

```

import streamlit as st
import numpy as np
import numpy.linalg as la
import pickle

"""
This is a starter code for Assignment 0 of the course, "Hands-on Master
Class on LLMs and ChatGPT | Autumn 2023"
taught by Dr. Karthik Mohan.

Computes closest category of a given word or sentence input into a search
bar.
The search is implemented through streamlit and can be hosted as a "web
app" on the cloud through streamlit as well
Example webpage and search demo: searchdemo.streamlit.app
"""

# Compute Cosine Similarity
def cosine_similarity(x,y):

    x_arr = np.array(x)
    y_arr = np.array(y)

    #####
    ### WRITE YOUR CODE HERE ###
    #####
    # Cosine similarity = dot product (how similar the directions of the
two vectors are) of the vectors divided by
    # the product of their euclidean norms, accounting for divide by 0
    return np.dot(x,y)/max(la.norm(x)*la.norm(y),1e-3)

# Function to Load Glove Embeddings
def load_glove_embeddings(glove_path="glove.6B.50d.txt"):
    """
    First step: Download the 50d Glove embeddings from here -
https://www.kaggle.com/datasets/adityajn105/glove6b50d
    Second step: Format the glove embeddings into a dictionary that goes
from a word to the 50d embedding.
    Third step: Store the 50d Glove embeddings in a pickle file of a
dictionary.
    Now load that pickle file back in this function
    """

    # Extract data from the txt file and inject into a dictionary
    embedding_dict = {}
    with open(glove_path, "r", encoding='utf-8') as f:
        for line in f:
            word = line.split(" ")[0]
            embedding = np.array([float(val) for val in line.split("
") [1:]])
            embedding_dict[word] = embedding

```

```

    # print(len(embedding_dict["cat"]))
    # print(embedding_dict["cat"])

    return embedding_dict

# Get Averaged Glove Embedding of a sentence
def averaged_glove_embeddings(sentence, embeddings_dict):
    """
    Simple sentence embedding: Embedding of a sentence is the average of
the word embeddings
    """
    words = sentence.split(" ")
    glove_embedding = np.zeros(50)
    count_words = 0

    #####
    ### WRITE YOUR CODE HERE ###
    #####

    for word in words:
        if word.lower() in embeddings_dict:
            glove_embedding = np.add(glove_embedding,
embeddings_dict[word.lower()])
            count_words += 1

    if count_words > 0:
        glove_embedding = glove_embedding / count_words

    print(glove_embedding)

    return glove_embedding


# Load glove embeddings
glove_embeddings = load_glove_embeddings()

# Gold standard words to search from
gold_words = ["flower", "mountain", "tree", "car", "building"]

# Text Search
st.title("Search Based Retrieval Demo")
st.subheader("Pass in an input word or even a sentence (e.g. jasmine or
mount adams)")
text_search = st.text_input("", value="")

# Find closest word to an input word
if text_search:
    input_embedding = averaged_glove_embeddings(text_search,
glove_embeddings)
    cosine_sim = {}
    for index in range(len(gold_words)):

```

```

        cosine_sim[index] = cosine_similarity(input_embedding,
glove_embeddings[gold_words[index]])
        print(cosine_sim)

#####
### WRITE YOUR CODE HERE ###
#####

# Sort the cosine similarities
sorted_cosine_sim = sorted(cosine_sim.items(), key=lambda x: x[1],
reverse=True)
print (sorted_cosine_sim)

st.write("(My search uses glove embeddings)")
st.write("Closest word I have between flower, mountain, tree, car and
building for your input is: ")
st.subheader(gold_words[sorted_cosine_sim[0][0]] )
st.write("")

```

This is a starter code for Assignment 0 of the course, "Hands-on Master Class on LLMs and ChatGPT | Autumn 2023" taught by Dr. Karthik Mohan.

Computes closest category of a given word or sentence input into a search bar. The search is implemented through streamlit and can be hosted as a "web app" on the cloud through streamlit as well Example webpage and search demo: searchdemo.streamlit.app

Search Based Retrieval Demo

Pass in an input word or even a sentence (e.g. jasmine or mount adams)

the police office

(My search uses glove embeddings)

Closest word I have between flower, mountain, tree, car and building for your input is:

building

