Import the library

```
from gensim.models import word2vec
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
```

Loading the corpus

```
sentences = [['human', 'interface', 'computer'],
             ['survey','user', 'computer', 'response', 'time'],
             ['eps', 'user', 'interface', 'system'],
             ['user', 'response', 'time'],
             ['trees'],
             ['graph', 'trees'],
             ['graph', 'minors', 'trees'],
             ['graph', 'minor', 'survey']]
```

Training the model

```
model = word2vec.Word2Vec(sentences = sentences, min_count = 1,
vector_size=10)

print(model.wv['trees'])
```

```
[ 0.07380505 -0.01533471 -0.04536613  0.06554051 -0.0486016  -
0.01816018
  0.0287658   0.00991874 -0.08285215 -0.09448818]
```

```
model.wv.similarity('human', 'computer')
```

```
0.00073069916
```

```
model.wv.most_similar('computer', topn = 10)
```

```
[('response', 0.6143981218338013),
 ('human', 0.3862057030200958),
 ('interface', 0.19734424352645874),
 ('eps', 0.11534241586923599),
 ('survey', -0.04264535382390022),
 ('user', -0.08911214768886566),
 ('time', -0.11387499421834946),
 ('trees', -0.1799870878458023),
 ('graph', -0.18973825871944427),
 ('minors', -0.19863170385360718)]
```

performing PCA

```
vectors = [ model.wv[word] for word in model.wv.index_to_key]
pca = PCA(n_components = 2)
final_vec = pca.fit_transform(vectors)

final_vec

array([[-0.15545549, -0.06809328],
       [-0.14398222, -0.01501524],
       [-0.04692788,  0.06700867],
       [-0.02100691,  0.09336288],
       [ 0.14448316, -0.13837725],
       [ 0.08479437, -0.01664304],
       [ 0.04851948, -0.0309391 ],
       [-0.03246829,  0.10978899],
       [-0.03208923, -0.01428178],
       [ 0.12869936,  0.05282802],
       [-0.03967166, -0.01215902],
       [-0.04580893, -0.09020352],
       [ 0.11091425,  0.06272367]])
```

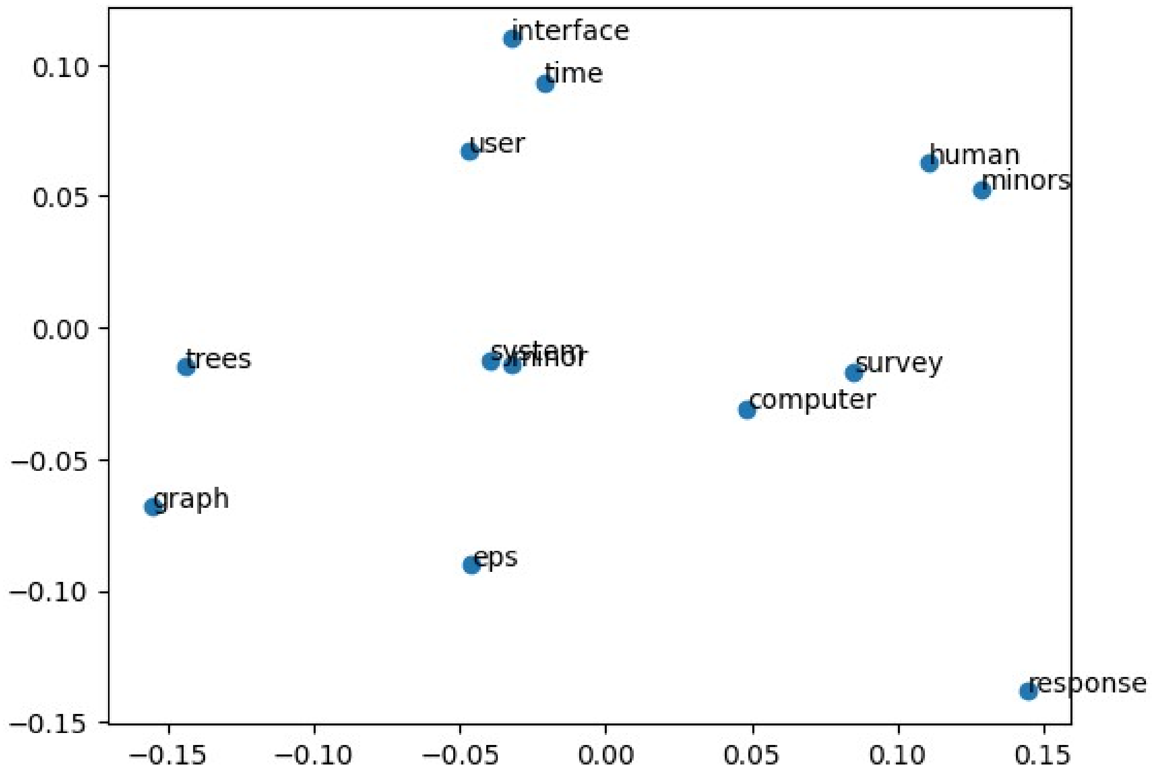
Visualising the word Vectors

```

plt.scatter(final_vec[:, 0], final_vec[:, 1])

words = list(model.wv.index_to_key)
for i, word in enumerate(words):
    plt.annotate(word, xy=(final_vec[i, 0], final_vec[i, 1]))

plt.show()
```



interface
time
user
human
minors
trees
system
minor
survey
computer
graph
eps
response
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.15
−0.10
−0.05
0.00
0.05
0.10
0.15