In [2]:

```
from gensim.models import Word2Vec

corpus = [
    ['king', 'queen', 'man', 'woman'],
    ['paris', 'france', 'london', 'england'],
    ['apple', 'orange', 'banana', 'fruit'],
    ['dog', 'cat', 'animal', 'pet'],
    ['car', 'bus', 'truck', 'vehicle']
]
```

In [3]:

```
# 2. Train Word2Vec model using CBOW (sg=0)
model = Word2Vec(
    sentences=corpus,
    vector_size=50,
    window=2,
    min_count=1,
    sg=0  # 0 = CBOW, 1 = Skip-gram
)
```

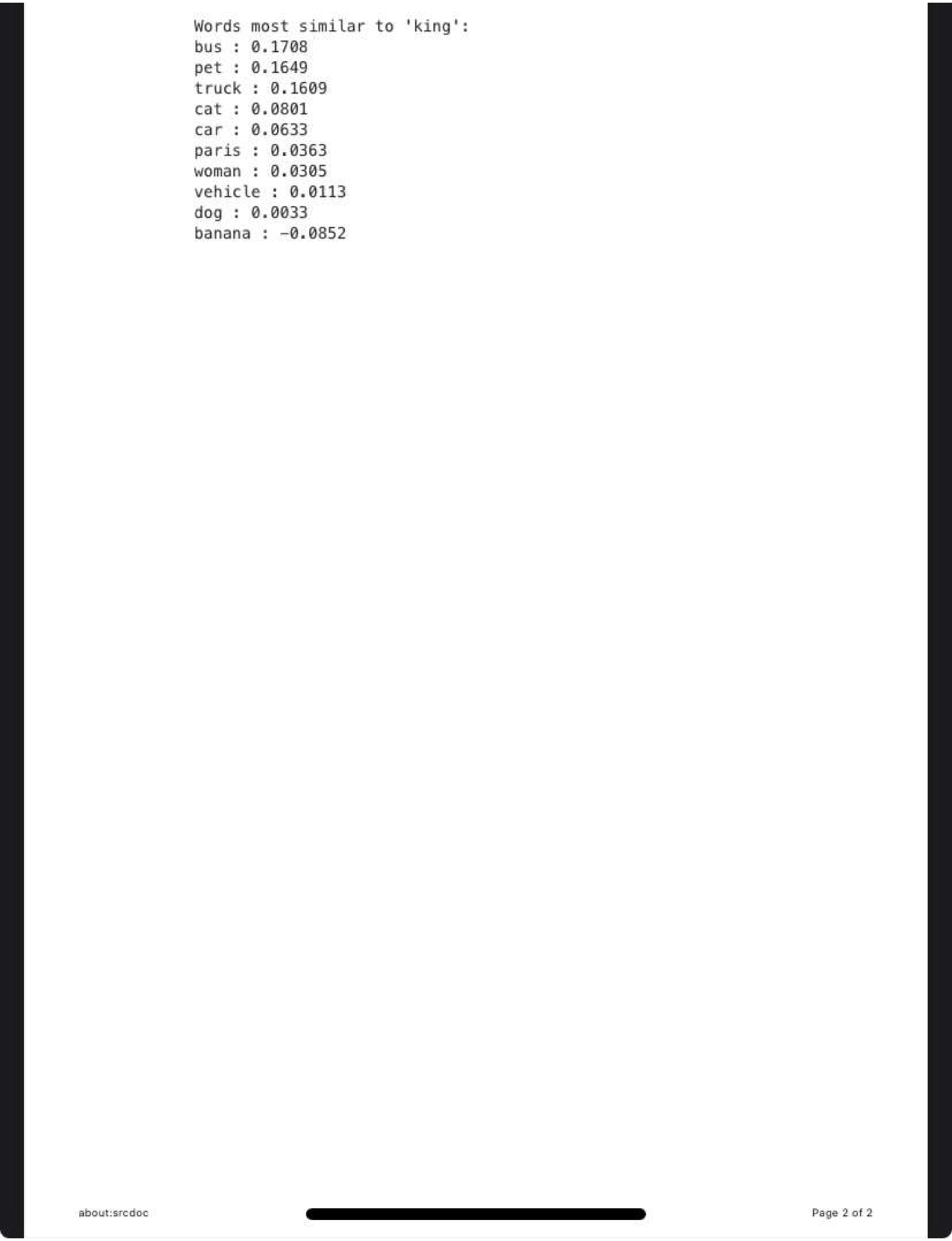
In [5]:

```
# 3. Find vector for a word
print("Vector for 'king':\n", model.wv['king'])
```

```
Vector for 'king':
 [ 0.00513854  0.01623337 -0.00277442  0.01615586  0.00743403 -0.016
09465
 -0.00786723 -0.00494376  0.00978609 -0.00174432 -0.00566182  0.0156
6741
  0.0186458  -0.00322986 -0.0103185  -0.00940352 -0.0096921  -0.0192
0565
  0.00274404 -0.00844983  0.00505342  0.01122897 -0.00813181 -0.0191
9316
  0.0030934  -0.01340024  0.00499035 -0.00756126  0.01415684  0.0012
8044
  0.00712188 -0.00547827 -0.00342109  0.01530559  0.00281536 -0.0117
009
 -0.01566899  0.00246537  0.01290926  0.0111127  -0.0179541   0.0171
8433
  0.00809396  0.01493921  0.01949267 -0.01457916 -0.01807993  0.0116
72
  0.01878243  0.00701385]
```

In [7]:

```
# 4. Find similar words
print("\nWords most similar to 'king':")
for word, score in model.wv.most_similar('king'):
    print(f"{word} : {score:.4f}")
```



```
Words most similar to 'king':
bus : 0.1708
pet : 0.1649
truck : 0.1609
cat : 0.0801
car : 0.0633
paris : 0.0363
woman : 0.0305
vehicle : 0.0113
dog : 0.0033
banana : -0.0852
```