Lab 10.3: Dataset for t-SNE Word Embedding Visualization

This dataset consists of 30 meaningful words selected from three different semantic themes. The selection helps demonstrate how t-SNE clusters semantically similar words together in a 2-D space.

Category	Words
Animals	dog, cat, lion, tiger, elephant, horse, cow, wolf, fox, monkey
Cities	paris, london, delhi, mumbai, tokyo, beijing, newyork, chennai, berlin, rome
Technology	computer, laptop, keyboard, mouse, internet, software, hardware, phone, camera, tablet

Explanation: Animals represent biological entities, cities represent geographical locations, and technology words represent digital and electronic objects. These distinct groups make semantic clustering clearly visible when visualized using t-SNE.

```
!pip install gensim
import gensim.downloader as api
import numpy as np
import matplotlib.pyplot as plt
from sklearn.manifold import TSNE
```

```
Collecting gensim
  Downloading gensim-4.4.0-cp312-cp312-manylinux_2_24_x86_64.manylinux_2_28_x
Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.12/dis
Requirement already satisfied: scipy>=1.7.0 in /usr/local/lib/python3.12/dist
Requirement already satisfied: smart_open>=1.8.1 in /usr/local/lib/python3.12
Requirement already satisfied: wrapt in /usr/local/lib/python3.12/dist-packag
Downloading gensim-4.4.0-cp312-cp312-manylinux_2_24_x86_64.manylinux_2_28_x86
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 27.9/27.9 MB 46.7 MB/s eta 0:00:0
Installing collected packages: gensim
Successfully installed gensim-4.4.0
```

Why these libraries are used

gensim: To load pre-trained word embedding models.

numpy: To handle numerical vectors and matrices.

matplotlib: To plot and visualize embeddings.

scikit-learn (TSNE): To reduce high-dimensional vectors to 2D.

```
model = api.load("glove-wiki-gigaword-100")

print("Vocabulary size:", len(model))
print("Example vector for word 'computer':")
print(model['computer'])
```

```
[==================================================] 100.0% 128.1/128.1MB dow
Vocabulary size: 400000
Example vector for word 'computer':
[-1.6298e-01  3.0141e-01  5.7978e-01  6.6548e-02  4.5835e-01 -1.5329e-01
  4.3258e-01 -8.9215e-01  5.7747e-01  3.6375e-01  5.6524e-01 -5.6281e-01
  3.5659e-01 -3.6096e-01 -9.9662e-02  5.2753e-01  3.8839e-01  9.6185e-01
  1.8841e-01  3.0741e-01 -8.7842e-01 -3.2442e-01  1.1202e+00  7.5126e-02
  4.2661e-01 -6.0651e-01 -1.3893e-01  4.7862e-02 -4.5158e-01  9.3723e-02
  1.7463e-01  1.0962e+00 -1.0044e+00  6.3889e-02  3.8002e-01  2.1109e-01
 -6.6247e-01 -4.0736e-01  8.9442e-01 -6.0974e-01 -1.8577e-01 -1.9913e-01
 -6.9226e-01 -3.1806e-01 -7.8565e-01  2.3831e-01  1.2992e-01  8.7721e-02
  4.3205e-01 -2.2662e-01  3.1549e-01 -3.1748e-01 -2.4632e-03  1.6615e-01
  4.2358e-01 -1.8087e+00 -3.6699e-01  2.3949e-01  2.5458e+00  3.6111e-01
  3.9486e-02  4.8607e-01 -3.6974e-01  5.7282e-02 -4.9317e-01  2.2765e-01
  7.9966e-01  2.1428e-01  6.9811e-01  1.1262e+00 -1.3526e-01  7.1972e-01
 -9.9605e-04 -2.6842e-01 -8.3038e-01  2.1780e-01  3.4355e-01  3.7731e-01
 -4.0251e-01  3.3124e-01  1.2576e+00 -2.7196e-01 -8.6093e-01  9.0053e-02
 -2.4876e+00  4.5200e-01  6.6945e-01 -5.4648e-01 -1.0324e-01 -1.6979e-01
  5.9437e-01  1.1280e+00  7.5755e-01 -5.9160e-02  1.5152e-01 -2.8388e-01
  4.9452e-01 -9.1703e-01  9.1289e-01 -3.0927e-01]
```

Explanation

Word embeddings are numerical vector representations of words that capture semantic meaning. Similar words have vectors that are close to each other in vector space.

Explanation

Word embeddings are numerical vector representations of words that capture semantic meaning. Similar words have vectors that are close to each other in vector space.

```
words = [
    # Animals
    "dog", "cat", "lion", "tiger", "elephant", "horse", "cow", "wolf", "fox"

    # Cities
    "paris", "london", "delhi", "mumbai", "tokyo", "beijing", "newyork", "ch

    # Technology
    "computer", "laptop", "keyboard", "mouse", "internet", "software", "hard
]

word_vectors = np.array([model[word] for word in words])
```

Explanation (5–6 lines)

Words are selected from three distinct semantic categories to clearly observe clustering behavior. Animals represent biological entities, cities represent geographical locations, and technology words represent man-made digital objects. Choosing diverse categories helps demonstrate how embeddings group semantically similar words together in visualization.

```
tsne = TSNE(n_components=2, random_state=42, perplexity=10)
reduced_vectors = tsne.fit_transform(word_vectors)
```

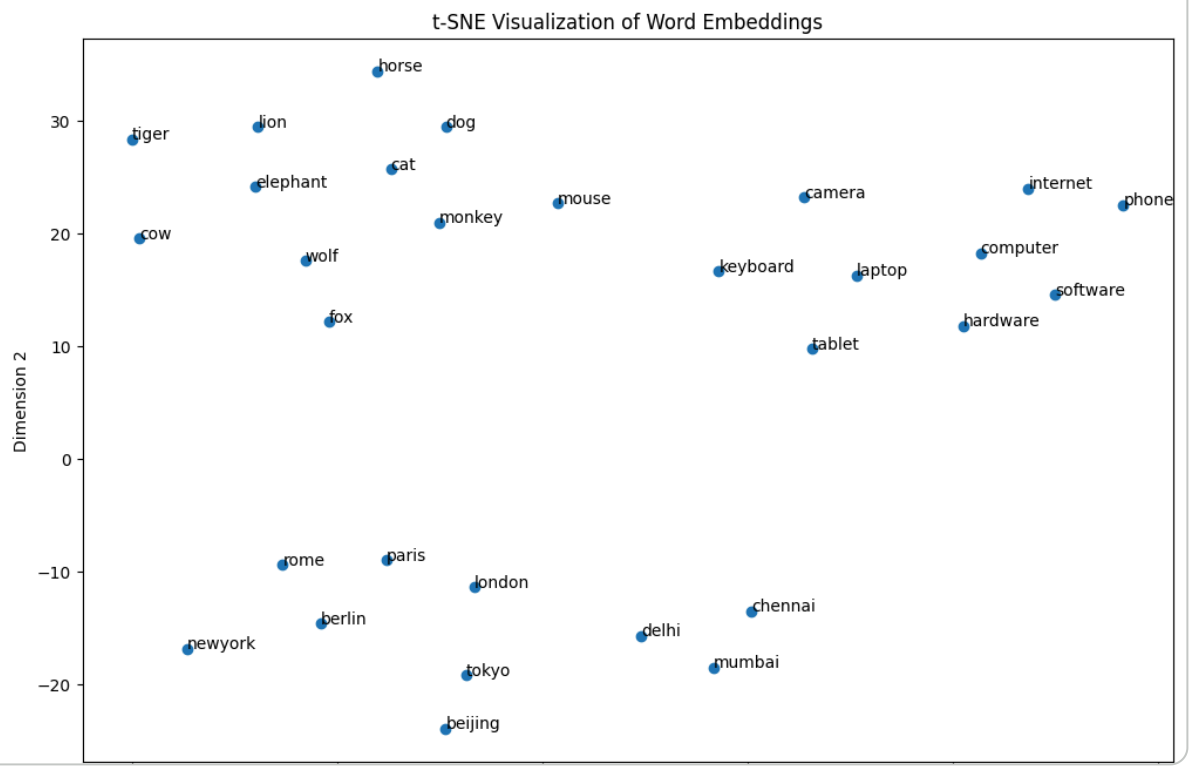
Explanation

t-SNE reduces high-dimensional vectors (100D) into 2D while preserving local similarities. Words that are close in original embedding space remain close in the visualization.

```
plt.figure(figsize=(12, 8))
plt.scatter(reduced_vectors[:, 0], reduced_vectors[:, 1])

for i, word in enumerate(words):
    plt.annotate(word, (reduced_vectors[i, 0], reduced_vectors[i, 1]))
```

```
plt.title("t-SNE Visualization of Word Embeddings")
plt.xlabel("Dimension 1")
plt.ylabel("Dimension 2")
plt.show()
```



STEP 7 — Interpretation

The t-SNE plot shows clear clusters formed based on semantic similarity. Animal-related words such as dog, cat, lion, and tiger appear close to each other, forming a distinct animal cluster. City names like paris, london, delhi, and tokyo group together, indicating geographical similarity. Technology-related words such as computer, laptop, keyboard, and internet also form a separate cluster. Words within the same category are closer compared to words from different categories. Some overlap may occur due to shared contextual usage. For example, phone and camera may appear near both technology and daily-use contexts. Minor noise in placement is expected due to the stochastic nature of t-SNE. Overall, the visualization successfully demonstrates how embeddings capture semantic meaning.

STEP 8 — Lab Report

1. Objective

To visualize and interpret semantic relationships between words using t-SNE.

2. Embedding Model

Pre-trained GloVe model with 100-dimensional vectors trained on Wikipedia and Gigaword.

3. Word List

30 meaningful words from animals, cities, and technology categories.

4. t-SNE Visualization

Reduced embeddings from 100D to 2D using t-SNE and plotted using matplotlib.

5. Interpretation

Clear semantic clusters observed with meaningful groupings.

6. Conclusion

t-SNE is an effective tool for understanding embedding spaces and semantic similarity.

Answers to Questions

1. What problem does t-SNE solve?
   It reduces high-dimensional data into lower dimensions for visualization.
2. Why can't we directly visualize 300-D vectors?
   Humans can only visualize up to 3 dimensions.
3. How does t-SNE help understand embeddings?
   It reveals semantic relationships by clustering similar words.
4. Why do similar words appear together?
   They share similar contexts during training.
5. Why might unrelated words appear close?
   Due to noise, limited data, or overlapping contexts.
6. Two real-world uses

NLP model debugging

Educational visualization of embeddings

Lab 10.3: Visualizing Word Embeddings using t-SNE

Objective
The objective of this lab is to visualize high-dimensional word embeddings in a two-dimensional space using t-SNE in order to understand semantic relationships between words.

Embedding Model
A pre-trained GloVe word embedding model with 100-dimensional vectors trained on Wikipedia and Gigaword corpus was used for this experiment. These embeddings capture semantic meaning based on contextual similarity.

Dataset / Word List
The dataset consists of 30 meaningful words selected from three semantic categories: animals, cities, and technology. This selection helps in observing clear semantic clusters during visualization.

t-SNE Visualization
The word vectors were reduced from 100 dimensions to 2 dimensions using the t-SNE algorithm. The reduced vectors were plotted using a scatter plot with word annotations.

Interpretation
The visualization shows clear clusters based on semantic similarity. Animal-related words form a distinct group, city names cluster together, and technology-related words appear close to each other. Words within the same category are closer than words from different categories. Minor overlaps occur due to shared contextual usage. Overall, the visualization demonstrates how word embeddings capture semantic relationships effectively.

Conclusion
t-SNE is an effective technique for visualizing and interpreting word embeddings. It helps in understanding semantic similarity and structure within high-dimensional embedding spaces.