PG3: Train a custom Word2Vec model on a small dataset. Train embeddings on a domain-specific corpus (e.g., legal, medical) and analyze how embeddings capture domain-specific semantics

Soln:

```
!pip install gensim
```

#Gensim: A Python library for NLP and word embeddings.

Important Steps

1. **Tokenization**: Converts sentences into lists of lowercase tokens for processing.
2. **Word2Vec Training**:
   - `vector_size`: Sets the embedding dimension to 50.
   - `window`: Uses a context window of 3 words.
   - `sg`: Skip-gram (sg=1) is used, which works better for smaller datasets.
   - `epochs`: The number of training iterations.
3. **Visualization**: PCA reduces the high-dimensional word vectors to 2D for visualization, helping to understand semantic relationships.
4. **Semantic Analysis**: The `most_similar` method identifies words that are semantically similar based on embeddings.

Example: Legal Corpus

```
from gensim.models import Word2Vec
from gensim.utils import simple_preprocess
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
```

```
legal_corpus = [
    "The court ruled in favor of the plaintiff.",
    "The defendant was found guilty of negligence.",
    "A breach of contract case was filed.",
    "The agreement between parties must be honored.",
    "The lawyer presented compelling evidence.",
    "Legal documents must be drafted carefully.",
    "The jury deliberated for several hours.",
    "A settlement was reached between the parties.",
    "The plaintiff claimed damages for losses incurred.",
    "The contract outlined the obligations of both parties."
]
```

```
# Example legal corpus
legal_corpus = [
    "The court ruled in favor of the plaintiff.",
    "The defendant was found guilty of negligence.",
    "A breach of contract case was filed.",
    "The agreement between parties must be honored.",
    "The lawyer presented compelling evidence.",
    "Legal documents must be drafted carefully.",
    "The jury deliberated for several hours.",
    "A settlement was reached between the parties.",
    "The plaintiff claimed damages for losses incurred.",
    "The contract outlined the obligations of both parties."
]

# Preprocess the corpus
tokenized_corpus = [simple_preprocess(sentence) for sentence in
legal_corpus]

# Train the Word2Vec model
legal_word2vec = Word2Vec(
    sentences=tokenized_corpus,
    vector_size=50,  # Embedding dimension
    window=3,        # Context window size
    min_count=1,     # Minimum word frequency
    sg=1,            # Skip-gram model
    epochs=100       # Training epochs
)

# Save the model for later use
legal_word2vec.save("legal_word2vec.model")
```

```
# Analyze embeddings: Display vector for a specific word
word = "lawyer"
if word in legal_word2vec.wv:
    print(f"Vector embedding for
'{word}':\n{legal_word2vec.wv[word]}\n")
else:
    print(f"Word '{word}' not found in the Word2Vec model.")
```

Output:

```
Vector embedding for 'lawyer':

[ 0.00373483  0.01353383  0.00585796 -0.01324683  0.01500349 -0.01261986

  0.01892563  0.00698961 -0.0087639  -0.01023367 -0.00875896 -0.01318524
```

```
 0.01972703 -0.00463062  0.01525868 -0.01837575  0.0055629  -0.00126356
 0.01417167 -0.01969541  0.01564029 -0.00948072 -0.0107858  -0.01128642
-0.00610619 -0.00604345 -0.00693252 -0.01396556  0.00086967 -0.00136903
-0.00358557  0.00685404 -0.01432065 -0.00657563  0.00952303  0.01720192
-0.01858611  0.01418636  0.01038651 -0.00818817  0.01832661 -0.01858529
 0.01404059  0.01154918  0.00326395 -0.01036671 -0.00841038 -0.00736812
 0.00374052  0.00413726]
```

```
# Visualize embeddings using PCA
words_to_visualize = ["court", "plaintiff", "defendant", "agreement",
"lawyer", "evidence", "contract", "settlement", "jury", "damages"]
word_vectors = [legal_word2vec.wv[word] for word in words_to_visualize]
```

```
word_vectors
```

Output:

```
[array([-0.01018794, -0.0037532 , -0.01479373,  0.00535417,  0.00549183,
        -0.00194653, -0.00904275, -0.00120178,  0.01239534,  0.005502  ,
        -0.01752885, -0.00888894,  0.00678894,  0.00598825, -0.01972261,
         0.01158325, -0.01438892, -0.01200779,  0.00463451, -0.01056976,
         0.00906795,  0.01991566, -0.00384839,  0.01845003,  0.00452612,
         0.02153785,  0.0106156 , -0.0164802 , -0.0075984 ,  0.01259563,
         0.01069134,  0.01610584,  0.01608272,  0.01619358, -0.02157517,
         0.00898223, -0.00762749,  0.00642556,  0.01106042,  0.00757853,
         0.01795846,  0.00227335, -0.00347768,  0.01356644, -0.00962057,
         0.00016249,  0.01841913, -0.01246461,  0.00897428, -0.01424266], dtype=float32),
 array([-0.01382369,  0.0011437 , -0.01449836, -0.00296123,  0.00624782,
         0.00982854,  0.00482432,  0.00674102, -0.01035763,  0.0125059 ,
        -0.01251031,  0.00691753, -0.01420641,  0.00583739, -0.01051113,
        -0.00608784, -0.00284186,  0.01448168,  0.00904517, -0.01370935,
         0.00321437, -0.01510567,  0.01918532,  0.01829434, -0.00426899,
         0.00343321,  0.00058916,  0.01309333, -0.0183534 ,  0.00069488,
         0.0132396 ,  0.0028656 ,  0.00451153, -0.01875341,  0.01468391,
        -0.01201477, -0.00313035,  0.00620906, -0.0025351 ,  0.00151398,
         0.0066815 , -0.01435055, -0.02045849,  0.01987134,  0.01433266,
        -0.01331776,  0.00661788, -0.00128313,  0.01081608, -0.01213262],
       dtype=float32),
 array([-0.01735372,  0.00324048, -0.00153466, -0.01745638, -0.01992291,
        -0.00444436,  0.01032289,  0.00879421, -0.01455397, -0.01536747,
        -0.01001598, -0.00653257, -0.0128883 , -0.01829896, -0.0059358 ,
        -0.01495283, -0.00974466, -0.00899372, -0.00697665, -0.00573702,
        -0.01700949,  0.0003172 ,  0.01874463,  0.01480774, -0.01384414,
        -0.00612229,  0.00565069, -0.01732042,  0.00195022,  0.01274919,
         0.01080692, -0.01920946, -0.00795812, -0.01638088, -0.00148547,
         0.01870203,  0.01399906,  0.00958245,  0.00941055, -0.00658938,
         0.02153141, -0.01520018, -0.01545056, -0.00327682,  0.00024155,
        -0.00606883, -0.00135896,  0.01406399,  0.00023136, -0.00163963],
       dtype=float32),
 array([ 0.00544287,  0.01555425, -0.00307019,  0.01717134,  0.00693973,
        -0.0170452 , -0.00679161, -0.00333116,  0.00903156, -0.00295565,
        -0.00578579,  0.01436892,  0.02001157, -0.00213262, -0.01079166,
        -0.007799  , -0.00751752, -0.01736892,  0.00095211, -0.01050753,
         0.00615652,  0.01262798, -0.00638232, -0.01966911,  0.00394809,
        -0.01237557,  0.0045896 , -0.00610946,  0.01346509,  0.00106505,
         0.00631289, -0.00667795, -0.00218376,  0.01535427,  0.00144457,
        -0.0117866 , -0.01405202,  0.00186158,  0.0125593 ,  0.0105592 ,
        -0.01650265,  0.01693893,  0.0074757 ,  0.0163192 ,  0.02056517,
        -0.01457632, -0.01834234,  0.01092377,  0.01994778,  0.00864314],
       dtype=float32),
 array([ 0.00373483,  0.01353383,  0.00585796, -0.01324683,  0.01500349,
        -0.01261986,  0.01892563,  0.00698961, -0.0087639 , -0.01023367,
        -0.00875896, -0.01318524,  0.01972703, -0.00463062,  0.01525868,
        -0.01837575,  0.0055629 , -0.00126356,  0.01417167, -0.01969541,
         0.01564029, -0.00948072, -0.0107858 , -0.01128642, -0.00610619,
        -0.00604345, -0.00693252, -0.01396556,  0.00086967, -0.00136903,
        -0.00358557,  0.00685404, -0.01432065, -0.00657563,  0.00952303,
         0.01720192, -0.01858611,  0.01418636,  0.01038651, -0.00818817,
         0.01832661, -0.01858529,  0.01404059,  0.01154918,  0.00326395,
        -0.01036671, -0.00841038, -0.00736812,  0.00374052,  0.00413726],
       dtype=float32),
 array([ 0.00550566,  0.00103798, -0.00515228,  0.01945088,  0.00499871,
         0.00736707, -0.0011947 ,  0.00298867,  0.01247635, -0.00248031,
         0.00660107, -0.00238972,  0.01178223,  0.00798718,  0.00505932,
        -0.00936528, -0.00755702,  0.00989482, -0.01304692, -0.00193519,
        -0.00039899,  0.0078729 , -0.01549838,  0.01741308, -0.0023178 ,
        -0.00983727,  0.00754468, -0.0027872 , -0.01603217, -0.00921708,
        -0.00134961, -0.01871502,  0.002125  ,  0.00480915, -0.00744796,
         0.00537565,  0.00629158,  0.01973929,  0.0024904 ,  0.00340102,
         0.00710946, -0.00441335, -0.01761757,  0.01698   , -0.0031966 ,
        -0.0194808 , -0.01307702, -0.00849545,  0.00867249,  0.01145031],
       dtype=float32),
 array([ 0.00298495, -0.00700375, -0.01431873, -0.01400242, -0.01991   ,
        -0.01428693, -0.00105788, -0.00551727, -0.0153189 , -0.0100668 ,
         0.00858984, -0.01069687,  0.01958971,  0.00508815, -0.01531299,
         0.02237322,  0.01962719,  0.01488377, -0.01710452,  0.00707861,
         0.01021231,  0.01304598,  0.01277774,  0.00337116, -0.00486931,
         0.01909359,  0.01800028,  0.01032766, -0.00758116, -0.00048564,
         0.00164387, -0.0189941 , -0.01410591, -0.00047871, -0.00010738,
        -0.01102702, -0.0061726 , -0.01550268,  0.0161471 , -0.00069464,
        -0.00545253,  0.01093123,  0.01718066, -0.00617879,  0.0186232 ,
         0.01143978,  0.01163601, -0.00062903,  0.01886317, -0.0114121 ],
       dtype=float32),
 array([-0.00976338, -0.00780081,  0.019559  ,  0.01830878, -0.00654693,
         0.01011876,  0.01784409, -0.00301464,  0.01761319,  0.01262996,
         0.00393919, -0.00980376, -0.00886089, -0.00519702,  0.01557352,
        -0.01077065, -0.00356288,  0.02101176,  0.00479641,  0.01005143,
        -0.01529913,  0.00012613,  0.01357427, -0.01018804,  0.01573833,
         0.02000298, -0.00148577, -0.00239202, -0.00189635, -0.01172749,
        -0.01742925, -0.00479667, -0.00404787,  0.00869905, -0.01522061,
         0.01094797, -0.01160657, -0.0163735 ,  0.01649981,  0.01770434,
        -0.00497504,  0.00637913, -0.01261914, -0.0161758 , -0.00964475,
         0.01381735,  0.01255536,  0.01808335,  0.01568656,  0.01504712],
       dtype=float32),
array([ 0.01492715,  0.01934095,  0.01774599, -0.00747902,  0.01891196,
        -0.0020531 ,  0.01053821,  0.00635226, -0.00197045,  0.00632444,
        -0.01059867, -0.01259004, -0.01428318,  0.00465197,  0.01267453,
         0.0028981 ,  0.00384051,  0.00779968,  0.01519439, -0.01727828,
         0.00548228, -0.01392599,  0.00899558,  0.01923842,  0.01556924,
         0.01372755,  0.01566461,  0.01412691,  0.01313736,  0.01728814,
        -0.01028677,  0.01760968,  0.01114872, -0.00440745,  0.01607866,
         0.01023387,  0.02058647,  0.00533376,  0.01917837,  0.00176341,
         0.01960336,  0.0070012 ,  0.01266869, -0.00624314,  0.01447076,
         0.01456392, -0.00432669, -0.00459186,  0.00780778, -0.01304201],
       dtype=float32),
 array([ 0.0158812 ,  0.0174168 ,  0.00195527, -0.01554414,  0.01595952,
        -0.00898288,  0.0134456 ,  0.01096715,  0.01782416, -0.02043355,
         0.01853634, -0.02043897, -0.01187125, -0.01672532, -0.01152777,
         0.01697107,  0.02129747, -0.00410723,  0.0053023 , -0.01103053,
         0.017639  ,  0.01337754,  0.0028419 , -0.00731513, -0.01343816,
         0.01128781,  0.00173393, -0.00338352,  0.01469343, -0.00834176,
        -0.01866035, -0.00566033,  0.00234445, -0.0135692 , -0.0126051 ,
        -0.01704373,  0.02071049,  0.0147259 , -0.00145971,  0.01323994,
         0.01840546, -0.0020906 ,  0.0126531 ,  0.0093615 ,  0.01196339,
         0.01458218, -0.00961088, -0.00608193,  0.00305689,  0.01033708],
       dtype=float32)]
```

```python
# Dimensionality reduction
pca = PCA(n_components=2)
reduced_vectors = pca.fit_transform(word_vectors)
reduced_vectors
```

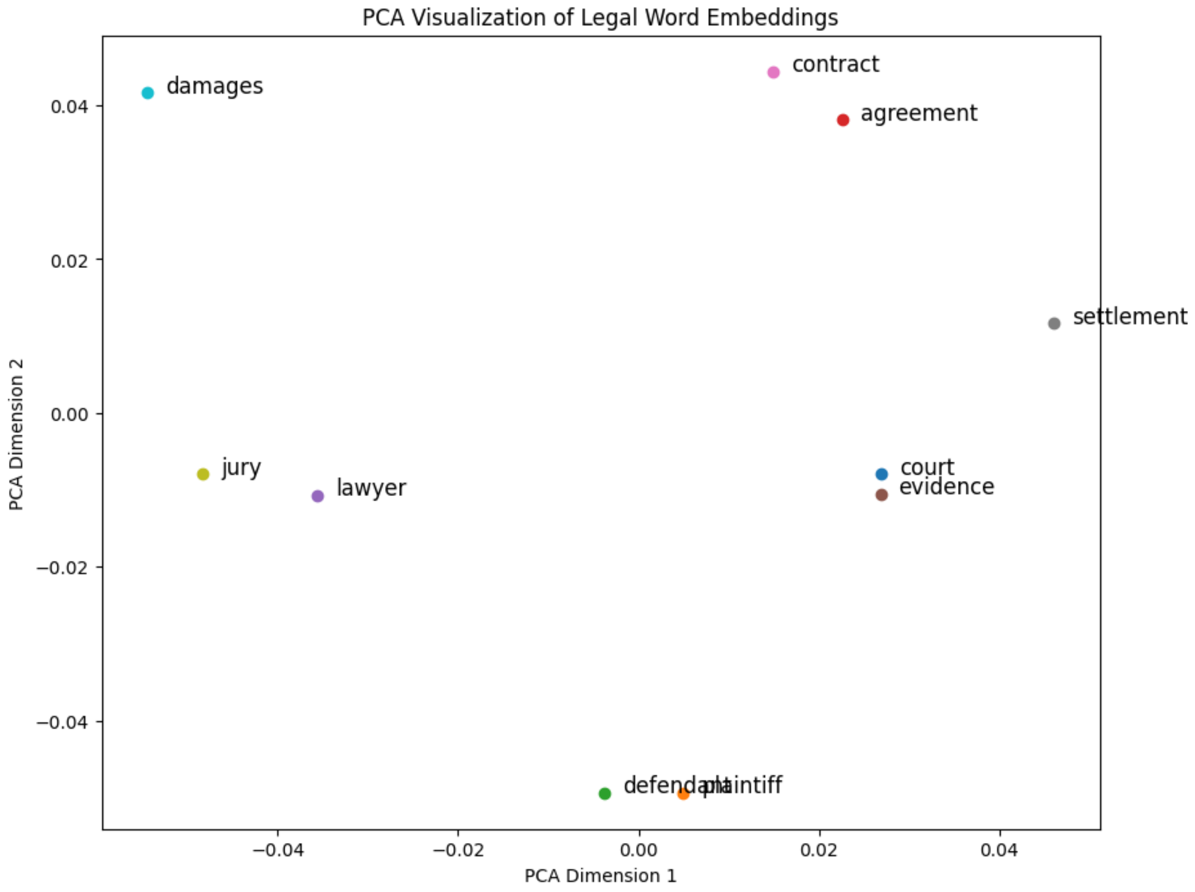
Output:

```
array([[ 0.02688162, -0.00792018],
       [ 0.00493226, -0.04934309],
       [-0.00377306, -0.04936944],
       [ 0.02256997,  0.03808062],
       [-0.0355795 , -0.01066101],
       [ 0.02682294, -0.01050709],
       [ 0.01486912,  0.0443972 ],
       [ 0.04605154,  0.01166099],
       [-0.0482769 , -0.0079725 ],
       [-0.05449799,  0.0416345 ]])
```

```python
# Plot embeddings
plt.figure(figsize=(10, 8))
for i, word in enumerate(words_to_visualize):
    plt.scatter(reduced_vectors[i, 0], reduced_vectors[i, 1])
    plt.text(reduced_vectors[i, 0] + 0.002, reduced_vectors[i, 1],
word, fontsize=12)
plt.title("PCA Visualization of Legal Word Embeddings")
plt.xlabel("PCA Dimension 1")
plt.ylabel("PCA Dimension 2")
plt.show()
```

Output:

```
array([[ 0.02688162, -0.00792018],
       [ 0.00493226, -0.04934309],
       [-0.00377306, -0.04936944],
       [ 0.02256997,  0.03808062],
       [-0.0355795 , -0.01066101],
       [ 0.02682294, -0.01050709],
       [ 0.01486912,  0.0443972 ],
       [ 0.04605154,  0.01166099],
       [-0.0482769 , -0.0079725 ],
       [-0.05449799,  0.0416345 ]])
```



```python
# Find similar words
similar_words = legal_word2vec.wv.most_similar("lawyer", topn=5)
print(f"Words similar to 'lawyer': {similar_words}")
```

```
Output:

Words similar to 'lawyer': [('carefully', 0.29186686873435974),
('claimed', 0.27888569235801697), ('jury', 0.21892617642879486),
('damages', 0.1961500644683838), ('negligence', 0.1820133775472641)]
```

```
### Example: Legal and Medical / Healthcare Corpus
```

Example: Legal and Medical / Healthcare Corpus

```python
from gensim.models import Word2Vec
from gensim.utils import simple_preprocess
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

# Enhanced legal and medical corpus
enhanced_corpus = [
    # Legal domain
    "The court ordered the immediate release of the detained individual due to lack of evidence.",
    "A new amendment was introduced to ensure the protection of intellectual property rights.",
    "The defendant pleaded not guilty, citing an alibi supported by credible witnesses.",
    "The plaintiff accused the company of violating environmental regulations.",
    "A settlement agreement was reached through arbitration, avoiding a lengthy trial.",
    "The legal team presented a compelling argument to overturn the previous judgment.",
    "Contractual obligations must be fulfilled unless waived by mutual consent.",
    "The jury found the accused guilty of fraud and embezzlement.",
    "The appeal was dismissed as the evidence presented was deemed inadmissible.",
    "The attorney emphasized the importance of adhering to constitutional rights.",

    # Medical domain
    "The patient was admitted to the emergency department with severe chest pain.",
    "The surgeon successfully performed a minimally invasive procedure to remove the tumor.",
    "Clinical trials showed significant improvement in patients treated with the experimental drug.",
    "Regular screening is essential for early detection of chronic illnesses such as diabetes.",
    "The doctor recommended physical therapy to improve mobility after surgery.",
    "The hospital implemented stringent protocols to prevent the spread of infectious diseases.",
    "The nurse monitored the patient's vital signs hourly to ensure stability.",
    "Vaccination campaigns have drastically reduced the prevalence of polio worldwide.",
    "The radiologist identified a small abnormality in the CT scan requiring further investigation.",
    "Proper nutrition and exercise are vital components of a healthy
lifestyle."
]


# Preprocess the corpus
tokenized_corpus = [simple_preprocess(sentence) for sentence in
enhanced_corpus]
tokenized_corpus
```

Output :

```
[['the',
  'court',
  'ordered',
  'the',
  'immediate',
  'release',
  'of',
  'the',
  'detained',
  'individual',
  'due',
  'to',
  'lack',
  'of',
  'evidence'],
 ['new',
  'amendment',
  'was',
  'introduced',
  'to',
  'ensure',
  'the',
  'protection',
  'of',
  'intellectual',
  'property',
  'rights'],
 ['the',
  'defendant',
  'pleaded',
  'not',
  'guilty',
  'citing',
  'an',
  'alibi',
  'supported',
  'by',
  'credible',
  'witnesses'],
 ['the',
  'plaintiff',
  'accused',
  'the',
  'company',
  'of',
  'violating',
  'environmental',
  'regulations'],
 ['settlement',
  'agreement',
  'was',
  'reached',
  'through',
  'arbitration',
  'avoiding',
  'lengthy',
  'trial'],
 ['the',
  'legal',
  'team',
  'presented',
  'compelling',
  'argument',
  'to',
  'overturn',
  'the',
  'previous',
  'judgment'],
 ['contractual',
  'obligations',
  'must',
  'be',
  'fulfilled',
  'unless',
  'waived',
  'by',
  'mutual',
  'consent'],
 ['the',
  'jury',
  'found',
  'the',
  'accused',
  'guilty',
  'of',
  'fraud',
  'and',
  'embezzlement'],
 ['the',
  'appeal',
  'was',
  'dismissed',
  'as',
  'the',
  'evidence',
  'presented',
  'was',
  'deemed',
  'inadmissible'],
 ['the',
  'attorney',
  'emphasized',
  'the',
  'importance',
  'of',
  'adhering',
  'to',
  'constitutional',
  'rights'],
 ['the',
  'patient',
  'was',
  'admitted',
  'to',
  'the',
  'emergency',
  'department',
  'with',
  'severe',
  'chest',
  'pain'],
 ['the',
  'surgeon',
  'successfully',
  'performed',
  'minimally',
  'invasive',
  'procedure',
  'to',
  'remove',
  'the',
  'tumor'],
 ['clinical',
  'trials',
  'showed',
  'significant',
  'improvement',
  'in',
  'patients',
  'treated',
  'with',
  'the',
  'experimental',
  'drug'],
 ['regular',
  'screening',
  'is',
  'essential',
  'for',
  'early',
  'detection',
  'of',
  'chronic',
  'illnesses',
  'such',
  'as',
  'diabetes'],
 ['the',
  'doctor',
  'recommended',
  'physical',
  'therapy',
  'to',
  'improve',
  'mobility',
  'after',
  'surgery'],
 ['the',
  'hospital',
  'implemented',
  'stringent',
  'protocols',
  'to',
  'prevent',
  'the',
  'spread',
  'of',
  'infectious',
  'diseases'],
 ['the',
  'nurse',
  'monitored',
  'the',
  'patient',
  'vital',
  'signs',
  'hourly',
  'to',
  'ensure',
  'stability'],
 ['vaccination',
  'campaigns',
  'have',
  'drastically',
  'reduced',
  'the',
  'prevalence',
  'of',
  'polio',
  'worldwide'],
 ['the',
  'radiologist',
  'identified',
  'small',
  'abnormality',
  'in',
  'the',
  'ct',
  'scan',
  'requiring',
  'further',
  'investigation'],
 ['proper',
  'nutrition',
  'and',
  'exercise',
  'are',
  'vital',
  'components',
  'of',
  'healthy',
  'lifestyle']]
```

```python
# Train Word2Vec
domain_word2vec = Word2Vec(
    sentences=tokenized_corpus,
    vector_size=100,  # Higher embedding dimension for better
representation
    window=5,         # Wider context window
    min_count=1,      # Include all words
    sg=1,             # Skip-gram model
    epochs=150        # More training iterations
)

# Save the model
domain_word2vec.save("enhanced_domain_word2vec.model")
```

```python
# Analyze embeddings: Get vectors for specific words
words_to_analyze = ["court", "plaintiff", "doctor", "patient",
"guilty", "surgery"]
for word in words_to_analyze:
    if word in domain_word2vec.wv:
        print(f"Vector embedding for
'{word}':\n{domain_word2vec.wv[word]}\n")
    else:
        print(f"Word '{word}' not found in the Word2Vec model.")
```

Output :

```
Vector embedding for 'court':

[-0.00520213  0.05436571  0.0196009   0.00766893  0.04851889 -0.22194375

  0.15068555  0.2671535  -0.16717364 -0.04062838 -0.054865   -0.17729442

 -0.06285486  0.16066416  0.00799252  0.00430546 -0.04130681 -0.11852198

 -0.11586928 -0.32001996  0.07377547  0.00634967  0.01555517 -0.04018658

 -0.05180506 -0.06574838  0.01809591 -0.04998898 -0.05094941  0.00987862

  0.17092119 -0.03111312  0.12419216 -0.07877786 -0.07952873  0.22328345

  0.12608306 -0.0951244  -0.07667849 -0.1501351   0.04725789 -0.15457962

 -0.06896634  0.13114625  0.11142956  0.03642106 -0.06946036 -0.02198208

  0.01422113  0.05933676  0.09983439 -0.12603386  0.07056595  0.02597529

 -0.02668819  0.0757888  -0.00033602  0.05289464 -0.16172495  0.12800941

  0.07429419  0.10103885  0.08504409 -0.01794797 -0.06241613  0.14987893

  0.15474467  0.18398537 -0.17408288  0.13962157 -0.11823418  0.09919562

  0.07957372 -0.05181967  0.15559544  0.0681076  -0.0985308   0.02557893

 -0.11090399 -0.02128516 -0.01085772  0.11211726 -0.14611867  0.20995773

 -0.10311343  0.06910679  0.14604773  0.10655196  0.10023539 -0.02284993

  0.14183174  0.13799591  0.00409749  0.11127966  0.21348046  0.03055387

  0.11364785 -0.1445034   0.11242675 -0.04190433]


Vector embedding for 'plaintiff':

[-0.03223411  0.06478627  0.00088969 -0.00806353  0.05694845 -0.21240263

  0.13640128  0.26523107 -0.13281158 -0.04770363 -0.02368818 -0.1402928

 -0.03685566  0.12257947  0.00039671  0.00741028 -0.01043882 -0.11464308

 -0.09540985 -0.3000543   0.0647751   0.00074026  0.00411286 -0.05273201

 -0.02684729 -0.04762366  0.02497391 -0.04300669 -0.04396778 -0.00184753
```

```
  0.14383827 -0.04924785  0.08860843 -0.08550214 -0.06152922  0.24551614
  0.10724474 -0.13455397 -0.05984696 -0.15700217  0.02755019 -0.14089336
 -0.07535081  0.0659988   0.11539416  0.020872   -0.05348673 -0.02727061
  0.01346072  0.03318129  0.09382757 -0.10529419  0.0414049   0.07656677
 -0.01830849  0.07164428  0.01196256  0.05545417 -0.13542365  0.1291954
  0.08052401  0.06550701  0.09594982 -0.03788032 -0.07346537  0.16846505
  0.13681169  0.14530386 -0.15170906  0.14640196 -0.09068518  0.0789521
  0.05557212 -0.02400086  0.11684093  0.06631403 -0.11164055  0.01440321
 -0.10535935 -0.00458972 -0.02664629  0.1090111  -0.12968238  0.18052402
 -0.09392222  0.08443088  0.12474449  0.09482376  0.11001488 -0.01367659
  0.12273199  0.1101999   0.02236929  0.09491293  0.19617565  0.01282949
  0.11568122 -0.1593218   0.10664962 -0.04113806]
Vector embedding for 'doctor':
[-3.76006439e-02  8.11468363e-02 -1.18198330e-02  1.22082625e-02
  5.35595044e-03 -2.21441105e-01  1.31108329e-01  3.10447901e-01
 -2.11071640e-01  7.52886664e-03 -6.67306557e-02 -1.76628768e-01
 -4.83631082e-02  1.88437983e-01 -2.80619003e-02  3.20329741e-02
 -2.19840016e-02 -1.36392176e-01 -1.02166705e-01 -3.58890593e-01
  4.39012572e-02  4.81801666e-03  1.11632412e-02 -6.98464885e-02
 -4.50425185e-02 -4.01994735e-02 -6.03534980e-04 -7.15099052e-02
 -7.36634061e-02  2.14629583e-02  2.10165456e-01 -6.25279024e-02
  1.19931854e-01 -1.26935437e-01 -8.21741298e-02  2.74210095e-01
  9.49538499e-02 -1.17289513e-01 -9.49264839e-02 -1.75545543e-01
  3.37264240e-02 -2.08480164e-01 -8.98559391e-02  1.35834515e-01
  1.21459514e-01  5.26671447e-02 -7.85357356e-02 -1.38883330e-02
  3.44770006e-03  5.95685691e-02  1.30519092e-01 -1.28386602e-01
  9.01534930e-02  7.31256530e-02 -1.94634255e-02  1.17376871e-01
 1.67697188e-04  4.33479100e-02 -1.57258630e-01  1.38467610e-01
 8.46170783e-02  7.77027458e-02  8.34437460e-02 -2.43678018e-02
-8.29226896e-02  1.89361051e-01  1.67503580e-01  2.07188442e-01
-1.92358971e-01  1.90954044e-01 -8.66395757e-02  8.63512680e-02
 8.16990361e-02 -2.30716318e-02  1.48350254e-01  9.33871120e-02
-1.03444301e-01  3.32759172e-02 -1.03499167e-01  2.95007881e-02
-4.18480560e-02  1.48850128e-01 -1.25358477e-01  2.33333096e-01
-1.20942295e-01  1.06142171e-01  1.28692985e-01  1.23203449e-01
 1.00113675e-01 -1.41250789e-02  1.63177848e-01  1.50014937e-01
-1.95683893e-02  1.19940504e-01  2.54336447e-01  2.12510210e-02
 1.35626718e-01 -1.89367294e-01  1.02768317e-01 -7.30541497e-02]

Vector embedding for 'patient':
[ 0.00135616  0.06625096  0.02714886 -0.03324671  0.05406597 -0.2076351
  0.14450136  0.27830392 -0.1474757  -0.05214735 -0.02860676 -0.218962
 -0.05803476  0.11022121 -0.03196976  0.0245685   0.0070367  -0.12605277
 -0.11396559 -0.3183468   0.07659787  0.01132763  0.00593386 -0.04407553
 -0.05708291 -0.05022431  0.03657781 -0.05108569 -0.0220301   0.00680075
  0.14817646 -0.03874053  0.13069744 -0.11300313 -0.10196024  0.2306353
  0.13352849 -0.12474146 -0.07811124 -0.14196448  0.03165774 -0.15317255
 -0.04029788  0.10843351  0.11978162  0.03644174 -0.07184896 -0.00125591
  0.01996329  0.04686815  0.12031849 -0.13361286  0.07784432  0.03898075
 -0.05535794  0.07788541  0.02375661  0.06319185 -0.13593689  0.13807625
  0.04011758  0.07736681  0.10920981 -0.01097703 -0.08413535  0.1694132
  0.1142689   0.17812304 -0.16391632  0.13841556 -0.08013699  0.09719803
  0.07872047 -0.04311903  0.14359443  0.06323478 -0.05998136  0.03068179
 -0.10644887  0.00854869 -0.04508544  0.13762434 -0.12336963  0.1855616
 -0.11391655  0.09752344  0.1405091   0.12214459  0.11253129 -0.01929942
  0.13898279  0.15566415  0.01292162  0.08838749  0.19901091  0.03416261
  0.12509196 -0.13636002  0.11566975 -0.02010318]

Vector embedding for 'guilty':
[-0.01413389  0.06656995 -0.00734866 -0.03095385  0.06509437 -0.2517697
  0.14954449  0.29895368 -0.15728544 -0.07182206 -0.06310162 -0.20050046
 -0.08547995  0.15693647 -0.0186175   0.01778842 -0.05446635 -0.12549472
 -0.11124176 -0.31952748  0.03580405  0.01365704  0.03395955 -0.03605738
 -0.06030127 -0.04814158  0.03859452 -0.09555041 -0.05513439  0.0372526
  0.19865839 -0.07835107  0.10888778 -0.11142128 -0.10577497  0.29005775
  0.11180676 -0.13126965 -0.07538164 -0.1596524   0.06402622 -0.17310387
 -0.09087672  0.04137763  0.09426072  0.02597058 -0.06627226 -0.02641308
  0.03379544  0.0561525   0.13159601 -0.16362782  0.08867155  0.10736878
 -0.04391972  0.10295371  0.04891674  0.00565069 -0.163432    0.08589575
  0.1108232   0.05997586  0.11241774 -0.04420831 -0.06642649  0.15975468
  0.1490166   0.12801382 -0.21193038  0.1502985  -0.10489336  0.09517636
  0.0673286  -0.03900745  0.15302955  0.0800889  -0.13577344  0.05731111
 -0.12092727  0.00424497 -0.00455176  0.11054221 -0.15298396  0.20722686
 -0.15278348  0.03610937  0.10936919  0.14354476  0.09363212 -0.00813364
  0.1714467   0.15730394 -0.02156785  0.11239511  0.24912179  0.03659537
  0.0892475  -0.202413    0.11249497 -0.05155509]

Vector embedding for 'surgery':
[-3.12990844e-02  6.58327192e-02  2.85430159e-03  1.10345073e-02
 -7.04743201e-03 -2.36223593e-01  1.33402810e-01  3.03116202e-01
 -2.05681935e-01  6.48758421e-03 -8.28733593e-02 -1.69779241e-01
```

```
-5.81854694e-02  1.80510432e-01 -4.00698669e-02  3.47116366e-02
-1.62971541e-02 -1.29537463e-01 -9.92213637e-02 -3.68670791e-01
 4.55319285e-02  8.06765445e-03 -1.78291200e-04 -6.00495152e-02
-5.73267005e-02 -4.28762138e-02 -3.84912407e-03 -6.40033185e-02
-7.08072856e-02  4.36537573e-03  2.26468816e-01 -4.98397388e-02
 1.30335823e-01 -1.16139121e-01 -8.42535719e-02  2.86336660e-01
 1.00505255e-01 -1.20256521e-01 -9.17292535e-02 -1.76113561e-01
 2.96843071e-02 -2.00398415e-01 -9.28441510e-02  1.45912632e-01
 1.11865871e-01  5.49624115e-02 -6.89490139e-02 -1.83873083e-02
-1.00601949e-02  6.59109801e-02  1.25353217e-01 -1.26397550e-01
 9.62558836e-02  5.71697466e-02 -2.06405111e-02  1.16529934e-01
-8.17977940e-04  2.92389747e-02 -1.62125885e-01  1.34710684e-01
 6.75722361e-02  8.40188041e-02  8.42126012e-02 -1.94504112e-02
-1.00880139e-01  1.89215228e-01  1.60290688e-01  2.12331533e-01
-2.03707144e-01  2.01542258e-01 -9.25249755e-02  9.14819315e-02
 8.59961137e-02 -2.71495730e-02  1.61703631e-01  9.22792554e-02
-1.11497119e-01  5.09562343e-02 -1.00743666e-01  3.40460427e-02
-5.15895225e-02  1.68939248e-01 -1.28210068e-01  2.49226272e-01
-1.33621320e-01  1.16187118e-01  1.42963469e-01  1.47219375e-01
 1.09663606e-01  5.80039807e-03  1.60661057e-01  1.45263568e-01
-1.83158442e-02  1.16535008e-01  2.47885883e-01  1.26237087e-02
 1.36337191e-01 -1.75651938e-01  1.01963326e-01 -7.20273107e-02]
```

```
# Visualization using PCA
selected_words = ["court", "plaintiff", "defendant", "guilty", "jury",
                  "patient", "doctor", "hospital", "surgery",
"emergency"]
word_vectors = [domain_word2vec.wv[word] for word in selected_words]

word_vectors
```

Output:

```
[array([-0.00520213,  0.05436571,  0.0196009 ,  0.00766893,  0.04851889,
        -0.22194375,  0.15068555,  0.2671535 , -0.16717364, -0.04062838,
        -0.054865  , -0.17729442, -0.06285486,  0.16066416,  0.00799252,
         0.00430546, -0.04130681, -0.11852198, -0.11586928, -0.32001996,
         0.07377547,  0.00634967,  0.01555517, -0.04018658, -0.05180506,
        -0.06574838,  0.01809591, -0.04998898, -0.05094941,  0.00987862,
         0.17092119, -0.03111312,  0.12419216, -0.07877786, -0.07952873,
         0.22328345,  0.12608306, -0.0951244 , -0.07667849, -0.1501351 ,
         0.04725789, -0.15457962, -0.06896634,  0.13114625,  0.11142956,
         0.03642106, -0.06946036, -0.02198208,  0.01422113,  0.05933676,
         0.09983439, -0.12603386,  0.07056595,  0.02597529, -0.02668819,
         0.0757888 , -0.00033602,  0.05289464, -0.16172495,  0.12800941,
         0.07429419,  0.10103885,  0.08504409, -0.01794797, -0.06241613,
         0.14987893,  0.15474467,  0.18398537, -0.17408288,  0.13962157,
        -0.11823418,  0.09919562,  0.07957372, -0.05181967,  0.15559544,
         0.0681076 , -0.0985308 ,  0.02557893, -0.11090399, -0.02128516,
        -0.01085772,  0.11211726, -0.14611867,  0.20995773, -0.10311343,
         0.06910679,  0.14604773,  0.10655196,  0.10023539, -0.02284993,
         0.14183174,  0.13799591,  0.00409749,  0.11127966,  0.21348046,
         0.03055387,  0.11364785, -0.1445034 ,  0.11242675, -0.04190433],
       dtype=float32),
 array([-0.03223411,  0.06478627,  0.00088969, -0.00806353,  0.05694845,
        -0.21240263,  0.13640128,  0.26523107, -0.13281158, -0.04770363,
        -0.02368818, -0.1402928 , -0.03685566,  0.12257947,  0.00039671,
         0.00741028, -0.01043882, -0.11464308, -0.09540985, -0.3000543 ,
         0.0647751 ,  0.00074026,  0.00411286, -0.05273201, -0.02684729,
        -0.04762366,  0.02497391, -0.04300669, -0.04396778, -0.00184753,
         0.14383827, -0.04924785,  0.08860843, -0.08550214, -0.06152922,
         0.24551614,  0.10724474, -0.13455397, -0.05984696, -0.15700217,
         0.02755019, -0.14089336, -0.07535081,  0.0659988 ,  0.11539416,
         0.020872  , -0.05348673, -0.02727061,  0.01346072,  0.03318129,
         0.09382757, -0.10529419,  0.0414049 ,  0.07656677, -0.01830849,
         0.07164428,  0.01196256,  0.05545417, -0.13542365,  0.1291954 ,
         0.08052401,  0.06550701,  0.09594982, -0.03788032, -0.07346537,
         0.16846505,  0.13681169,  0.14530386, -0.15170906,  0.14640196,
        -0.09068518,  0.0789521 ,  0.05557212, -0.02400086,  0.11684093,
         0.06631403, -0.11164055,  0.01440321, -0.10535935, -0.00458972,
        -0.02664629,  0.1090111 , -0.12968238,  0.18052402, -0.09392222,
         0.08443088,  0.12474449,  0.09482376,  0.11001488, -0.01367659,
         0.12273199,  0.1101999 ,  0.02236929,  0.09491293,  0.19617565,
         0.01282949,  0.11568122, -0.1593218 ,  0.10664962, -0.04113806],
       dtype=float32),
 array([ 0.00656709,  0.07256435, -0.0084228 , -0.02586134,  0.07641555,
        -0.2732658 ,  0.1540303 ,  0.32865882, -0.17496191, -0.06661771,
        -0.06085587, -0.22411431, -0.08474998,  0.18789086, -0.02127556,
         0.03096173, -0.05577651, -0.12937057, -0.11135948, -0.36175218,
         0.04432205,  0.00878906,  0.02296932, -0.05328603, -0.07712711,
        -0.06075291,  0.04381331, -0.10575836, -0.06409874,  0.04152325,
         0.21431115, -0.08531993,  0.14578514, -0.11424538, -0.11725931,
         0.29418284,  0.10676998, -0.15401532, -0.09160217, -0.16645099,
         0.06118093, -0.19756706, -0.08612581,  0.06556768,  0.1085471 ,
         0.04415575, -0.06776308, -0.04802901,  0.04284215,  0.0638606 ,
         0.13356939, -0.17036071,  0.08767819,  0.10464148, -0.03167466,
         0.11624619,  0.03233933,  0.01407737, -0.15678538,  0.10963659,
         0.11469187,  0.07420997,  0.09665452, -0.03110271, -0.07621247,
         0.17188032,  0.18252161,  0.14339091, -0.22514871,  0.18041831,
        -0.12061799,  0.07872933,  0.06301736, -0.04593184,  0.16801536,
         0.08114434, -0.13756713,  0.06104114, -0.13863237,  0.01738747,
        -0.01658883,  0.13528904, -0.1735411 ,  0.24808215, -0.17541201,
         0.04516907,  0.11772847,  0.14438275,  0.12152614, -0.00914207,
         0.16980448,  0.15530077, -0.0296784 ,  0.13635741,  0.24644977,
         0.03516944,  0.11169897, -0.21215999,  0.10724142, -0.03329436],
       dtype=float32),
  array([-0.01413389,  0.06656995, -0.00734866, -0.03095385,  0.06509437,
        -0.2517697 ,  0.14954449,  0.29895368, -0.15728544, -0.07182206,
        -0.06310162, -0.20050046, -0.08547995,  0.15693647, -0.0186175 ,
         0.01778842, -0.05446635, -0.12549472, -0.11124176, -0.31952748,
         0.03580405,  0.01365704,  0.03395955, -0.03605738, -0.06030127,
        -0.04814158,  0.03859452, -0.09555041, -0.05513439,  0.0372526 ,
         0.19865839, -0.07835107,  0.10888778, -0.11142128, -0.10577497,
         0.29005775,  0.11180676, -0.13126965, -0.07538164, -0.1596524 ,
         0.06402622, -0.17310387, -0.09087672,  0.04137763,  0.09426072,
         0.02597058, -0.06627226, -0.02641308,  0.03379544,  0.0561525 ,
         0.13159601, -0.16362782,  0.08867155,  0.10736878, -0.04391972,
         0.10295371,  0.04891674,  0.00565069, -0.163432  ,  0.08589575,
         0.1108232 ,  0.05997586,  0.11241774, -0.04420831, -0.06642649,
         0.15975468,  0.1490166 ,  0.12801382, -0.21193038,  0.1502985 ,
        -0.10489336,  0.09517636,  0.0673286 , -0.03900745,  0.15302955,
         0.0800889 , -0.13577344,  0.05731111, -0.12092727,  0.00424497,
        -0.00455176,  0.11054221, -0.15298396,  0.20722686, -0.15278348,
         0.03610937,  0.10936919,  0.14354476,  0.09363212, -0.00813364,
         0.1714467 ,  0.15730394, -0.02156785,  0.11239511,  0.24912179,
         0.03659537,  0.0892475 , -0.202413  ,  0.11249497, -0.05155509],
       dtype=float32),
  array([-0.00793692,  0.04061861, -0.01272589, -0.0216382 ,  0.05832693,
        -0.20959468,  0.1335544 ,  0.23678838, -0.1236183 , -0.03556946,
        -0.02290245, -0.1449683 , -0.06467045,  0.1215817 , -0.01873406,
         0.01126286, -0.01548086, -0.10774158, -0.09506714, -0.26566857,
         0.04896098, -0.00084279,  0.01825006, -0.05337542, -0.03144738,
        -0.05701503,  0.03505556, -0.05904163, -0.04336127, -0.00061382,
         0.16488056, -0.05326047,  0.09773479, -0.07977082, -0.06476859,
         0.22621374,  0.09030687, -0.11633091, -0.06397521, -0.13930038,
         0.04200654, -0.13733749, -0.06120953,  0.06062739,  0.0891199 ,
         0.02058195, -0.06635275, -0.00753717,  0.01779128,  0.05422449,
         0.10981765, -0.1124575 ,  0.06291748,  0.08702539, -0.03922568,
         0.08803029,  0.03010272,  0.03673965, -0.13025954,  0.10378475,
         0.07686505,  0.06131725,  0.09677017, -0.02306653, -0.05867948,
         0.14983074,  0.11522046,  0.13194208, -0.16845842,  0.130707  ,
        -0.09688476,  0.0888531 ,  0.05799359, -0.03768088,  0.1355294 ,
         0.04766061, -0.1006932 ,  0.02873053, -0.10280664, -0.01355723,
        -0.0322897 ,  0.125441  , -0.13379173,  0.1888993 , -0.1090064 ,
         0.05351871,  0.11118538,  0.09934786,  0.08427987, -0.01916844,
         0.11553331,  0.12629525, -0.00194136,  0.09477089,  0.19319633,
         0.01517526,  0.08618706, -0.15499583,  0.09854038, -0.04697568],
       dtype=float32),
  array([ 0.00135616,  0.06625096,  0.02714886, -0.03324671,  0.05406597,
        -0.2076351 ,  0.14450136,  0.27830392, -0.1474757 , -0.05214735,
        -0.02860676, -0.218962  , -0.05803476,  0.11022121, -0.03196976,
         0.0245685 ,  0.0070367 , -0.12605277, -0.11396559, -0.3183468 ,
         0.07659787,  0.01132763,  0.00593386, -0.04407553, -0.05708291,
        -0.05022431,  0.03657781, -0.05108569, -0.0220301 ,  0.00680075,
         0.14817646, -0.03874053,  0.13069744, -0.11300313, -0.10196024,
         0.2306353 ,  0.13352849, -0.12474146, -0.07811124, -0.14196448,
         0.03165774, -0.15317255, -0.04029788,  0.10843351,  0.11978162,
         0.03644174, -0.07184896, -0.00125591,  0.01996329,  0.04686815,
         0.12031849, -0.13361286,  0.07784432,  0.03898075, -0.05535794,
         0.07788541,  0.02375661,  0.06319185, -0.13593689,  0.13807625,
         0.04011758,  0.07736681,  0.10920981, -0.01097703, -0.08413535,
         0.1694132 ,  0.1142689 ,  0.17812304, -0.16391632,  0.13841556,
        -0.08013699,  0.09719803,  0.07872047, -0.04311903,  0.14359443,
         0.06323478, -0.05998136,  0.03068179, -0.10644887,  0.00854869,
        -0.04508544,  0.13762434, -0.12336963,  0.1855616 , -0.11391655,
         0.09752344,  0.1405091 ,  0.12214459,  0.11253129, -0.01929942,
         0.13898279,  0.15566415,  0.01292162,  0.08838749,  0.19901091,
         0.03416261,  0.12509196, -0.13636002,  0.11566975, -0.02010318],
       dtype=float32),
  array([-3.76006439e-02,  8.11468363e-02, -1.18198330e-02,  1.22082625e-02,
         5.35595044e-03, -2.21441105e-01,  1.31108329e-01,  3.10447901e-01,
        -2.11071640e-01,  7.52886664e-03, -6.67306557e-02, -1.76628768e-01,
        -4.83631082e-02,  1.88437983e-01, -2.80619003e-02,  3.20329741e-02,
        -2.19840016e-02, -1.36392176e-01, -1.02166705e-01, -3.58890593e-01,
         4.39012572e-02,  4.81801666e-03,  1.11632412e-02, -6.98464885e-02,
        -4.50425185e-02, -4.01994735e-02, -6.03534980e-04, -7.15099052e-02,
        -7.36634061e-02,  2.14629583e-02,  2.10165456e-01, -6.25279024e-02,
         1.19931854e-01, -1.26935437e-01, -8.21741298e-02,  2.74210095e-01,
         9.49538499e-02, -1.17289513e-01, -9.49264839e-02, -1.75545543e-01,
         3.37264240e-02, -2.08480164e-01, -8.98559391e-02,  1.35834515e-01,
         1.21459514e-01,  5.26671447e-02, -7.85357356e-02, -1.38883330e-02,
         3.44770006e-03,  5.95685691e-02,  1.30519092e-01, -1.28386602e-01,
         9.01534930e-02,  7.31256530e-02, -1.94634255e-02,  1.17376871e-01,
         1.67697188e-04,  4.33479100e-02, -1.57258630e-01,  1.38467610e-01,
         8.46170783e-02,  7.77027458e-02,  8.34437460e-02, -2.43678018e-02,
        -8.29226896e-02,  1.89361051e-01,  1.67503580e-01,  2.07188442e-01,
        -1.92358971e-01,  1.90954044e-01, -8.66395757e-02,  8.63512680e-02,
         8.16990361e-02, -2.30716318e-02,  1.48350254e-01,  9.33871120e-02,
        -1.03444301e-01,  3.32759172e-02, -1.03499167e-01,  2.95007881e-02,
        -4.18480560e-02,  1.48850128e-01, -1.25358477e-01,  2.33333096e-01,
        -1.20942295e-01,  1.06142171e-01,  1.28692985e-01,  1.23203449e-01,
         1.00113675e-01, -1.41250789e-02,  1.63177848e-01,  1.50014937e-01,
        -1.95683893e-02,  1.19940504e-01,  2.54336447e-01,  2.12510210e-02,
         1.35626718e-01, -1.89367294e-01,  1.02768317e-01, -7.30541497e-02],
       dtype=float32),
  array([-0.02265889,  0.05778723, -0.01179005, -0.00284239,  0.04882376,
        -0.23377152,  0.13176689,  0.3085964 , -0.16563822, -0.00594464,
        -0.06069683, -0.16251391, -0.04316762,  0.19339406, -0.00984798,
         0.00349556, -0.03423214, -0.15440044, -0.15881209, -0.36713216,
         0.0536154 , -0.02835809, -0.01769544, -0.04901092, -0.05845137,
        -0.06207271,  0.01982044, -0.07527879, -0.0646024 ,  0.04246312,
         0.17291646, -0.03934861,  0.15174052, -0.11018316, -0.1073219 ,
         0.25728288,  0.10440008, -0.15727909, -0.0763182 , -0.1741864 ,
          0.02225032, -0.2151592 , -0.09898599,  0.08515406,  0.14156315,
          0.04146844, -0.080073  , -0.02897993,  0.03221606,  0.05149914,
          0.13052906, -0.13760065,  0.0776631 ,  0.07678478, -0.00785946,
          0.10275181, -0.01308092,  0.06987558, -0.16492371,  0.15031946,
          0.07514932,  0.06996216,  0.08479748,  0.00843247, -0.08604642,
          0.19633524,  0.18013164,  0.14563482, -0.19318359,  0.18252854,
         -0.10180707,  0.07443867,  0.04677813, -0.07124216,  0.16272344,
          0.05405647, -0.0953992 ,  0.04183496, -0.10839124, -0.01641163,
         -0.00933946,  0.11554954, -0.13658553,  0.22137882, -0.13358647,
          0.06070266,  0.13194123,  0.11046906,  0.12901476, -0.01263191,
          0.16108233,  0.17001428,  0.02732584,  0.10106397,  0.21696223,
          0.00993073,  0.13067529, -0.19392748,  0.11318995, -0.02350748],
        dtype=float32),
  array([-3.12990844e-02,  6.58327192e-02,  2.85430159e-03,  1.10345073e-02,
         -7.04743201e-03, -2.36223593e-01,  1.33402810e-01,  3.03116202e-01,
         -2.05681935e-01,  6.48758421e-03, -8.28733593e-02, -1.69779241e-01,
         -5.81854694e-02,  1.80510432e-01, -4.00698669e-02,  3.47116366e-02,
         -1.62971541e-02, -1.29537463e-01, -9.92213637e-02, -3.68670791e-01,
          4.55319285e-02,  8.06765445e-03, -1.78291200e-04, -6.00495152e-02,
         -5.73267005e-02, -4.28762138e-02, -3.84912407e-03, -6.40033185e-02,
         -7.08072856e-02,  4.36537573e-03,  2.26468816e-01, -4.98397388e-02,
          1.30335823e-01, -1.16139121e-01, -8.42535719e-02,  2.86336660e-01,
          1.00505255e-01, -1.20256521e-01, -9.17292535e-02, -1.76113561e-01,
          2.96843071e-02, -2.00398415e-01, -9.28441510e-02,  1.45912632e-01,
          1.11865871e-01,  5.49624115e-02, -6.89490139e-02, -1.83873083e-02,
         -1.00601949e-02,  6.59109801e-02,  1.25353217e-01, -1.26397550e-01,
          9.62558836e-02,  5.71697466e-02, -2.06405111e-02,  1.16529934e-01,
        -8.17977940e-04,  2.92389747e-02, -1.62125885e-01,  1.34710684e-01,
         6.75722361e-02,  8.40188041e-02,  8.42126012e-02, -1.94504112e-02,
        -1.00880139e-01,  1.89215228e-01,  1.60290688e-01,  2.12331533e-01,
        -2.03707144e-01,  2.01542258e-01, -9.25249755e-02,  9.14819315e-02,
         8.59961137e-02, -2.71495730e-02,  1.61703631e-01,  9.22792554e-02,
        -1.11497119e-01,  5.09562343e-02, -1.00743666e-01,  3.40460427e-02,
        -5.15895225e-02,  1.68939248e-01, -1.28210068e-01,  2.49226272e-01,
        -1.33621320e-01,  1.16187118e-01,  1.42963469e-01,  1.47219375e-01,
         1.09663606e-01,  5.80039807e-03,  1.60661057e-01,  1.45263568e-01,
        -1.83158442e-02,  1.16535008e-01,  2.47885883e-01,  1.26237087e-02,
         1.36337191e-01, -1.75651938e-01,  1.01963326e-01, -7.20273107e-02],
       dtype=float32),
  array([-0.01135001,  0.05962004,  0.03272124, -0.01768819,  0.04998965,
        -0.20870571,  0.14685056,  0.2836007 , -0.18323691, -0.03109219,
        -0.03263121, -0.22973996, -0.04514541,  0.15115191, -0.02573079,
         0.00774679, -0.00233633, -0.1304845 , -0.14386515, -0.31636477,
         0.06311441,  0.00088838, -0.02058102, -0.03525062, -0.05741948,
        -0.08361816,  0.04948161, -0.06476792, -0.02724299,  0.00202177,
         0.18204965, -0.03211843,  0.15414716, -0.11622933, -0.12767726,
         0.25150353,  0.13327569, -0.16982682, -0.09414072, -0.16327456,
         0.01518478, -0.16644533, -0.0390787 ,  0.10758833,  0.1320721 ,
         0.01700985, -0.06482255, -0.01300713,  0.01951688,  0.04992113,
         0.12908201, -0.15448081,  0.05776305,  0.0506245 , -0.04097727,
         0.08827188,  0.02562736,  0.04502805, -0.13268901,  0.15970598,
         0.0415643 ,  0.10894849,  0.1138011 , -0.03124123, -0.09126465,
         0.17592564,  0.11290699,  0.18183088, -0.17974737,  0.15896654,
        -0.07602367,  0.10534283,  0.06435065, -0.05782789,  0.17650633,
   0.06925058, -0.07527371,  0.04818593, -0.11926682, -0.00753901,

  -0.03426384,  0.13999003, -0.1504484 ,  0.19165526, -0.14357014,

   0.09853069,  0.1328372 ,  0.13577645,  0.12897931, -0.0323601 ,

   0.11762244,  0.17293827, -0.00854158,  0.07778574,  0.22550419,

   0.03845395,  0.1585336 , -0.14676552,  0.14424598, -0.03719835],

  dtype=float32)]
```

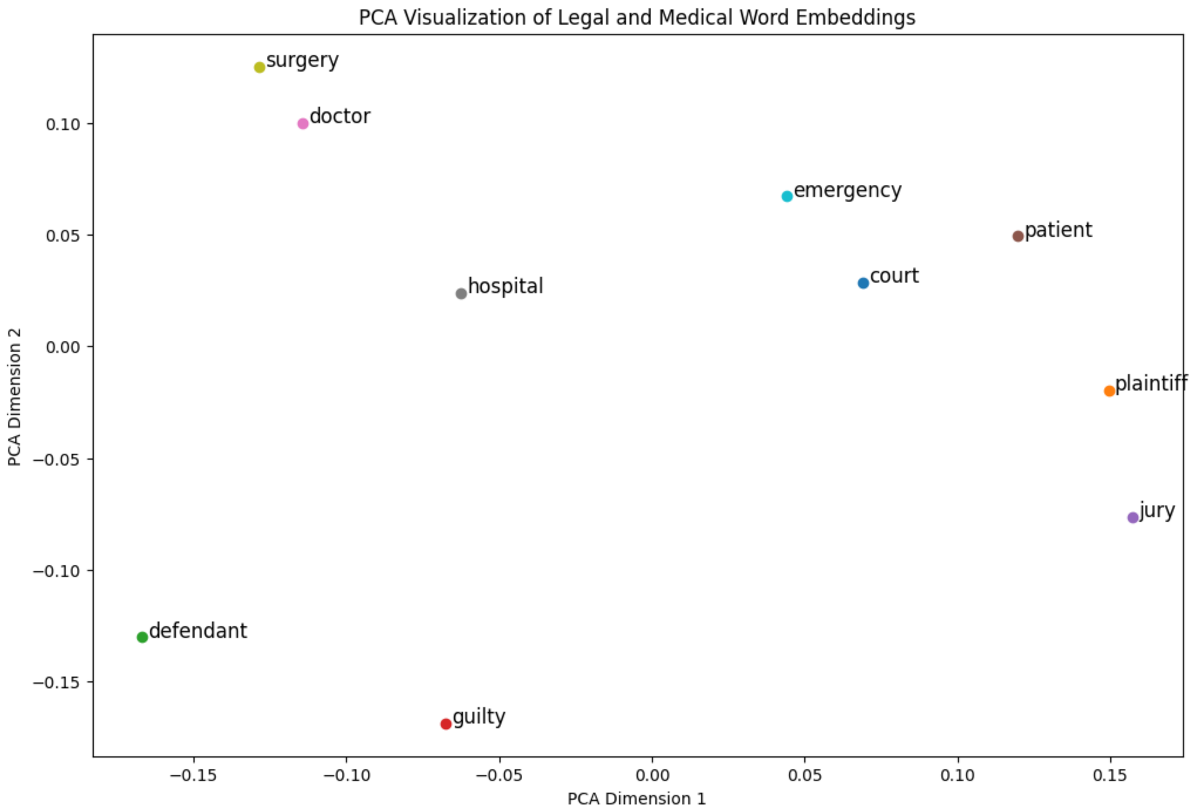
```python
pca = PCA(n_components=2)
reduced_vectors = pca.fit_transform(word_vectors)

reduced_vectors
```

```
array([[ 0.06908131,  0.0287464 ],
       [ 0.14953919, -0.01964696],
       [-0.16693931, -0.13033459],
       [-0.0674262 , -0.16882876],
       [ 0.1574409 , -0.07644414],
       [ 0.11961383,  0.04956438],
       [-0.1142199 ,  0.10011148],
       [-0.06259862,  0.0240761 ],
       [-0.12848931,  0.12535115],
       [ 0.04399811,  0.06740492]])
```

```python
plt.figure(figsize=(12, 8))
for i, word in enumerate(selected_words):
    plt.scatter(reduced_vectors[i, 0], reduced_vectors[i, 1])
    plt.text(reduced_vectors[i, 0] + 0.002, reduced_vectors[i, 1],
word, fontsize=12)
plt.title("PCA Visualization of Legal and Medical Word Embeddings")
plt.xlabel("PCA Dimension 1")
plt.ylabel("PCA Dimension 2")
plt.show()
```

Output :


PCA Visualization of Legal and Medical Word Embeddings
surgery
doctor
emergency
patient
court
hospital
plaintiff
jury
defendant
guilty
PCA Dimension 2
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
PCA Dimension 1