In [2]:

```
1 pip install gensim
```

```
Requirement already satisfied: gensim in c:\users\kiran\anaco
nda3\lib\site-packages (4.1.2)Note: you may need to restart t
he kernel to use updated packages.

Requirement already satisfied: numpy>=1.17.0 in c:\users\kira
n\anaconda3\lib\site-packages (from gensim) (1.23.5)
Requirement already satisfied: scipy>=0.18.1 in c:\users\kira
n\anaconda3\lib\site-packages (from gensim) (1.9.1)
Requirement already satisfied: smart-open>=1.8.1 in c:\users
\kiran\anaconda3\lib\site-packages (from gensim) (5.2.1)
```

In [3]:

```
1 pip install python-Levenshtein
```

```
Collecting python-Levenshtein
  Downloading python_Levenshtein-0.21.0-py3-none-any.whl (9.4
kB)
Collecting Levenshtein==0.21.0 (from python-Levenshtein)
  Downloading Levenshtein-0.21.0-cp39-cp39-win_amd64.whl (101
kB)
     ---------------------------------- 101.0/101.0 kB 832.
8 kB/s eta 0:00:00
Collecting rapidfuzz<4.0.0,>=2.3.0 (from Levenshtein==0.21.0-
>python-Levenshtein)
  Downloading rapidfuzz-3.0.0-cp39-cp39-win_amd64.whl (1.8 M
B)
     ---------------------------------------- 1.8/1.8 MB 5.0
MB/s eta 0:00:00
Installing collected packages: rapidfuzz, Levenshtein, python
-Levenshtein
Successfully installed Levenshtein-0.21.0 python-Levenshtein-
0.21.0 rapidfuzz-3.0.0
Note: you may need to restart the kernel to use updated packa
ges.
```

In [5]:

```
1 import gensim
2 import pandas as pd
```

Reading and Exploring the Dataset

The dataset we are using here is a subset of Amazon reviews from the Cell Phones & Accessories category. The Data is stored as a JSON file and can be read using pandas.

Link to the Dataset:
http://snap.stanford.edu/data/amazon/productGraph/categoryFiles/reviews_Cell_Phones_and_
(http://snap.stanford.edu/data/amazon/productGraph/categoryFiles/reviews_Cell_Phones_and

In [11]:

```
1 df=pd.read_json("Cell_Phones_and_Accessories_5.json",lines=True)
```

In [12]:

```
1 df.head()
```

Out[12]:

	reviewerID	asin	reviewerName	helpful	reviewText	overa
0	A30TL5EWN6DFXT	120401325X	christina	[0, 0]	They look good and stick good! I just don't li...	
1	ASY55RVNIL0UD	120401325X	emily l.	[0, 0]	These stickers work like the review says they ...	
2	A2TMXE2AFO7ONB	120401325X	Erica	[0, 0]	These are awesome and make my phone look so st...	
3	AWJ0WZQYMYFQ4	120401325X	JM	[4, 4]	Item arrived in great time and was in perfect ...	
4	ATX7CZYFXI1KW	120401325X	patrice m rogoza	[2, 3]	awesome! stays on, and looks great. can be use...	

In [13]:

```
1 df.shape
```

Out[13]:

```
(194439, 9)
```

In [15]:

```
1 df.reviewText[0]
```

Out[15]:

```
"They look good and stick good! I just don't like the rounded
shape because I was always bumping it and Siri kept popping u
p and it was irritating. I just won't buy a product like this
again"
```

In [17]:

```
1 gensim.utils.simple_preprocess("They look good and stick good! I just d
2 )
```

Out[17]:

```
['they',
 'look',
 'good',
 'and',
 'stick',
 'good',
 'just',
 'don',
 'like',
 'the',
 'rounded',
 'shape',
 'because',
 'was',
 'always',
 'bumping',
 'it',
 'and',
 'siri',
 'kept',
 'popping',
 'up',
 'and',
 'it',
 'was',
 'irritating',
 'just',
 'won',
 'buy',
 'product',
 'like',
 'this',
 'again']
```

In [18]:

```
review_text=df.reviewText.apply(gensim.utils.simple_preprocess)
review_text
```

Out[18]:

```
0         [they, look, good, and, stick, good, just, don...
1         [these, stickers, work, like, the, review, say...
2         [these, are, awesome, and, make, my, phone, lo...
3         [item, arrived, in, great, time, and, was, in,...
4         [awesome, stays, on, and, looks, great, can, b...
                                ...
194434    [works, great, just, like, my, original, one, ...
194435    [great, product, great, packaging, high, quali...
194436    [this, is, great, cable, just, as, good, as, t...
194437    [really, like, it, becasue, it, works, well, w...
194438    [product, as, described, have, wasted, lot, of...
Name: reviewText, Length: 194439, dtype: object
```

In [19]:

```
#building a model
model=gensim.models.Word2Vec(
    window=10,
    min_count=2,
    workers=4
)
```

In [20]:

```
#building vocabulary
model.build_vocab(review_text, progress_per=1000)
```

In [21]:

```
model.epochs
```

Out[21]:

```
5
```

In [22]:

```
model.corpus_count
```

Out[22]:

```
194439
```

In [23]:

```
#train the model
model.train(review_text,total_examples=model.corpus_count,epochs=model.
```

Out[23]:

```
(61503826, 83868975)
```

In [24]:

```
#saving model to the current directory
model.save('./word2vec-amazon-cell-accessories-reviews-short.model')
```

In [33]:

```
#find similar words
model.wv.most_similar('bad')
```

Out[33]:

```
[('terrible', 0.6739454865455627),
 ('shabby', 0.6507302522659302),
 ('horrible', 0.647965669631958),
 ('good', 0.5997362732887268),
 ('crappy', 0.5657022595405579),
 ('okay', 0.5574902296066284),
 ('awful', 0.5489994883537292),
 ('cheap', 0.5367509126663208),
 ('ok', 0.5237422585487366),
 ('poor', 0.5218642950057983)]
```

In [34]:

```
#function to find out the similarity between two words
model.wv.similarity(w1='cheap',w2='inexpensive')
```

Out[34]:

```
0.5192925
```

In [28]:

```
model.wv.similarity(w1='great',w2='good')
```

Out[28]:

```
0.7805511
```

In [29]:

```
1 model.wv.similarity(w1='great',w2='product')
```

Out[29]:

-0.030929063

In [30]:

```
1 model.wv.similarity(w1='great',w2='awesome')
```

Out[30]:

0.7473884

In [31]:

```
1 model.wv.similarity(w1='great',w2='nice')
```

Out[31]:

0.6849427

In [32]:

```
1 model.wv.similarity(w1='great',w2='iphone')
```

Out[32]:

0.10205598

In [ ]:

```
1
```