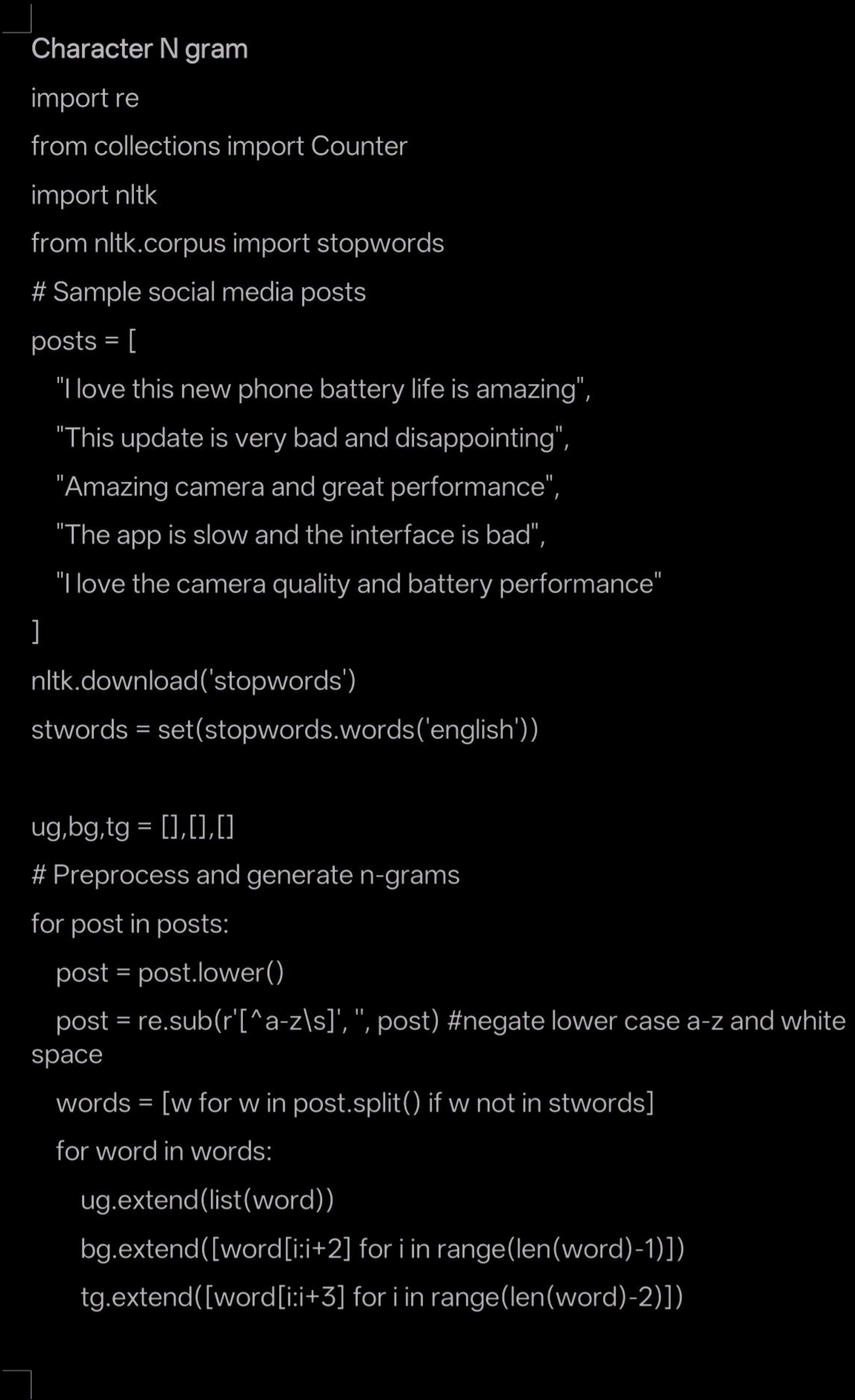
```
import re
from collections import Counter
import nltk
from nltk.corpus import stopwords
# Sample social media posts
posts = [
    "I love this new phone battery life is amazing",
    "This update is very bad and disappointing",
    "Amazing camera and great performance",
    "The app is slow and the interface is bad",
    "I love the camera quality and battery performance"
]
nltk.download('stopwords')
stwords = set(stopwords.words('english'))
ug,bg,tg = [],[],[]
# Preprocess and generate n-grams
for post in posts:
    post = post.lower()
    post = re.sub(r'[^a-z\s]', '', post) #negate lower case a-z and white space
    words = [w for w in post.split() if w not in stwords]
    ug.extend(words)
    bg.extend(zip(words, words[1:]))
    tg.extend(zip(words, words[1:], words[2:]))
# Count frequencies
ugc = Counter(ug)
bgc = Counter(bg)
tgc = Counter(tg)
```

```
# Display trending phrases
print("Top Unigrams:",ugc.most_common(3))
print("\nTop Bigrams:",bgc.most_common(3))
print("\nTop Trigrams:",tgc.most_common(3))
# Simple sentiment analysis
from textblob import TextBlob
print("\nSentiment Analysis:")
for post in posts:
    blob = TextBlob(post)
    polarity = blob.sentiment.polarity
    if polarity > 0:
        print("Positive")
    elif polarity < 0:
        print("Negative")
    else:
        print("Neutral")
    print(f"Post: '{post}' → Polarity: {polarity}")
```

Output

```
Top Unigrams: [('love', 2), ('battery', 2), ('amazing', 2)]

Top Bigrams: [(('love', 'new'), 1), (('new', 'phone'), 1), (('phone', 'battery'), 1)]Top Trigrams: [(('love', 'new', 'phone'), 1), (('new', 'phone', 'battery'), 1), (('phone', 'battery', 'life'), 1)]


Sentiment Analysis:

Positive

Post: 'I love this new phone battery life is amazing' → Polarity: 0.4121212121212121

Negative

Post: 'This update is very bad and disappointing' → Polarity: -0.7549999999999999

Positive

Post: 'Amazing camera and great performance' → Polarity: 0.7000000000000001

Negative

Post: 'The app is slow and the interface is bad' → Polarity: -0.49999999999999994

Positive

Post: 'I love the camera quality and battery performance' → Polarity: 0.5
```



Character N gram

```
import re
from collections import Counter
import nltk
from nltk.corpus import stopwords
# Sample social media posts
posts = [
    "I love this new phone battery life is amazing",
    "This update is very bad and disappointing",
    "Amazing camera and great performance",
    "The app is slow and the interface is bad",
    "I love the camera quality and battery performance"
]
nltk.download('stopwords')
stwords = set(stopwords.words('english'))

ug,bg,tg = [],[],[]
# Preprocess and generate n-grams
for post in posts:
    post = post.lower()
    post = re.sub(r'[^a-z\s]', '', post) #negate lower case a-z and white space
    words = [w for w in post.split() if w not in stwords]
    for word in words:
        ug.extend(list(word))
        bg.extend([word[i:i+2] for i in range(len(word)-1)])
        tg.extend([word[i:i+3] for i in range(len(word)-2)])
```

```
# Count frequencies
ugc = Counter(ug)
bgc = Counter(bg)
tgc = Counter(tg)
# Display trending phrases
print("Top Unigrams:",ugc.most_common(3))
print("\nTop Bigrams:",bgc.most_common(3))
print("\nTop Trigrams:",tgc.most_common(3))
# Simple sentiment analysis
from textblob import TextBlob
print("\nSentiment Analysis:")
for post in posts:
    blob = TextBlob(post)
    polarity = blob.sentiment.polarity
    if polarity > 0:
        print("Positive")
    elif polarity < 0:
        print("Negative")
    else:
        print("Neutral")
    print(f"Post: '{post}' → Polarity: {polarity}")
```

Output

```
Top Unigrams: [('a', 20), ('e', 17), ('r', 10)]
Top Bigrams: [('er', 7), ('in', 5), ('ba', 4)]
Top Trigrams: [('ter', 3), ('ing', 3), ('erf', 3)]
Sentiment Analysis:
Positive
Post: 'I love this new phone battery life is amazing' → Polarity: 0.4121212121212121
Negative
Post: 'This update is very bad and disappointing' → Polarity: -0.7549999999999999
Positive
Post: 'Amazing camera and great performance' → Polarity: 0.7000000000000001
Negative
Post: 'The app is slow and the interface is bad' → Polarity: -0.49999999999999994
Positive
Post: 'I love the camera quality and battery performance' → Polarity: 0.5
```