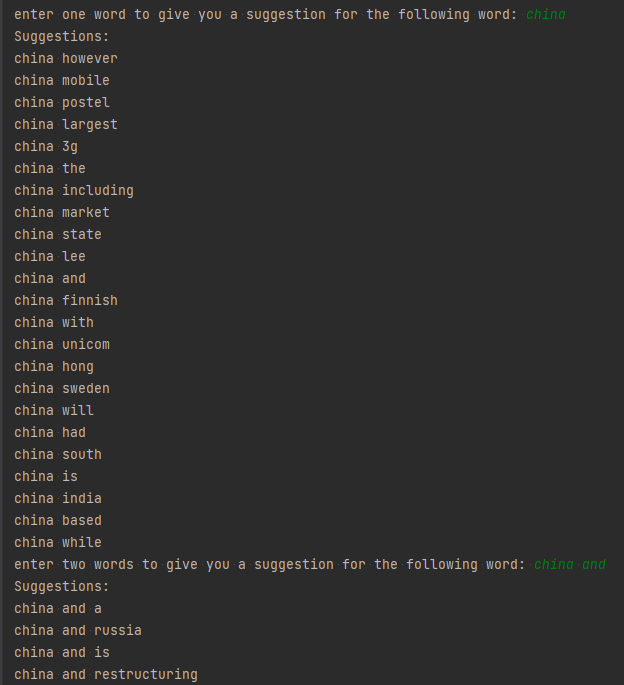
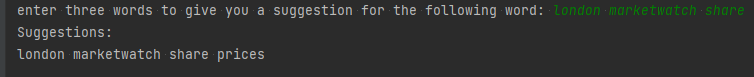
Lab 11

Word Suggestion with N-Grams

1. What do you notice about the number of suggestions you get for a single word, versus the number of suggestions you get when you enter two words.
   1. There are more suggestions for single words than for two words
2. For example, enter the single word china, and see how many suggestions you get. Then enter the two words china and, and see how many suggestions you get.
   1. 23
   2. 4



1. Can you modify the code to generate 4-grams, and generate suggestions for the fourth word when the user enters 3 words?
   1. Yes



import string

import re

fp = open('text\_only.csv', 'r')

clean\_text = []

for line in fp:

# remove 's

line = line.replace(' \'s', '')

# string.punctuation contains all special characters.

# this for loop removes all special characters

for i in string.punctuation:

line = line.replace(i, '')

# removing special characters results in some extra white spaces.

# replace mutliple white spaces with a single white space

# use a regular expression (re) to define the multiple space pattern

line = re.sub(' +', ' ', line)

# remove the last two characters of the line which are: white space, new line break

line = line[:-2]

# convert into all lower-case letters

line = line.lower()

# add line to cleaned data

clean\_text.append(line)

unigrams = {}

bigrams = {}

trigrams = {}

quadgrams = {}

unigram\_cnts = {}

bigram\_cnts = {}

trigram\_cnts = {}

quadgram\_cnts = {}

#word suggestion indices

one\_wrd\_sug = {}

two\_wrd\_sug = {}

three\_wrd\_sug = {}

for t in clean\_text:

# split the words in each sentence

words = [i for i in t.split(' ')]

# add single words to list of unigrams

for w in words:

unigrams[w] = 0

unigram\_cnts[w] = unigram\_cnts.get(w, 0) + 1

# add every two successive words to the list of bigrams

bi = zip(words[:-1], words[1:])

for b in bi:

x = ' '.join(b)

bigrams[x] = 0

bigram\_cnts[x] = bigram\_cnts.get(x, 0) + 1

# add entry to one word suggestion list

if bigram\_cnts[x] == 1:

if b[0] in one\_wrd\_sug:

(one\_wrd\_sug[b[0]]).append(x)

else:

one\_wrd\_sug[b[0]] = [x]

# add every three successive words to the list of trigrams

tri = zip(words[:-2], words[1:-1], words[2:])

for t in tri:

x = ' '.join(t)

trigrams[x] = 0

trigram\_cnts[x] = trigram\_cnts.get(x, 0) + 1

# add entry to two-word suggestion list

if trigram\_cnts[x] == 1:

k = t[0] + " " + t[1]

if k in two\_wrd\_sug:

(two\_wrd\_sug[k]).append(x)

else:

two\_wrd\_sug[k] = [x]

# add every four successive words to the list of quadgrams

quad = zip(words[:-3], words[1:-1], words[2:-2], words[3:])

# print(list(quad))

for q in quad:

x = ' '.join(q)

quadgrams[x] = 0

quadgram\_cnts[x] = quadgram\_cnts.get(x, 0) + 1

# add entry to four-word suggestion list

if quadgram\_cnts[x] == 1:

k = q[0] + " " + q[1] + " " + q[2]

if k in three\_wrd\_sug:

(three\_wrd\_sug[k]).append(x)

else:

three\_wrd\_sug[k] = [x]

print(three\_wrd\_sug)

#print some sample words for which a suggestion can be made

three\_wrds=['the gypsii mobile', 'london marketwatch share', 'operating profits in',

'and fixed infrastructure', 'for return on', 'understand the decision']

print("These are some of the three-word phrases that the agent can suggest a word after:")

for i in three\_wrds:

print(i)

print("==================================")

#print some sample words for which a suggestion can be made

two\_wrds=['goldman sachs', 'similar to', 'deal includes',

'private equity', 'china and', 'the longterm', 'kind of']

print("These are some of the two-word phrases that the agent can suggest a word after:")

for i in two\_wrds:

print(i)

print("==================================")

one\_wrd = ['longer', 'identity', 'know', 'protocol',

'realtime', 'motorola', 'highquality', 'defense']

print("These are some of the one-word phrases that the agent can suggest a word after:")

for i in one\_wrd:

print(i)

print("==================================")

#ask the user to enter a word, output a couple of suggestions

wrd = input('enter one word to give you a suggestion for the following word: ')

if wrd in one\_wrd\_sug:

print('Suggestions:')

for i in one\_wrd\_sug[wrd]:

print(i)

#ask the user to enter two words, output a couple of suggestions

wrd = input('enter two words to give you a suggestion for the following word: ')

if wrd in two\_wrd\_sug:

print('Suggestions:')

for i in two\_wrd\_sug[wrd]:

print(i)

#ask the user to enter two words, output a couple of suggestions

wrd = input('enter three words to give you a suggestion for the following word: ')

if wrd in three\_wrd\_sug:

print('Suggestions:')

for i in three\_wrd\_sug[wrd]:

print(i)

# charrel5

# 4/23/2023



