show()

    # Join the list back into a string with spaces between each word
    new_contents = " ".join(new_list)
```

```
  File "<tokenize>", line 6
    darker_list = contents.split()
    ^
IndentationError: unindent does not match any outer indentation level
```

In [14]:

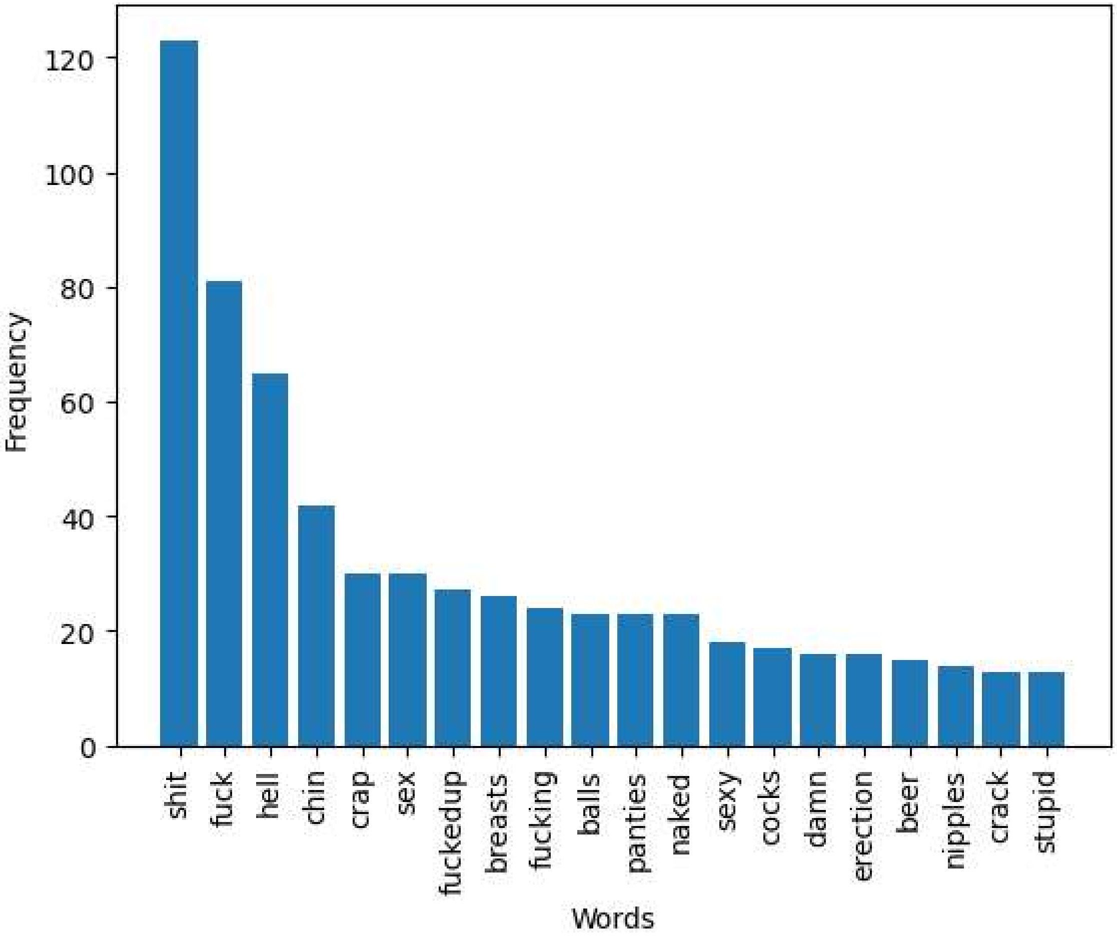
```
import string
from collections import Counter
import matplotlib.pyplot as plt

def plot_word_frequencies(file_name, alert_words_file):
    with open(alert_words_file, "r") as f:
        # Read the alert words from the file
        alert_words = [word.strip() for word in f]

    with open(file_name, "r") as file:
        # Read the entire contents of the file
        contents = file.read()

    # Create a list of the words in the file
    darker_list = contents.lower().translate(str.maketrans("", "", string.punctua

    # Replace any alert words with asterisks
    new_list = [word if word not in alert_words else "****" for word in darker_li

    # Count the frequency of each asterisk-replaced word
    word_counts = {}
    for word in darker_list:
        if word in alert_words:
            if word in word_counts:
                word_counts[word] += 1
            else:
                word_counts[word] = 1


    # Find the most repeated words
    top_words = sorted(word_counts, key=word_counts.get, reverse=True)[:20]
    top_word_counts = {word: word_counts[word] for word in top_words}

    # Plot a bar graph of the word frequencies
    plt.bar(top_word_counts.keys(), top_word_counts.values())
    plt.xticks(rotation=90)
    plt.xlabel("Words")
    plt.ylabel("Frequency")
    plt.show()

    # Join the list back into a string with spaces between each word
    new_contents = " ".join(new_list)
```

```
In [15]: plot_word_frequencies("50.txt", "full-list-of-bad-words_text-file_2022_05_05.txt'
```



In [16]:

```
plot_word_frequencies("Fifty-Shades-Darker.txt", "full-list-of-bad-words_text-fil
```



In [ ]:

```
```