```
# Numerical computations
import numpy as np

# Word embedding handling
from gensim.models import KeyedVectors

# Visualization
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA

# Download files from internet
import requests
```

```
url = "https://github.com/mmihaltz/word2vec-GoogleNews-vectors/raw/master/GoogleNews-vectors-negative300.bin.gz"
filename = "GoogleNews-vectors-negative300.bin.gz"

r = requests.get(url)

with open(filename, "wb") as f:
    f.write(r.content)

print("Model downloaded successfully")
```

```
Model downloaded successfully
```

```
model = KeyedVectors.load_word2vec_format(
    "GoogleNews-vectors-negative300.bin.gz",
    binary=True
)

print("Vocabulary size:", len(model.key_to_index))
print("Example vector for 'king':")
print(model["king"])
```

```
 -1.77734375e-01  8.59375000e-02 -2.18505859e-02  2.05078125e-02
 -1.39648438e-01  2.51464844e-02  1.38671875e-01 -1.05468750e-01
  1.38671875e-01  8.88671875e-02 -7.51953125e-02 -2.13623047e-02
  1.72851562e-01  4.63867188e-02 -2.65625000e-01  8.91113281e-03
  1.49414062e-01  3.78417969e-02  2.38281250e-01 -1.24511719e-01
 -2.17773438e-01 -1.81640625e-01  2.97851562e-02  5.71289062e-02
 -2.89306641e-02  1.24511719e-02  9.66796875e-02 -2.31445312e-01
  5.81054688e-02  6.68945312e-02  7.08007812e-02 -3.08593750e-01
 -2.14843750e-01  1.45507812e-01 -4.27734375e-01 -9.39941406e-03
  1.54296875e-01 -7.66601562e-02  2.89062500e-01  2.77343750e-01
 -4.86373901e-04 -1.36718750e-01  3.24218750e-01 -2.46093750e-01
 -3.03649902e-03 -2.11914062e-01  1.25000000e-01  2.69531250e-01
  2.04101562e-01  8.25195312e-02 -2.01171875e-01 -1.60156250e-01
 -3.78417969e-02 -1.20117188e-01  1.15234375e-01 -4.10156250e-02
 -3.95507812e-02 -8.98437500e-02  6.34765625e-03  2.03125000e-01
  1.86523438e-01  2.73437500e-01  6.29882812e-02  1.41601562e-01
 -9.81445312e-02  1.38671875e-01  1.82617188e-01  1.73828125e-01
  1.73828125e-01 -2.37304688e-01  1.78710938e-01  6.34765625e-02
  2.36328125e-01 -2.08984375e-01  8.74023438e-02 -1.66015625e-01
 -7.91015625e-02  2.43164062e-01 -8.88671875e-02  1.26953125e-01
 -2.16796875e-01 -1.73828125e-01 -3.59375000e-01 -8.25195312e-02
 -6.49414062e-02  5.07812500e-02  1.35742188e-01 -7.47070312e-02
 -1.64062500e-01  1.15356445e-02  4.45312500e-01 -2.15820312e-01
 -1.11328125e-01 -1.92382812e-01  1.70898438e-01 -1.25000000e-01
  2.65502930e-03  1.92382812e-01 -1.74804688e-01  1.39648438e-01
  2.92968750e-01  1.13281250e-01  5.95703125e-02 -6.39648438e-02
  9.96093750e-02 -2.72216797e-02  1.96533203e-02  4.27246094e-02
 -2.46093750e-01  6.39648438e-02 -2.25585938e-01 -1.68945312e-01
```

```
 -9.37500000e-02 -6.68945312e-02  2.27050781e-02  7.61718750e-02
  2.89062500e-01  3.10546875e-01 -5.37109375e-02  2.28515625e-01
  2.51464844e-02  6.78710938e-02 -1.21093750e-01 -2.15820312e-01
 -2.73437500e-01 -3.07617188e-02 -3.37890625e-01  1.53320312e-01
  2.33398438e-01 -2.08007812e-01  3.73046875e-01  8.20312500e-02
  2.51953125e-01 -7.61718750e-02 -4.66308594e-02 -2.23388672e-02
  2.99072266e-02 -5.93261719e-02 -4.66918945e-03 -2.44140625e-01
 -2.09960938e-01 -2.87109375e-01 -4.54101562e-02 -1.77734375e-01
 -2.79296875e-01 -8.59375000e-02  9.13085938e-02  2.51953125e-01]
```

```
pairs = [
    ("doctor", "nurse"),
    ("cat", "dog"),
    ("car", "bus"),
    ("king", "queen"),
    ("apple", "orange"),
    ("teacher", "student"),
    ("computer", "laptop"),
    ("city", "village"),
    ("man", "woman"),
    ("sun", "moon")
]

for w1, w2 in pairs:
    print(w1, "-", w2, ":", model.similarity(w1, w2))
```

```
doctor - nurse : 0.63195235
cat - dog : 0.76094574
car - bus : 0.4693371
king - queen : 0.6510957
apple - orange : 0.39203462
teacher - student : 0.63013655
computer - laptop : 0.66404927
city - village : 0.47896868
man - woman : 0.76640123
sun - moon : 0.4262834
```

```
words = ["king", "university", "car", "computer", "music"]

for word in words:
    print("\nSimilar words for", word)
    print(model.most_similar(word, topn=5))
```

```

Similar words for king
[('kings', 0.7138045430183411), ('queen', 0.6510956883430481), ('monarch', 0.6413194537162781), ('crown_prince', 0.620422005

Similar words for university
[('universities', 0.7003918886184692), ('faculty', 0.6780907511711121), ('unversity', 0.6758289933204651), ('undergraduate',

Similar words for car
[('vehicle', 0.7821096181869507), ('cars', 0.7423831224441528), ('SUV', 0.7160962224006653), ('minivan', 0.6907036900520325)

Similar words for computer
[('computers', 0.7979379892349243), ('laptop', 0.6640493273735046), ('laptop_computer', 0.6548868417739868), ('Computer', 0.

Similar words for music
[('classical_music', 0.7197794318199158), ('jazz', 0.6834640502929688), ('Music', 0.6595720648765564), ('Without_Donny_Kirsh
```

```
print(model.most_similar(
    positive=['king','woman'],
    negative=['man'],
    topn=1))

print(model.most_similar(
    positive=['paris','india'],
    negative=['france'],
    topn=1))

print(model.most_similar(
    positive=['teacher','hospital'],
    negative=['school'],
    topn=1))
```

```
[('queen', 0.7118193507194519)]
[('chennai', 0.5442505478858948)]
[('Hospital', 0.6331106424331665)]
```

```
words = ["king","queen","man","woman",
         "car","bus","train","plane",
         "apple","banana","orange","fruit",
         "dog","cat","horse","animal",
         "teacher","student","school","college"]
```

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

pca = PCA(n_components=2)
reduced = pca.fit_transform(vectors)

plt.figure()
plt.scatter(reduced[:,0], reduced[:,1])

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

plt.title("Word Embeddings Visualization")
plt.show()
```