Intro to NLP

Devesh Marwah (2020115005)

March 2023

Pages: 7

1 Questions

1.1 Question 1

Explain negative sampling. How do we approximate the word2vec training computation using this technique?

Word2vec involves three steps which are the following:

- Look up embeddings
- Calculate prediction
- Project to output vocab

The third part is actually very expensive from a computation point of view and needs to be done for every word which can approximate to millions. Given this problem, negative sampling was introduced which aims to approximate the calculation by introducing some samples which are NOT in the given context and aims on labelling them as 0 or 1.

The need was needed so that model doesn't learn just neighbours and will output 100% accuracy for all values, learning nothing.
With the introduction of these negative samples, instead of calculating for entire vocab we can do it for a small number of samples. A new function can be defined which tries to maximises probability of neighbour words and keep them as 1 while keeping probability of negative samples as 0.
Hence few words are randomly sampled from the vocab and added to context with say label 0. The actual context is given with label 1.

1.2 Question 2

Explain the concept of semantic similarity and how it is measured using word embeddings. Describe at least two techniques for measuring semantic similarity using word embeddings.

Semantic similarity can be thought to be the likeness between the two words in terms of actual meaning gained by learning context of the word. When we project them to embeddings we intend to learn the hidden meaning behind the words by projecting them to higher vector spaces and each vector space representing a dimension learnt by it. Hence what I mean when I say I have 128 length embedding is that I have 128 different features which can be learnt.

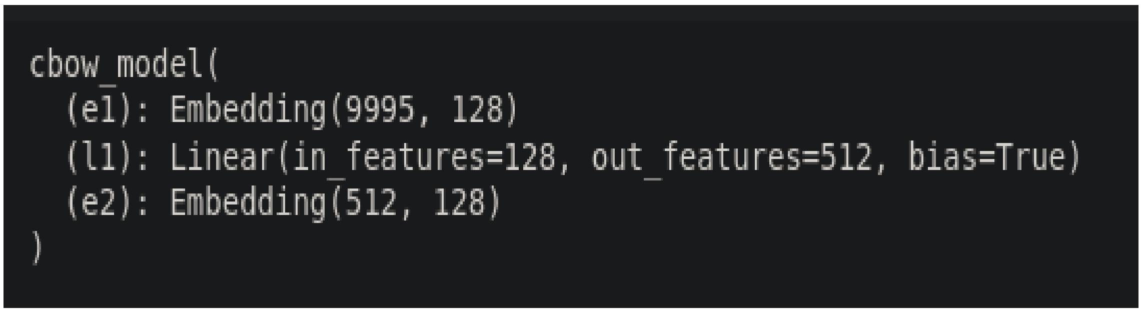
- Cosine Similarity: It takes the dot product of two vectors and calculates the degree of similarity. It ranges from 0 to 1 and the closer to 1, the better it is.
- Euclidean Distance: Euclidean distance as we know is the root square distance between two vectors. It however is used less because it shows differences more.

Cosine similarity is used more than often than Euclidean and hence in assignment we have chosen to do the same.

2 Models

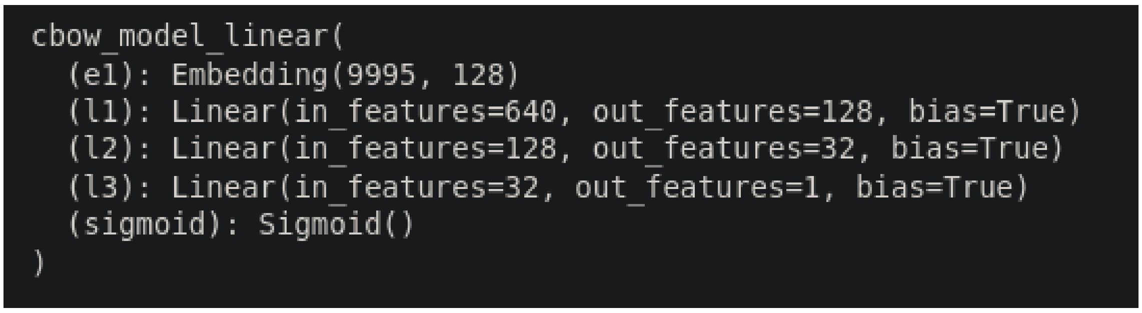
2.1 Implementationd details

I have experimented with two types of models.
Model 1 is an implementation of the model specified by here by taking the dot product in batchnorm after summing them up.



```
cbow_model(
  (e1): Embedding(9995, 128)
  (l1): Linear(in_features=128, out_features=512, bias=True)
  (e2): Embedding(512, 128)
)
```

Model 2: The second is an interesting approach. I have reduced the problem to a binary classification problem. It involves training the embedding layer by first calculating embeddings and then putting three basic linear layers to output a probability which denotes whether the sample is negative or positive. We have the actual labels of them and then can train the binary classification model simply using Binary Cross Entropy Loss function.



```
cbow_model_linear(
  (e1): Embedding(9995, 128)
  (l1): Linear(in_features=640, out_features=128, bias=True)
  (l2): Linear(in_features=128, out_features=32, bias=True)
  (l3): Linear(in_features=32, out_features=1, bias=True)
  (sigmoid): Sigmoid()
)
```

2.2 Results

The results for the same can be looked up here for the word "titanic":

Model 1:

style
judging
depends
shannon
addicts
powerfully
procedures
communities
authenticity
tongue

In this at first glance the outputs seem to be a bit random but they are related to a book by a person named shannon who was also a character in the movie. Titanic: A Love Story . Similarly the dataset is of movie reviews, the word judging is coming.
Model 2:

posting
tolerated
dreyer
melts
epilepsy
wwi
chased
washed
leonardo
sanctity

This gives better output as we see words like melts, leonardo i.e. the main lead of movie, wwi, the time when movie occurred and washed. It seems to work fine. Even the name dreyer is from Titanic 1984 hence the results seem decent for this model.

The limitiation seems to be off compute power and small dataset as only 40k sentences were picked for training.

SVD:

For the word titanic, it seems to perform pretty poorly.

inducing', 'cheek', 'objects', 'wisely', 'identification', 'wholesale', 'mulligan', 'soften', 'histories', 'docudrama

Original Word2Vec Embeddings closest to titanic can be seen here:

epic 0.600616455078125

colossal 0.5896502137184143

gargantuan 0.5718227028846741

titanic_proportions 0.5610266923904419

titantic 0.5592556595802307

monumental 0.5530510544776917

monstrous 0.5457675457000732

epic_proportions 0.5437003970146179

gigantic 0.5176911950111389

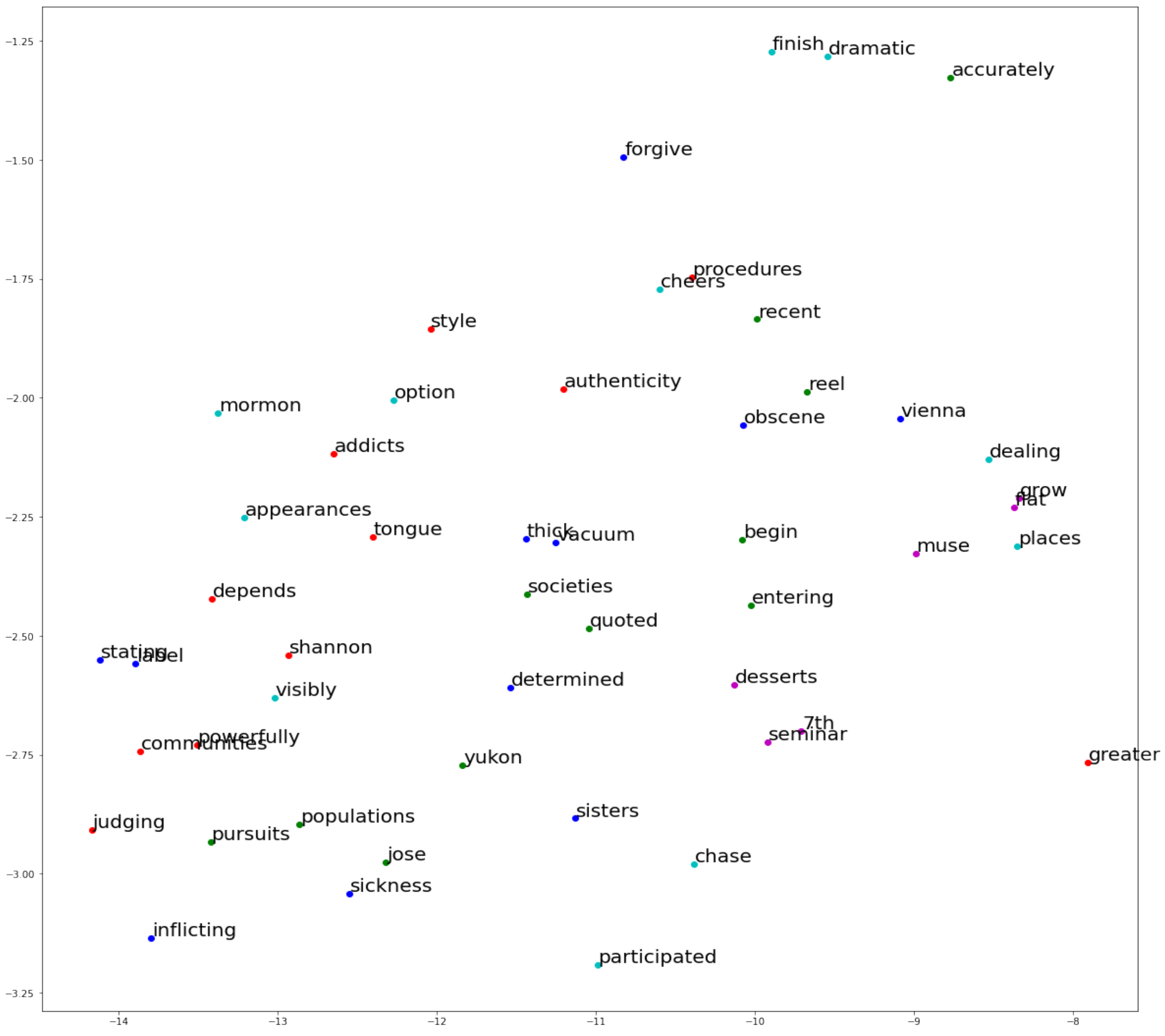
mighty 0.5088781118392944)

The SVD performs pretty poorly as expected and the original Word2Vec embeddings produce a general result, given our dataset was specifically for movie reviews, it performed a bit better than the general model.

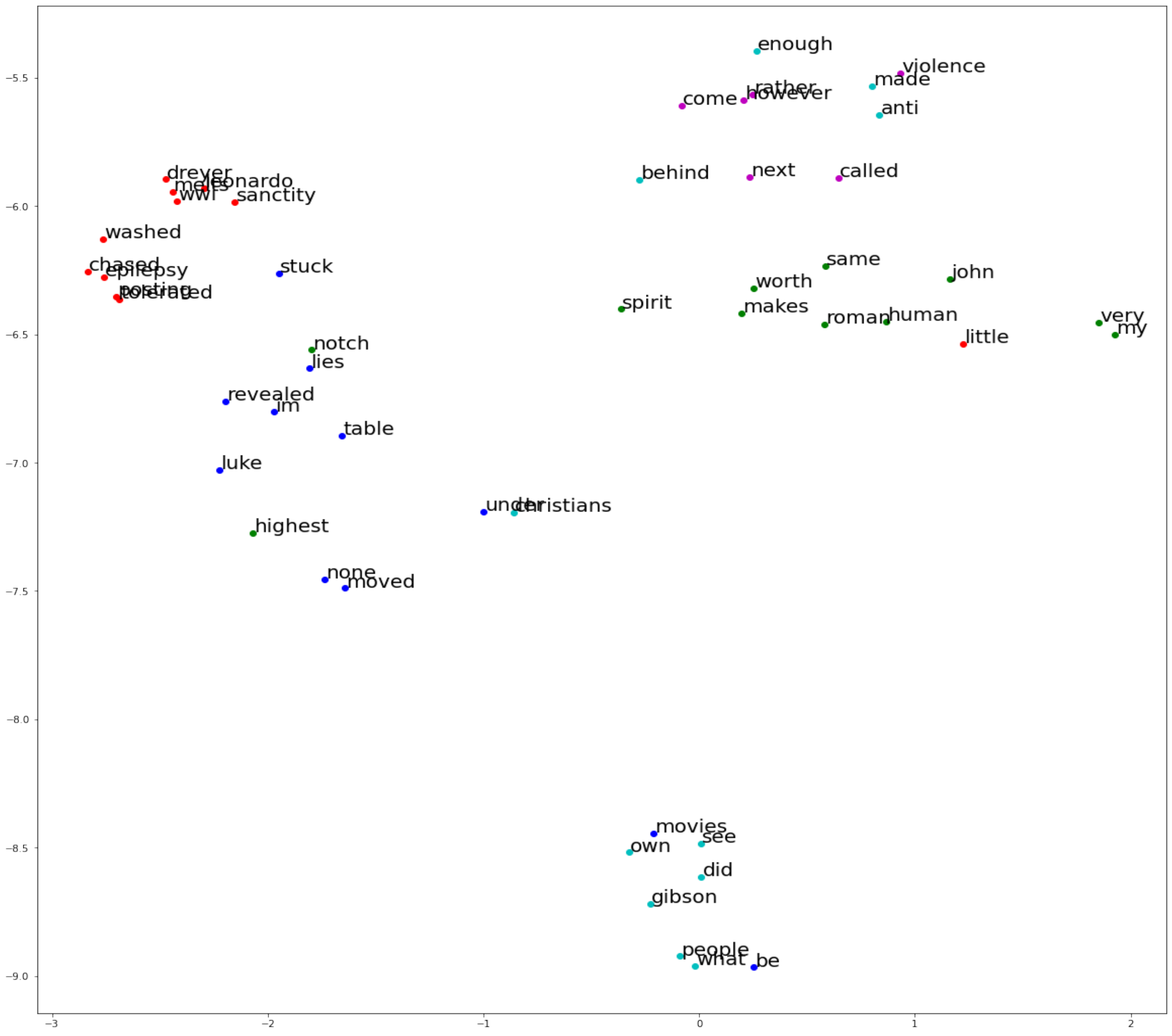
2.3 TSNE

The TSNE for each of them can be seen here. The words are titanic,great,running,movie,music

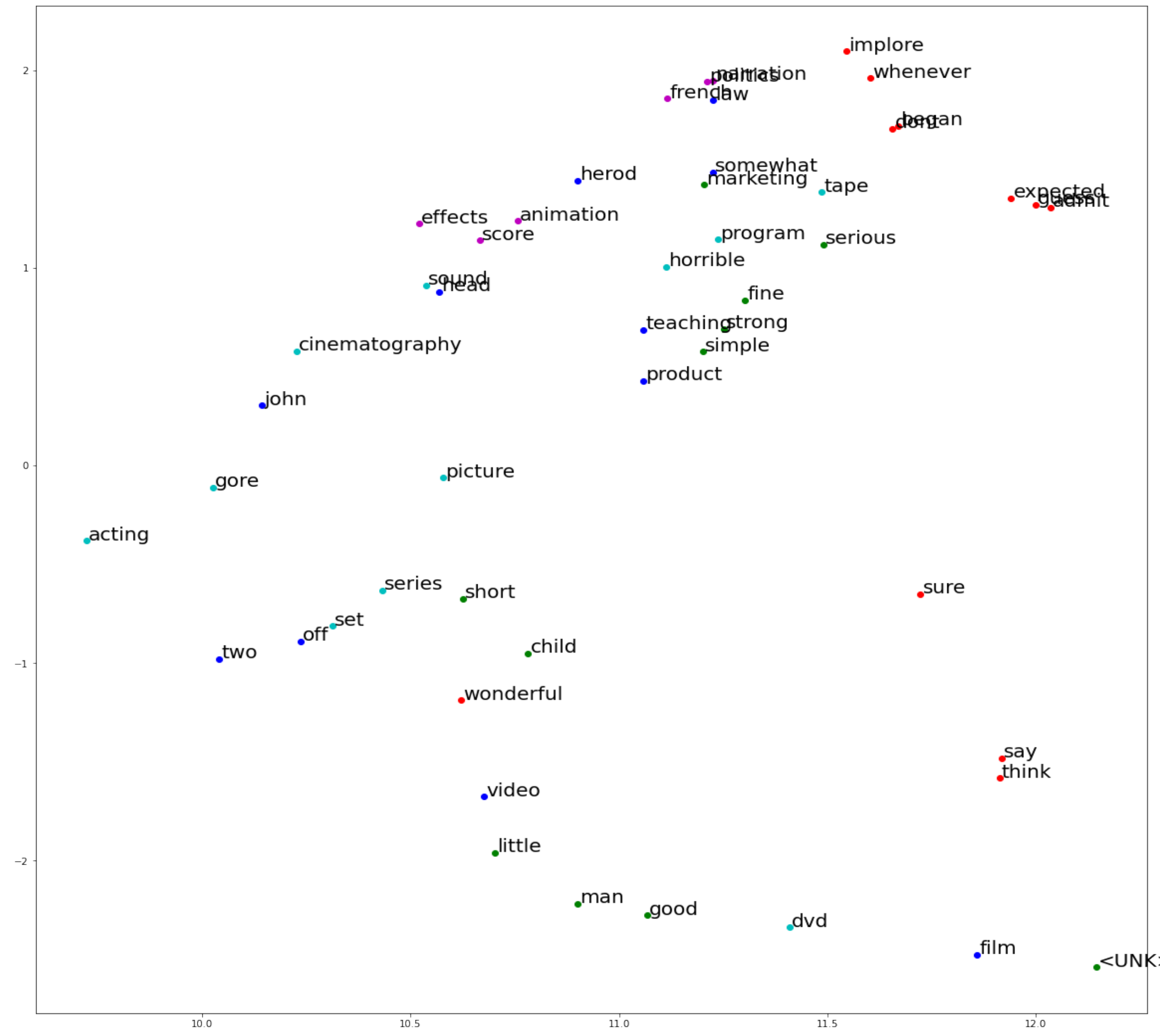
2.4 Model 1



2.4.1 Model 2


enough
violence
made
come
rather
however
anti
behind
next
called
dreyer
leonardo
melies
wwi
sanctity
washed
stuck
same
john
worth
chased
epilepsy
spirit
makes
posting
tolerated
roman
human
very
my
little
notch
lies
revealed
im
table
luke
under
christians
highest
none
moved
movies
see
own
did
gibson
people
what
be
−5.5
−6.0
−6.5
−7.0
−7.5
−8.0
−8.5
−9.0
−3
−2
−1
0
1
2

2.4.2 SVD



2.4.3 Analysis

As we observe the similar color words are together showing a similarity between them (as it should) but there is a proper clustering in the neural models, unlike the SVD.