Tokenization

In [1]:

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

```
[nltk_data] Downloading collection 'popular'
[nltk_data]    |
[nltk_data]    | Downloading package cmudict to
[nltk_data]    |     C:\Users\admin\AppData\Roaming\nltk_data...
[nltk_data]    |   Package cmudict is already up-to-date!
[nltk_data]    | Downloading package gazetteers to
[nltk_data]    |     C:\Users\admin\AppData\Roaming\nltk_data...
[nltk_data]    |   Package gazetteers is already up-to-date!
[nltk_data]    | Downloading package genesis to
[nltk_data]    |     C:\Users\admin\AppData\Roaming\nltk_data...
[nltk_data]    |   Package genesis is already up-to-date!
[nltk_data]    | Downloading package gutenberg to
[nltk_data]    |     C:\Users\admin\AppData\Roaming\nltk_data...
[nltk_data]    |   Package gutenberg is already up-to-date!
[nltk_data]    | Downloading package inaugural to
[nltk_data]    |     C:\Users\admin\AppData\Roaming\nltk_data...
[nltk_data]    |   Unzipping corpora\inaugural.zip.
[nltk_data]    | Downloading package movie_reviews to
[nltk_data]    |     C:\Users\admin\AppData\Roaming\nltk_data...
```

In [21]:

```
paragraph="""A true friend loves you unconditionally, understands you, but never judges you and always t
The friendship of Krishna and Sudama is a great example of true friendship.
A true friend is one who will always be there when you need someone.
He will leave all his important work, but will never leave you alone, especially in your difficult times
That is why it is said a friend in need is a friend indeed.
Difficult times are the best time to realize who your true friends are.
Blessed are the souls who have true friends. """
```

In [22]:

```
sentences=nltk.sent_tokenize(paragraph)
print(sentences,end=" ")
```

```
['A true friend loves you unconditionally, understands you, but never judges you and always tries to su
pport you and give you good advice.', 'The friendship of Krishna and Sudama is a great example of true
friendship.', 'A true friend is one who will always be there when you need someone.', 'He will leave al
l his important work, but will never leave you alone, especially in your difficult times.', 'That is wh
y it is said a friend in need is a friend indeed.', 'Difficult times are the best time to realize who y
our true friends are.', 'Blessed are the souls who have true friends.']
```

In [23]:

```
words=nltk.word_tokenize(paragraph)
print(words)
```

```
['A', 'true', 'friend', 'loves', 'you', 'unconditionally', ',', 'understands', 'you', ',', 'but', 'neve
r', 'judges', 'you', 'and', 'always', 'tries', 'to', 'support', 'you', 'and', 'give', 'you', 'good', 'a
dvice', '.', 'The', 'friendship', 'of', 'Krishna', 'and', 'Sudama', 'is', 'a', 'great', 'example', 'o
f', 'true', 'friendship', '.', 'A', 'true', 'friend', 'is', 'one', 'who', 'will', 'always', 'be', 'ther
e', 'when', 'you', 'need', 'someone', '.', 'He', 'will', 'leave', 'all', 'his', 'important', 'work',
',', 'but', 'will', 'never', 'leave', 'you', 'alone', ',', 'especially', 'in', 'your', 'difficult', 'ti
mes', '.', 'That', 'is', 'why', 'it', 'is', 'said', 'a', 'friend', 'in', 'need', 'is', 'a', 'friend',
'indeed', '.', 'Difficult', 'times', 'are', 'the', 'best', 'time', 'to', 'realize', 'who', 'your', 'tru
e', 'friends', 'are', '.', 'Blessed', 'are', 'the', 'souls', 'who', 'have', 'true', 'friends', '.']
```

Part Of Speech Tagging

In [24]:

```
from nltk.corpus import stopwords
```

In [25]:

```
stop_words = set(stopwords.words('english'))
```

In [26]:
```
for i in sentences:
    wordsList = nltk.word_tokenize(i)
    wordsList = [w for w in wordsList if not w in stop_words]
    tagged = nltk.pos_tag(wordsList)
    print(tagged)
```

```
[('A', 'DT'), ('true', 'JJ'), ('friend', 'NN'), ('loves', 'VBZ'), ('unconditionally', 'RB'), (',',
','), ('understands', 'NNS'), (',', ','), ('never', 'RB'), ('judges', 'NNS'), ('always', 'RB'), ('trie
s', 'VBZ'), ('support', 'NN'), ('give', 'VB'), ('good', 'JJ'), ('advice', 'NN'), ('.', '.')]
[('The', 'DT'), ('friendship', 'NN'), ('Krishna', 'NNP'), ('Sudama', 'NNP'), ('great', 'JJ'), ('exampl
e', 'NN'), ('true', 'JJ'), ('friendship', 'NN'), ('.', '.')]
[('A', 'DT'), ('true', 'JJ'), ('friend', 'NN'), ('one', 'CD'), ('always', 'RB'), ('need', 'VB'), ('some
one', 'NN'), ('.', '.')]
[('He', 'PRP'), ('leave', 'VBP'), ('important', 'JJ'), ('work', 'NN'), (',', ','), ('never', 'RB'), ('l
eave', 'VBP'), ('alone', 'RB'), (',', ','), ('especially', 'RB'), ('difficult', 'JJ'), ('times', 'NN
S'), ('.', '.')]
[('That', 'DT'), ('said', 'VBD'), ('friend', 'VBP'), ('need', 'MD'), ('friend', 'VB'), ('indeed', 'R
B'), ('.', '.')]
[('Difficult', 'NNP'), ('times', 'NNS'), ('best', 'JJS'), ('time', 'NN'), ('realize', 'VB'), ('true',
'JJ'), ('friends', 'NNS'), ('.', '.')]
[('Blessed', 'VBN'), ('souls', 'NNS'), ('true', 'JJ'), ('friends', 'NNS'), ('.', '.')]
```

Stop Word Removal

In [36]:
```
for i in range (len(sentences)):
    words=nltk.word_tokenize(sentences[i])
    words=[word for word in words if word not in set(stopwords.words('english'))]
    sentences[i]=' '.join(words)
print(sentences)
```

```
['A true friend loves unconditionally , understands , never judges always tries support give good advic
e .', 'The friendship Krishna Sudama great example true friendship .', 'A true friend one always need s
omeone .', 'He leave important work , never leave alone , especially difficult times .', 'That said fri
end need friend indeed .', 'Difficult times best time realize true friends .', 'Blessed souls true frie
nds .']
```

Stemming

In [37]:
```
from nltk.stem import PorterStemmer
stemmer=PorterStemmer()
for i in range (len(sentences)):
    words=nltk.word_tokenize(sentences[i])
    words=[stemmer.stem(word) for word in words if word not in set(stopwords.words('english'))]
    sentences[i]=' '.join(words)
print(sentences)
```

```
['A true friend love uncondit , understand , never judg alway tri support give good advic .', 'the frie
ndship krishna sudama great exampl true friendship .', 'A true friend one alway need someon .', 'He lea
v import work , never leav alon , especi difficult time .', 'that said friend need friend inde .', 'dif
ficult time best time realiz true friend .', 'bless soul true friend .']
```

Lemmatization

In [38]:
```
from nltk.stem import WordNetLemmatizer
import re
```

In [39]:
```
sentences=nltk.sent_tokenize(paragraph)
lemmatizer=WordNetLemmatizer()
```

In [40]:
```
for i in range (len(sentences)):
    words=nltk.word_tokenize(sentences[i])
    words=[lemmatizer.lemmatize(word) for word in words if word not in set(stopwords.words('english'))]
    sentences[i]=' '.join(words)
print(sentences)
```

```
['A true friend love unconditionally , understands , never judge always try support give good advice
.', 'The friendship Krishna Sudama great example true friendship .', 'A true friend one always need som
eone .', 'He leave important work , never leave alone , especially difficult time .', 'That said friend
need friend indeed .', 'Difficult time best time realize true friend .', 'Blessed soul true friend .']
```

Term Frequency and Inverse Document Frequency

In [41]:
```
ps=PorterStemmer()
wordnet=WordNetLemmatizer()
sentences=nltk.sent_tokenize(paragraph)
corpus=[]
for i in range (len(sentences)):
    review=re.sub('[^a-zA-Z]',' ',sentences[i])
    review=review.lower()
    review=review.split()
    review=[wordnet.lemmatize(word) for word in review if word not in set(stopwords.words('english'))]
    review=' '.join(review)
    corpus.append(review)
```

In [42]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
cv=TfidfVectorizer()
x=cv.fit_transform(corpus).toarray()
```

In [43]:
```
print(x)
```

```
[[0.30205477 0.         0.2507314  0.         0.         0.
  0.         0.         0.16299352 0.         0.30205477 0.30205477
  0.         0.         0.         0.30205477 0.         0.
  0.30205477 0.         0.2507314  0.         0.         0.
  0.         0.         0.         0.30205477 0.         0.16299352
  0.30205477 0.30205477 0.30205477 0.        ]
 [0.         0.         0.         0.         0.         0.
  0.         0.34728953 0.         0.69457906 0.         0.
  0.34728953 0.         0.         0.         0.34728953 0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.34728953 0.         0.         0.18740291
  0.         0.         0.         0.        ]
 [0.         0.         0.41710986 0.         0.         0.
  0.         0.         0.27115153 0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.41710986 0.         0.50249001 0.         0.
  0.50249001 0.         0.         0.         0.         0.27115153
  0.         0.         0.         0.        ]
 [0.         0.3151717  0.         0.         0.         0.26161958
  0.3151717  0.         0.         0.         0.         0.
  0.         0.3151717  0.         0.         0.         0.6303434
  0.         0.         0.26161958 0.         0.         0.
  0.         0.         0.         0.         0.26161958 0.
  0.         0.         0.         0.3151717 ]
 [0.         0.         0.         0.         0.         0.
  0.         0.         0.54975728 0.         0.         0.
  0.         0.         0.50939697 0.         0.         0.
  0.         0.42284323 0.         0.         0.         0.50939697
  0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.        ]
 [0.         0.         0.         0.40731311 0.         0.33810486
  0.         0.         0.21979258 0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.40731311 0.
  0.         0.         0.         0.         0.67620973 0.21979258
  0.         0.         0.         0.        ]
 [0.         0.         0.         0.         0.62228701 0.
  0.         0.         0.33579589 0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.62228701 0.         0.         0.         0.33579589
  0.         0.         0.         0.        ]]
```