Introduction to NLP
Assignment-3

2.1

1) Negative sampling

Negative sampling is a technique used in training word embeddings like word2vec, which aims to approximate the training process with a much smaller number of samples than traditional methods. The goal of negative sampling is to improve training efficiency by reducing the number of samples needed to update the word vectors.In negative sampling, instead of treating every word in the vocabulary as a positive example for a given target word, we treat only a small subset of words as negative examples. These negative examples are randomly sampled from the vocabulary, and their goal is to provide contrastive information for the positive examples.More specifically, the negative sampling objective is to maximise the probability of the context word given the centre word, and minimise the probability of negative samples given the centre word. To achieve this, the model is trained to predict the context word given the centre word, and to predict that the negative samples are not related to the centre word. This is done by updating the embeddings in such a way that the dot product between the centre and context vectors is maximised, while the dot product between the centre and negative vectors is minimized.in terms of computation, negative sampling allows us to approximate the training process with fewer samples by only updating a small subset of the weights during each iteration. Specifically, we only need to update the weights for the positive and negative samples, which are a much smaller set than the full vocabulary. This can significantly reduce the computational cost of training, especially for large vocabularies.Overall, negative sampling is an effective and efficient way to train word embeddings like word2vec. By focusing on a small subset of negative examples, we can reduce the computational cost of training while still obtaining high-quality word embeddings.

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 is the degree of relatedness between two words or phrases based on their meaning. It measures how similar the concepts represented by the words are, regardless of their surface form or context. For example, "cat" and "dog" are semantically similar because they both represent common household pets, whereas "cat" and "car" are not similar because they

represent different concepts.There are many techniques for measuring semantic similarity using word embeddings common one is cosine similarity and Jaccard similarity.

Cosine similarity computes the cosine of the angle between the vectors of two words, where a higher cosine similarity indicates a greater semantic similarity between the two words.Jaccard similarity is a measure of the similarity between two sets. In the context of word embeddings, the Jaccard similarity between two words is defined as the intersection of the sets of words that co-occur with each word, divided by the union of the sets. In other words, it measures the proportion of shared context between two words. The Jaccard similarity ranges from 0 (no shared context) to 1 (identical context), and can be used as a simple and computationally efficient measure of semantic similarity.

2.3 Analysis

1)Display the top-10 word vectors for five different words (a combination of nouns, verbs, adjectives, etc.)using t-SNE (or such methods) on a 2D plot.

Words considered wife,movies,looked,slowly,titanic
This are top 10 words obtained using frequency-based method(svd)
Wife
Husband 0.985487188157307
sister 0.9828463935650195
girlfriend 0.9718198862188344
fiancee 0.9700866510943292
grandmother 0.9658922875815549
guidance 0.9582280177749865
boyfriend 0.9581529658305455
fiance 0.9553508794280906
tortured 0.9535062468558226
marie 0.9528629591421948

Titanic
commentaries 0.9881415507595523
villains 0.9840770447270877
bikes 0.9828140040497172
design 0.9827990518778984
tapping 0.9823525519286914
cameras 0.9813541662461548
upstairs 0.9811869103596048

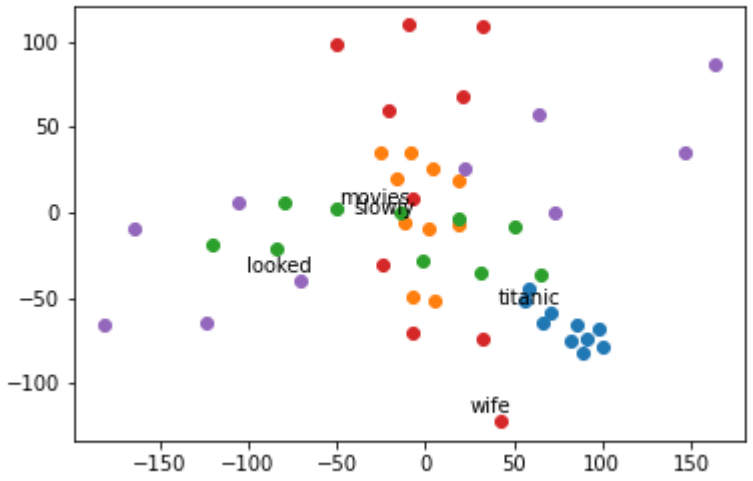
spirits 0.980278726714613
cramped 0.9802369903447897
whipping 0.9799018105263257

looked
stopped 0.9799577184250623
worked 0.9718543795974616
skipped 0.9609941191888641
ended 0.9587844726389516
started 0.9575905207903478
suggested 0.9566271235076385
kept 0.956351199964012
stayed 0.9517456419787269
regularly 0.9514797723523446
immediately 0.947578568942189

Movies
westerns 0.9883329732705343
films 0.985016510733088
dvds 0.9845896432727335
videos 0.979581577532621
classics 0.9768051881612617
productions 0.9746669949678277
requirements 0.9713967758982741
hearing 0.9692460502871707
crap 0.9687144607196418
specials 0.9671905135494688

Slowly
handed 0.9728916267679668
gradually 0.967944937252596
rushed 0.9658522676519956
clean 0.965740158129507
faster 0.9644591714969256
vulnerable 0.9599982811645812
threatening 0.9567234028501485
warm 0.9550595749003752
lazy 0.9534778323685218
convincingly 0.9525483961541933

This is plot for 5 words top 10 words obtained from svd model plotted using t-SNE on a 2D plot.



This are the 10 top words obtained using CBOW with negative sampling
Wife
mother 0.8421778619983942
son 0.8381759153545235
daughter 0.8303794070235114
father 0.8187882483182476
grandmother 0.8183667357057584
head 0.7893419530565484
cousin 0.7847278443386525
friend 0.7791003552564401
year 0.7771843254373153
dad 0.7760963752227213

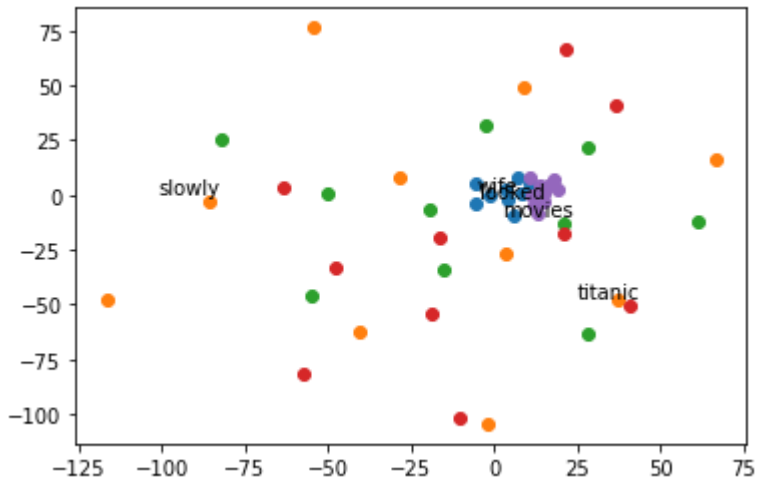
Movies
films 0.8715889749288915
scenes 0.8481934721756851
actors 0.8307508770191535
people 0.8091008726223698
things 0.7946318775619157
characters 0.7899318478287267
effects 0.7775506855013022
reviews 0.7706203836526727
moments 0.7651478353709072
three 0.7647242681522949

Titanic
specifically 0.780895079940175
pressure 0.7676086141115588
murderous 0.7640468278308074
legal 0.7618817863286271
attack 0.7545885096191082
flashy 0.7530739594899807
jimmy 0.752766156306393
representing 0.7526304086710389
birth 0.7414942088636557
enduring 0.7405153082179541

Looked
got 0.7499830681689217
- 0.7331721554487804
experienced 0.7301137699925524
played 0.729416209534631
started 0.7227574476920754
given 0.721132168084981
was 0.716513360370569
been 0.7144200179547083
look 0.7131752838200649
done 0.7109472434026929

Slowly
refreshing 0.7131013061866629
pleasant 0.7010440267092884
scream 0.6970864794007708
tear 0.6901650341005443
beside 0.6892575281467328
lively 0.6854865505389988
pilot 0.6848231135064464
funeral 0.6842089289980562
(julia 0.6823797109582035
everywhere 0.6780081835765736

Plot obtained using CBOW model



2)What are the top 10 closest words for the word ‘titanic’ in the embeddings generated by your program

Top 10 word embeddings for word titanic using svd
commentaries 0.9881415507595523
villains 0.9840770447270877
bikes 0.9828140040497172
design 0.9827990518778984
tapping 0.9823525519286914
cameras 0.9813541662461548
upstairs 0.9811869103596048
spirits 0.980278726714613
cramped 0.9802369903447897
whipping 0.9799018105263257
Top 10 word embeddings for word titanic using CBOW
specifically 0.780895079940175
pressure 0.7676086141115588
murderous 0.7640468278308074
legal 0.7618817863286271
attack 0.7545885096191082
flashy 0.7530739594899807
jimmy 0.752766156306393
representing 0.7526304086710389
birth 0.7414942088636557
enduring 0.7405153082179541

Top 10 word embeddings for word titanic using pre-trained gensim model

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

Models comparison

gensim(pretrained model)>svd(frequency-based model)>CBOW with negative sampling(prediction based model)

We can observe that gensim pretrained model gave words which are making some sense because that model is trained on huge data set as compared to svd and CBOW models .Both svd and CBOW models trained approximately nearly 45k sentences .compared to data set which is used for training gensim it is too small.After we got good results with svd embedded model i.e frequency based model reason lies same as the training data is less CBOW model didn't performed well compared to svd model.we can also observe svd gave good results for 'wife' and 'movies'.for wife it gave very similar words like husband ,sister,fiance all given words are relevant to wife .with cbow also it gave good results but there are some irrelevant words like year in top 10 .similarly for movies also svd performed really well compared to cbow with negative sampling.model performances depends on what we are doing and on size of dataset ,as our train data is 45k sentences frequency based performed well if we train on large dataset may be CBOW will perform better than svd.