```
1 # !pip install gensim
```

1. Library

```
1 from gensim.models import Word2Vec
2 import nltk
3 from nltk.tokenize import word_tokenize
4 import re
5 import gensim.downloader
6 from sklearn.manifold import TSNE
7 from sklearn.decomposition import PCA
8 import plotly.express as px
9 import numpy as np
```

```
1 for model in list(gensim.downloader.info()['models'].keys()):
2   print(model)
```

```
fasttext-wiki-news-subwords-300
conceptnet-numberbatch-17-06-300
word2vec-ruscorpora-300
word2vec-google-news-300
glove-wiki-gigaword-50
glove-wiki-gigaword-100
glove-wiki-gigaword-200
glove-wiki-gigaword-300
glove-twitter-25
glove-twitter-50
glove-twitter-100
glove-twitter-200
__testing_word2vec-matrix-synopsis
```

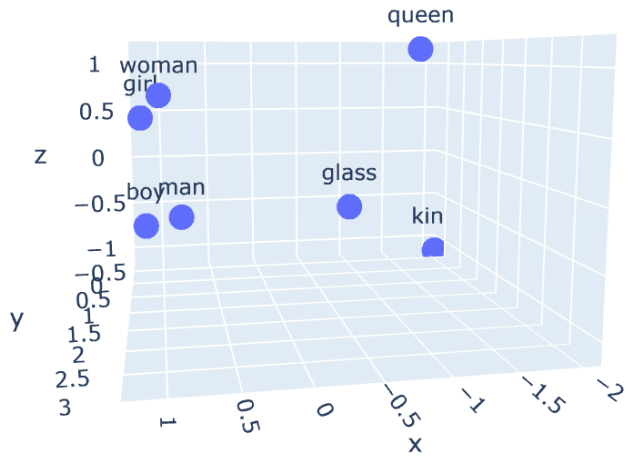
```
1 word2vec_vector = gensim.downloader.load("word2vec-google-news-300")
2 glove_vector = gensim.downloader.load("glove-wiki-gigaword-300")
3 fasttext_vector = gensim.downloader.load("fasttext-wiki-news-subwords-300")
```

```
[==================================================] 100.0% 1662.8/1662.8MB downloaded
[==================================================] 100.0% 376.1/376.1MB downloaded
```

```
def get_visual_vector(plot_words, vector):
  word_matrix = np.array([vector[word] for word in plot_words])
  num_components = 3
  pca = PCA(n_components=num_components)
  pca_result = pca.fit_transform(word_matrix)
  return pca_result

def visual_pca(vector, plot_words):
  data = {
    'x': vector[:, 0],
    'y': vector[:, 1],
    'z': vector[:, 2],
    'word': plot_words
  }

  fig = px.scatter_3d(data, x='x', y='y', z='z', text='word')

  fig.update_traces(marker=dict(size=8))

  fig.show()
```

Word2Vec

```
plot_words = ["king", "queen", "man", "woman", "glass","boy","girl"]

vector = get_visual_vector(plot_words, word2vec_vector)
visual_pca(vector, plot_words)
```



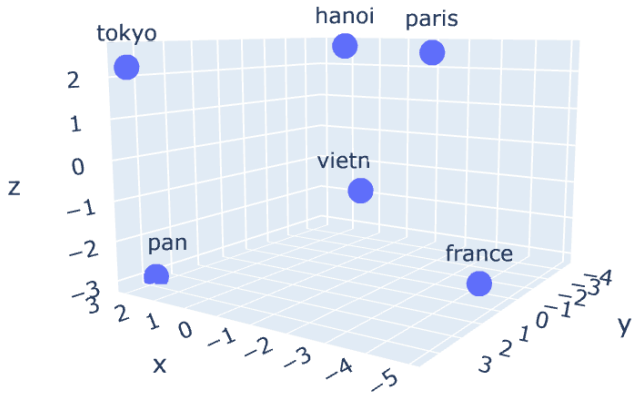
==> Quan sát biểu đồ trực quan có thể thấy được hướng của các cặp vector: man-woman, king-queen, girl-boy dường như rất gần nhau. Cho thấy khả năng biểu diễn vector ngữ nghĩa tương đối tốt của Word2Vec

Glove

```
# plot_words = ["king", "queen", "man", "woman", "glass","boy","girl"]
plot_words = ["vietnam", "hanoi", "japan", "tokyo", "france", "paris"]


```

```
vector = get_visual_vector(plot_words, glove_vector)
print(vector)

visual_pca(vector, plot_words)
```

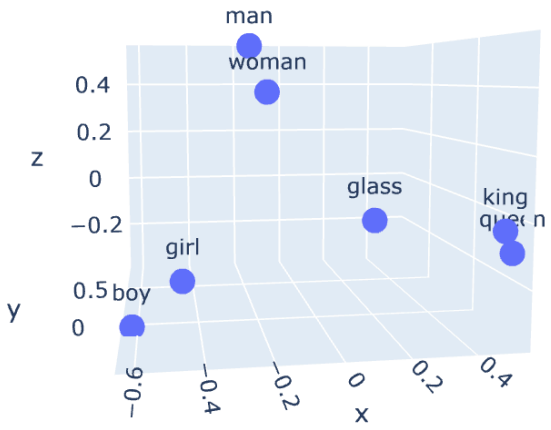
```
[[ 1.5482424 -4.529238  -1.9185411]
 [ 1.3882579 -3.4289033  2.618058 ]
 [ 3.0012653  2.7940404 -3.0955675]
 [ 2.9049764  3.5540733  2.2414753]
 [-4.96844    0.6396516 -2.480383 ]
 [-3.8743029  0.9703737  2.6349564]]
```



==> Quan sát biểu đồ trực quan có thể thấy được hướng của các cặp vector thủ đô-đất nước: hanoi-vietnam, tokyo-japan, paris-france dường như rất gần nhau.

Fasttext

```
plot_words = ["king", "queen", "man", "woman", "glass","boy","girl"]

vector = get_visual_vector(plot_words, fasttext_vector)
visual_pca(vector, plot_words)

```



==> Khác với 2 phương pháp vector hoá ở trên, tạo thành cặp các vector tương đối gần nhau, các vector gần nghĩa khi biểu diễn trong fastText thì thành cụm gần nhau hơn so với các cụm còn lại (cụm boy-girl, king-queen, man-woman)

1

1

1

1

1

1