import nltk

# nltk.download('punkt')

# nltk.download('wordnet')

# nltk.download('stopwords')

# nltk.download('averaged\_perceptron\_tagger')

d1 = "Yong is a good guy, he is not bad"

d2 = "feet wolves cooked boys girls ,!<@!"

d3 = "Yong is not a good guy, he is bad"

c1 = [d1, d2, d3]

###TOKENIZE

token\_d1 = nltk.word\_tokenize(d1)

print(token\_d1)

tokenizer2 = nltk.tokenize.WhitespaceTokenizer()

token\_d12 = tokenizer2.tokenize(d1)

print(token\_d12)

##BOW Frequency

from sklearn.feature\_extraction.text import CountVectorizer

vectorizer1 = CountVectorizer()

vectorizer1.fit(c1)

print(vectorizer1.vocabulary\_)

v1 = vectorizer1.transform(c1)

print(v1.toarray())

###STEMMER

token\_d2 = nltk.word\_tokenize(d2.lower())

stemmer = nltk.stem.PorterStemmer()

stemmered\_token\_d2 = [stemmer.stem(token) for token in token\_d2 if token.isalpha()]

print(token\_d2)

print(stemmered\_token\_d2)

lemmatizer = nltk.stem.WordNetLemmatizer()

lemmatized\_token\_d2 = [lemmatizer.lemmatize(token) for token in token\_d2 if token.isalpha()]

print(lemmatized\_token\_d2)

###remove stop words

from nltk.corpus import stopwords

stop\_words\_removed = [token for token in token\_d1 if not token in stopwords.words('english') if token.isalpha()]

print(token\_d1)

print(stop\_words\_removed)

### low frequency words

from sklearn.feature\_extraction.text import CountVectorizer

vectorizer1 = CountVectorizer(min\_df=2)

vectorizer1.fit(c1)

print(vectorizer1.vocabulary\_)

v1 = vectorizer1.transform(c1)

print(v1.toarray())

###TFI-IDF

from sklearn.feature\_extraction.text import TfidfVectorizer

vectorizer2 = TfidfVectorizer()

vectorizer2.fit(c1)

print(vectorizer2.vocabulary\_)

v2 = vectorizer2.transform(c1)

print(v2.toarray())

c2 = ["hello world", "Yong is calling"]

v\_c2 = vectorizer2.transform(c2)

print(v\_c2.toarray())

### Bag of 2-grams

vectorizer3 = TfidfVectorizer(ngram\_range=(1, 2), min\_df=2)

vectorizer3.fit(c1)

v3 = vectorizer3.transform(c1)

print(v3.toarray())

print(vectorizer3.vocabulary\_)

### POS TAG

d4 = "I drink water in parties"

d5 = "I grab a drink in parties"

token4 = nltk.word\_tokenize(d4)

POS\_token4 = nltk.pos\_tag(token4)

c2 = [d4, d5]

POS\_c2 = []

for doc in c2:

token\_doc = nltk.word\_tokenize(doc)

POS\_token\_doc = nltk.pos\_tag(token\_doc)

POS\_token\_temp = []

for i in POS\_token\_doc:

POS\_token\_temp.append(i[0] + i[1])

POS\_c2.append(" ".join(POS\_token\_temp))

vectorizer4 = TfidfVectorizer()

vectorizer4.fit(POS\_c2)

print(vectorizer4.vocabulary\_)

POS\_v3 = vectorizer4.transform(POS\_c2)

print(POS\_v3.toarray())