Example 1:

PaperID = 1887

Title: 'Breast cancer cell-derived exosomes and macrophage polarization are associated with lymph node metastasis.'

Step 1: Stopword removal and lower case transformation

'breast cancer cell derived exosomes macrophage polarization associated lymph node metastasis'

Step 2: Single words, Biwords, triwords generation and Assigning numbering to generated single words, biwords and triwords.

['breast', 'cancer', 'cell', 'derived', 'exosomes', 'macrophage', 'polarization', 'associated', 'lymph', 'node', 'metastasis', 'breast cancer', 'cancer cell', 'cell derived', 'derived exosomes', 'exosomes macrophage', 'macrophage polarization', 'polarization associated', 'associated lymph', 'lymph node', 'node metastasis', 'breast cancer cell', 'cancer cell derived', 'cell derived exosomes', 'derived exosomes macrophage', 'exosomes macrophage polarization', 'macrophage polarization associated', 'polarization associated lymph', 'associated lymph node', 'lymph node metastasis']	[[179], [189], [252], [488], [753], [1349], [1728], [96], [1340], [1612], [1410], [180], [193], [258], [497], [799], [1354], [1729], [99], [1341], [1613], [181], [194], [259], [503], [800], [1355], [1730], [100], [1342]]

Step 3: Searching Keywords in above created table

['cancer', 'lymph node', 'exosomes', 'breast cancer', 'metastasis', 'macrophage']	[189, 1341, 753, 180, 1410, 1349]

Step 4 : Searching words which are not keywords in table created in step 2.

['lymph node metastasis', 'exosomes macrophage', 'associated', 'node', 'node metastasis', 'cell derived', 'derived', 'associated lymph', 'polarization associated lymph', 'polarization associated', 'exosomes macrophage polarization', 'derived exosomes', 'associated lymph node', 'breast', 'cell derived exosomes', 'breast cancer cell', 'cancer cell derived', 'polarization', 'cell', 'lymph', 'derived exosomes macrophage', 'cancer cell', 'macrophage polarization', 'macrophage polarization associated']	[1342, 799, 96, 1612, 1613, 258, 488, 99, 1730, 1729, 800, 497, 100, 179, 259, 181, 194, 1728, 252, 1340, 503, 193, 1354, 1355]

Step 5: Create following (179 x 3) dataframe or table representing keywords by 2 and non-keyword words by 1.

Row Number	Paper Index	Word ID	Keywords
1.	1887	179	1
2.	1887	189	2
3.	1887	252	1
4.	1887	488	1
5.	1887	753	2
6.	1887	1349	2
7.	1887	1728	1
8.	1887	96	1
9.	1887	1340	1
10.	1887	1612	1
11.	1887	1410	2
12.	1887	180	2
13.	1887	193	1
14.	1887	258	1
15.	1887	497	1
16.	1887	799	1
17.	1887	1354	1
18.	1887	1729	1
19.	1887	99	1
20.	1887	1341	2
21.	1887	1613	1
22.	1887	181	1
23.	1887	194	1
24.	1887	259	1
25.	1887	503	1
26.	1887	800	1
27.	1887	1355	1
28.	1887	1730	1
29.	1887	100	1
30.	1887	1342	1
31.	1887	753	2
32.	1887	488	1
33.	1887	753	2
34.	1887	1728	1
35.	1887	753	2
36.	1887	252	1
37.	1887	753	2
38.	1887	488	1
39.	1887	189	2
40.	1887	252	1
41.	1887	488	1
42.	1887	753	2

43.	1887	193	1
44.	1887	258	1
45.	1887	497	1
46.	1887	194	1
47.	1887	259	1
48.	1887	753	2
49.	1887	488	1
50.	1887	753	2
51.	1887	1349	2
52.	1887	252	1
53.	1887	1728	1
54.	1887	497	1
55.	1887	753	2
56.	1887	488	1
57.	1887	753	2
58.	1887	497	1
59.	1887	252	1
60.	1887	488	1
61.	1887	753	2
62.	1887	258	1
63.	1887	497	1
64.	1887	259	1
65.	1887	488	1
66.	1887	753	2
67.	1887	1410	2
68.	1887	96	1
69.	1887	488	1
70.	1887	753	2
71.	1887	497	1
72.	1887	189	2
73.	1887	488	1
74.	1887	252	1
75.	1887	753	2
76.	1887	753	2
77.	1887	1728	1
78.	1887	189	2
79.	1887	1349	2
80.	1887	753	2
81.	1887	488	1
82.	1887	753	2
83.	1887	252	1
84.	1887	753	2
85.	1887	488	1
86.	1887	753	2
87.	1887	488	1
88.	1887	753	2

89.	1887	252	1
90.	1887	753	2
91.	1887	753	2
92.	1887	189	2
93.	1887	753	2
94.	1887	488	1
95.	1887	753	2
96.	1887	753	2
97.	1887	252	1
98.	1887	96	1
99.	1887	488	1
100.	1887	753	2
101.	1887	189	2
102.	1887	497	1
103.	1887	189	2
104.	1887	488	1
105.	1887	753	2
106.	1887	252	1
107.	1887	497	1
108.	1887	252	1
109.	1887	753	2
110.	1887	252	1
111.	1887	189	2
112.	1887	753	2
113.	1887	753	2
114.	1887	488	1
115.	1887	252	1
116.	1887	753	2
117.	1887	488	1
118.	1887	189	2
119.	1887	488	1
120.	1887	753	2
121.	1887	497	1
122.	1887	189	2
123.	1887	252	1
124.	1887	193	1
125.	1887	488	1
126.	1887	753	2
127.	1887	497	1
128.	1887	753	2
129.	1887	488	1
130.	1887	252	1
131.	1887	753	2
132.	1887	189	2
133.	1887	252	1
134.	1887	753	2

135.	1887	189	2
136.	1887	189	2
137.	1887	252	1
138.	1887	193	1
139.	1887	488	1
140.	1887	753	2
141.	1887	189	2
142.	1887	96	1
143.	1887	753	2
144.	1887	189	2
145.	1887	488	1
146.	1887	753	2
147.	1887	252	1
148.	1887	497	1
149.	1887	753	2
150.	1887	488	1
151.	1887	252	1
152.	1887	488	1
153.	1887	753	2
154.	1887	258	1
155.	1887	497	1
156.	1887	259	1
157.	1887	96	1
158.	1887	189	2
159.	1887	252	1
160.	1887	488	1
161.	1887	753	2
162.	1887	258	1
163.	1887	497	1
164.	1887	259	1
165.	1887	252	1
166.	1887	252	1
167.	1887	488	1
168.	1887	252	1
169.	1887	258	1
170.	1887	753	2
171.	1887	189	2
172.	1887	189	2
173.	1887	96	1
174.	1887	753	2
175.	1887	189	2
176.	1887	753	2
177.	1887	189	2
178.	1887	252	1
179.	1887	252	1

Step 6: Use 50% random information for training and 50% for testing from above table.
Please Note that here we are selecting random information from all papers, therefore we are getting 94 rows for training and remaining 85 for testing. 50% randomness makes a selection of 94 rows for training and 85 rows for testing.

Training Data

Row Number	Paper ID	Word ID	Keyword yes(2)/No(1)
1.	1887	488	1
2.	1887	189	2
3.	1887	189	2
4.	1887	753	2
5.	1887	753	2
6.	1887	497	1
7.	1887	753	2
8.	1887	189	2
9.	1887	753	2
10.	1887	194	1
11.	1887	753	2
12.	1887	189	2
13.	1887	753	2
14.	1887	488	1
15.	1887	488	1
16.	1887	488	1
17.	1887	252	1
18.	1887	96	1
19.	1887	497	1
20.	1887	488	1
21.	1887	258	1
22.	1887	180	2
23.	1887	1341	2
24.	1887	753	2
25.	1887	488	1
26.	1887	753	2
27.	1887	1349	2
28.	1887	488	1
29.	1887	1410	2
30.	1887	753	2
31.	1887	252	1
32.	1887	189	2
33.	1887	753	2
34.	1887	96	1
35.	1887	252	1
36.	1887	259	1
37.	1887	753	2

38.	1887	753	2
39.	1887	252	1
40.	1887	252	1
41.	1887	259	1
42.	1887	252	1
43.	1887	488	1
44.	1887	1728	1
45.	1887	1354	1
46.	1887	488	1
47.	1887	252	1
48.	1887	96	1
49.	1887	189	2
50.	1887	1730	1
51.	1887	252	1
52.	1887	753	2
53.	1887	488	1
54.	1887	258	1
55.	1887	1355	1
56.	1887	1728	1
57.	1887	488	1
58.	1887	488	1
59.	1887	189	2
60.	1887	753	2
61.	1887	258	1
62.	1887	753	2
63.	1887	497	1
64.	1887	753	2
65.	1887	497	1
66.	1887	189	2
67.	1887	753	2
68.	1887	252	1
69.	1887	252	1
70.	1887	753	2
71.	1887	252	1
72.	1887	753	2
73.	1887	252	1
74.	1887	1349	2
75.	1887	753	2
76.	1887	189	2
77.	1887	252	1
78.	1887	193	1
79.	1887	488	1
80.	1887	753	2
81.	1887	497	1
82.	1887	497	1
83.	1887	179	1

84.	1887	753	2
85.	1887	100	1
86.	1887	753	2
87.	1887	181	1
88.	1887	189	2
89.	1887	488	1
90.	1887	259	1
91.	1887	753	2
92.	1887	258	1
93.	1887	252	1
94.	1887	800	1

Testing Data

	Paper Index	Word ID	Key Rating
1.	1887	252	1
2.	1887	189	2
3.	1887	753	2
4.	1887	488	1
5.	1887	488	1
6.	1887	753	2
7.	1887	1612	1
8.	1887	193	1
9.	1887	497	1
10.	1887	753	2
11.	1887	753	2
12.	1887	189	2
13.	1887	252	1
14.	1887	497	1
15.	1887	189	2
16.	1887	1728	1
17.	1887	189	2
18.	1887	99	1
19.	1887	252	1
20.	1887	488	1
21.	1887	497	1
22.	1887	259	1
23.	1887	799	1
24.	1887	753	2
25.	1887	189	2
26.	1887	252	1
27.	1887	488	1
28.	1887	753	2
29.	1887	189	2
30.	1887	1349	2

31.	1887	189	2
32.	1887	189	2
33.	1887	488	1
34.	1887	252	1
35.	1887	252	1
36.	1887	488	1
37.	1887	497	1
38.	1887	96	1
39.	1887	497	1
40.	1887	1613	1
41.	1887	488	1
42.	1887	96	1
43.	1887	194	1
44.	1887	753	2
45.	1887	488	1
46.	1887	259	1
47.	1887	497	1
48.	1887	753	2
49.	1887	252	1
50.	1887	753	2
51.	1887	753	2
52.	1887	753	2
53.	1887	1342	1
54.	1887	1728	1
55.	1887	193	1
56.	1887	96	1
57.	1887	497	1
58.	1887	753	2
59.	1887	753	2
60.	1887	488	1
61.	1887	753	2
62.	1887	488	1
63.	1887	258	1
64.	1887	753	2
65.	1887	753	2
66.	1887	488	1
67.	1887	193	1
68.	1887	252	1
69.	1887	1340	1
70.	1887	753	2
71.	1887	753	2
72.	1887	488	1
73.	1887	252	1
74.	1887	503	1
75.	1887	252	1
76.	1887	189	2

77.	1887	753	2
78.	1887	1410	2
79.	1887	189	2
80.	1887	753	2
81.	1887	252	1
82.	1887	1729	1
83.	1887	258	1
84.	1887	488	1
85.	1887	753	2

Step 7: Converting training and testing data into an array with paper in lines and word corpus in columns.

We create the word corpus by adding single words, biwords and triwords from paper titles. And then numbering them in alphabetical ordering. Following is the example of word corpus created. For 100 paper titles there are 2451 words created. These words contains single words, biwords and triwords. Following table shows first 100 words and their numbers. These word number is used in all above steps to represent corresponding words.

1.	accelerating
2.	accelerating cell
3.	accelerating cell cycle
4.	activates
5.	activates latent
6.	activates latent hiv
7.	acute
8.	acute ischaemic
9.	acute ischaemic stroke
10.	acute rejection
11.	acute rejection mouse
12.	adipose
13.	adipose derived
14.	adipose derived exosomes
15.	adipose derived mesenchymal
16.	adipose mesenchymal
17.	adipose mesenchymal stem
18.	advances
19.	advances exosomal
20.	advances exosomal protein
21.	affinity
22.	affinity peptide
23.	affinity peptide vn
24.	age
25.	age related
26.	age related macular
27.	aggregation

28.	aggregation competent
29.	aggregation competent tau
30.	aggressive
31.	aggressive chemoresistant
32.	aggressive chemoresistant ovarian
33.	akt
34.	akt pathway
35.	alloantigens
36.	alloantigens lessens
37.	alloantigens lessens alloreactivity
38.	allogeneic
39.	allogeneic heart
40.	allogeneic heart transplantation
41.	allograft
42.	allograft rejection
43.	allograft tolerance
44.	alloreactivity
45.	alloreactivity recipients
46.	alloreactivity recipients lymphocytes
47.	als
48.	als patients
49.	alter
50.	alter host
51.	alter host cell
52.	altered
53.	altered cerebrospinal
54.	altered cerebrospinal fluid
55.	alzheimer
56.	alzheimer disease
57.	amniotic
58.	amniotic fluid
59.	amniotic fluid exosome
60.	analysis
61.	analysis liquid
62.	analysis liquid biopsy
63.	analysis patients
64.	analysis patients type
65.	anaplastic
66.	anaplastic lymphoma
67.	anaplastic lymphoma kinase
68.	angiogenesis
69.	angiogenesis repressing
70.	angiogenesis repressing novel
71.	angiogenesis vascular
72.	angiogenesis vascular permeability
73.	anti

74.	anti glioma
75.	anti glioma strategy
76.	antiestrogen
77.	antiestrogen drugs
78.	antitumor
79.	antitumor immune
80.	antitumor immune response
81.	apolipoprotein
82.	apolipoprotein e
83.	applications
84.	applications exosomes
85.	applications exosomes cancer
86.	architecture
87.	architecture myocardial
88.	architecture myocardial infarction
89.	arsenite
90.	arsenite transformed
91.	arsenite transformed cells
92.	arthritis
93.	asbestos
94.	asbestos exposed
95.	asbestos exposed cells
96.	asbestos exposure
97.	associated
98.	associated fibroblasts
99.	associated fibroblasts tgf
100.	associated lymph

We then create array where columns are representing these 2451 words and rows are representing 100 papers. If word in column1(C1) is present in paper of row1(R1), then the value of array for [R1 C1] becomes 1 for keywords and 0 for non-keywords; if it is not present then it becomes -1.
Following table shows this concept:

	Word 1	**Word 2**	**. . . . . .**	**Word 2451**
Paper 1	**1** (Keyword)	**0**(Non-keyword)		**-1** (Absent)
Paper 2	**-1** (Absent)	**1** (Keyword)		**-1** (Absent)
Paper 3	**0**(Non-keyword)	**0**(Non-keyword)		**1** (Keyword)
Paper 4	**1** (Keyword)	**-1** (Absent)		**0**(Non-keyword)
. . . .				
Paper 100	**0**(Non-keyword)	**0**(Non-keyword)		**1** (Keyword)

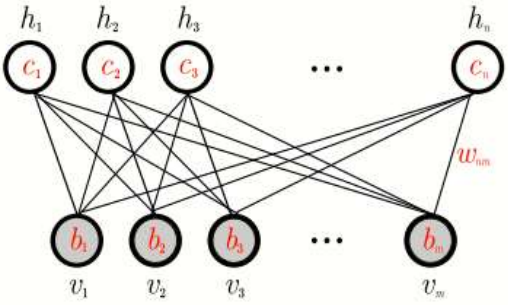
Step 8: Training Restricted Boltzmann Machine.

Batch size = 10,
Hidden nodes used = 100

Restricted Boltzmann machines (RBM) are Markov Random Fields (MRFs) with hidden variables and RBM learning algorithms are based on gradient ascent on the log-likelihood. Markov chains play an important role in RBM training because they provide a method to draw samples from 'complex' probability distributions like the Gibbs distribution of an MRF.

Gibbs Sampling belongs to the class of Metropolis-Hastings algorithms. It is a simple Markov Chain Monte Carlo (MCMC) algorithm for producing samples from the joint probability distribution of multiple random variables. The basic idea is to update each variable subsequently based on its conditional distribution given the state of the others.

Restricted Boltzmann machines (RBM) are Markov Random Fields (MRFs) associated with a bipartite undirected graph as shown in figure below



The undirected graph of an RBM with n hidden and m visible variables

The graph of an RBM has only connections between the layer of hidden and visible variables but not between two variables of the same layer.

In binary RBMs the random variables (V, H) take values (v, h) ∈ {0, 1}$^{m+n}$ and the joint probability distribution under the model is given by the Gibbs distribution with following energy function:

$$E(\boldsymbol{v}, \boldsymbol{h}) = -\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij} h_i v_j - \sum_{j=1}^{m} b_j v_j - \sum_{i=1}^{n} c_i h_i \ .$$

Contrastive Divergence:
Obtaining unbiased estimates of log-likelihood gradient using MCMC methods typically requires many sampling steps. However, recently it was shown that estimates obtained after running the chain for just a few steps can be sufficient for model training. This leads to contrastive divergence (CD) learning, which has become a standard way to train RBMs.

The idea of k-step contrastive divergence learning (CD-k) is quite simple: Instead of approximating the second term in the log-likelihood gradient by a sample from the RBM-distribution (which would require to run a Markov chain until the stationary distribution is reached), a Gibbs chain is run for only k steps (and for our problem k = batch size = 10). The Gibbs chain is initialized with a training example v(0) of the training set and yields the sample v(k) after k steps. At each step t such that 0 < t < k+1, we use **sigmoid activation function** to create P(h/v) and P(v/h). After k step, we update weight of edge connecting V and H in a way that we approximate the log-likelihood of one training pattern v(0).

A batch version of CD-k is shown in following algorithm:

Algorithm 1. k-step contrastive divergence

Input: RBM $(V_1, \ldots, V_m, H_1, \ldots, H_n)$, training batch S
Output: gradient approximation Δw_{ij}, Δb_j and Δc_i for $i = 1, \ldots, n$, $j = 1, \ldots, m$

```
init Δw_ij = Δb_j = Δc_i = 0 for i = 1,...,n, j = 1,...,m
forall the v ∈ S do
    v^(0) ← v
    for t = 0,...,k − 1 do
        for i = 1,...,n do sample h_i^(t) ~ p(h_i | v^(t))
        for j = 1,...,m do sample v_j^(t+1) ~ p(v_j | h^(t))
    for i = 1,...,n, j = 1,...,m do
        Δw_ij ← Δw_ij + p(H_i = 1 | v^(0)) · v_j^(0) − p(H_i = 1 | v^(k)) · v_j^(k)
        Δb_j ← Δb_j + v_j^(0) − v_j^(k)
        Δc_i ← Δc_i + p(H_i = 1 | v^(0)) − p(H_i = 1 | v^(k))
```

We implemented the above k-step contrastive divergence in RBM () class.

Long story short we use k = 10 rows (i.e Papers) for contrastive divergence and update weights to the direction of v(0). i.e we update weights in such a way that after training of 10 rows values, the result should create minimum errors or loss to the initial visible node values. In each step of k-step, we propogate information from visible nodes to hidden nodes and back from hidden nodes to visible nodes using sigmoid activation fuction. We use bernoulli distribution to generate binary values from sigmoid activation function. Thus, we train all 100 papers for sequencially in bucket of 10, where at each bucket 10 papers are present for 10-step contrastive divergence process.

In our example the value is generated in step 7 for paper titles. We create a list from paper 1 for training data.

List = [0., 0., 0., 1., 0., 1., 0., 0., 0., 0., 0., 0., 1., 0., 1., 1., 0., 0., 1., 0., 0.]

Total no. of 1's = 6
Total no. of 0's = 16

Here we only shows the 0 and 1 values. -1 is not shown because with including -1 value in the above list, it's size grows to 2451.
Here each index of list is representing a word. It can be single word, biword or triword.

['cancer', 'lymph node', 'exosomes', 'breast cancer', 'metastasis', 'macrophage']	[189, 1341, 753, 180, 1410, 1349]

Above are the keywords reperesed by 1 in the list. Though we are thinking that we are very lucky to have all 6 keywords in training of paper 1, but it is not true because in reality only two keywords are present in paper 1 training data and they formed duplicates for positional indexing which I think is very important for prediction.

['macrophage polarization', 'cancer cell derived', 'polarization associated', 'associated', 'derived', 'node metastasis', 'cancer cell', 'exosomes macrophage polarization', 'cell', 'polarization associated lymph', 'cell derived exosomes', 'lymph node metastasis', 'exosomes macrophage', 'derived exosomes macrophage', 'derived exosomes', 'polarization', 'associated lymph', 'macrophage polarization associated', 'associated lymph node', 'node', 'cell derived', 'breast', 'breast cancer cell', 'lymph']	[1342, 799, 96, 1612, 1613, 258, 488, 99, 1730, 1729, 800, 497, 100, 179, 259, 181, 194, 1728, 252, 1340, 503, 193, 1354, 1355]

Above are the non-keywords represented by 0 in the list. Here just for the sake of argument, consider that we are having 16 non-keywords in training data and emaining 8 non-keywords are placed in testing data.

After training for 10 time (10 epoch) we updated the weight to reduce errors explained earlier.This creates concepts related to EVs in hidden nodes. After training visible node list is updated to the following:

List = [0., 0., 0., 1., 0., 1., 0., 0., 0., 0., 0., 0., 0., 1., 0., 1., 1., 0., 0., 1., 0., 0.]

Let us compare these two list of paper 1:

List_before_training = [0., 0., 0., 1., 0., 1., 0., 0., 0., 0., 0., 0., 1., 0., 1., 1., 0., 0., 1., 0., 0.]

List_after_training = [0., 0., 0., 1., 0., 1., 0., 0., 0., 0., 0., 0., 1., 0., 1., 1., 0., 0., 1., 0., 0.]

Clearly our model made no error for 0 and 1 values before and after training the model. But I still got some mean of absolute difference error of 0.00326. This is due to the fact that our list also makes some prediction for -1 values. This clearly shows that our model is trained by keywords and non-keywords words of 100 papers and generate a concept in hidden nodes which are responsible to make prediction for keywords and non-keywords words which are not present in the paper 1.

Now since our model is trained we can use it for testing.

Let's do it.

For testing data our training model changes into Trained_list. This list have the value of the trained model but representing only value for which testing data is having 1 or 0.

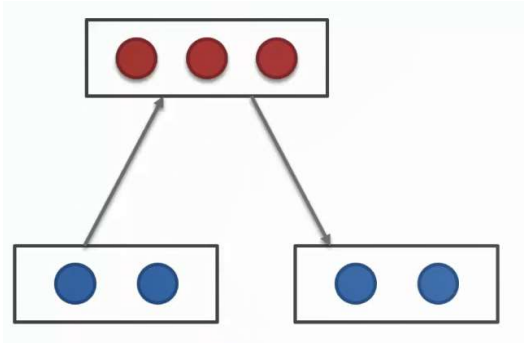
The value -1 shows that it is not used for training but still is a part of paper1 title because it has corresponding 0 or 1 value in Test_list. This shows that some information used for testing is not used in training.

Trained_list = [0., -1., 1., 0., 0., 0., 0., 0., 0., 0., -1., 1., -1., -1., -1., 1., 1., -1., -1., 0., -1.]

Test_list = [0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 1., 1., 0., 0., 0., 0.]

Remember there are remaining 8 non-keywords word present in the testing data. Lets us see that they are present in the testing data or not.
There are 17 zeros which is greater than 8. This happens because our analysis also include duplicates of words from paper 1. This duplicates are generated due to the occurance of same word but in different position. Therefore our training data may have 16 non-keyword words or less than 16 non-keyword words with some duplicates.

For above Trained_list we traverse this information to hidden layers using activation function (sigmoid) and bernoulli function to generates values in hidden layers. From hidden layer concepts the information is traverse from hidden layer to visible nodes using again activation function of sigmoid and lastly applied this information to bernoulli function to generate values for visible nodes. The value generated are shown in the Prediction_list.



Prediction _list = [0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 1., 1., 1., 1., 1., 1., 1., 0., 0., 0., 0.]
Now let us compare our prediction with original result in testing data.

Prediction _list = [0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 1., 1., 1., 1., 1., 1., 1., 0., 0., 0., 0.]

Test_list = [0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 1., 1., 0., 0., 0., 0.]

Now Let us calculated number of 1's matched in prediction_list and Test_list.

We got 4 ones matching, which shows that the 4 EV related keywords are present in the paper.

We make this kind of prediction for all 100 papers. And then calculate their score by summing the number of keywords matched in the prediction_list and Test_list. We then normalize the score of all papers to generate value ranging from 0 to 1. When the normalized score > 0.5, we can put the papers into related to EVs. And vice versa.