Learning Vector Embeddings for Words

Mallika Subramanian

2018101041

This report describes the methods adopted to train the word embeddings on the *Stanford Amazon Electronics Product Reviews* corpus using SVD decomposition of co-occurrence matrix and Continuous Bag of Words.

Analysis for CBOW & SVD:

The dimensionality of each word vector is chosen as **50** (50 principle components/singular vectors) and a window size of **5 words** (left and right of the center word) is selected.

A very high dimension of the word vectors can cause unwanted effects since increasing the dimension to an extremely high value may not penalize the less frequent words, and hence typos or missing punctuation words which occur in the same context as a query word may become closer (which is not desired).

For SVD, a sparse matrix of the co-occurence matrix was constructed so as to fit the entire matrix into memory and since many of the entries in the VxV co-occ matrix would be 0.

Closest words for the word "Camera":

SVD

```
Closest words via Custom model:
           Word  Distance
0  opportunities  0.016979
1         shoots  0.019403
2          camer  0.019489
3        cameras  0.023204
4         taking  0.025289
5          shoot  0.028736
6         s100fs  0.031011
7       pictures  0.031387
8          photo  0.033768
9           dlsr  0.034868
```

CBOW

```
Closest words via Custom model:
         Word  Distance
0    shooting  0.308741
1         t1i  0.324106
2       shoot  0.339060
3         slr  0.350236
4      shoots  0.351036
5        c180  0.352362
6   picbridge  0.353198
7  viewfinder  0.353304
8   thecamera  0.354191
9       shots  0.357362
```

GENSIM

```
Closest words via Gensim model:
          Word  Distance
0      cameras  0.847485
1       screen  0.754730
2        video  0.697745
3      screens  0.678826
4   microphone  0.662158
5      digital  0.633627
6       images  0.628702
7       device  0.627248
8       window  0.624143
9       sensor  0.621488
10       photo  0.620934
```

Observations:

- The **custom** model has words such as *photos*, *shoots*, *dslr*, etc. which are semantically very close to the query word "camera".
- Some exceptions are *opportunities* with the lowest distance, indicating that this word occurs several times in similar contexts as the word "camera", and the word *camer* which is a typographical mistake that the model has learnt as well.
- On comparing the word embeddings from SVD and CBOW with Gensim, firstly it can be observed that the vocabulary in gensim based word embeddings is richer - this is because it is trained on a much larger corpus as compared to ~64K words in the custom models. Further, there are no typos in

the gensim words. Lastly, due to a larger number of vectors in the Gensim vector space, it can be seen that the distances between the top words is also larger as compared to the custom model.

Inferences from results via the encoder decoder architechture

On training the CBOW model using the enc-dec architechture of 2 Weight matrices, without an `nn.Embedding` layer, there are two weight matrices (W1 and W2) that are learnt by the model. W1 represents the *input word* representation while W2 represents the *output word* representation. The output word representation produces much better semantically closer embeddings since the input word rep captures more of contextual information than direct meaning representation.

Word that are extremely similar in terms of meaning to "camera" such as "camcorder", "cannon" etc. appear in the output rep, and as observed, apart from a few meaning similar words, many possible neighbouring words such as "enjoy", "very" occur in the input rep.

Input Word Representation

```
----------------------------------
For query word: CAMERA
----------------------------------

Closest words via Custom model:
     Word  Distance
0  result  0.436552
1    very  0.438986
2    lens  0.452688
3   helps  0.457526
4    pero  0.459920
5   enjoy  0.463503
6  optics  0.470779
7    zoom  0.471201
8    fuji  0.472382
9   ultra  0.475835
```

Output Word Representation

```
----------------------------------
For query word: CAMERA
----------------------------------

Closest words via Custom model:
          Word  Distance
0      franiec  0.458251
1      richard  0.473028
2  microswitch  0.483453
3       sx10is  0.488104
4          g1x  0.497506
5          len  0.498702
6      unheard  0.505282
7    camcorder  0.505454
8    cameraman  0.508244
9       cannon  0.515874
```

Similar words to a mix of words:

The top most similar (geometrically closest) words for 5 words - adjectives, nouns, verbs combined - are shown below:

Word : Bought, POS: Verb

SVD

```
Closest words via Custom model:
         Word  Distance
0   purchased  0.007462
1       bough  0.010110
2  originally  0.011880
3         got  0.013839
4        whim  0.014652
5      wanted  0.016165
6    thrilled  0.016198
7    recently  0.017205
8       happy  0.017303
9   delighted  0.017783
```

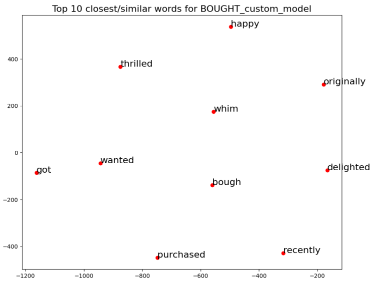
CBOW

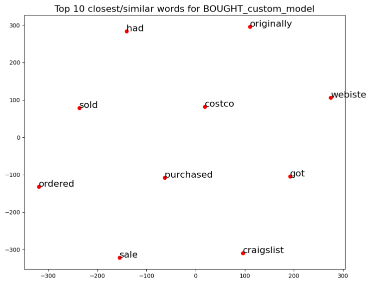
```
Closest words via Custom model:
         Word  Distance
0   purchased  0.179871
1  originally  0.235424
2     ordered  0.239623
3        sale  0.314945
4         had  0.316879
5         got  0.318878
6        sold  0.324557
7      costco  0.335277
8     webiste  0.336655
9  craigslist  0.339215
```

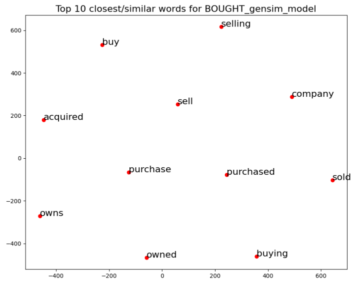
GENSIM

```
Closest words via Gensim model:
         Word  Distance
0        sold  0.904530
1   purchased  0.889572
2        sell  0.787863
3         buy  0.776478
4    acquired  0.761537
5       owned  0.738812
6    purchase  0.729229
7     company  0.717462
8     selling  0.701326
9        owns  0.700094
10     buying  0.694530
```

SVD	CBOW	GENSIM







Word : Broken, POS: Adjective

SVD

Closest words via **Custom** model:

	Word	Distance
0	busted	0.011194
1	eventually	0.016946
2	wound	0.017332
3	treated	0.018182
4	broke	0.018336
5	destroyed	0.018943
6	damaged	0.019393
7	threw	0.019408
8	tossed	0.019420
9	trash	0.019555

CBOW

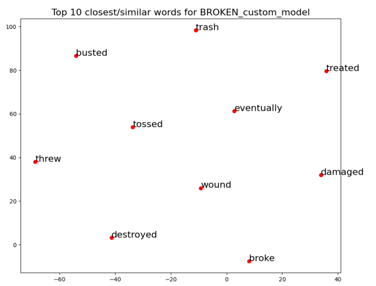
Closest words via **Custom** model:

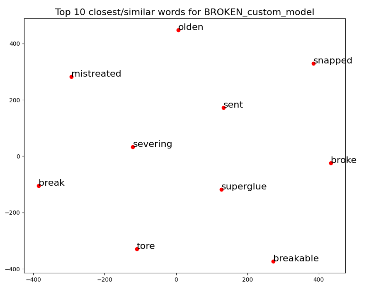
	Word	Distance
0	broke	0.175180
1	tore	0.279162
2	superglue	0.282920
3	olden	0.284691
4	snapped	0.295339
5	severing	0.300505
6	break	0.300851
7	mistreated	0.301243
8	sent	0.302186
9	breakable	0.309855

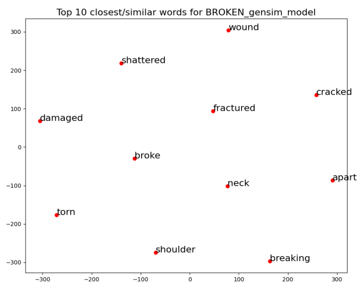
GENSIM

Closest words via **Gensim** model:

	Word	Distance
0	breaking	0.728136
1	broke	0.718399
2	apart	0.717275
3	cracked	0.704760
4	fractured	0.695731
5	neck	0.677935
6	damaged	0.677219
7	shattered	0.670413
8	torn	0.667543
9	shoulder	0.661755
10	wound	0.658630







Word : Computer, POS: Noun

SVD

Closest words via **Custom** model:

	Word	Distance
0	pc	0.012715
1	instantly	0.021464
2	run	0.026124
3	fuss	0.026765
4	desktop	0.028909
5	computers	0.029388
6	solution	0.029891
7	device	0.030501
8	devise	0.030616
9	comp	0.032868

CBOW

Closest words via **Custom** model:

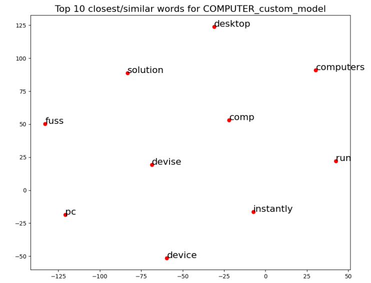
	Word	Distance
0	pc	0.223677
1	unplug	0.324378
2	usp	0.362745
3	hardrives	0.366824
4	sneakernet	0.375913
5	ubs	0.379100
6	xp	0.384929
7	pug	0.388496
8	laptop	0.391597
9	computers	0.392434

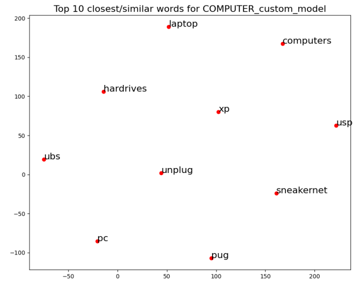
GENSIM

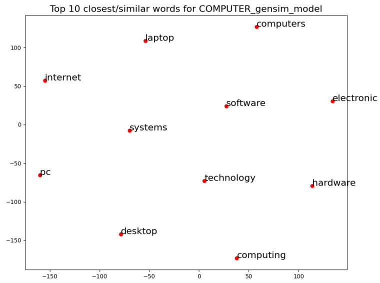
Closest words via **Gensim** model:

	Word	Distance
0	computers	0.875198
1	software	0.837312
2	technology	0.764216
3	pc	0.736645
4	hardware	0.729039
5	internet	0.728678
6	desktop	0.723444
7	electronic	0.722183
8	systems	0.719792
9	computing	0.714173
10	laptop	0.702416

SVD	CBOW	GENSIM







Word : Excellent, POS: Adjective

SVD

Closest words via **Custom** model:

	Word	Distance
0	outstanding	0.010217
1	terrific	0.011257
2	exceptionally	0.011279
3	quality	0.012172
4	fantastic	0.012651
5	exceptional	0.013373
6	reasonably	0.015661
7	good	0.016082
8	superb	0.016338
9	winner	0.016390

CBOW

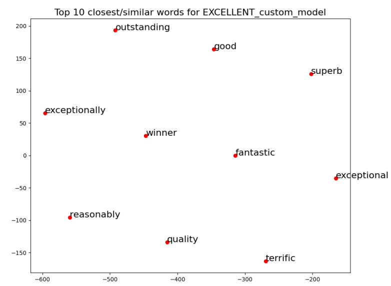
Closest words via **Custom** model:

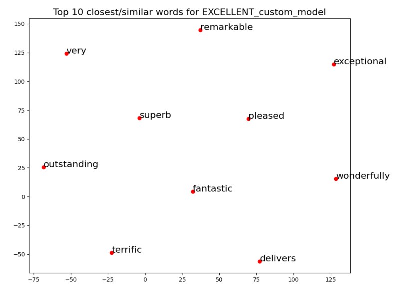
	Word	Distance
0	exceptional	0.167847
1	terrific	0.171288
2	superb	0.174636
3	outstanding	0.188234
4	wonderfully	0.195855
5	fantastic	0.242061
6	very	0.250670
7	delivers	0.252763
8	pleased	0.264613
9	remarkable	0.269314

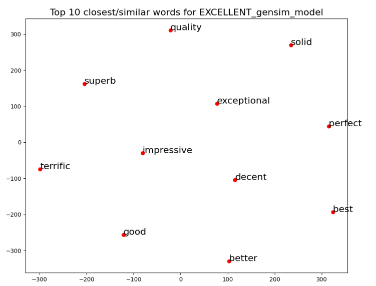
GENSIM

Closest words via **Gensim** model:

	Word	Distance
0	good	0.793624
1	quality	0.760627
2	terrific	0.741562
3	superb	0.740296
4	best	0.728757
5	impressive	0.721752
6	better	0.710246
7	solid	0.699920
8	decent	0.690701
9	perfect	0.683730
10	exceptional	0.677119







Word : iPhone, POS: Noun

SVD

Closest words via **Custom** model:

	Word	Distance
0	iphone4	0.012280
1	4s	0.024213
2	iphones	0.027344
3	idevices	0.029014
4	itouch	0.034241
5	docked	0.038826
6	3gs	0.043476
7	blackberry	0.045569
8	dock	0.046432
9	smartphone	0.046924

CBOW

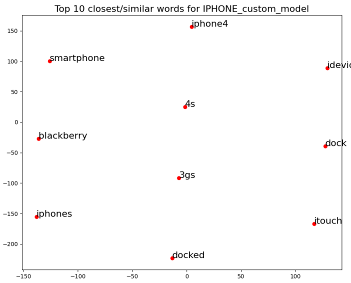
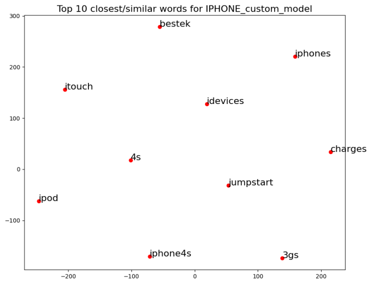
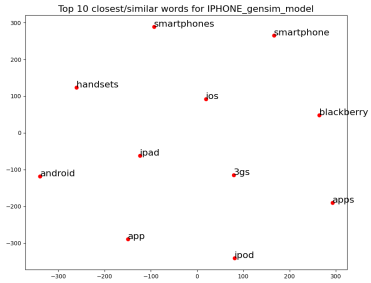
Closest words via **Custom** model:

	Word	Distance
0	3gs	0.217977
1	iphones	0.300831
2	jumpstart	0.308777
3	4s	0.317262
4	charges	0.317569
5	bestek	0.323118
6	idevices	0.323731
7	iphone4s	0.324984
8	ipod	0.330806
9	itouch	0.334271

GENSIM

Closest words via **Gensim** model:

	Word	Distance
0	ipad	0.922029
1	ipod	0.836901
2	smartphone	0.798461
3	android	0.760697
4	app	0.749143
5	apps	0.740781
6	ios	0.734827
7	smartphones	0.717577
8	handsets	0.716835
9	blackberry	0.716263
10	3gs	0.703737

SVD	CBOW	GENSIM
		

Word : Terrible, POS: Adjective

SVD	CBOW	GENSIM

Closest words via Custom model:

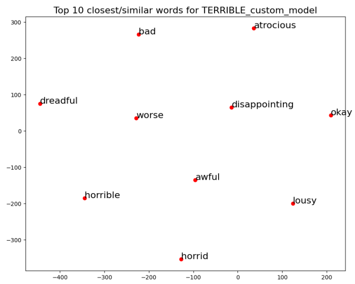
	Word	Distance
0	horrible	0.002366
1	awful	0.003485
2	horrid	0.008626
3	worse	0.009293
4	disappointing	0.010193
5	lousy	0.010394
6	atrocious	0.010860
7	bad	0.011257
8	dreadful	0.011409
9	okay	0.011708

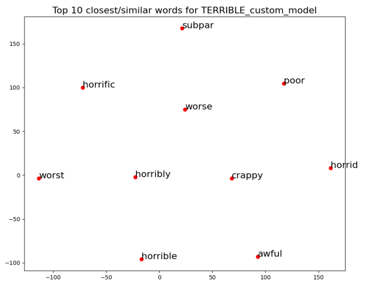
Closest words via Custom model:

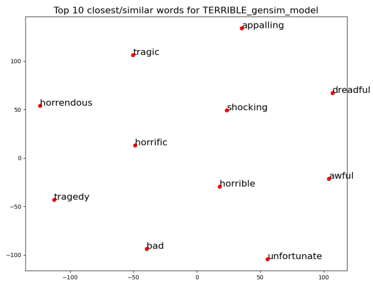
	Word	Distance
0	horrible	0.097584
1	poor	0.136019
2	horribly	0.142454
3	awful	0.179958
4	horrid	0.182003
5	worst	0.220708
6	subpar	0.222893
7	horrific	0.224972
8	crappy	0.235017
9	worse	0.240263

Closest words via Gensim model:

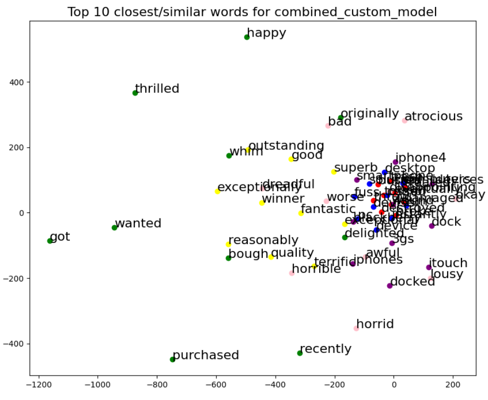
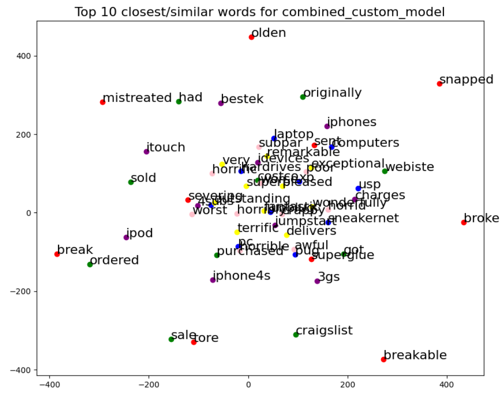
	Word	Distance
0	horrible	0.919477
1	awful	0.874217
2	dreadful	0.782151
3	horrendous	0.778243
4	horrific	0.764398
5	tragic	0.759758
6	appalling	0.732745
7	tragedy	0.725573
8	bad	0.707213
9	unfortunate	0.704123
10	shocking	0.698123







Combined plot of the 6 words:

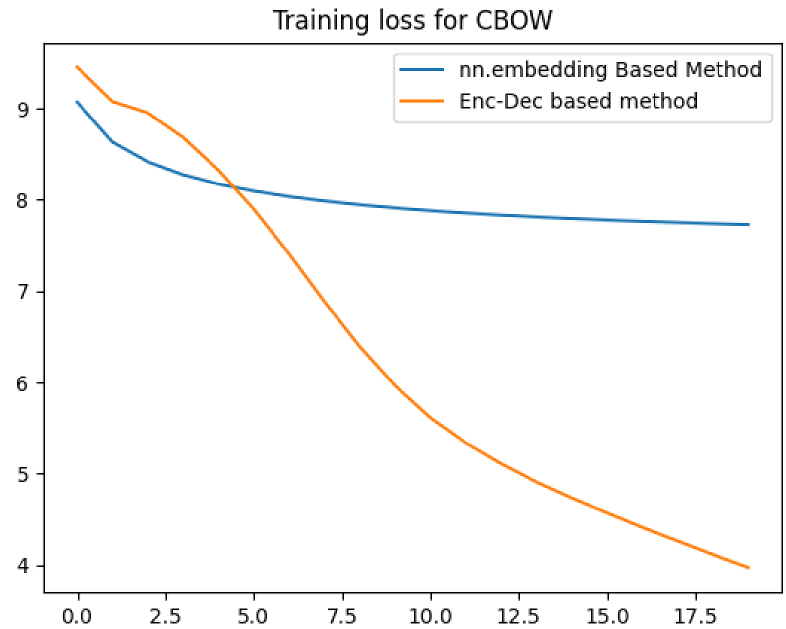
SVD	CBOW
	

The TSNE parameters can be fine-tuned further, so as to get perfect clusters of words. The current parameters do not preserve the geometric information of the 50 or 100 dimension vector space perfectly, and some representational info is lost while reducing dimensionality.

Loss Plot for CBOW Training.

The loss plot for CBOW training is shown below. The ***nn.Embedding*** based method uses 1M reviews to train, while the ***enc-dec*** based method uses 20K reviews.

The loss decreases from ~8-9 to roughly ~4-5 and can be further decreased if trained on a larger corpus and for more number of epochs.



Computational constraints:

- ***Building vocabulary from 1M reviews (instead of 1.6M)***: To construct the entire vocabulary, it takes around 20 hours or so. Further, requesting for a higher memory and cpu configuration causes a SLURM error on ADA:

```
(base) mallika.subramanian@gnode71:~/courses/anlp/Learning-Word-Embeddings$ cat train.sh
#!/bin/bash
#SBATCH -A research
#SBATCH -c 1
#SBATCH --gres=gpu:1
#SBATCH --mem-per-cpu=64G
#SBATCH -o wv_op.txt
#SBATCH --job-name=wv_train
#SBATCH --time=3-00:00:00


python3 run.py Electronics_5.json 0
(base) mallika.subramanian@gnode71:~/courses/anlp/Learning-Word-Embeddings$ sbatch train.sh
sbatch: error: QOSMaxMemoryPerUser
sbatch: error: Batch job submission failed: Job violates accounting/QOS policy (job submit limit, user's size and/or time limits)
(base) mallika.subramanian@gnode71:~/courses/anlp/Learning-Word-Embeddings$
```

- ***Training CBOW enc-dec on a smaller sample***: Since the compute time to train 1 epoch of enc-dec CBOW on the entire corpus of 1.6M reviews was nearly 14 hours, to train 20 epochs would take 280 hours which is > 10 days. 10 epochs also would take 140 hours which is ~5 days. Hence the enc-dec CBOW architechture was trained on a smaller subset of 20K reviews. The screenshot of runtime is :

```
(base) mallika.subramanian@gnode53:~/courses/anlp/Learning-Word-Embeddings$ cat wv_op.txt

----------------------------------------
Performing CBOW...
Generating context-center train loader...
tokenized_corpus_len =  990502
100%|██████████| 990502/990502 [01:27<00:00, 11351.41it/s]
Total num training samples =  13442043
Batch size =  1024
Instantiating CBOW Model ...
Beginning Training Now ...
Device :  cuda
  0%|          | 0/20 [00:00<?, ?it/s]               (base) mallika.subramanian@gnode53:~/courses/anlp/Learning-Word-Embeddings$
  0%|          | 37/13127 [02:30<14:53:36,  4.10s/it]
```