Experiment 03 NLP DLOC Jayesh Ingale CSE(DS)

# Library required

!pip install nltk

Requirement already satisfied: nltk in c:\users\admin\appdata\local\programs\python\python37\lib\site-packages (3.6.2)

Requirement already satisfied: joblib in c:\users\admin\appdata\local\programs\python\python37\lib\site-packages (from nltk) (1.0.0 Requirement already satisfied: click in c:\users\admin\appdata\local\programs\python\python37\lib\site-packages (from nltk) (7.1.2) Requirement already satisfied: regex in c:\users\admin\appdata\local\programs\python\python37\lib\site-packages (from nltk) (2021.4 Requirement already satisfied: tqdm in c:\users\admin\appdata\local\programs\python\python37\lib\site-packages (from nltk) (4.60.0) WARNING: You are using pip version 22.0; however, version 23.2.1 is available.

You should consider upgrading via the 'c:\users\admin\appdata\local\programs\python\python37\python.exe -m pip install --upgrade pi



# Text

text = 'TON 618 is a hyperluminous, broad-absorption-line, radio-loud quasar and Lyman-alpha blob located near the border of the constell

text

'TON 618 is a hyperluminous, broad-absorption-line, radio-loud quasar and Lyman-alpha blob located near the border of the

constellations Canes Venatici and Coma Berenices, with the projected comoving distance of approximately 18.2 billion light-years from Earth.'

# Stopwords

from nltk.corpus import stopwords

stop\_words = stopwords.words('english')

from nltk.tokenize import word\_tokenize words = word\_tokenize(text)

## Applying stop words

holder = list() for w in words:

if w not in set(stop\_words): holder.append(w)

holder

['TON',

'618',

'hyperluminous', ',',

'broad-absorption-line', ',',

'radio-loud',

'quasar',

'Lyman-alpha', 'blob',

'located', 'near',

'border',

'constellations', 'Canes',

'Venatici', 'Coma',

'Berenices',

',',

'projected', 'comoving', 'distance',

'approximately', '18.2',

'billion',

'light-years', 'Earth',

'.']

## List Comprehension for stop words

holder = [w for w in words if w not in set(stop\_words)] print(holder)

['TON', '618', 'hyperluminous', ',', 'broad-absorption-line', ',', 'radio-loud', 'quasar', 'Lyman-alpha', 'blob', 'located', 'near'



# Stemming

from nltk.stem import PorterStemmer, SnowballStemmer, LancasterStemmer

porter = PorterStemmer()

snow = SnowballStemmer(language = 'english') lancaster = LancasterStemmer()

words = ['play', 'plays', 'played', 'playing', 'player']

## Porter Stemmer

porter\_stemmed = list() for w in words:

stemmed\_words = porter.stem(w)

porter\_stemmed.append(stemmed\_words)

porter\_stemmed

['play', 'play', 'play', 'play', 'player']

## Porter Stemmer List Comprehension

porter\_stemmed = [porter.stem(x) for x in words] print (porter\_stemmed)

['play', 'play', 'play', 'play', 'player']

## Snowball Stemmer

snow\_stemmed = list() for w in words:

stemmed\_words = snow.stem(w)

snow\_stemmed.append(stemmed\_words)

snow\_stemmed

['play', 'play', 'play', 'play', 'player']

## Snowball Stemmer List Comprehension

snow\_stemmed = [snow.stem(x) for x in words] print (snow\_stemmed)

['play', 'play', 'play', 'play', 'player']

## Lancaster Stemmer

lancaster\_stemmed = list() for w in words:

stemmed\_words = lancaster.stem(w)

lancaster\_stemmed.append(stemmed\_words)

lancaster\_stemmed

['play', 'play', 'play', 'play', 'play']

## Lancaster Stemmer List Comprehension

lancaster\_stemmed = [lancaster.stem(x) for x in words] print (lancaster\_stemmed)

['play', 'play', 'play', 'play', 'play']

# Lemmatization : This has a more expansive vocabulary than Stemming

from nltk.stem import WordNetLemmatizer wordnet = WordNetLemmatizer()

lemmatized = [wordnet.lemmatize(x) for x in words]

lemmatized

['play', 'play', 'played', 'playing', 'player']