In [1]:
```
with open('twitter-shares-jump-after-apples-privacy-changes-have-minimal-impact-on-quarterly-earnings.t
xt') as f:
    contents = f.read()
    # print(contents)
```

1. Remove all company names shown in the article

In [2]:
```
# (Please do not use hardcode to specify the company name)
contents = contents.replace('Twitter', '')
contents = contents.replace('Apple', '')
```

2. Regular Expression/Normalization

In [3]:
```
# lowercase the words,
contents = contents.lower()

# remove punctuation
punc = '''!()-[]{};:\,<>./?@#$%^&*_'‘’“”|~'''
def removePunc(str_lst):
    for j in str_lst:
        if j in punc:
            str_lst = str_lst.replace(j, "")
    return str_lst
contents = removePunc(contents)

# remove numbers
import re
contents = re.sub(r'\d+', '', contents)
# print(contents)
```

3.Tokenization

In [4]:
```
from nltk.tokenize import word_tokenize
tok_contt = word_tokenize(contents)
```

4.Remove stop words

In [5]:
```
import nltk
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\37251\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[5]:
```
True
```

In [25]:
```
# !  pip install stop-words
```

In [6]:
```
from stop_words import get_stop_words
from nltk.corpus import stopwords

stop_words = list(get_stop_words('en'))         #About 900 stopwords
nltk_words = list(stopwords.words('english')) #About 150 stopwords
stop_words.extend(nltk_words)

tok_contt = [w for w in tok_contt if not w in stop_words]
```

5. Lemmatization

In [7]:
```
import nltk
nltk.download('wordnet')
import pandas as pd
from nltk.stem import WordNetLemmatizer
from nltk import pos_tag

wordnet_lemmatizer = WordNetLemmatizer() #defining the object for Lemmatization

def lemmatizer(text):
    # lemmatize include more than plural nouns
    lemm_text = [wordnet_lemmatizer.lemmatize(i,j[0].lower()) if j[0].lower() in ['a','n','v'] else wor
dnet_lemmatizer.lemmatize(i) for i,j in pos_tag(text)]
    return lemm_text
tok_contt = lemmatizer(tok_contt)
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\37251\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

6. Any other pre-processing steps you think is necessary to prepare this input for topic modelling.

In [8]:
```
# remove short words
tok_contt = [w for w in tok_contt if len(w) > 3]
```

In [9]:
```
file_object = open(r"proc_twitter-shares-jump-after-apples-privacy.txt",'w')
file_object.write('\n'.join(tok_contt))
file_object.close()
```

In [10]:
```
from sklearn.feature_extraction.text import TfidfVectorizer,CountVectorizer
from sklearn.decomposition import LatentDirichletAllocation

vectorizer = CountVectorizer()
X = vectorizer.fit_transform(tok_contt)

# # LDA
# lda = LatentDirichletAllocation(n_components=3,  # num_topic
#                                 # max_iter=5,    # EM max iter
#                                 # learning_method='online',
#                                 random_state=0)
# docres = lda.fit_transform(X)
```