Name:Akhil.B

ROLL_NO:2403a52344

lab:10

step 1: Import Libraries

```
# Load pre-trained embeddings
import gensim.downloader as api
# Handle numerical arrays
import numpy as np
import pandas as pd
# Dimensionality reduction
from sklearn.manifold import TSNE
# Visualization
import matplotlib.pyplot as plt
```

step 2:Load Pre-trained Embedding Model

```
model = api.load("glove-wiki-gigaword-100")
print("Vocabulary size:", len(model.key_to_index))
# Example vector
print("Vector for 'king':")
print(model["king"])
```

```
Vocabulary size: 400000
Vector for 'king':
[-0.32307  -0.87616   0.21977   0.25268   0.22976   0.7388   -0.37954
 -0.35307  -0.84369  -1.1113   -0.30266   0.33178  -0.25113   0.30448
 -0.077491 -0.89815   0.092496 -1.1407   -0.58324   0.66869  -0.23122
 -0.95855   0.28262  -0.078848  0.75315   0.26584   0.3422   -0.33949
  0.95608   0.065641  0.45747   0.39835   0.57965   0.39267  -0.21851
  0.58795  -0.55999   0.63368  -0.043983 -0.68731  -0.37841   0.38026
  0.61641  -0.88269  -0.12346  -0.37928  -0.38318   0.23868   0.6685
 -0.43321  -0.11065   0.081723  1.1569    0.78958  -0.21223  -2.3211
 -0.67806   0.44561   0.65707   0.1045    0.46217   0.19912   0.25802
  0.057194  0.53443  -0.43133  -0.34311   0.59789  -0.58417   0.068995
  0.23944  -0.85181   0.30379  -0.34177  -0.25746  -0.031101 -0.16285
  0.45169  -0.91627   0.64521   0.73281  -0.22752   0.30226   0.044801
 -0.83741   0.55006  -0.52506  -1.7357    0.4751   -0.70487   0.056939
 -0.7132    0.089623  0.41394  -1.3363   -0.61915  -0.33089  -0.52881
  0.16483  -0.98878 ]
```

step 3:Select Word List (30–50 Words)

```
animals = ["cat", "dog", "lion", "tiger", "wolf", "horse"]
countries = ["india", "china", "france", "germany", "brazil", "japan"]
fruits = ["apple", "banana", "orange", "mango", "grape", "pineapple"]
technology = ["computer", "laptop", "internet", "software", "hardware", "keyboard"]
vehicles = ["car", "bus", "train", "airplane", "bicycle", "truck"]
professions = ["doctor", "engineer", "teacher", "lawyer", "nurse", "scientist"]

word_list = animals + countries + fruits + technology + vehicles + professions
```

step 4:Apply t-SNE

```
vectors_array = np.array(vectors)
tsne = TSNE(n_components=2, random_state=42, perplexity=5)
reduced_vectors = tsne.fit_transform(vectors_array)
```

step 5:Plot Visualization

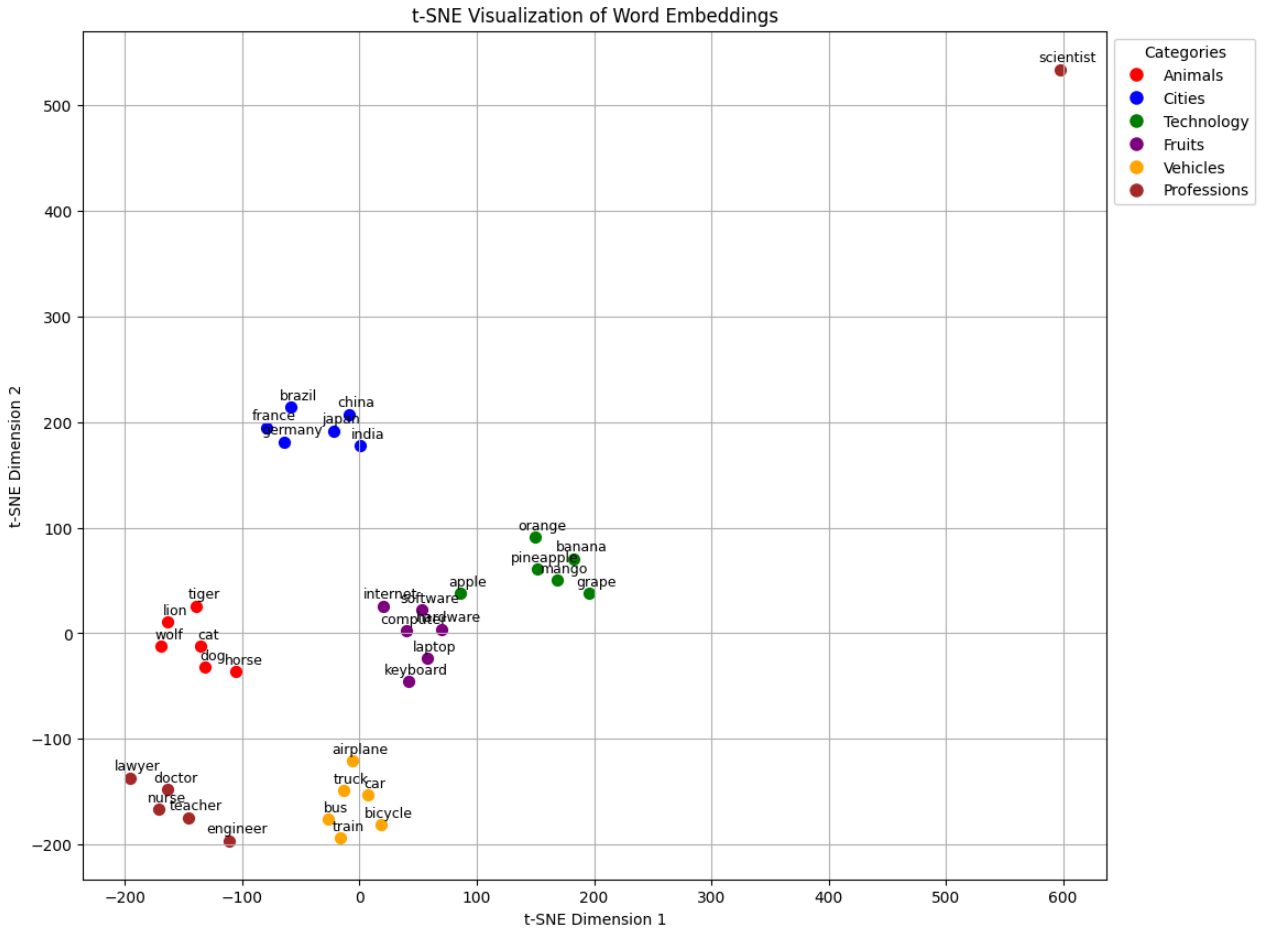
```
word_categories = [
    "Animals", "Animals", "Animals", "Animals", "Animals", "Animals",
    "Cities", "Cities", "Cities", "Cities", "Cities", "Cities",
    "Technology", "Technology", "Technology", "Technology", "Technology", "Technology",
    "Fruits", "Fruits", "Fruits", "Fruits", "Fruits", "Fruits",
    "Vehicles", "Vehicles", "Vehicles", "Vehicles", "Vehicles", "Vehicles",
    "Professions", "Professions", "Professions", "Professions", "Professions", "Professions"
]

category_colors = {
```

```
    "Animals": "red",
    "Cities": "blue",
    "Technology": "green",
    "Fruits": "purple",
    "Vehicles": "orange",
    "Professions": "brown"
}

plt.figure(figsize=(12, 10))
# Plotting points and annotations
for i, word in enumerate(word_list):
    category = word_categories[i]
    color = category_colors[category]
    plt.scatter(reduced_vectors[i, 0], reduced_vectors[i, 1], c=color, s=50) # Plot point
    plt.annotate(word, (reduced_vectors[i, 0], reduced_vectors[i, 1]), textcoords="offset points", xytext=(5,5), ha='center'

# Create custom legend handles to avoid duplicate labels
legend_handles = []
for category, color in category_colors.items():
    legend_handles.append(plt.Line2D([0], [0], marker='o', color='w', label=category,
                                     markerfacecolor=color, markersize=10))
plt.legend(handles=legend_handles, title="Categories", loc='upper left', bbox_to_anchor=(1, 1))
plt.title('t-SNE Visualization of Word Embeddings')
plt.xlabel('t-SNE Dimension 1')
plt.ylabel('t-SNE Dimension 2')
plt.grid(True)
plt.show()
```



step 6: Interpretation

The t-SNE visualization shows clear clustering of semantically related words. Animal names such as "cat," "dog," and "lion" appear close together, forming a distinct group. Similarly, fruits like "apple," "banana," and "mango" are positioned near each other. Technology-related words cluster in another region of the plot, indicating semantic similarity. Countries also appear relatively close to one another. Some professions appear near technology terms, possibly due to contextual overlap in training data. A few words may

appear slightly distant from their expected cluster due to how they are used in different contexts. Overall, the visualization confirms that word embeddings capture semantic meaning. t-SNE effectively preserves local relationships between words. The clusters demonstrate how vector representations encode similarity.

step 7:Lab Report Structure

A lab report for this analysis would typically include the following sections:

1. **Objective:** Clearly state the purpose of the experiment, which is to visualize word embeddings and understand semantic relationships using t-SNE.
2. **Methodology:** Detail the steps taken:
    - **Data Source:** Specify the pre-trained word embedding model used (e.g., GloVe-wiki-gigaword-100).
    - **Word Selection:** Describe the process of selecting a diverse list of words belonging to various semantic categories.
    - **Dimensionality Reduction:** Explain the application of t-SNE, including key parameters like `n_components` (2) and `perplexity` (5).
    - **Visualization:** Mention the use of `matplotlib` for creating the scatter plot and annotating words.
3. **Results:** Present the t-SNE visualization and discuss observations:
    - **Clustering:** Describe how words belonging to similar semantic categories (e.g., animals, cities, technology) form distinct clusters.
    - **Semantic Proximity:** Explain that the proximity of words in the 2D space reflects their semantic similarity in the original high-dimensional embedding space.
    - **Outliers/Nuances:** Briefly discuss any words that might appear slightly out of place from their expected clusters, possibly due to contextual nuances in the training data.
4. **Conclusion:** Summarize the key findings and their implications:
    - Reiterate that word embeddings effectively capture semantic relationships.
    - Conclude that t-SNE is a valuable tool for visualizing and interpreting these high-dimensional embeddings.

This structure provides a comprehensive overview of the experiment, from its goal to its interpretation.