Author_1: Faraz Gurramkonda

Author_2: Kritika sharma

Artificial Intelligence

Final Project on NLP Techniques of English Language

```
import gensim
from gensim.models import Word2Vec, KeyedVectors
import pandas as pd

# Import train_rel_2.tsv into Python
with open('/content/Train_data.txt', 'r') as f:
    lines = f.readlines()
    columns = lines[0].split('\t')
    response = []
    for line in lines[1:]:
        temp = line.split('\t')
        print(temp)
        response.append(temp[-1])  # Select "EssayText" as a corpus

# Construct a dataframe ("data") which includes only response column
data = pd.DataFrame(list(zip(response)))
data.columns = ['response']
print(data)
```

```
['Education, as the bedrock of personal and societal development, extends beyond the confines of traditiona
['The state of the environment is a critical global concern that demands concerted efforts to address press
['The pursuit of holistic health encompasses a multifaceted approach to well-being, emphasizing physical, m
['Literature and the arts stand as timeless expressions of the human experience, capturing the intricacies
['The imperative of embracing diversity and promoting inclusion has become central to fostering equitable a
["Globalization, as a transformative force, has interconnected the world's economies, cultures, and societi
['Politics and governance form the backbone of societal structures, influencing the formulation of policies
['The economy, as the engine of societal progress, operates within a complex web of interconnected factors.
['Science and innovation serve as catalysts for progress, pushing the boundaries of human knowledge and tec
["Social media platforms, as integral components of contemporary communication, serve as digital arenas whe
["Travel, as a multifaceted experience, encompasses various dimensions that extend beyond mere movement fro
['The intricate web of relationships woven throughout our lives encompasses diverse connections, each chara
['Sports and recreation play multifaceted roles in society, serving as sources of entertainment, physical a
['Philanthropy and social responsibility are cornerstones of a compassionate and equitable society, reflect
['Artificial Intelligence, a transformative field, encompasses the development of algorithms and computatio
['Space exploration, fueled by scientific curiosity and technological advancements, involves the investigat
["Quantum computing, a cutting-edge field, leverages the principles of quantum mechanics to perform complex
["Renewable energy sources, crucial for sustainable development, harness natural resources to generate powe
['Biotechnology, a multidisciplinary field, applies biological principles to develop technologies and produ
['Cybersecurity, critical in the digital age, involves protecting computer systems, networks, and data from
['Blockchain technology, a decentralized and distributed ledger system, revolutionizes how data is stored a
["Neuroscience, the scientific study of the nervous system, delves into understanding the structure and fun
["Climate science investigates Earth's climate system, analyzing long-term patterns and variations. Climato
['Robotics, an interdisciplinary field, involves the design, construction, and operation of robots for vari
['These diverse fields represent the forefront of human knowledge and innovation, each contributing to the
                                            response
0   Education, as the bedrock of personal and soci...
1   The state of the environment is a critical glo...
2   The pursuit of holistic health encompasses a m...
3   Literature and the arts stand as timeless expr...
4   The imperative of embracing diversity and prom...
5   Globalization, as a transformative force, has ...
6   Politics and governance form the backbone of s...
7   The economy, as the engine of societal progres...
8   Science and innovation serve as catalysts for ...
9   Social media platforms, as integral components...
10  Travel, as a multifaceted experience, encompas...
11  The intricate web of relationships woven throu...
12  Sports and recreation play multifaceted roles ...
13  Philanthropy and social responsibility are cor...
14  Artificial Intelligence, a transformative fiel...
15  Space exploration, fueled by scientific curios...
```

```
16  Quantum computing, a cutting-edge field, lever...
17  Renewable energy sources, crucial for sustaina...
18  Biotechnology, a multidisciplinary field, appl...
19  Cybersecurity, critical in the digital age, in...
20  Blockchain technology, a decentralized and dis...
21  Neuroscience, the scientific study of the nerv...
22  Climate science investigates Earth's climate s...
23  Robotics, an interdisciplinary field, involves...
24  These diverse fields represent the forefront o...
```

```
data.response[0]
```

```
'Education, as the bedrock of personal and societal development, extends beyond the confines of traditiona
l classrooms. Modern pedagogy incorporates diverse teaching methodologies to cater to various learning sty
les, ensuring that students engage actively with the educational process. The curriculum spans a spectrum
of subjects, encompassing not only core academic disciplines but also practical life skills. Online learni
ng platforms have revolutionized education, offering flexibility and accessibility to learners globally. E
xtracurricular activities, including sports, arts, and community service, contribute to holistic developme
nt, fostering teamwork, creativity, and leadership skills. Educational institutions, as hubs of intellectu
al growth, empower individuals to become lifelong learners, equipped with the knowledge and skills necessa
rv for success in a dvnamic world.\n'
```

```
new_response =data.response.apply(gensim.utils.simple_preprocess)
new_response
```

```
0     [education, as, the, bedrock, of, personal, an...
1     [the, state, of, the, environment, is, critica...
2     [the, pursuit, of, holistic, health, encompass...
3     [literature, and, the, arts, stand, as, timele...
4     [the, imperative, of, embracing, diversity, an...
5     [globalization, as, transformative, force, has...
6     [politics, and, governance, form, the, backbon...
7     [the, economy, as, the, engine, of, societal, ...
8     [science, and, innovation, serve, as, catalyst...
9     [social, media, platforms, as, integral, compo...
10    [travel, as, multifaceted, experience, encompa...
11    [the, intricate, web, of, relationships, woven...
12    [sports, and, recreation, play, multifaceted, ...
13    [philanthropy, and, social, responsibility, ar...
14    [artificial, intelligence, transformative, fie...
15    [space, exploration, fueled, by, scientific, c...
16    [quantum, computing, cutting, edge, field, lev...
17    [renewable, energy, sources, crucial, for, sus...
18    [biotechnology, field, applies, biological, pr...
19    [cybersecurity, critical, in, the, digital, ag...
20    [blockchain, technology, decentralized, and, d...
21    [neuroscience, the, scientific, study, of, the...
22    [climate, science, investigates, earth, climat...
23    [robotics, an, field, involves, the, design, c...
24    [these, diverse, fields, represent, the, foref...
Name: response, dtype: object
```

```
model=gensim.models.Word2Vec(window=5, min_count=2, workers=4, sg=0)
model.build_vocab(new_response, progress_per=1000)
model.train(new_response, total_examples=model.corpus_count, epochs=model.epochs)
model.save("./respon.model")
```

```
model.wv["critical"]
```

```
array([ 4.32500103e-03, -8.31272919e-03,  7.04971375e-03, -5.13712876e-04,
        7.76144397e-03,  3.56046297e-03,  2.08832952e-03, -3.65709211e-03,
        3.89940338e-03,  8.38649366e-03, -2.52647651e-03,  3.56219313e-03,
       -5.17730461e-03,  6.18395908e-03,  7.01050041e-03,  2.40609937e-04,
        7.77229434e-03,  5.38561027e-03, -1.08325435e-02, -9.07585211e-03,
        9.79025476e-03, -8.04353767e-05,  1.16034504e-03, -1.00492425e-02,
       -8.38325638e-03,  1.03995693e-03,  9.50688124e-03, -3.41803255e-03,
        2.02667114e-04,  1.42607908e-03, -4.76968754e-03,  7.59376306e-03,
       -8.36957153e-03, -2.70615914e-03, -5.81921730e-03,  2.21615611e-03,
        3.55401263e-03,  8.34462233e-03, -7.01466249e-03,  8.05554300e-05,
        4.04854678e-03,  4.93612001e-03,  5.68741700e-03,  3.98773281e-03,
```

```
          3.15052387e-03, -6.04526652e-03, -6.49901456e-04,  7.57036917e-03,
          3.84374778e-03, -2.25750706e-03, -6.49794238e-03,  6.28059218e-03,
         -1.03915262e-03, -1.29142730e-03,  7.18723517e-03, -5.02170809e-03,
         -3.82145937e-03,  2.92392517e-03,  9.38211509e-04, -4.86196019e-03,
         -7.69773684e-03,  4.19869646e-03,  2.99940875e-04, -1.04911048e-02,
         -4.95469803e-03,  3.29639413e-03,  9.58075467e-03,  5.76495659e-03,
         -1.07559534e-02, -4.03550267e-03, -5.91792678e-03, -8.20143428e-03,
          3.38554411e-04,  2.67039263e-03, -7.20999576e-03,  9.91194136e-03,
          8.31554551e-03,  9.07232519e-03, -9.30631999e-03, -5.11380797e-03,
          3.02333664e-03,  9.08982847e-03,  6.81517925e-03, -4.82444558e-03,
          7.56018655e-03,  3.88040295e-04, -5.48547599e-03, -5.67554822e-03,
          7.69285252e-03,  5.81688154e-03,  9.67339054e-03,  9.52744018e-03,
          2.78044818e-03, -3.51795601e-03,  1.41977705e-03,  1.04954075e-02,
          5.83395315e-03,  1.97344855e-03, -5.85879723e-04,  4.06180881e-03],
        dtype=float32)
```

```
from scipy.spatial.distance import cosine

import math
import numpy as np

with open("/content/drive/MyDrive/src6/vectors/words.txt") as f:
    words = dict()
    for line in f:
        row = line.split()
        word = row[0]
        print(word)
        print("printing row:",row[1:])
        vector = np.array([float(x) for x in row[1:]])
        words[word] = vector


def distance(w1, w2):
    return cosine(w1, w2)


def closest_words(embedding):
    distances = {
        w: distance(embedding, words[w])
        for w in words
    }
    return sorted(distances, key=lambda w: distances[w])[:10]


def closest_word(embedding):
    return closest_words(embedding)[0]
```

```
printing row: ['-0.246558', '0.129303', '0.619182', '-0.062265', '0.135429', '0.214962', '0.187126', '-0.493
k
printing row: ['0.315315', '-0.136819', '0.014305', '-0.231770', '0.621387', '-0.513426', '0.408154', '-0.37
oseph
printing row: ['-0.230092', '0.028531', '0.190542', '0.006344', '0.019684', '-0.102341', '0.388414', '-0.686
opy
printing row: ['0.095604', '-0.073530', '0.002328', '-0.018020', '-0.036423', '-0.255175', '-0.316821', '-0.
lress
printing row: ['0.314422', '0.211590', '0.767197', '-0.577775', '-0.168600', '0.241912', '-0.354339', '-0.42
teve
printing row: ['-0.373911', '0.352274', '0.305589', '0.004621', '0.123921', '-0.090294', '0.349335', '-0.473
istoric
printing row: ['0.151468', '-0.590410', '0.064291', '-0.473943', '0.638503', '0.321229', '0.204183', '-0.266
uest
printing row: ['-0.441456', '-0.391519', '-0.320185', '-0.261476', '-0.072683', '0.024620', '-0.089089', '0.
ighest
printing row: ['-0.488281', '-0.684173', '0.439936', '-0.213453', '-0.010348', '0.019881', '0.064387', '0.26
ue
printing row: ['-0.722499', '0.528998', '0.580823', '-0.352876', '-0.331112', '-0.985724', '0.393869', '-0.1
queen
printing row: ['-0.276391', '-0.321548', '0.209976', '-0.210990', '0.204694', '0.528204', '0.101802', '-0.32
ri
printing row: ['-0.756497', '0.402558', '0.644678', '-0.351589', '-0.429424', '-0.970667', '0.410093', '-0.1
hu
printing row: ['-0.672181', '0.475444', '0.608542', '-0.273672', '-0.350316', '-1.036378', '0.508223', '-0.1
.
printing row: ['-0.445492', '0.470921', '-0.126177', '-0.658319', '0.571067', '0.165267', '0.334657', '0.007
elpful
printing row: ['0.268769', '0.404737', '0.432300', '-0.388760', '0.012313', '0.257592', '0.282085', '-0.6338
lasses
printing row: ['0.373640', '-0.135607', '0.094374', '-0.426261', '0.077984', '-0.272527', '-0.101329', '-0.1

printing row: ['-0.036231', '0.096113', '0.158572', '-0.635889', '0.789277', '-0.047941', '0.564974', '0.091
cademic
printing row: ['0.219915', '-0.536523', '0.173129', '-0.448697', '0.005654', '0.542077', '-0.113983', '-0.18
4
```

```
print(distance(words["book"],words["library"]))
```

```
0.49818406861648856
```

```
print(closest_words(words["book"])[:10])
```

```
['book', 'books', 'essay', 'memoir', 'essays', 'novella', 'anthology', 'blurb', 'autobiography', 'audiobook
```

```
print(closest_words(words["king"]- words["man"]+ words["woman"])[:1])
```

```
['queen']
```

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

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
True
```

```
import nltk



def main():

    # Read data from files


    directory_name = input("enter the directory_name:")
    positives, negatives = load_data(directory_name)

    # Create a set of all words
    words = set()
    for document in positives:
        words.update(document)
    for document in negatives:
        words.update(document)

    # Extract features from text
    training = []
    training.extend(generate_features(positives, words, "Positive"))
    training.extend(generate_features(negatives, words, "Negative"))

    # Classify a new sample
    classifier = nltk.NaiveBayesClassifier.train(training)
    s = input("s: ")
    result = (classify(classifier, s, words))
    for key in result.samples():
        print(f"{key}: {result.prob(key):.4f}")


def extract_words(document):
    return set(
        word.lower() for word in nltk.word_tokenize(document)
        if any(c.isalpha() for c in word)
    )


def load_data(directory):
    result = []
    directory = "/content/drive/MyDrive/src6/sentiment/"+ directory
    for filename in ["positives.txt","negatives.txt"]:
        with open(directory+"/"+filename) as f:
            result.append([
                extract_words(line)
                for line in f.read().splitlines()
            ])
    return result


def generate_features(documents, words, label):
    features = []
    for document in documents:
        features.append(({
            word: (word in document)
            for word in words
        }, label))
    return features


def classify(classifier, document, words):
    document_words = extract_words(document)
    features = {
        word: (word in document_words)
        for word in words
    }
```

```
    return classifier.prob_classify(features)


if __name__ == "__main__":
    main()
```

```
enter the directory_name:corpus
s: you are so bad
Positive: 0.2220
Negative: 0.7780
```

```
pip install markovify
```

```
Collecting markovify
  Downloading markovify-0.9.4.tar.gz (27 kB)
  Preparing metadata (setup.py) ... done
Collecting unidecode (from markovify)
  Downloading Unidecode-1.3.7-py3-none-any.whl (235 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 235.5/235.5 kB 2.6 MB/s eta 0:00:00
Building wheels for collected packages: markovify
  Building wheel for markovify (setup.py) ... done
  Created wheel for markovify: filename=markovify-0.9.4-py3-none-any.whl size=18606 sha256=74dab41099f7cdf5
  Stored in directory: /root/.cache/pip/wheels/ca/8c/c5/41413e24c484f883a100c63ca7b3b0362b7c6f6eb6d7c9cc7f
Successfully built markovify
Installing collected packages: unidecode, markovify
Successfully installed markovify-0.9.4 unidecode-1.3.7
```

```
from collections import Counter

import math
import nltk



def main():
    """Calculate top term frequencies for a corpus of documents."""


    print("Loading data...")

    n = int(input("number of ngrams:"))
    data_choosen = input("enter the doirectory file to be loaded:")
    corpus = load_data(data_choosen)

    # Compute n-grams
    ngrams = Counter(nltk.ngrams(corpus, n))

    # Print most common n-grams
    for ngram, freq in ngrams.most_common(10):
        print(f"{freq}: {ngram}")


def load_data(directory):
    contents = []

    # Read all files and extract words
    directory  = "/content/drive/MyDrive/src6/ngrams/" + directory
    for filename in ["bachelor.txt","blaze.txt","blaze.txt","bohemia.txt","boscombe.txt","carbuncle.txt","clerk
        with open(directory + "/" +filename) as f:
            contents.extend([
                word.lower() for word in
                nltk.word_tokenize(f.read())
                if any(c.isalpha() for c in word)
            ])
    return contents

# /content/drive/MyDrive/src6/ngrams/holmes
if __name__ == "__main__":
    main()
```

```
Loading data...
number of ngrams:3
enter the doirectory file to be loaded:holmes
79: ('it', 'was', 'a')
73: ('one', 'of', 'the')
67: ('i', 'think', 'that')
62: ('out', 'of', 'the')
62: ('that', 'he', 'had')
60: ('there', 'was', 'a')
56: ('that', 'he', 'was')
54: ('that', 'it', 'was')
52: ('it', 'is', 'a')
51: ('i', 'can', 'not')
```

```
import markovify

# /content/drive/MyDrive/src6/markov/shakespeare.txt
directory = "/content/drive/MyDrive/src6/markov/"
filename = input("please enter the file name:")
path = directory + filename
with open(path) as f:
    text = f.read()

# Train model
text_model = markovify.Text(text)

# Generate sentences
print()
for i in range(10):
    print(text_model.make_sentence())
    print()
```

```
 please enter the file name:shakespeare.txt

 I hope to speed alone.

 Know you of this, Helena; go to, very well.

 My nobler part to heaven, Whiles, like a brib'd buck, each a haunch; I will walk till thou be pardoned.

 We need no more of your daughter for a traitor.

 Frenchmen, I'll be with you, if he were a king transformed to a chaos, or an aglet-baby, or an aglet-baby,

 It is no other answer make but thanks, And thanks, and make her scorn you still.

 O, it is done; the bell have told my Lord Lackbeard there, he shall come to me a light.

 Sweet men, come to keep his vow and this his humble suit.

 No sooner had they from thy burgonet I'll rend thy faith be not four by the DUKE OF VENICE.

 Speak well of him I will not miss my sense: I mean is promis'd by this crime he owes the prince, Even such
```

```
import nltk

grammar = nltk.CFG.fromstring("""
    S -> NP VP

    NP -> D N | N
    VP -> V | V NP

    D -> "the" | "a"
    N -> "she" | "city" | "car"
    V -> "saw" | "walked"
""")

parser = nltk.ChartParser(grammar)

sentence = input("Sentence: ").split()
try:
    for tree in parser.parse(sentence):
        tree.pretty_print()
except ValueError:
    print("No parse tree possible.")
```

```
 Sentence: she walked
      S
  ____|____
 NP       VP
```

```
        |       |
        N       V
        |       |
       she    walked
```