NLP - Topic Analysis ¶

In [53]:
```
import pandas as pd
import nltk
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
from string import punctuation
from nltk.corpus import stopwords
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
import re
import gensim
import gensim.corpora as corpora
from gensim.models import CoherenceModel
from gensim.models import ldamodel
```

A popular mobile phone brand has launched their smartphone in market. The client wants to understand the VOC (voice of the customer) on the product. This will be useful to not just evaluate the current product, but to also get some direction for developing the product pipeline. The client is particularly interested in the different aspects that customers care about. Product reviews by customers on a leading e-commerce site should provide a good view.

```
Sentiment: The sentiment against the review (4,5 star reviews are positive, 1,2 ar
e negative)
Reviews: The main text of the review
```

1.Read the review data.

In [3]:
```
topicData = pd.read_csv("reviews.csv")
topicData.head()
```

Out[3]:

	sentiment	review
0	1	Good but need updates and improvements
1	0	Worst mobile i have bought ever, Battery is dr...
2	1	when I will get my 10% cash back.... its alrea...
3	1	Good
4	0	The worst phone everThey have changed the last...

2.Normalize casings for the review text and extract the text into a list for easier manipulation.

In [4]:
```
#Casing Normalize
reviewsExtract =  [review.lower() for review in topicData.review.values]
print(reviewsExtract[0:5])
```

```
['good but need updates and improvements', "worst mobile i have bought ever, battery is draining like hell, backup is only 6 to 7 hours with internet use s, even if i put mobile idle its getting discharged.this is biggest lie from amazon & lenove which is not at all expected, they are making full by saying that battery is 4000mah & booster charger is fake, it takes at least 4 to 5 hours to be fully charged.don't know how lenovo will survive by making full of us.please don;t go for this else you will regret like me.", 'when i will get my 10% cash back.... its already 15 january..', 'good', 'the worst phone everthey have changed the last phone but the problem is still same and the a mazon is not returning the phone .highly disappointing of amazon']
```

3.Tokenize the reviews using NLTKs word_tokenize function.

```
Tokenization is essentially splitting a phrase, sentence, paragraph, or an entire
 text document into smaller units, such as individual words or terms
```

In [5]:
```
#Tokenize using NLTK's
reviewTokens =  [word_tokenize( review ) for review in reviewsExtract]
print(reviewTokens[0])
```

```
['good', 'but', 'need', 'updates', 'and', 'improvements']
```

4.Perform parts-of-speech tagging on each sentence using the NLTK POS tagger.

```
POS-tagging simply implies labelling words with their appropriate Part-Of-Speech
 (Noun, Verb, Adjective, Adverb, Pronoun, …). POS tagging can be really useful, pa
rticularly if you have words or tokens that can have multiple POS tags
```

In [6]:
```
#Parts-Of-Speech Tagging
reviewTags =  [nltk.pos_tag(review) for review in reviewTokens]
print(reviewTags[0])
```

```
[('good', 'JJ'), ('but', 'CC'), ('need', 'VBP'), ('updates', 'NNS'), ('and', 'CC'), ('improvements', 'NNS')]
```

5.For the topic model, include all the POS tags that correspond to nouns and Limit the data to only terms with these tags.

```
As corpus tends have a broad and varied vocab-ulary, that can be time consuming to
topic model, limiting articles to only the nouns also offers the advantage of redu
cing the size of the vocabulary to be modelled.Topic Modelling is more efficient i
n noun only approach
```

In [54]:

```
#extraction of tags starts with NN
reviewsNoun = []
for nountag in reviewTags :  reviewsNoun.append([token for token in nountag i
f re.search("NN.*",token[1])])
print(reviewsNoun[0])
```

```
[('updates', 'NNS'), ('improvements', 'NNS')]
```

6.Lemmatize :

```
   lemmatizing means to extract the ‘lemma’ from a given word after its morphologi
cal analysis. For example: If we lemmatize ‘studies’ and ‘studying’, we will end u
p with ‘study’ as its lemma.
```

In [13]:

```
#Lemmatize
reviewsLemm = []
lemm = WordNetLemmatizer()
for data in reviewsNoun :  reviewsLemm.append([lemm.lemmatize(word[0]) for wo
rd in data])
print(reviewsLemm[0])
```

```
['update', 'improvement']
```

7.Remove stopwords and punctuation (if there are any).

In [19]:

```
#Removing Stop Words and Punctuations
stoplist =stopwords.words('english')
stopupdated = stoplist + list(punctuation) + ["..."] + [".."]
reviewsStopRemoved = []
for data in reviewsLemm : reviewsStopRemoved.append([word for word in data if
word not in stopupdated])
print(reviewsStopRemoved[0:3]) #displaying the results
```

```
[['update', 'improvement'], ['mobile', 'battery', 'hell', 'backup', 'hour',
'us', 'idle', 'discharged.this', 'lie', 'amazon', 'lenove', 'battery', 'char
ger', 'hour'], ['cash']]
```

8.Creating a topic model using LDA on the cleaned-up data - with 12 topics.

```
Topic Modeling is a technique to extract the hidden topics from large volumes of t
ext.
Latent Dirichlet Allocation(LDA) is a popular algorithm for topic modeling.

LDA’s considers each document as a collection of topics in a certain proportion. A
nd each topic as a collection of keywords, again, in a certain proportion.

Once you provide the algorithm with the number of topics, all it does it to rearra
nge the topics distribution within the documents and keywords distribution within
 the topics to obtain a good composition of topic-keywords distribution.
```

In [50]:
```python
#index Mapping
wordId = corpora.Dictionary(reviewsStopRemoved)
corpus = [wordId.doc2bow(review) for review in reviewsStopRemoved]

lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus, id2word=wordId,num
_topics=12,random_state=42,passes=10,per_word_topics=True)
list = lda_model.print_topics()
for topic in list:
    print(topic)

#coherence calculations
co_lda_model = CoherenceModel(model=lda_model, texts=reviewsStopRemoved,dicti
onary=wordId,coherence='c_v')
co_lda = co_lda_model.get_coherence()
print('\nResult : ',co_lda)
```

```
(0, '0.138*"mobile" + 0.040*"call" + 0.036*"screen" + 0.031*"feature" + 0.03
0*"option" + 0.020*"music" + 0.017*"software" + 0.016*"app" + 0.015*"video"
+ 0.015*"card"')
(1, '0.151*"money" + 0.128*"...." + 0.071*"waste" + 0.056*"value" + 0.046*"g
lass" + 0.038*"speaker" + 0.024*"gorilla" + 0.022*"set" + 0.022*"ok" + 0.020
*"piece"')
(2, '0.216*"note" + 0.113*"k8" + 0.090*"lenovo" + 0.030*"sound" + 0.023*"dol
by" + 0.020*"killer" + 0.018*"gallery" + 0.018*"system" + 0.018*"atmos" + 0.
018*"excellent"')
(3, '0.078*"phone" + 0.040*"day" + 0.038*"amazon" + 0.035*"service" + 0.034
*"issue" + 0.027*"time" + 0.027*"lenovo" + 0.026*"battery" + 0.024*"month" +
0.023*"device"')
(4, '0.280*"product" + 0.176*"problem" + 0.080*"network" + 0.075*"issue" +
0.066*"heating" + 0.021*"jio" + 0.021*"sim" + 0.019*"volta" + 0.010*"connect
ion" + 0.009*"signal"')
(5, '0.093*"heat" + 0.070*"....." + 0.052*"processor" + 0.038*"everything" +
0.038*"budget" + 0.031*"...." + 0.030*"core" + 0.025*"display" + 0.017*"cel
l" + 0.016*"hr"')
(6, '0.126*"range" + 0.075*"price" + 0.046*"work" + 0.038*"mobile" + 0.038
*"specification" + 0.035*"super" + 0.034*"......" + 0.030*"bit" + 0.026*"ca
m" + 0.025*"k"')
(7, '0.118*"charger" + 0.059*"hai" + 0.056*"handset" + 0.037*"box" + 0.029
*"turbo" + 0.027*"charge" + 0.021*"plz" + 0.016*"hi" + 0.016*"cable" + 0.013
*"bhi"')
(8, '0.242*"price" + 0.065*"superb" + 0.046*"buy" + 0.045*"headphone" + 0.03
9*"thanks" + 0.036*"worth" + 0.034*"feature" + 0.029*"smartphone" + 0.026*"e
xpectation" + 0.017*"offer"')
(9, '0.158*"camera" + 0.136*"battery" + 0.064*"quality" + 0.061*"phone" + 0.
045*"performance" + 0.029*"backup" + 0.019*"issue" + 0.017*"life" + 0.017*"d
ay" + 0.015*"mode"')
(10, '0.548*"phone" + 0.021*"h" + 0.014*"ram" + 0.013*"hang" + 0.012*"gb" +
0.011*"game" + 0.010*"ho" + 0.007*"u" + 0.006*"lot" + 0.006*"interface"')
(11, '0.106*"feature" + 0.061*"delivery" + 0.060*"time" + 0.035*"star" + 0.0
34*"experience" + 0.029*"camera" + 0.023*"condition" + 0.018*"cost" + 0.018
*"class" + 0.017*"awesome"')

Result :  0.475339388396195
```

9.From the business lens, the topics can combine in below ways,

1. Topic 2,5,7 possibly talks about - Pricing
2. Topic 4, 6 and 10 talks about - battery quality Issues
3. Topic 3 and 11 are talks about - Performances

10.Creating the topic model using LDA with the optimal number of topics (Here, I choose 8)

Coherence provides a convenient measure to judge how good a given topic model is.

For finding the optimal number of topics, build many LDA models with different values of number of topics(k) & pick the one that gives the highest coherence value. Choosing a ‘k’ that marks the end of a rapid growth of topic coherence usually offers meaningful & interpretable topics. Picking an even higher value can sometimes provide more granular sub-topics.

In [34]:
```
#Creating model with 8 topics
lda8model = gensim.models.ldamodel.LdaModel(corpus=corpus, id2word=wordId,num_topics=8,random_state=42,passes=10,per_word_topics=True)
co_lda8model = CoherenceModel(model=lda8model, texts=reviewsStopRemoved,dictionary=wordId,coherence='c_v')
co8lda = co_lda8model.get_coherence()
print('\nScore Result : ',co8lda)
```

```
Score Result :  0.5351527233521374
```

The coherence is now 0.53 which is a significant increase from previous value 0.47

1. Creating a table with the topic name and the top 10 terms in each to present to the business.

In [46]:
```
results = lda8model.show_topics(formatted=False)
t_words = [(topics[0],[name[0] for name in topics[1]]) for topics in results]
for t,w in t_words:
    print(str(t) + " : "+str(w))
```

```
0 : ['mobile', 'charging', 'hour', 'charger', 'charge', 'battery', 'turbo', 'hr', 'card', 'notification']
1 : ['money', 'waste', 'value', 'screen', 'glass', 'speaker', 'call', 'handset', 'box', 'headphone']
2 : ['note', 'camera', 'quality', 'k8', 'feature', 'lenovo', 'sound', 'phone', 'music', 'speaker']
3 : ['phone', 'day', 'issue', 'time', 'battery', 'lenovo', 'month', 'problem', 'service', 'update']
4 : ['product', 'problem', 'network', 'issue', 'heating', 'amazon', 'sim', 'return', '....', 'delivery']
5 : ['camera', 'battery', 'phone', 'performance', 'quality', 'backup', '....', 'issue', 'life', 'processor']
6 : ['price', 'phone', 'range', 'superb', 'device', 'super', 'feature', 'excellent', 'specification', 'k']
7 : ['charger', 'hai', 'h', 'ho', 'cable', 'bill', 'bhi', 'hi', 'offer', 'ye']
```

Topic	Business Name
Topic1	Battery Charging capacity
Topic2	Phone Features
Topic3	Camera Quality
Topic4	Battery Related Issues
Topic5	Amazon
Topic6	Performance Issues
Topic7	Pricing
Topic8	Phone Performance

```
In [ ]:
```