Lab 10.4 – Visualizing Word Embeddings using t-SNE

Name: GUJJA RAJU Roll No: 2403A52018

```
# Import Libraries
!pip install gensim
import numpy as np
import matplotlib.pyplot as plt
from gensim.models import KeyedVectors
from sklearn.manifold import TSNE
```

```
Requirement already satisfied: gensim in /usr/local/lib/python3.12/dist-packages (4.4.0)
Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.12/dist-packages (from gensim) (2.0.2)
Requirement already satisfied: scipy>=1.7.0 in /usr/local/lib/python3.12/dist-packages (from gensim) (1.16.3)
Requirement already satisfied: smart_open>=1.8.1 in /usr/local/lib/python3.12/dist-packages (from gensim) (7.5.0)
Requirement already satisfied: wrapt in /usr/local/lib/python3.12/dist-packages (from smart_open>=1.8.1->gensim) (2.1.1)
```

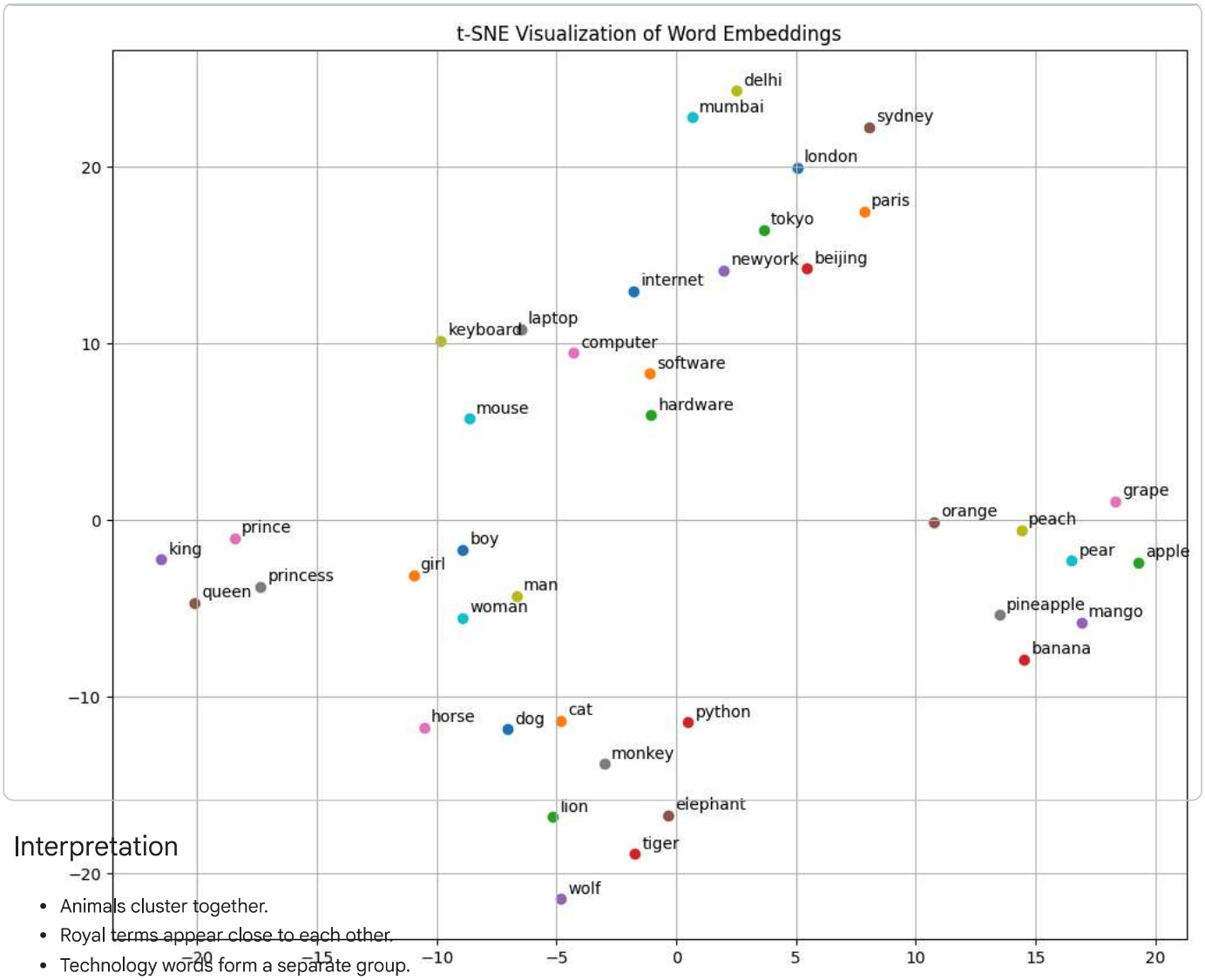
```
import gensim.downloader as api

# Load pretrained Word2Vec model directly in Colab
model = api.load("word2vec-google-news-300")

print("Vocabulary Size:", len(model.key_to_index))
print("Example Vector for king (first 10 dims):")
print(model["king"][:10])
```

```
[==================================================] 100.0% 1662.8/1662.8MB downloaded
Vocabulary Size: 3000000
Example Vector for king (first 10 dims):
[ 0.12597656  0.02978516  0.00860596  0.13964844 -0.02563477 -0.03613281
  0.11181641 -0.19824219  0.05126953  0.36328125]
```

```
# Select Words
words = [
    'dog','cat','lion','tiger','wolf','elephant','horse','monkey',
    'delhi','mumbai','london','paris','tokyo','beijing','newyork','sydney',
    'computer','laptop','keyboard','mouse','internet','software','hardware','python',
    'king','queen','prince','princess','man','woman','boy','girl',
    'apple','banana','mango','orange','grape','pineapple','peach','pear'
]

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

```
# Apply t-SNE
tsne = TSNE(n_components=2, random_state=42, perplexity=10)
reduced_vectors = tsne.fit_transform(vectors)
```

```
# Plot Visualization
plt.figure(figsize=(12,10))

for i, word in enumerate(words):
    x, y = reduced_vectors[i]
    plt.scatter(x, y)
    plt.text(x+0.3, y+0.3, word)

plt.title('t-SNE Visualization of Word Embeddings')
plt.grid(True)
plt.show()
```



Interpretation

- Animals cluster together.
- Royal terms appear close to each other.
- Technology words form a separate group.
- Cities are grouped based on semantic similarity.
- Some words may overlap due to multiple meanings.