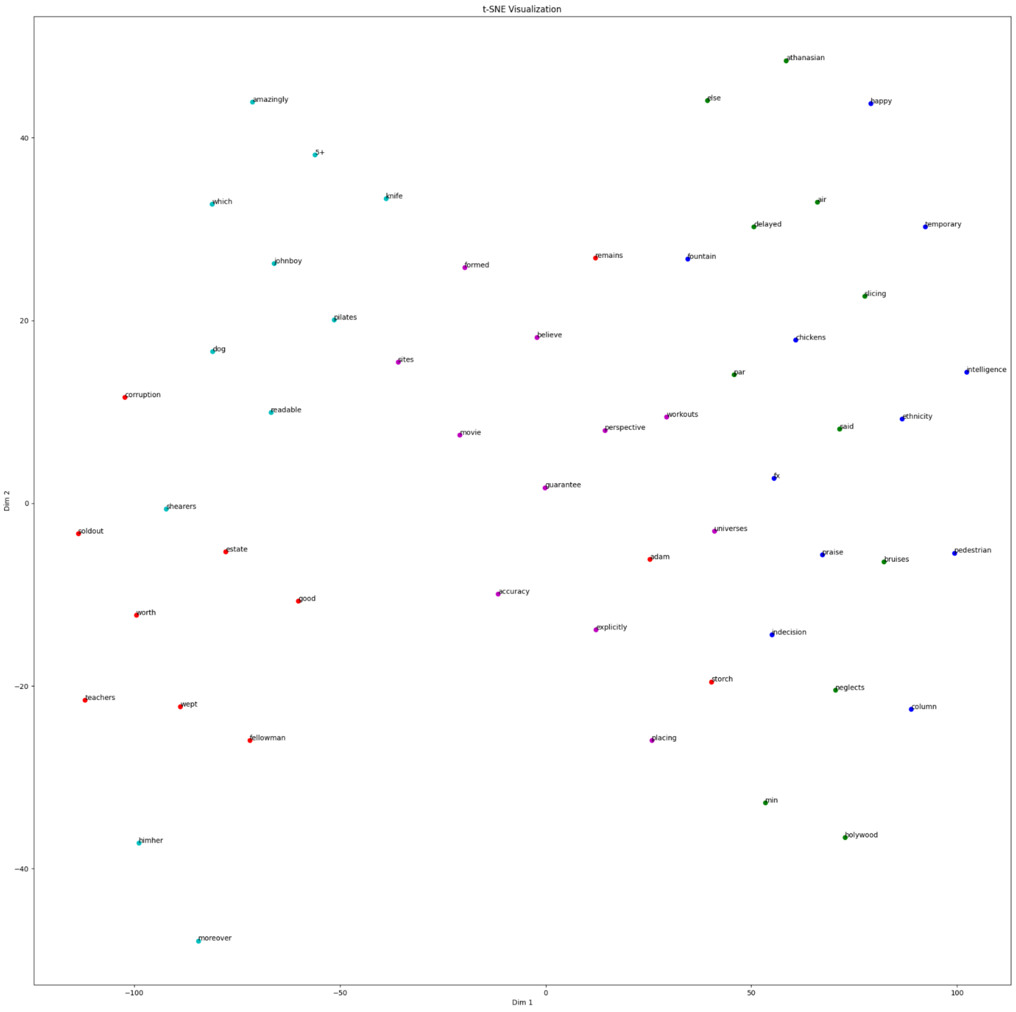
Report for CBOW and SVD

Srijan Chakraborty
2020115001

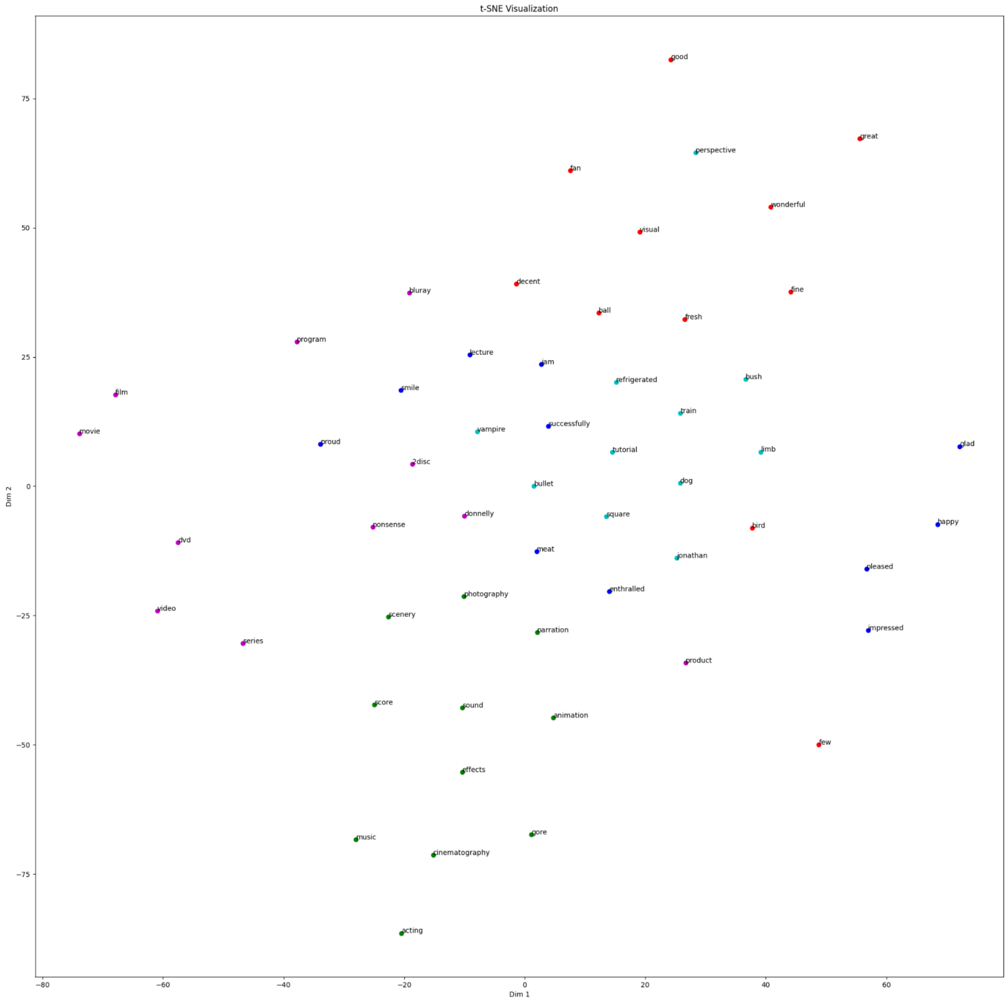
Analysis (2.3):

Part 1:

tSNE for CBOW with Negative Sampling



tSNE for SVD:

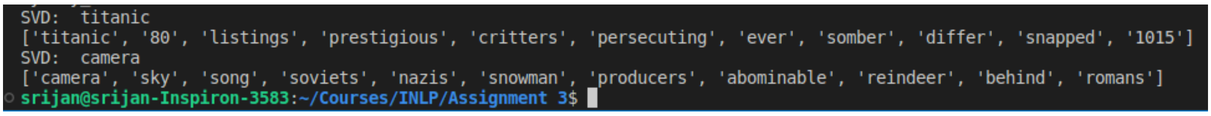


Part 2:

We have examined and worked with three models for this assignment. The three models are SVD, CBOW with negative sampling and finally, the pre-trained models.

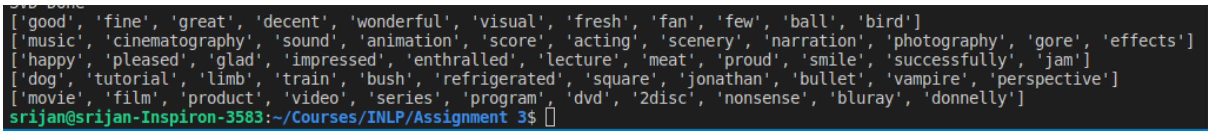
Typically, SVD helps to identify the underlying semantic structure in these matrices by decomposing them into a lower-dimensional representation, which captures the most important information. Through our results, which gives some similar words like somber, prestigious for titanic, and producers for camera, we can see that our SVD has caught a little bit of the semantic structure of the related phrases. These give movie-related outputs (such as producer/listings) because of the nature of our dataset. However, the SVD has a clear drawback and does not produce satisfactory outputs, largely due to the limitations of our dataset and the fact that we were only able to train them on 45,000 sentences because of insufficient computational resources.

Results for the trained SVD:



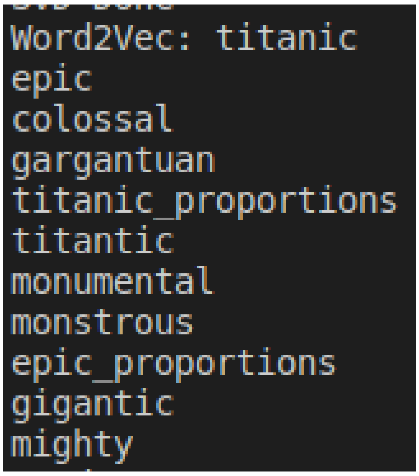
```
SVD:  titanic
['titanic', '80', 'listings', 'prestigious', 'critters', 'persecuting', 'ever', 'somber', 'differ', 'snapped', '1015']
SVD:  camera
['camera', 'sky', 'song', 'soviets', 'nazis', 'snowman', 'producers', 'abominable', 'reindeer', 'behind', 'romans']
srijan@srijan-Inspiron-3583:~/Courses/INLP/Assignment 3$
```

These are the results for more commonly occurring words:

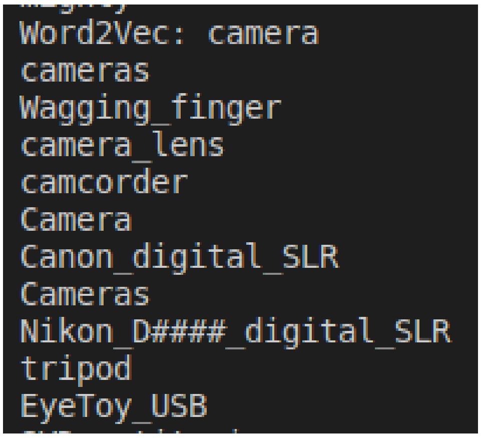


```
['good', 'fine', 'great', 'decent', 'wonderful', 'visual', 'fresh', 'fan', 'few', 'ball', 'bird']
['music', 'cinematography', 'sound', 'animation', 'score', 'acting', 'scenery', 'narration', 'photography', 'gore', 'effects']
['happy', 'pleased', 'glad', 'impressed', 'enthralled', 'lecture', 'meat', 'proud', 'smile', 'successfully', 'jam']
['dog', 'tutorial', 'limb', 'train', 'bush', 'refrigerated', 'square', 'jonathan', 'bullet', 'vampire', 'perspective']
['movie', 'film', 'product', 'video', 'series', 'program', 'dvd', '2disc', 'nonsense', 'bluray', 'donnelly']
srijan@srijan-Inspiron-3583:~/Courses/INLP/Assignment 3$
```

Reviews for the pre-trained Word2Vec model:



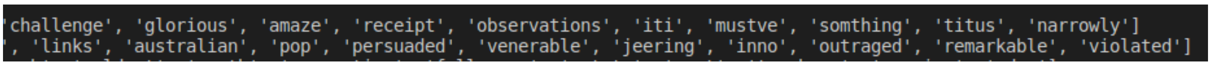
```
Word2Vec: titanic
epic
colossal
gargantuan
titanic_proportions
titantic
monumental
monstrous
epic_proportions
gigantic
mighty
```



```
Word2Vec: camera
cameras
Wagging_finger
camera_lens
camcorder
Camera
Canon_digital_SLR
Cameras
Nikon_D####_digital_SLR
tripod
EyeToy_USB
```

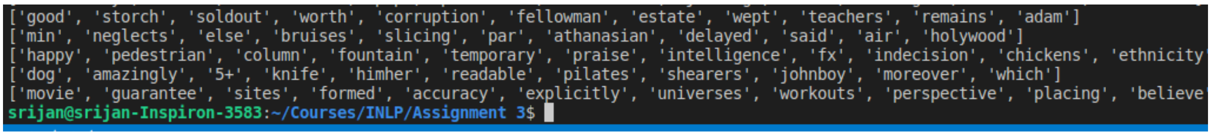
We see that the CBOW Model, which is a neural network architecture used in Natural Language Processing (NLP) to generate word embeddings and capture their semantic meaning, gives similar results, showing that the semantic structures and similarities are slightly captured, but the results are still poor. We understand that the drawback is similar to the SVD and it is because of the small corpus size we are considering because of our computational limitations

Results of the CBOW model:



```
'challenge', 'glorious', 'amaze', 'receipt', 'observations', 'iti', 'mustve', 'somthing', 'titus', 'narrowly']
', 'links', 'australian', 'pop', 'persuaded', 'venerable', 'jeering', 'inno', 'outraged', 'remarkable', 'violated']
```

To make an apple-to-apple comparison with SVD:



```
['good', 'storch', 'soldout', 'worth', 'corruption', 'fellowman', 'estate', 'wept', 'teachers', 'remains', 'adam']
['min', 'neglects', 'else', 'bruises', 'slicing', 'par', 'athanasian', 'delayed', 'said', 'air', 'holywood']
['happy', 'pedestrian', 'column', 'fountain', 'temporary', 'praise', 'intelligence', 'fx', 'indecision', 'chickens', 'ethnicity
['dog', 'amazingly', '5+', 'knife', 'himher', 'readable', 'pilates', 'shearers', 'johnboy', 'moreover', 'which']
['movie', 'guarantee', 'sites', 'formed', 'accuracy', 'explicitly', 'universes', 'workouts', 'perspective', 'placing', 'believe
srijan@srijan-Inspiron-3583:~/Courses/INLP/Assignment 3$ 
```

Theory Questions (2.1):

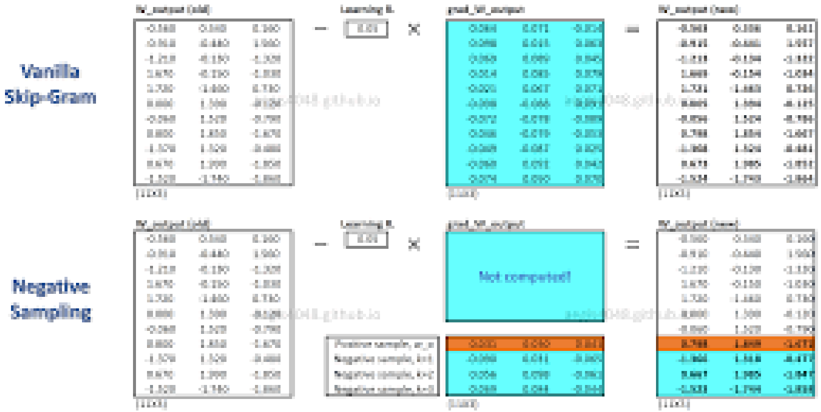
Question 1:

Negative Sampling is a technique used to enhance the computational efficiency and training performance of the Word2Vec model when training word embeddings.

In conventional Word2Vec model training, the objective is to predict the probability of a context word given a target word or vice versa. This necessitates the computation of the softmax function over the entire vocabulary for each training example, which can be computationally costly for large vocabularies.

Negative Sampling approximates this computation by arbitrarily sampling a small subset of words that are not contextually related to the target word and computing the probability that the target word is associated with these negative samples. The Word2Vec model is then trained to maximise the likelihood of correctly identifying true context words and minimise the likelihood of erroneously identifying negative samples.

The model is first initialised with random weights to approximate the Word2Vec training computation using negative sampling. The target word and its context words are then used to compute the neural network's output for each training example. Next, negative samples are selected at random from the vocabulary, and their probabilities are determined. The weights of the neural network are then modified using stochastic gradient descent in order to minimise the loss function, which is a combination of the probability of the true context terms and the negative samples. This procedure is repeated for each example of training until convergence is attained.



Question 2:

Semantic similarity is a measure of how closely related the meaning of two words or phrases is. In natural language processing, word embeddings are often used to represent words in a high-dimensional vector space where words with similar meanings are located closer together.

There are several techniques for measuring semantic similarity using word embeddings. Two common methods are cosine similarity and Euclidean distance.

1. Cosine Similarity: This method measures the cosine angle between two word vectors. If the angle between two word vectors is small, then the cosine similarity value will be high, indicating that the two words are semantically similar. On the other hand, if the angle between two word vectors is large, the cosine similarity value will be low, indicating that the two words are semantically dissimilar.

   Cosine similarity can be calculated with the formula: (A.B)/(|A|.|B|)

2. Euclidean Distance: This method measures the distance between two-word vectors in a multi-dimensional space. If the distance between two-word vectors is small, then the words are semantically similar. On the other hand, if the distance between two-word vectors is large, then the words are semantically dissimilar.

   Euclidean Distance can be calculated with the formula: $\sqrt{(\Sigma(A_i - B_i)^2)}$