Name : Dhanush A

Roll No: 717821i111

Dept : AD

Course Code:21ID31

Course Name:Speech and Language Processing

Assignment :1

Word Embeddings and similarity between senetences

CODE:

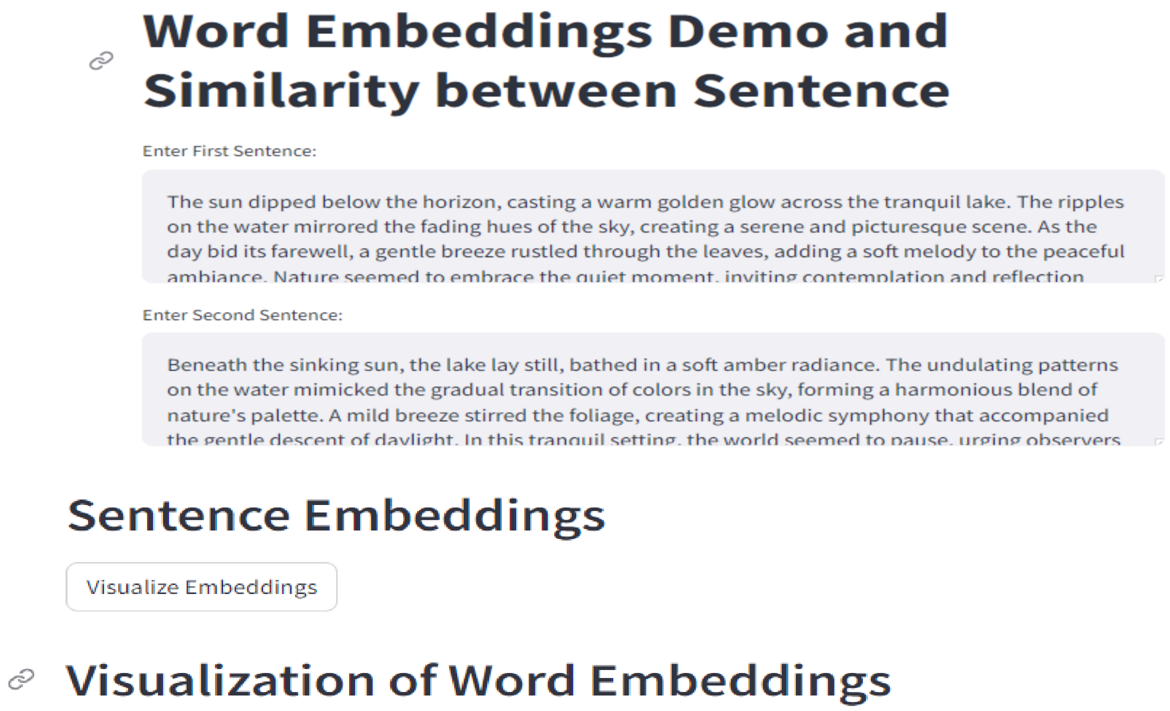
```
from transformers import DistilBertTokenizer, DistilBertForQuestionAnswering,DistilBertModel
import torch
import numpy as np
from sklearn.manifold import TSNE
import matplotlib.pyplot as plt
from sklearn.metrics.pairwise import cosine_similarity
# Load pre-trained model and tokenizer
tokenizer1 = DistilBertTokenizer.from_pretrained('distilbert-base-cased')
model1 = DistilBertModel.from_pretrained('distilbert-base-cased')
def encode_sentence(sentence):
    # Tokenize input
    inputs = tokenizer1(sentence, return_tensors='pt')
    # Get model output
    outputs = model1(**inputs)
    # Extract the embeddings from 'last_hidden_state'
    embeddings = outputs.last_hidden_state
    return embeddings
def encode_sentence_sim(sentence, max_length=512):
    # Tokenize input
    inputs = tokenizer1(sentence, return_tensors='pt', truncation=True, max_length=max_length)
    # Get model output
    outputs = model1(**inputs)
    # Extract the embeddings from 'last_hidden_state'
    embeddings = outputs.last_hidden_state
    # Reduce embeddings to a fixed size (e.g., by mean pooling)
    embeddings = torch.mean(embeddings, dim=1)
    return embeddings

def visualize_word_embeddings(embeddings, tokens):
    # Use t-SNE to reduce dimensionality for visualization
    tsne_model = TSNE(n_components=2, random_state=42, perplexity=min(5, len(tokens)-1))
    word_vectors_2D = tsne_model.fit_transform(embeddings)

    # Plotting the words in 2D
    plt.figure(figsize=(8, 6))
    for i, token in enumerate(tokens):
        plt.scatter(word_vectors_2D[i, 0], word_vectors_2D[i, 1], marker='o', color='b')
        plt.text(word_vectors_2D[i, 0] + 0.02, word_vectors_2D[i, 1] + 0.02, token, fontsize=9)

    plt.show()
def calculate_cosine_similarity(embeddings1, embeddings2):
    # Detach tensors before converting to numpy arrays
    embeddings1 = embeddings1.detach().numpy()
    embeddings2 = embeddings2.detach().numpy()

    # Reshape embeddings if needed
    embeddings1 = embeddings1.reshape(1, -1)
    embeddings2 = embeddings2.reshape(1, -1)

    # Calculate cosine similarity
    similarity_matrix = cosine_similarity(embeddings1, embeddings2)
    # Extract the similarity score
    similarity_score = similarity_matrix[0, 0]

    return similarity_score
# Example usage
context =input()
question =input()

# Encode the sentence to get the embeddings
embeddings = encode_sentence(question)

# Extract word vectors directly from the model
word_vectors = embeddings[0].detach().numpy()
# Example usage
context_embeddings = encode_sentence_sim(context)
question_embeddings = encode_sentence_sim(question)
similarity_score = calculate_cosine_similarity(context_embeddings, question_embeddings)
print(f"Cosine Similarity between context and question: {similarity_score}")


# Tokenize input for visualization
tokens = tokenizer.tokenize(tokenizer.decode(tokenizer.encode(question)))

```

```
# Visualize word embeddings
print("visualization of word embedding")
visualize_word_embeddings(word_vectors, tokens)
```

Requirements.txt

```
streamlit
transformers
torch
numpy
scikit-learn
matplotlib
```

OUTPUT:



Word Embeddings Demo and Similarity between Sentence

Enter First Sentence:

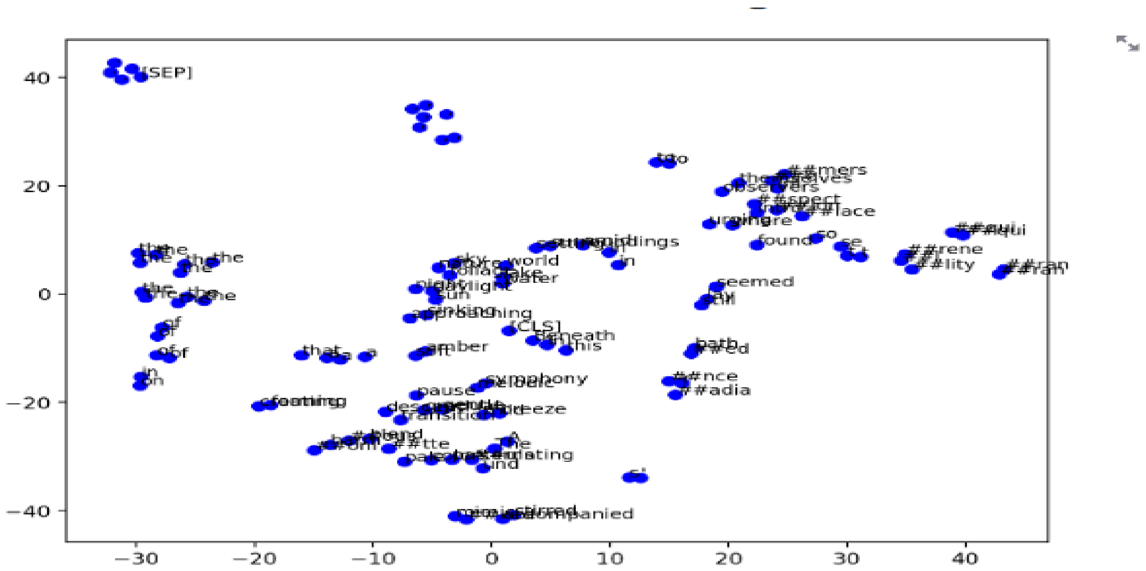
The sun dipped below the horizon, casting a warm golden glow across the tranquil lake. The ripples on the water mirrored the fading hues of the sky, creating a serene and picturesque scene. As the day bid its farewell, a gentle breeze rustled through the leaves, adding a soft melody to the peaceful ambiance. Nature seemed to embrace the quiet moment, inviting contemplation and reflection

Enter Second Sentence:

Beneath the sinking sun, the lake lay still, bathed in a soft amber radiance. The undulating patterns on the water mimicked the gradual transition of colors in the sky, forming a harmonious blend of nature's palette. A mild breeze stirred the foliage, creating a melodic symphony that accompanied the gentle descent of daylight. In this tranquil setting, the world seemed to pause, urging observers

Sentence Embeddings

Visualization of Word Embeddings



Similarity Between Sentence

Cosine Similarity between context and question: 0.9806268811225891

Github link : https://github.com/DhanushAshok04/wordembedding
Streamlit link: https://wordembedding-cosinesimilarity.streamlit.app/
Colab link : https://colab.research.google.com/drive/1YYSlxNRnNqJfKajUiOaPxTyoAoz4RoEn#scrollTo=Iz_J2-wCQ-af