NLP Assignment 3 Report

Theory

Q1. Explain negative sampling. How do we approximate the word2vec training computation using this technique?

Negative sampling is a technique used in natural language processing, particularly in training word embedding models like word2vec. It reduces the computational cost of training the model while still maintaining accuracy.

In the standard skip-gram model used by word2vec, the goal is to predict context words given a target word. This involves computing the softmax function over the entire vocabulary, which can be computationally expensive as the vocabulary size grows. Negative sampling addresses this issue by training the model to distinguish the target word from randomly sampled "negative" words that are not in the context.

For each training example (target word, context word), negative sampling samples a fixed number of negative examples (typically around 5-20) from a noise distribution. The noise distribution is usually a simple unigram distribution based on word frequencies. These negative examples are then used to train the model to distinguish the target word from the negative words.

To approximate word2vec training computation using negative sampling, follow these steps:

1. Get the embeddings of the context words and the target word using a randomly initialized embedding matrix.
2. Average the embeddings of the context words to get the context embedding.
3. Calculate the cosine similarity (dot product) between the context embedding and the target embedding.
4. Use the sigmoid function to get the probability of the target word being the correct word for the context words.
5. Calculate the loss using binary cross entropy.

6. Backpropagate the loss to update the weights until convergence is reached.

Q2. 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 is a measure of how similar the meanings of two words or phrases are. Word embeddings are often used to represent words as vectors in a high-dimensional space, where the distance between vectors can be used as a measure for semantic similarity.

Here are two commonly used approaches to measure semantic similarity:

1. Cosine similarity:
   Cosine similarity measures the similarity between two non-zero vectors of an inner product space as the cosine of the angle between them. In the context of word embeddings, the cosine similarity between two word vectors can be used to measure their semantic similarity. The cosine similarity is calculated as follows:

$$\text{cosine similarity}(\mathbf{w}_1, \mathbf{w}_2) = \frac{(\mathbf{w}_1.\mathbf{w}_2)}{(||\mathbf{w}_1||||\mathbf{w}_2||)}$$

   where w1 and w2 are the word embeddings for the two words.

2. Word Mover's Distance (WMD):
   WMD measures the distance between two text documents by taking into account the distances between the individual words in the documents. It can be used to measure the semantic similarity between two phrases or sentences. The WMD between two phrases is calculated by finding the minimum distance between each word in one phrase and every word in the other phrase, weighted by their vector distances. This measure is computed using pre-trained embeddings and requires an optimization algorithm to calculate. The WMD is defined as follows:

$$\text{WMD}(w_1, w_2) = min \sum_i \sum_j d(w_{1i}, w_{2j}) * f(i, j)$$

   where w1 and w2 are the two phrases, d(w1i, w2j) is the Euclidean distance between the word embeddings of the i-th word in w1 and the j-th word in w2, and f(i, j) is a measure of the "flow" between words i and j, which is based on the distance between all pairs of words in w1 and w2.

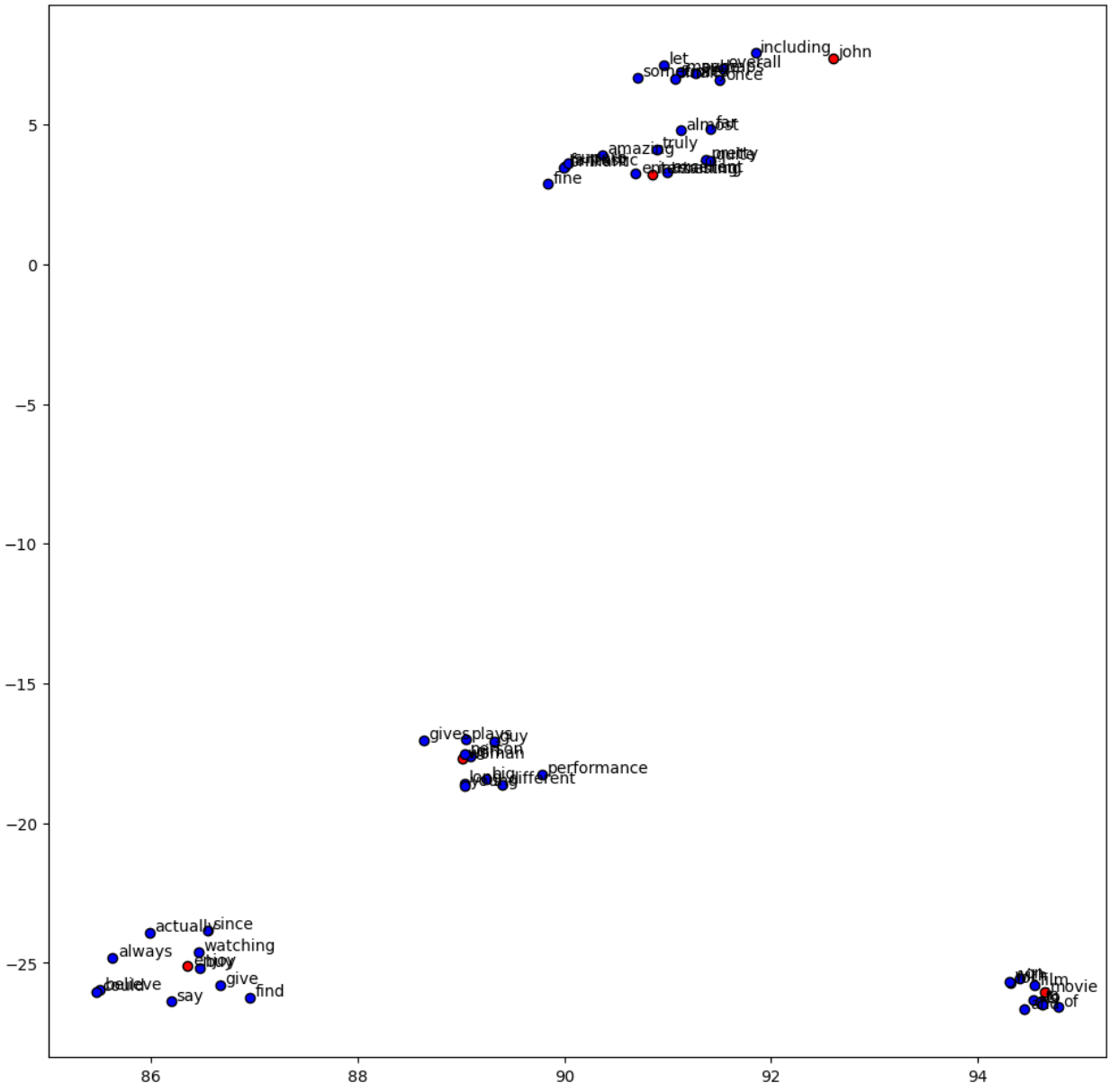
Cosine similarity and WMD are widely used in various NLP applications, such as information retrieval, document clustering, and question-answering systems.

Analysis

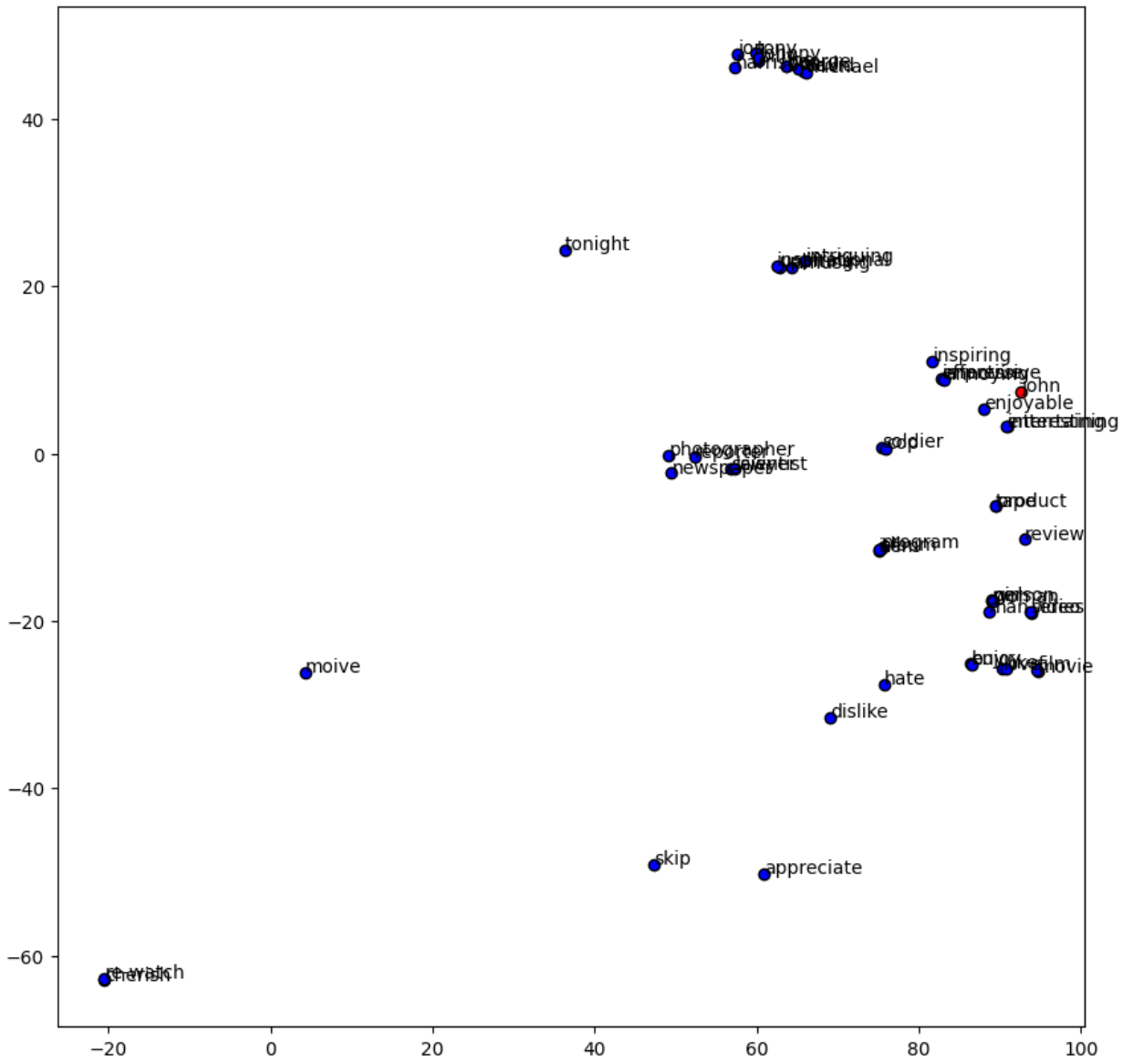
1. Display top 10 vectors for 5 words

SVD Model :

Euclidean


including
john
let
once
almost
truly
amazing
fine
performance
different
actually
since
always
watching
give
find
say
film
movie
of
5
0
−5
−10
−15
−20
−25
86
88
90
92
94

Cosine :




```
The 10 closest (cosine) words to woman are:
man
scientist
soldier
person
photographer
girl
cop
lawyer
reporter
newspaper
The 10 closest (cosine) words to interesting are:
```

```
enjoyable
impressive
intriguing
effective
uplifting
inspiring
entertaining
inspirational
annoying
amusing
The 10 closest (cosine) words to enjoy are:
buy
love
hate
appreciate
cherish
like
dislike
re-watch
skip
tonight
The 10 closest (cosine) words to john are:
joe
jon
harrison
billy
george
david
tony
paul
michael
johnny
The 10 closest (cosine) words to movie are:
video
film
product
program
item
tape
album
series
moive
review
```

Word2Vec Model (model 1) :

```
for woman :
haters    13.481234550476074
wears   12.53993034362793
```

```
pecking   12.535730361938477
girl    12.282276153564453
white   11.942014694213867
betrays   11.890693664550781
enlisted    11.703506469726562
k-nigits    11.626981735229492
young   11.410850524902344
'young    11.325740814208984

for interesting :
textured    13.549972534179688
andersons   13.31401252746582
plotless    11.429566383361816
years.this    11.39962100982666
intriguing    11.233503341674805
very    10.636001586914062
sumptuous   10.63503646850586
poltergeist   10.536170959472656
painfully   10.43954849243164
eye-popping   10.336181640625

for enjoy :
grown-ups   11.728203773498535
gals    11.535062789916992
brassy    11.191598892211914
loath   11.03962230682373
prehistoric   10.600914001464844
timelessly    10.540472030639648
self-doubt    10.210888862609863
analogies   9.657747268676758
dagger    9.644522666931152
pronto    9.56373405456543

for john :
ortberg   31.10955047607422
saxon   30.462928771972656
hancock   30.282419204711914
wayne   30.02106285095215
hinckley    29.893695831298828
sole    29.617042541503906
milius    29.349750518798828
travolta    28.467334747314453
larroquette   27.223203659057617
mctiernan   26.169795989990234

for movie :
i   41.74229049682617
it    40.50118637084961
this    31.16708755493164
you   30.292802810668945
'<pad>'   29.952171325683594
film    28.937746047973633
a   25.20366668701172
;   24.518835067749023
```

```
is    23.531322479248047
,   23.40203857421875
```

Word2vec (model 2) :

```
Most similar words to woman:
1. woman
2. jewish
3. roman
4. black
5. shaw
6. surprisingly
7. files
8. beneath
9. commentary
10. chord
Most similar words to interesting:
1. interesting
2. very
3. important
4. effective
5. educational
6. more
7. in-depth
8. artists
9. attractive
10. disturbing
Most similar words to enjoy:
1. enjoy
2. gentiles
3. back
4. bring
5. christians
6. after
7. loved
8. relax
9. seeing
10. miss
Most similar words to john:
1. john
2. gilbert
3. baptist
4. charlie
5. bay
6. grief
7. prior
8. robert
9. boorman
10. ortberg
```

```
Most similar words to movie:
1. movie
2. film
3. it
4. this
5. i
6. you
7. that
8. but
9. ;
10. <pad>
```

2. 10 Closest words to ‘titanic’

SVD Model :

```
The 10 closest (cosine) words to titanic are:
wild
x
den
halloween
blade
red
rear
casino
braveheart
alias

The 10 closest (Euclidean) words to titanic are:
blade
braveheart
halloween
blues
runner
rear
hornblower
iron
snow
trilogy
```

Word2vec Model 1 :

```
chairman    32.929290771484375
henning   28.767919540405273
fischer   28.385936737060547
caddyshack    27.96255874633789
```

```
rollins   27.829641342163086
parks   26.75896453857422
pieced    26.406631469726562
happenings    26.12584686279297
by-the-numbers    25.86142349243164
jody    25.46945571899414
```

Word2vec Model 2 :

```
1. titanic
2. modes
3. dan
4. fade
5. rewind
6. najimy
7. tennessee
8. metronome
9. netflix
10. sadist
```

GoogleNews pre-trained embeddings :

```
[('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)]
```

We can clearly see that the context of the meaning of ‘titanic’ in the pre-trianed model is completely different. In our models, we see that the meaning is in the context of it being the name of a movie. Which is why the closest words are different.

Note : due to lack of compute, the cbow was only able to train on 60k sentences. thus its performance is not at its maximum potential.

Observation :

In the 2D plot of the embeddings, if euclidean distance is used as the measure of similarity, then the 10 closest words for each word appear in a pocket around it. When cosine similarity is used as the measure, then, the closest words appear in straight lines along the word vector.