In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import requests
from tqdm import tqdm_notebook
from bs4 import BeautifulSoup
```

In [2]:

```
import warnings
warnings.filterwarnings("ignore")
```

In [3]:

```
url = "https://www.amazon.in/Redmi-Horizon-Qualcomm%C2%AE-SnapdragonTM-Included/product-revi
headers = {"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTM
             "X-Amzn-Trace-Id": "Root=1-63cf0052-760b19697375364569cfc0d0"}
```

In [4]:

```
def get_soup(url):
    r = requests.get(url, headers=headers,params={'url': url, 'wait': 2})
    soup = BeautifulSoup(r.text, 'html.parser')
    return soup
```

In [5]:

```
reviewlist = []

def get_reviews(soup):
    reviews = soup.find_all('div', {'data-hook': 'review'})
    try:
        for item in reviews:
            review = {
            'Reviews': item.find('span', {'data-hook': 'review-body'}).text.strip(),
            }
            reviewlist.append(review)
    except:
        pass
```

In [6]:

```
for x in tqdm_notebook(range(1,500)):
    soup = get_soup(f'https://www.amazon.in/Redmi-Horizon-Qualcomm%C2%AE-SnapdragonTM-Includ
    get_reviews(soup)
    if not soup.find('li', {'class': 'a-disabled a-last'}):
        pass
    else:
        break
```

```
100%                                499/499 [05:53<00:00, 1.50it/s]
```

In [7]:

```
df = pd.DataFrame(reviewlist)
```

In [8]:

```
df.head()
```

Out[8]:

	Reviews
0	The first look of this starbust design is eye ...
1	Camera is not so good it's averageBack camera...
2	50 days usage...1. Good battery LIFE. Bqest fo...
3	Good phone for average users. (4/64 GB)Battery...
4	Good performance... Good touch response... Goo...

In [9]:

```
df.tail()
```

Out[9]:

	Reviews
4985	If one needs to use this phone only for daily ...
4986	Battery issue battery gets drain i don't know ...
4987	Speaker quality is not good
4988	Quite a smooth daily driver phone. The overall...
4989	Don’t buy this phone its a wrong choice its no...

In [10]:

```
df.shape
```

Out[10]:

```
(4990, 1)
```

In [11]:

```
df.columns
```

Out[11]:

```
Index(['Reviews'], dtype='object')
```

In [12]:

```
df.duplicated().sum()
```

Out[12]:

```
4980
```

In [13]:

```
df.isnull().sum()
```

Out[13]:

```
Reviews    0
dtype: int64
```

In [14]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4990 entries, 0 to 4989
Data columns (total 1 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   Reviews  4990 non-null   object
dtypes: object(1)
memory usage: 39.1+ KB
```

In [15]:

```
text_total = " ".join(df["Reviews"])
```

In [16]:

```
text_total[:2000]
```

Out[16]:

```
"The first look of this starbust design is eye catchy...no word to express its
design,in hand feel and and it's all physical dimension. the second this which
impress me, it's battery backup.....even you playing game hardly for a long ti
me....it gives you sufficient battey backup and charging speed also 33W, is al
so a good deal at this price. The third impressive factor is it's dual speake
r....I have an another phone of 40k and the sound quality is almost same as i
t. Media playback experience is also enhanced by it's amoled display, very col
our charming display...... literally the display quality is unbeatable. The ca
mera notch and dual speaker increase the media playing experience. Other Perfo
rmance like gaming is also good at this price point. Fingerprint button is als
o very responsive. All other things and features are good to the price point.O
nly one thing is very disappointing bcoz the hype of its camera that company s
hows is very up to the mark and professional photography but the camera is low
er grade...even my old Redmi Y2 which I exchanged with this, had better camera
than this ....So plz I request you to all .... don't buy this for camera....ju
st don't. The images are shown of camera pictures on the website is illustrati
on according to me......I don't recommend you to buy this phone as your camera
phone.If you are non camera user then you will surely go for it.Pros- design,
in-hand feel, battery and charging, amoled display,dual speakerCons- camera😤
😤 Camera is not so good it's  averageBack camera is good at this price 💙🌀
Look is cool back and front both 🌀Battery 🔋 is great with 33watt fast charg
ingMobile exchange is great 💙 it's simple and easy 💙🌀Overall good  for no
rmal user i purchased for my sister 💙❤Thank you Amazon for great deal and s
ervice 💙🌀 50 days usage...1. Good battery LIFE. Bqest for day-day use. 33W
charger also good.2. Best display but should've 120hz instead of 90.3. Sturdy
in-hand feel4. Stereo speaker is loud and crisp(65-35 split).5. Camer"
```

In [17]:

```
chars = sorted(list(set(text_total)))
vocab_size = len(chars)
print(''.join(chars))
print(vocab_size)
```

```
 "&'()*+,-./0123456789:ABCDEFGHIKLMNOPQRSTUVWYabcdefghijklmnopqrstuvwxyz’❤🌀
💙🔋😤
79
```

In [18]:

```
df_new = df.copy()
```

In [19]:

```
import re
import string
```

In [20]:

```
def cleaning_text(text):
    text = text.lower()
    text = re.sub('😤', "", text)
    text = re.sub('\[.*?\]', '', text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub('[‘’“”…]', '', text)
    return text

cleaning = lambda x: cleaning_text(x)
```

In [21]:

```
df_new['Cleaned_Reviews'] = df_new.Reviews.apply(cleaning)
```

In [22]:

```
df_new
```

Out[22]:

	Reviews	Cleaned_Reviews
0	The first look of this starbust design is eye ...	the first look of this starbust design is eye ...
1	Camera is not so good it's averageBack camera...	camera is not so good its averageback camera ...
2	50 days usage...1. Good battery LIFE. Bqest fo...	50 days usage1 good battery life bqest for day...
3	Good phone for average users. (4/64 GB)Battery...	good phone for average users 464 gbbattery bac...
4	Good performance... Good touch response... Goo...	good performance good touch response good batt...
...	...	...
4985	If one needs to use this phone only for daily ...	if one needs to use this phone only for daily ...
4986	Battery issue battery gets drain i don't know ...	battery issue battery gets drain i dont know w...
4987	Speaker quality is not good	speaker quality is not good
4988	Quite a smooth daily driver phone. The overall...	quite a smooth daily driver phone the overall ...
4989	Don’t buy this phone its a wrong choice its no...	dont buy this phone its a wrong choice its not...

4990 rows × 2 columns

In [23]:

```
clean_total = " ".join(df_new["Cleaned_Reviews"])
chars = sorted(list(set(clean_total)))
vocab_size = len(chars)
print(''.join(chars))
print(vocab_size)
```

```
 0123456789abcdefghijklmnopqrstuvwxyz❤🌀💙🔋
42
```

In [24]:

```
df_new = df_new[df_new['Cleaned_Reviews']!='']
```

In [25]:

```
df_new
```

Out[25]:

	Reviews	Cleaned_Reviews
0	The first look of this starbust design is eye ...	the first look of this starbust design is eye ...
1	Camera is not so good it's averageBack camera...	camera is not so good its averageback camera ...
2	50 days usage...1. Good battery LIFE. Bqest fo...	50 days usage1 good battery life bqest for day...
3	Good phone for average users. (4/64 GB)Battery...	good phone for average users 464 gbbattery bac...
4	Good performance... Good touch response... Goo...	good performance good touch response good batt...
...	...	...
4985	If one needs to use this phone only for daily ...	if one needs to use this phone only for daily ...
4986	Battery issue battery gets drain i don't know ...	battery issue battery gets drain i dont know w...
4987	Speaker quality is not good	speaker quality is not good
4988	Quite a smooth daily driver phone. The overall...	quite a smooth daily driver phone the overall ...
4989	Don't buy this phone its a wrong choice its no...	dont buy this phone its a wrong choice its not...

4990 rows × 2 columns

In [26]:

```
from textblob import TextBlob
```

In [28]:

```
df_new['Cleaned_Reviews'][:10].apply(lambda x: str(TextBlob(x).correct()))
```

Out[28]:

```
0    the first look of this starbust design is eye ...
1    camera is not so good its  averageback camera ...
2    50 days usage good battery life best for daddy...
3    good phone for average users 464 battery back ...
4    good performance good touch response good batt...
5    if one needs to use this phone only for daily ...
6    battery issue battery gets drain i dont know w...
7                          speaker quality is not good
8    quite a smooth daily driver phone the overall ...
9    dont buy this phone its a wrong choice its not...
Name: Cleaned_Reviews, dtype: object
```

In [30]:

```
df_new
```

Out[30]:

	Reviews	Cleaned_Reviews
0	The first look of this starbust design is eye ...	the first look of this starbust design is eye ...
1	Camera is not so good it's averageBack camera...	camera is not so good its averageback camera ...
2	50 days usage...1. Good battery LIFE. Bqest fo...	50 days usage1 good battery life bqest for day...
3	Good phone for average users. (4/64 GB)Battery...	good phone for average users 464 gbbattery bac...
4	Good performance... Good touch response... Goo...	good performance good touch response good batt...
...	...	...
4985	If one needs to use this phone only for daily ...	if one needs to use this phone only for daily ...
4986	Battery issue battery gets drain i don't know ...	battery issue battery gets drain i dont know w...
4987	Speaker quality is not good	speaker quality is not good
4988	Quite a smooth daily driver phone. The overall...	quite a smooth daily driver phone the overall ...
4989	Don’t buy this phone its a wrong choice its no...	dont buy this phone its a wrong choice its not...

4990 rows × 2 columns

In [31]:

```
df_new.shape
```

Out[31]:

```
(4990, 2)
```

In [32]:

```
df_new.duplicated().sum()
```

Out[32]:

```
4980
```

In [33]:

```
df_new['Cleaned_Reviews'].duplicated().sum()
```

Out[33]:

```
4980
```

In [34]:

```
df_new.drop_duplicates(subset=['Cleaned_Reviews'], keep=False)
```

Out[34]:

Reviews	Cleaned_Reviews

In [35]:

```
df_new.shape
```

Out[35]:

```
(4990, 2)
```

In [37]:

```
freq = pd.Series(' '.join(df_new['Cleaned_Reviews']).split()).value_counts()[:10]
```

In [38]:

```
freq
```

Out[38]:

```
is        13972
the        9980
for        9980
good       9980
this       6986
camera     6986
and        6986
to         5988
its        5489
you        4491
dtype: int64
```

In [39]:

```
import nltk
nltk.download("popular")
```

```
[nltk_data] Downloading collection 'popular'
[nltk_data]    |
[nltk_data]    | Downloading package cmudict to
[nltk_data]    |     C:\Users\hp5cd\AppData\Roaming\nltk_data...
[nltk_data]    |   Unzipping corpora\cmudict.zip.
[nltk_data]    | Downloading package gazetteers to
[nltk_data]    |     C:\Users\hp5cd\AppData\Roaming\nltk_data...
[nltk_data]    |   Unzipping corpora\gazetteers.zip.
[nltk_data]    | Downloading package genesis to
[nltk_data]    |     C:\Users\hp5cd\AppData\Roaming\nltk_data...
[nltk_data]    |   Unzipping corpora\genesis.zip.
[nltk_data]    | Downloading package gutenberg to
[nltk_data]    |     C:\Users\hp5cd\AppData\Roaming\nltk_data...
[nltk_data]    |   Unzipping corpora\gutenberg.zip.
[nltk_data]    | Downloading package inaugural to
[nltk_data]    |     C:\Users\hp5cd\AppData\Roaming\nltk_data...
[nltk_data]    |   Unzipping corpora\inaugural.zip.
[nltk_data]    | Downloading package movie_reviews to
[nltk_data]    |     C:\Users\hp5cd\AppData\Roaming\nltk_data...
[nltk_data]    |   Unzipping corpora\movie_reviews.zip.
[nltk_data]    | Downloading package names to
[nltk_data]    |     C:\Users\hp5cd\AppData\Roaming\nltk_data...
[nltk_data]    |   Unzipping corpora\names.zip.
[nltk_data]    | Downloading package shakespeare to
[nltk_data]    |     C:\Users\hp5cd\AppData\Roaming\nltk_data...
[nltk_data]    |   Unzipping corpora\shakespeare.zip.
[nltk_data]    | Downloading package stopwords to
[nltk_data]    |     C:\Users\hp5cd\AppData\Roaming\nltk_data...
[nltk_data]    |   Package stopwords is already up-to-date!
[nltk_data]    | Downloading package treebank to
[nltk_data]    |     C:\Users\hp5cd\AppData\Roaming\nltk_data...
[nltk_data]    |   Unzipping corpora\treebank.zip.
[nltk_data]    | Downloading package twitter_samples to
[nltk_data]    |     C:\Users\hp5cd\AppData\Roaming\nltk_data...
[nltk_data]    |   Unzipping corpora\twitter_samples.zip.
[nltk_data]    | Downloading package omw to
[nltk_data]    |     C:\Users\hp5cd\AppData\Roaming\nltk_data...
[nltk_data]    | Downloading package omw-1.4 to
[nltk_data]    |     C:\Users\hp5cd\AppData\Roaming\nltk_data...
[nltk_data]    |   Package omw-1.4 is already up-to-date!
[nltk_data]    | Downloading package wordnet to
[nltk_data]    |     C:\Users\hp5cd\AppData\Roaming\nltk_data...
[nltk_data]    |   Package wordnet is already up-to-date!
[nltk_data]    | Downloading package wordnet2021 to
[nltk_data]    |     C:\Users\hp5cd\AppData\Roaming\nltk_data...
[nltk_data]    | Downloading package wordnet31 to
[nltk_data]    |     C:\Users\hp5cd\AppData\Roaming\nltk_data...
[nltk_data]    | Downloading package wordnet_ic to
[nltk_data]    |     C:\Users\hp5cd\AppData\Roaming\nltk_data...
[nltk_data]    |   Unzipping corpora\wordnet_ic.zip.
[nltk_data]    | Downloading package words to
[nltk_data]    |     C:\Users\hp5cd\AppData\Roaming\nltk_data...
[nltk_data]    |   Package words is already up-to-date!
[nltk_data]    | Downloading package maxent_ne_chunker to
[nltk_data]    |     C:\Users\hp5cd\AppData\Roaming\nltk_data...
[nltk_data]    |   Package maxent_ne_chunker is already up-to-date!
[nltk_data]    | Downloading package punkt to
[nltk_data]    |     C:\Users\hp5cd\AppData\Roaming\nltk_data...
[nltk_data]    |   Package punkt is already up-to-date!
[nltk_data]    | Downloading package snowball_data to
[nltk_data]    |     C:\Users\hp5cd\AppData\Roaming\nltk_data...
[nltk_data]    | Downloading package averaged_perceptron_tagger to
```

```
[nltk_data]    |     C:\Users\hp5cd\AppData\Roaming\nltk_data...
[nltk_data]    |   Package averaged_perceptron_tagger is already up-
[nltk_data]    |       to-date!
[nltk_data]    |
[nltk_data]  Done downloading collection popular
```

Out[39]:

```
True
```

In [40]:

```
from nltk.corpus import stopwords
stop = stopwords.words('english')
```

In [41]:

```
df_new['Cleaned_Reviews'] = df_new['Cleaned_Reviews'].apply(lambda x: " ".join(x for x in x.
```

In [42]:

```
df_new
```

Out[42]:

	Reviews	Cleaned_Reviews
0	The first look of this starbust design is eye ...	first look starbust design eye catchyno word e...
1	Camera is not so good it's averageBack camera...	camera good averageback camera good price 💙 🌀 lo...
2	50 days usage...1. Good battery LIFE. Bqest fo...	50 days usage1 good battery life bqest dayday ...
3	Good phone for average users. (4/64 GB)Battery...	good phone average users 464 gbbattery backup ...
4	Good performance... Good touch response... Goo...	good performance good touch response good batt...
...	...	...
4985	If one needs to use this phone only for daily ...	one needs use phone daily usage purpose best p...
4986	Battery issue battery gets drain i don't know ...	battery issue battery gets drain dont know thi...
4987	Speaker quality is not good	speaker quality good
4988	Quite a smooth daily driver phone. The overall...	quite smooth daily driver phone overall experi...
4989	Don’t buy this phone its a wrong choice its no...	dont buy phone wrong choice working 5months ic...

4990 rows × 2 columns

In [44]:

```
TextBlob(df_new['Cleaned_Reviews'][4]).ngrams(1)
```

Out[44]:

```
[WordList(['good']),
 WordList(['performance']),
 WordList(['good']),
 WordList(['touch']),
 WordList(['response']),
 WordList(['good']),
 WordList(['battery']),
 WordList(['life']),
 WordList(['good']),
 WordList(['charging']),
 WordList(['speedaverage']),
 WordList(['camera']),
 WordList(['clarity']),
 WordList(['average']),
 WordList(['led']),
 WordList(['flash']),
 WordList(['lightnote']),
 WordList(['selfie']),
 WordList(['photo']),
 WordList(['lovers']),
 WordList(['dont']),
 WordList(['go']),
 WordList(['average']),
 WordList(['mobile']),
 WordList(['money'])]
```

In [47]:

```
TextBlob(df_new['Cleaned_Reviews'][2]).ngrams(2)
```

```
Out[47]:
```

```
[WordList(['50', 'days']),
 WordList(['days', 'usage1']),
 WordList(['usage1', 'good']),
 WordList(['good', 'battery']),
 WordList(['battery', 'life']),
 WordList(['life', 'bqest']),
 WordList(['bqest', 'dayday']),
 WordList(['dayday', 'use']),
 WordList(['use', '33w']),
 WordList(['33w', 'charger']),
 WordList(['charger', 'also']),
 WordList(['also', 'good2']),
 WordList(['good2', 'best']),
 WordList(['best', 'display']),
 WordList(['display', 'shouldve']),
 WordList(['shouldve', '120hz']),
 WordList(['120hz', 'instead']),
 WordList(['instead', '903']),
 WordList(['903', 'sturdy']),
 WordList(['sturdy', 'inhand']),
 WordList(['inhand', 'feel4']),
 WordList(['feel4', 'stereo']),
 WordList(['stereo', 'speaker']),
 WordList(['speaker', 'loud']),
 WordList(['loud', 'crisp6535']),
 WordList(['crisp6535', 'split5']),
 WordList(['split5', 'camera']),
 WordList(['camera', 'average']),
 WordList(['average', 'good']),
 WordList(['good', 'concerning']),
 WordList(['concerning', 'price6']),
 WordList(['price6', 'buggy']),
 WordList(['buggy', 'uiminor']),
 WordList(['uiminor', 'expected']),
 WordList(['expected', 'ui13']),
 WordList(['ui13', 'way']),
 WordList(['way', 'better']),
 WordList(['better', 'bloody']),
 WordList(['bloody', 'miui']),
 WordList(['miui', '1257']),
 WordList(['1257', '4g']),
 WordList(['4g', 'chipset']),
 WordList(['chipset', 'sd']),
 WordList(['sd', '680']),
 WordList(['680', 'fine']),
 WordList(['fine', '4g']),
 WordList(['4g', 'always']),
 WordList(['always', 'available8']),
 WordList(['available8', 'minor']),
 WordList(['minor', 'issue']),
 WordList(['issue', 'wifi']),
 WordList(['wifi', 'reception']),
 WordList(['reception', 'dont']),
 WordList(['dont', 'know']),
 WordList(['know', 'may']),
 WordList(['may', 'fix']),
 WordList(['fix', 'ota']),
 WordList(['ota', 'phone']),
 WordList(['phone', 'people']),
 WordList(['people', 'clicks']),
 WordList(['clicks', 'photos']),
 WordList(['photos', 'casual']),
 WordList(['casual', 'gamer']),
 WordList(['gamer', 'binge']),
 WordList(['binge', 'watchers']),
 WordList(['watchers', 'web']),
 WordList(['web', 'showsmovies']),
 WordList(['showsmovies', 'intended']),
 WordList(['intended', 'heavy']),
 WordList(['heavy', 'gamingnot']),
 WordList(['gamingnot', 'pubg']),
 WordList(['pubg', 'cod']),
 WordList(['cod', 'lovers'])]
```

In [48]:

```
TextBlob(df_new['Cleaned Reviews'][3]).ngrams(3)
```

Out[48]:

```
[WordList(['good', 'phone', 'average']),
 WordList(['phone', 'average', 'users']),
 WordList(['average', 'users', '464']),
 WordList(['users', '464', 'gbbattery']),
 WordList(['464', 'gbbattery', 'backup']),
 WordList(['gbbattery', 'backup', 'good']),
 WordList(['backup', 'good', 'last']),
 WordList(['good', 'last', 'one']),
 WordList(['last', 'one', 'day']),
 WordList(['one', 'day', 'normal']),
 WordList(['day', 'normal', 'usage']),
 WordList(['normal', 'usage', 'use']),
 WordList(['usage', 'use', 'continue']),
 WordList(['use', 'continue', 'gps']),
 WordList(['continue', 'gps', 'results']),
 WordList(['gps', 'results', 'may']),
 WordList(['results', 'may', 'vary']),
 WordList(['may', 'vary', 'get']),
 WordList(['vary', 'get', '1011']),
 WordList(['get', '1011', 'hrs']),
 WordList(['1011', 'hrs', 'backupheating']),
 WordList(['hrs', 'backupheating', 'issues']),
 WordList(['backupheating', 'issues', 'much']),
 WordList(['issues', 'much', 'lessbattery']),
 WordList(['much', 'lessbattery', 'charging']),
 WordList(['lessbattery', 'charging', 'fast']),
 WordList(['charging', 'fast', 'gets']),
 WordList(['fast', 'gets', 'full']),
 WordList(['gets', 'full', 'charge']),
 WordList(['full', 'charge', '0']),
 WordList(['charge', '0', '100']),
 WordList(['0', '100', 'within']),
 WordList(['100', 'within', '1']),
 WordList(['within', '1', 'hr']),
 WordList(['1', 'hr', '10']),
 WordList(['hr', '10', 'minsback']),
 WordList(['10', 'minsback', 'camera']),
 WordList(['minsback', 'camera', 'good']),
 WordList(['camera', 'good', 'sunlight']),
 WordList(['good', 'sunlight', 'conditionsfront']),
 WordList(['sunlight', 'conditionsfront', 'camera']),
 WordList(['conditionsfront', 'camera', 'average']),
 WordList(['camera', 'average', 'poor']),
 WordList(['average', 'poor', 'makes']),
 WordList(['poor', 'makes', 'blurry']),
 WordList(['makes', 'blurry', 'photos']),
 WordList(['blurry', 'photos', 'daylight']),
 WordList(['photos', 'daylight', 'toonot']),
 WordList(['daylight', 'toonot', 'recomended']),
 WordList(['toonot', 'recomended', 'photography']),
 WordList(['recomended', 'photography', 'heavy']),
 WordList(['photography', 'heavy', 'userssatsfactory']),
 WordList(['heavy', 'userssatsfactory', 'performance']),
 WordList(['userssatsfactory', 'performance', 'daily']),
 WordList(['performance', 'daily', 'taskers'])]
```

In [49]:

```
freq_Sw = pd.Series(' '.join(df_new['Cleaned_Reviews']).split()).value_counts()[:20]
```

In [50]:

```
freq_Sw
```

Out[50]:

```
good           9980
camera         6986
dont           4491
battery        4491
also           3493
phone          3493
average        2495
charging       2495
price          2495
display        1996
quality        1996
great          1996
life           1996
use            1497
120hz          1497
experience     1497
speaker        1497
performance    1497
quite          1497
daily          1497
dtype: int64
```

In [51]:

```
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer(min_df = 1, max_df = 0.9)
X = vectorizer.fit_transform(df_new["Cleaned_Reviews"])
word_freq_df = pd.DataFrame({'term': vectorizer.get_feature_names(), 'occurrences':np.asarra
word_freq_df['frequency'] = word_freq_df['occurrences']/np.sum(word_freq_df['occurrences'])
print(word_freq_df.sort_values('occurrences',ascending = False).head())
```

```
        term  occurrences  frequency
117     good         9980   0.048193
42    camera         6986   0.033735
28   battery         4990   0.024096
78      dont         4491   0.021687
185    phone         3493   0.016867
```

In [53]:

```
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer(stop_words='english', max_features= 1000, max_df = 0.5, smooth_
doc_vec = vectorizer.fit_transform(df_new["Cleaned_Reviews"])
names_features = vectorizer.get_feature_names()
dense = doc_vec.todense()
denselist = dense.tolist()
```

In [54]:

```
df1 = pd.DataFrame(denselist, columns = names_features)
```

In [55]:

```
df1
```

Out[55]:

	10	100	1011	120hz	1257	33w	33watt	40k	464	
0	0.000000	0.000000	0.000000	0.000000	0.000000	0.067587	0.000000	0.085519	0.000000	(
1	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.212641	0.000000	0.000000	(
2	0.000000	0.000000	0.000000	0.101258	0.128125	0.101258	0.000000	0.000000	0.000000	(
3	0.163288	0.163288	0.163288	0.000000	0.000000	0.000000	0.000000	0.000000	0.163288	(
4	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	(
...	...	...	...	...	...	...	...	...	...	...
4985	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	(
4986	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	(
4987	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	(
4988	0.000000	0.000000	0.000000	0.261404	0.000000	0.000000	0.000000	0.000000	0.000000	(
4989	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	(

4990 rows × 251 columns

In [56]:

```
def get_top_n2_words(corpus, n=None):
    vec1 = CountVectorizer(ngram_range=(2,2),  #for tri-gram, put ngram_range=(3,3)
            max_features=2000).fit(corpus)
    bag_of_words = vec1.transform(corpus)
    sum_words = bag_of_words.sum(axis=0)
    words_freq = [(word, sum_words[0, idx]) for word, idx in
                  vec1.vocabulary_.items()]
    words_freq =sorted(words_freq, key = lambda x: x[1],
                reverse=True)
    return words_freq[:n]
```

In [58]:

```
top2_words = get_top_n2_words(df_new["Cleaned_Reviews"], n=200)
top2_df = pd.DataFrame(top2_words)
top2_df.columns=["Bi-gram", "Freq"]
top2_df.head()
```

Out[58]:

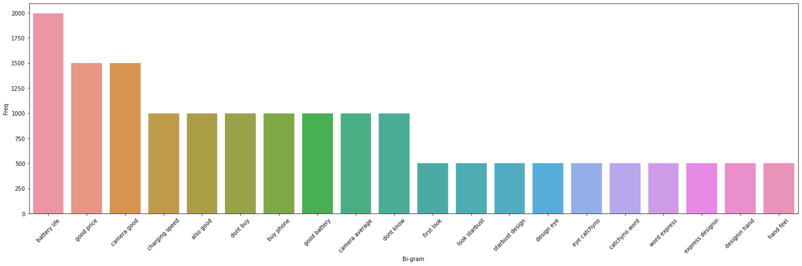
	Bi-gram	Freq
0	battery life	1996
1	good price	1497
2	camera good	1497
3	charging speed	998
4	also good	998

In [59]:

```
top20_bigram = top2_df.iloc[0:20,:]
fig = plt.figure(figsize = (25, 7))
plot=sns.barplot(x=top20_bigram["Bi-gram"],y=top20_bigram["Freq"])
plot.set_xticklabels(rotation=45,labels = top20_bigram["Bi-gram"])
```

Out[59]:

```
[Text(0, 0, 'battery life'),
 Text(1, 0, 'good price'),
 Text(2, 0, 'camera good'),
 Text(3, 0, 'charging speed'),
 Text(4, 0, 'also good'),
 Text(5, 0, 'dont buy'),
 Text(6, 0, 'buy phone'),
 Text(7, 0, 'good battery'),
 Text(8, 0, 'camera average'),
 Text(9, 0, 'dont know'),
 Text(10, 0, 'first look'),
 Text(11, 0, 'look starbust'),
 Text(12, 0, 'starbust design'),
 Text(13, 0, 'design eye'),
 Text(14, 0, 'eye catchyno'),
 Text(15, 0, 'catchyno word'),
 Text(16, 0, 'word express'),
 Text(17, 0, 'express designin'),
 Text(18, 0, 'designin hand'),
 Text(19, 0, 'hand feel')]
```



In [60]:

```
def get_top_n3_words(corpus, n=None):
    vec1 = CountVectorizer(ngram_range=(3,3),
           max_features=2000).fit(corpus)
    bag_of_words = vec1.transform(corpus)
    sum_words = bag_of_words.sum(axis=0)
    words_freq = [(word, sum_words[0, idx]) for word, idx in
                  vec1.vocabulary_.items()]
    words_freq =sorted(words_freq, key = lambda x: x[1],
                reverse=True)
    return words_freq[:n]
```

In [61]:

```
top3_words = get_top_n3_words(df_new["Cleaned_Reviews"], n=200)
top3_df = pd.DataFrame(top3_words)
top3_df.columns=["Tri-gram", "Freq"]
top3_df
```

Out[61]:

	Tri-gram	Freq
0	good battery life	998
1	first look starbust	499
2	look starbust design	499
3	starbust design eye	499
4	design eye catchyno	499
...	...	...
195	price6 buggy uiminor	499
196	buggy uiminor expected	499
197	uiminor expected ui13	499
198	expected ui13 way	499
199	ui13 way better	499

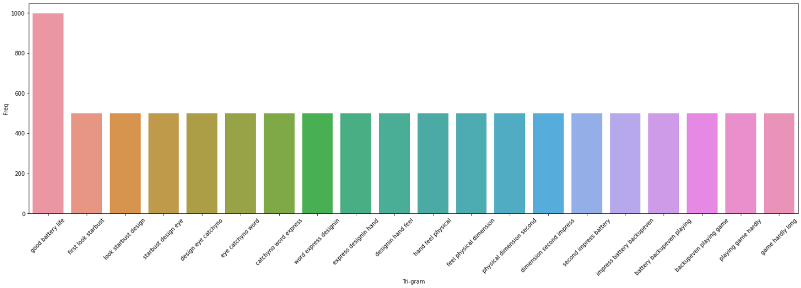
200 rows × 2 columns

In [62]:

```
top20_trigram = top3_df.iloc[0:20,:]
fig = plt.figure(figsize = (25, 7))
plot=sns.barplot(x=top20_trigram["Tri-gram"],y=top20_trigram["Freq"])
plot.set_xticklabels(rotation=45,labels = top20_trigram["Tri-gram"])
```

Out[62]:

```
[Text(0, 0, 'good battery life'),
 Text(1, 0, 'first look starbust'),
 Text(2, 0, 'look starbust design'),
 Text(3, 0, 'starbust design eye'),
 Text(4, 0, 'design eye catchyno'),
 Text(5, 0, 'eye catchyno word'),
 Text(6, 0, 'catchyno word express'),
 Text(7, 0, 'word express designin'),
 Text(8, 0, 'express designin hand'),
 Text(9, 0, 'designin hand feel'),
 Text(10, 0, 'hand feel physical'),
 Text(11, 0, 'feel physical dimension'),
 Text(12, 0, 'physical dimension second'),
 Text(13, 0, 'dimension second impress'),
 Text(14, 0, 'second impress battery'),
 Text(15, 0, 'impress battery backupeven'),
 Text(16, 0, 'battery backupeven playing'),
 Text(17, 0, 'backupeven playing game'),
 Text(18, 0, 'playing game hardly'),
 Text(19, 0, 'game hardly long')]
```



In [63]:

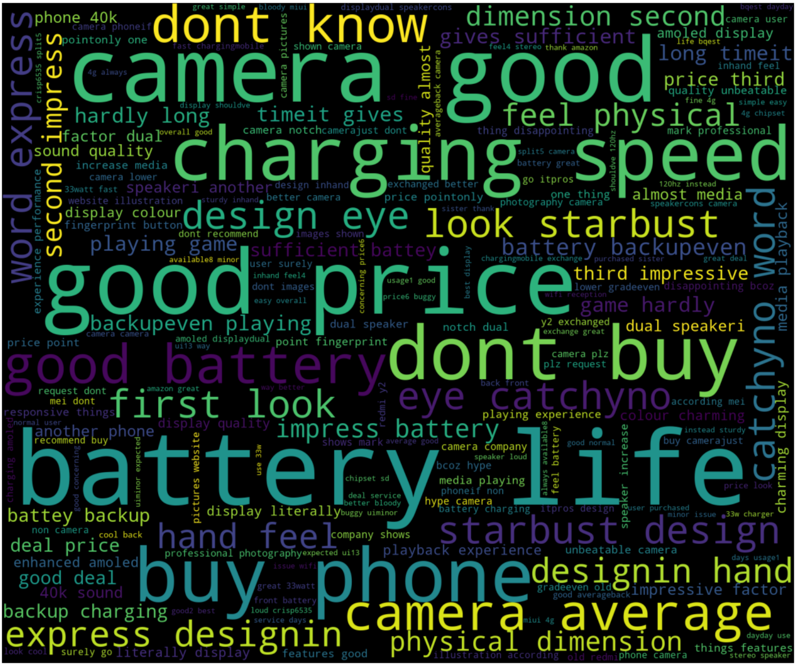
```
string_total = " ".join(df_new["Cleaned_Reviews"])
string_total[:2000]
```

Out[63]:

```
'first look starbust design eye catchyno word express designin hand feel physi
cal dimension second impress battery backupeven playing game hardly long timei
t gives sufficient battey backup charging speed also 33w also good deal price
third impressive factor dual speakeri another phone 40k sound quality almost m
edia playback experience also enhanced amoled display colour charming display
literally display quality unbeatable camera notch dual speaker increase media
playing experience performance like gaming also good price point fingerprint b
utton also responsive things features good price pointonly one thing disappoin
ting bcoz hype camera company shows mark professional photography camera lower
gradeeven old redmi y2 exchanged better camera plz request dont buy camerajust
dont images shown camera pictures website illustration according mei dont reco
mmend buy phone camera phoneif non camera user surely go itpros design inhand
feel battery charging amoled displaydual speakercons camera camera good averag
eback camera good price 💙🌀look cool back front 🌀battery 🔋 great 33watt f
ast chargingmobile exchange great 💙 simple easy 💙🌀overall good normal use
r purchased sister 💙❤️thank amazon great deal service 💙🌀 50 days usage1 g
ood battery life bqest dayday use 33w charger also good2 best display shouldve
120hz instead 903 sturdy inhand feel4 stereo speaker loud crisp6535 split5 cam
era average good concerning price6 buggy uiminor expected ui13 way better bloo
dy miui 1257 4g chipset sd 680 fine 4g always available8 minor issue wifi rece
ption dont know may fix ota phone people clicks photos casual gamer binge watc
hers web showsmovies intended heavy gamingnot pubg cod lovers good phone avera
ge users 464 gbbattery backup good last one day normal usage use continue gps
results may vary get 1011 hrs backupheating issues much lessbattery charging f
ast gets full charge 0 100 within 1 hr 10 minsback camera good sunlight condit
ionsfront camera average poor makes blurry photos daylight toono'
```

In [65]:

```
from wordcloud import WordCloud
wordcloud_generator = WordCloud(background_color= 'black', width = 1800, height = 1500).gene
fig = plt.figure(figsize = (20,20))
plt.imshow(wordcloud_generator)
plt.axis('off')
plt.show()
```



In [66]:

```
def fetch_sentiment_using_textblob(text):
    analysis = TextBlob(text)
    return 'positive' if analysis.sentiment.polarity >= 0 else 'negative'
```

In [68]:

```
fetch_sentiment_using_textblob = df_new.Cleaned_Reviews.apply(lambda tweet: fetch_sentiment_
pd.DataFrame(fetch_sentiment_using_textblob.value_counts())
```

Out[68]:

	Cleaned_Reviews
positive	4491
negative	499

In [69]:

```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
nltk.download('vader_lexicon')
```

```
[nltk_data] Downloading package vader_lexicon to
[nltk_data]     C:\Users\hp5cd\AppData\Roaming\nltk_data...
```

Out[69]:

```
True
```

In [70]:

```
sid = SentimentIntensityAnalyzer()
sid.polarity_scores(df_new.Cleaned_Reviews[4])
```

Out[70]:

```
{'neg': 0.0, 'neu': 0.518, 'pos': 0.482, 'compound': 0.9493}
```

In [71]:

```
df_score=pd.DataFrame()
df_score['Cleaned_Reviews'] = df_new.Cleaned_Reviews
df_score['scores'] = df_new['Cleaned_Reviews'].apply(lambda review: sid.polarity_scores(revi
df_score['compound']  = df_score['scores'].apply(lambda scores: scores['compound'])
df_score['sentiment'] = df_score['compound'].apply(lambda c: 'Positive' if c >=0.75 else ('N
```

In [72]:

```
df_score
```

Out[72]:

	Cleaned_Reviews	scores	compound	sentiment
0	first look starbust design eye catchyno word e...	{'neg': 0.045, 'neu': 0.717, 'pos': 0.238, 'co...	0.9824	Positive
1	camera good averageback camera good price 💙 🌀lo...	{'neg': 0.0, 'neu': 0.441, 'pos': 0.559, 'comp...	0.9796	Positive
2	50 days usage1 good battery life bqest dayday ...	{'neg': 0.033, 'neu': 0.743, 'pos': 0.224, 'co...	0.9432	Positive
3	good phone average users 464 gbbattery backup ...	{'neg': 0.071, 'neu': 0.791, 'pos': 0.138, 'co...	0.6369	Neutral
4	good performance good touch response good batt...	{'neg': 0.0, 'neu': 0.518, 'pos': 0.482, 'comp...	0.9493	Positive
...	...	...	...	...
4985	one needs use phone daily usage purpose best p...	{'neg': 0.111, 'neu': 0.694, 'pos': 0.194, 'co...	0.4203	Neutral
4986	battery issue battery gets drain dont know thi...	{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound...	0.0000	Neutral
4987	speaker quality good	{'neg': 0.0, 'neu': 0.408, 'pos': 0.592, 'comp...	0.4404	Neutral
4988	quite smooth daily driver phone overall experi...	{'neg': 0.0, 'neu': 0.712, 'pos': 0.288, 'comp...	0.9434	Positive
4989	dont buy phone wrong choice working 5months ic...	{'neg': 0.0, 'neu': 0.704, 'pos': 0.296, 'comp...	0.6192	Neutral

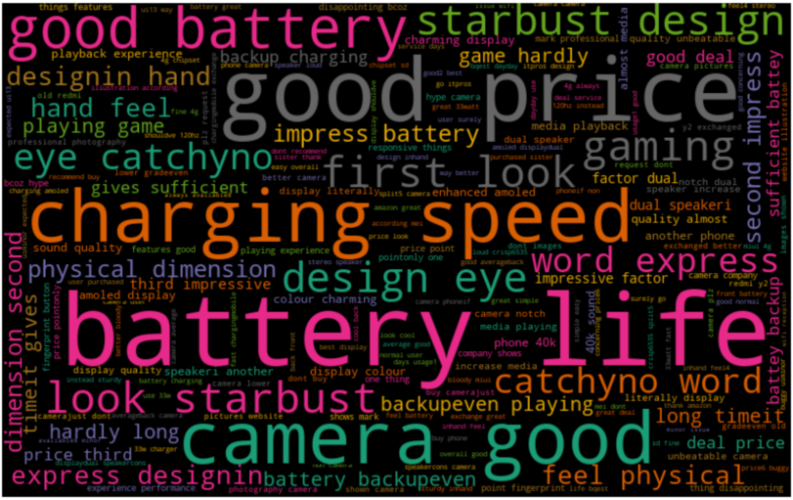
4990 rows × 4 columns

In [73]:

```
def generate_wordcloud(all_words):
    wordcloud = WordCloud(width=800, height=500, random_state=21, max_font_size=100, relativ
    plt.figure(figsize=(14, 10))
    plt.imshow(wordcloud, interpolation="bilinear")
    plt.axis('off')
    plt.show()
```

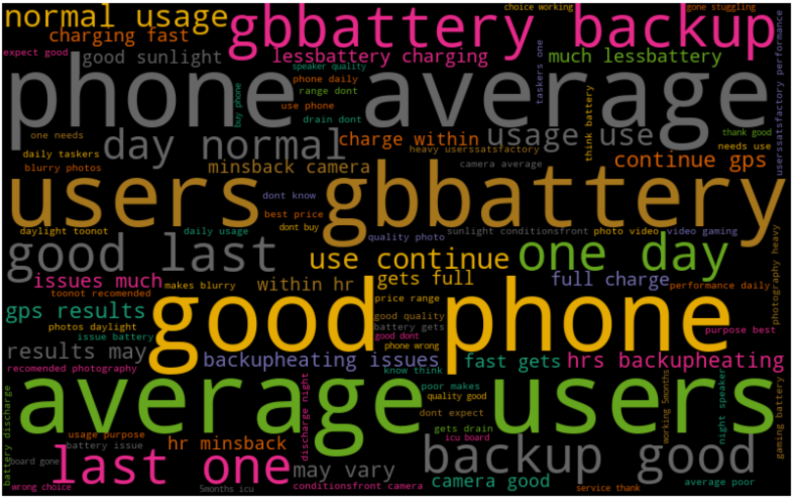
In [74]:

```
all_words = ' '.join([text for text in df_score['Cleaned_Reviews'][df_score.sentiment == 'Po
generate_wordcloud(all_words)
```



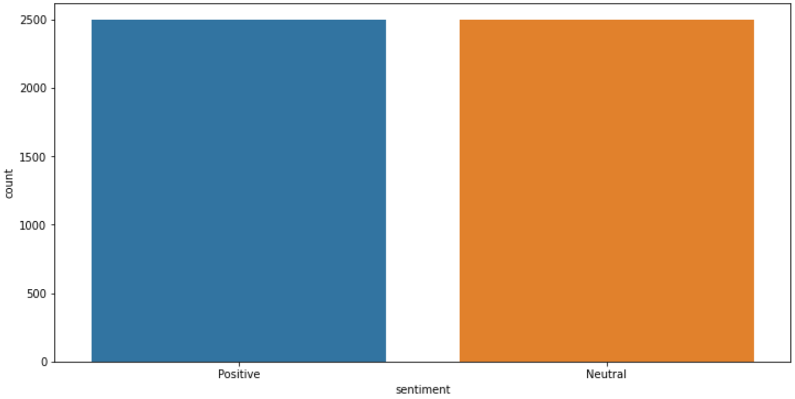
In [75]:

```
all_words = ' '.join([text for text in df_score['Cleaned_Reviews'][df_score.sentiment == 'Ne
generate_wordcloud(all_words)
```



In [77]:

```
plt.figure(figsize=(12,6))
sns.countplot(x='sentiment',data=df_score)
plt.show()
```



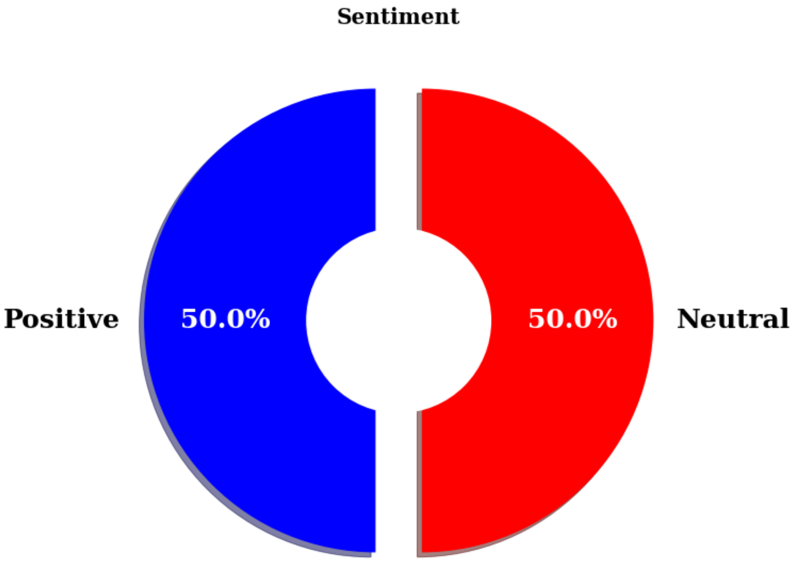
In [78]:

```
label_data = df_score['sentiment'].value_counts()

explode = (0.1, 0.1)
plt.figure(figsize=(14, 10))
patches, texts, pcts = plt.pie(label_data,
                               labels = label_data.index,
                               colors = ['blue', 'red'],
                               pctdistance = 0.65,
                               shadow = True,
                               startangle = 90,
                               explode = explode,
                               autopct = '%1.1f%%',
                               textprops={ 'fontsize': 25,
                                           'color': 'black',
                                           'weight': 'bold',
                                           'family': 'serif' })
plt.setp(pcts, color='white')

hfont = {'fontname':'serif', 'weight': 'bold'}
plt.title('Sentiment', size=20, **hfont)

centre_circle = plt.Circle((0,0),0.40,fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)
plt.show()
```


Sentiment
Positive
Neutral