In [13]:
```
string_list="        Jack and jill have made a delicious,        dish. Then they started to play some12 game! a
import string
import re


#function to remove punctuation and digits from sample string
def remove_punction(text):
    text_nonpunction = [re.sub('\d',"",char) for char in text if char not in string.punctuation]
    return text_nonpunction

nonpun_string_list=remove_punction(string_list)
print(nonpun_string_list)

# a function to combine all elements
def remove_punct_combine(text):
    text_nonpunct = "".join([char for char in text if char not in string.punctuation])
    return text_nonpunct

clean_string_list=remove_punct_combine(nonpun_string_list)

print(clean_string_list)
```

```
[' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', 'J', 'a', 'c', 'k', ' ', 'a', 'n', 'd', ' ', 'j', 'i', 'l', 'l',
 ' ', 'h', 'a', 'v', 'e', ' ', 'm', 'a', 'd', 'e', ' ', 'a', ' ', 'd', 'e', 'l', 'i', 'c', 'i', 'o', 'u', 's',
 ' ', ' ', ' ', ' ', ' ', ' ', ' ', 'd', 'i', 's', 'h', '.', ' ', 'T', 'h', 'e', 'n', ' ', 't', 'h', 'e', 'y',
 ' ', 's', 't', 'a', 'r', 't', 'e', 'd', ' ', 't', 'o', ' ', 'p', 'l', 'a', 'y', ' ', 's', 'o', 'm', 'e', '', '',
 '', 'g', 'a', 'm', 'e', '!', ' ', 'a', 'n', 'd', ' ', 'j', 'i', 'l', 'l', ' ', 'h', 'a', 's', ' ', 'a', 't', 't',
 'a', 'c', 'h', 'e', 'd', ' ', 'a', ' ', 'p', 'h', 'o', 't', 'o', ' ', 't', 'o', ' ', 't', 'h', 'e', ' ', 's',
 't', 'r', 'a', 'i', 'g', 'h', 't', ' ', 'w', 'a', 'l', 'l', ' ', 'a', 'n', 'd',
 ' ', 's', 'w', 'u', 'n', 'g', ' ', 'o', 'n', ' ', 's', 'e', 'a', 's', 'a', 'w', ' ', 'S', 'h', 'e', ' ', 'w',
 'a', 's', ' ', 'v', 'e', 'r', 'y', ' ', 'h', 'a', 'p', 'p', 'y', ' ', 'A', 'f', 't', 'e', 'r', ' ', 't', 'h',
 'e', ' ', 'g', 'a', 'm', 'e', ' ', 't', 'h', 'e', 'y', ' ', 'b', 'o', 't', 'h', ' ', 'w', 'e', 'n', 't', ' ',
 'c', 'e', 'n', 't', 'r', 'a', 'l', ' ', 't', 'o', ' ', 'e', 'n', 'j', 'o', 'y', ',', ' ', 't', 'o', ' ', 't',
 'h', 'e', ' ', 'f', 'a', 's', 't', ' ', 'f', 'o', 'o', 'd']
        Jack and jill have made a delicious        dish Then they started to play some game and jill has attacha
cd a photo frame to the straight wall and swung on seasaw She was very happy After the game they both went to c
entral London to enjoy some fast food
```

In [3]:
```
#tokenize Sentence
! pip install nltk
import nltk
from nltk.tokenize import word_tokenize, sent_tokenize
sents = sent_tokenize(clean_string_list)
print(sents)
```

```
Requirement already satisfied: nltk in c:\users\yogi\anaconda3\lib\site-packages (3.6.5)
Requirement already satisfied: click in c:\users\yogi\anaconda3\lib\site-packages (from nltk) (8.0.3)
Requirement already satisfied: joblib in c:\users\yogi\anaconda3\lib\site-packages (from nltk) (1.1.0)
Requirement already satisfied: regex>=2021.8.3 in c:\users\yogi\anaconda3\lib\site-packages (from nltk) (2021.
8.3)
Requirement already satisfied: tqdm in c:\users\yogi\anaconda3\lib\site-packages (from nltk) (4.62.3)
Requirement already satisfied: colorama in c:\users\yogi\anaconda3\lib\site-packages (from click->nltk) (0.4.4)
['        Jack and jill have made a delicious        dish Then they started to play some game and jill has atta
hacd a photo frame to the straight wall and swung on seasaw She was very happy After the game they both went to
central London to enjoy some fast food']
```

In [4]:
```
#tokenize words
words = [word_tokenize(sent) for sent in sents]
print(words)
```

```
[['Jack', 'and', 'jill', 'have', 'made', 'a', 'delicious', 'dish', 'Then', 'they', 'started', 'to', 'play', 'so
me', 'game', 'and', 'jill', 'has', 'attahacd', 'a', 'photo', 'frame', 'to', 'the', 'straight', 'wall', 'and',
'swung', 'on', 'seasaw', 'She', 'was', 'very', 'happy', 'After', 'the', 'game', 'they', 'both', 'went', 'to',
'central', 'London', 'to', 'enjoy', 'some', 'fast', 'food']]
```

In [5]:
```
#stopword removed
from nltk.corpus import stopwords
nltk.download('stopwords')
from string import punctuation
customStopWords = set(stopwords.words('english')+list(punctuation))

stopwords_list = [word for word in word_tokenize(clean_string_list) if word not in customStopWords]
print(stopwords_list)
```

```
['Jack', 'jill', 'made', 'delicious', 'dish', 'Then', 'started', 'play', 'game', 'jill', 'attahacd', 'photo',
'frame', 'straight', 'wall', 'swung', 'seasaw', 'She', 'happy', 'After', 'game', 'went', 'central', 'London',
'enjoy', 'fast', 'food']
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Yogi\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

In [6]:
```
#stemming
ps = nltk.PorterStemmer()
def stemming(tokenized_text):
    text =[ps.stem(word) for word in tokenized_text]
    return text

stemming(stopwords_list)
```

Out[6]:
```
['jack',
 'jill',
 'made',
 'delici',
 'dish',
 'then',
 'start',
 'play',
 'game',
 'jill',
 'attahacd',
 'photo',
 'frame',
 'straight',
 'wall',
 'swung',
 'seasaw',
 'she',
 'happi',
 'after',
 'game',
 'went',
 'central',
 'london',
 'enjoy',
 'fast',
 'food']
```

In [7]:
```
#lemmatization
nltk.download('wordnet')
wn = nltk.WordNetLemmatizer()
def lemmatizing(tokenized_text):
    text = [wn.lemmatize(word) for word in tokenized_text]
    return text

final_string=lemmatizing(stopwords_list)
print(final_string)
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\Yogi\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
['Jack', 'jill', 'made', 'delicious', 'dish', 'Then', 'started', 'play', 'game', 'jill', 'attahacd', 'photo',
'frame', 'straight', 'wall', 'swung', 'seasaw', 'She', 'happy', 'After', 'game', 'went', 'central', 'London',
'enjoy', 'fast', 'food']
```

In [14]:
```
#CountVectorizer

import pandas as pd
import re
stopwords = nltk.corpus.stopwords.words('english')
ps = nltk.PorterStemmer()

#creating dataframe here of sample string
df=pd.DataFrame({"string": [string_list]})
print(df)
def clean_text(text):
    #removing digits here
    text = "".join([re.sub('\d',"",word.lower()) for word in text if word not in string.punctuation])
    tokens = re.split('\W+', text)
    text1 = [ps.stem(word) for word in tokens if word not in stopwords]
    return text1


import sklearn

from sklearn.feature_extraction.text import CountVectorizer


count_vect = CountVectorizer(analyzer=clean_text)


X_counts = count_vect.fit_transform(df['string'])

print(X_counts.shape)
print(count_vect.get_feature_names())
X_counts_df = pd.DataFrame(X_counts.toarray())
print(X_counts_df)
```

```
                                              string
0          Jack and jill have made a delicious, ...
(1, 23)
['', 'attahacd', 'central', 'delici', 'dish', 'enjoy', 'fast', 'food', 'frame', 'game', 'happi', 'jack', 'jil
l', 'london', 'made', 'photo', 'play', 'seasaw', 'start', 'straight', 'swung', 'wall', 'went']
   0   1   2   3   4   5   6   7   8   9  ...  13  14  15  16  17  18  19  \
0   1   1   2   1   1   1   1   1   1   2  ...   1   1   1   1   1   1   1

   20  21  22
0   1   1   1

[1 rows x 23 columns]
```

In [10]:
```
#ngram(2,2)
ngram_vect = CountVectorizer(ngram_range=(2,2))
X_counts = ngram_vect.fit_transform(df['string'])
print(X_counts.shape)
print(ngram_vect.get_feature_names())
X_counts_df = pd.DataFrame(X_counts.toarray())
X_counts_df.columns = ngram_vect.get_feature_names()
print(X_counts_df)
```

```
(1, 45)
['after the', 'and jill', 'and swung', 'attahacd photo', 'both went', 'central london', 'delicious dish', 'dish
 then', 'enjoy some', 'fast food', 'frame to', 'game and', 'game they', 'happy after', 'has attahacd', 'have mad
e', 'jack and', 'jill has', 'jill have', 'london to', 'made delicious', 'on sea', 'photo frame', 'play some12',
'saw she', 'sea saw', 'she was', 'some fast', 'some12 game', 'started to', 'straight9 wall', 'swung on', 'the g
ame', 'the straight9', 'then they', 'they both', 'they started', 'to central', 'to enjoy', 'to play', 'to the',
 'very happy', 'wall and', 'was very', 'went to']
   after the  and jill  and swung  attahacd photo  both went  central london  \
0          1          2          1                1          1                1

   delicious dish  dish then  enjoy some  fast food  ...  they both  \
0               1          1           1          1  ...          1

   they started  to central  to enjoy  to play  to the  very happy  wall and  \
0             1           1         1        1       1           1         1

   was very  went to
0         1        1

[1 rows x 45 columns]
```

In [11]:
```
#ngram(2,3)
ngram_vect = CountVectorizer(ngram_range=(2,3))
X_counts = ngram_vect.fit_transform(df['string'])
print(X_counts.shape)
print(ngram_vect.get_feature_names())
X_counts_df = pd.DataFrame(X_counts.toarray())
X_counts_df.columns = ngram_vect.get_feature_names()
print(X_counts_df)
```

```
(1, 90)
['after the', 'after the game', 'and jill', 'and jill has', 'and jill have', 'and swung', 'and swung on', 'atta
hacd photo', 'attahacd photo frame', 'both went', 'both went to', 'central london', 'central london to', 'delic
ious dish', 'delicious dish then', 'dish then', 'dish then they', 'enjoy some', 'enjoy some fast', 'fast food',
'frame to', 'frame to the', 'game and', 'game and jill', 'game they', 'game they both', 'happy after', 'happy a
fter the', 'has attahacd', 'has attahacd photo', 'have made', 'have made delicious', 'jack and', 'jack and jil
l', 'jill has', 'jill has attahacd', 'jill have', 'jill have made', 'london to', 'london to enjoy', 'made delic
ious', 'made delicious dish', 'on sea', 'on sea saw', 'photo frame', 'photo frame to', 'play some12', 'play som
e12 game', 'saw she', 'saw she was', 'sea saw', 'sea saw she', 'she was', 'she was very', 'some fast', 'some fa
st food', 'some12 game', 'some12 game and', 'started to', 'started to play', 'straight9 wall', 'straight9 wall
 and', 'swung on', 'swung on sea', 'the game', 'the game they', 'the straight9', 'the straight9 wall', 'then the
y', 'then they started', 'they both', 'they both went', 'they started', 'they started to', 'to central', 'to ce
ntral london', 'to enjoy', 'to enjoy some', 'to play', 'to play some12', 'to the', 'to the straight9', 'very ha
ppy', 'very happy after', 'wall and', 'wall and swung', 'was very', 'was very happy', 'went to', 'went to centr
al']
   after the  after the game  and jill  and jill has  and jill have  \
0          1               1         1             1              1

   and swung  and swung on  attahacd photo  attahacd photo frame  both went  \
0          1             1                1                     1          1

   ...  to the  to the straight9  very happy  very happy after  wall and  \
0  ...       1                 1           1                 1         1

   wall and swung  was very  was very happy  went to  went to central
0               1         1               1        1                1

[1 rows x 90 columns]
```

In [12]:
```
#TF-IDF

from sklearn.feature_extraction.text import TfidfVectorizer

tfidf_vect = TfidfVectorizer(analyzer=clean_text)
X_tfidf = tfidf_vect.fit_transform(df['string'])
print(X_tfidf.shape)
print(tfidf_vect.get_feature_names())
X_tfidf_df = pd.DataFrame(X_tfidf.toarray())
X_tfidf_df.columns = tfidf_vect.get_feature_names()
print(X_tfidf_df)
```

```
(1, 23)
['', 'attahacd', 'central', 'delici', 'dish', 'enjoy', 'fast', 'food', 'frame', 'game', 'happi', 'jack', 'jil
l', 'london', 'made', 'photo', 'play', 'seasaw', 'start', 'straight', 'swung', 'wall', 'went']
   attahacd   central     delici      dish     enjoy      fast  \
0  0.185695  0.185695  0.185695  0.185695  0.185695  0.185695

       food     frame      game  ...    london      made     photo      play  \
0  0.185695  0.185695  0.371391  ...  0.185695  0.185695  0.185695  0.185695

     seasaw     start  straight      swung      wall      went
0  0.185695  0.185695  0.185695  0.185695  0.185695  0.185695

[1 rows x 23 columns]
```

In [ ]:
```
```