Intro to NLP

Assignment -3

Suyash Sethia

Theory

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

What is Negative Sampling?

Negative sampling is a technique used in Word2Vec models to train word embeddings by taking a simplified approach to the computation of word co-occurrences. Instead of considering all possible words in the vocabulary, negative sampling randomly selects a small subset of words, often referred to as negative samples, that do not appear in the context of a given word. The objective of negative sampling is to approximate the probability distribution of word co-occurrences while reducing the overall computational cost.

How does Negative Sampling work in Word2Vec?

In Word2Vec, negative sampling modifies the loss function of the Skip-gram model. The loss function for the Skip-gram model is defined as the log-likelihood of the context words given the target word. Negative sampling replaces this loss function with a new objective function that tries to differentiate between the true context words and randomly selected negative samples. The model then tries to maximize the probability of the true context words and minimize the probability of the randomly selected negative samples.

this gives us a major advantage of computational cost and also allows for better handling of rare words and reduces the dominance of frequent words in the embeddings .

negative sampling can lead to better quality embeddings by reducing the noise introduced by irrelevant words.

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

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

Understanding Semantic Similarity using Word Embeddings

What is Semantic Similarity?

Semantic similarity is the measure of how similar two words or phrases are in terms of meaning. It is a fundamental concept in NLP and is used in various applications such as information retrieval, question answering, and machine translation. For instance, consider the words "car" and "automobile." These two words are semantically similar, as they both refer to a vehicle used for transportation. On the other hand, the words "car" and "banana" are not semantically similar, as they have completely different meanings.

Measuring Semantic Similarity using Word Embeddings

Word embeddings are used to measure semantic similarity between words. Word embeddings represent words as dense vectors in a high-dimensional space, where

words with similar meanings are closer to each other. This allows us to use mathematical operations such as cosine similarity to measure the semantic similarity between words.

Technique 1: Cosine Similarity

Cosine similarity is a widely used technique for measuring semantic similarity between words using word embeddings. It is based on the cosine of the angle between the two vectors representing the words. The cosine similarity ranges from -1 to 1, where a value of 1 indicates that the words are identical, a value of 0 indicates that the words are completely unrelated, and a value of -1 indicates that the words are opposites.

For instance, consider the words "car" and "automobile" represented as vectors in a high-dimensional space. The cosine similarity between these two vectors would be close to 1, indicating that these words are highly semantically similar.

Technique 2: Euclidean Distance

Euclidean distance is another technique used to measure semantic similarity between words using word embeddings. It measures the distance between the two vectors representing the words in a high-dimensional space. The smaller the distance between the two vectors, the more semantically similar the words are.

For instance, consider the words "car" and "automobile" represented as vectors in a high-dimensional space. The Euclidean distance between these two vectors would be small, indicating that these words are highly semantically similar.

Word Embeddings using Singular value decomposition

To create word embeddings using SVD, a co-occurrence matrix is first constructed based on the frequency of word co-occurrences in a given corpus. The matrix is then factorized using SVD to obtain a low-rank approximation of the original matrix. The resulting embedding matrix consists of rows representing individual words and columns

representing the embedding dimensions. These dimensions capture underlying semantic relationships between words and can be used for downstream NLP tasks.

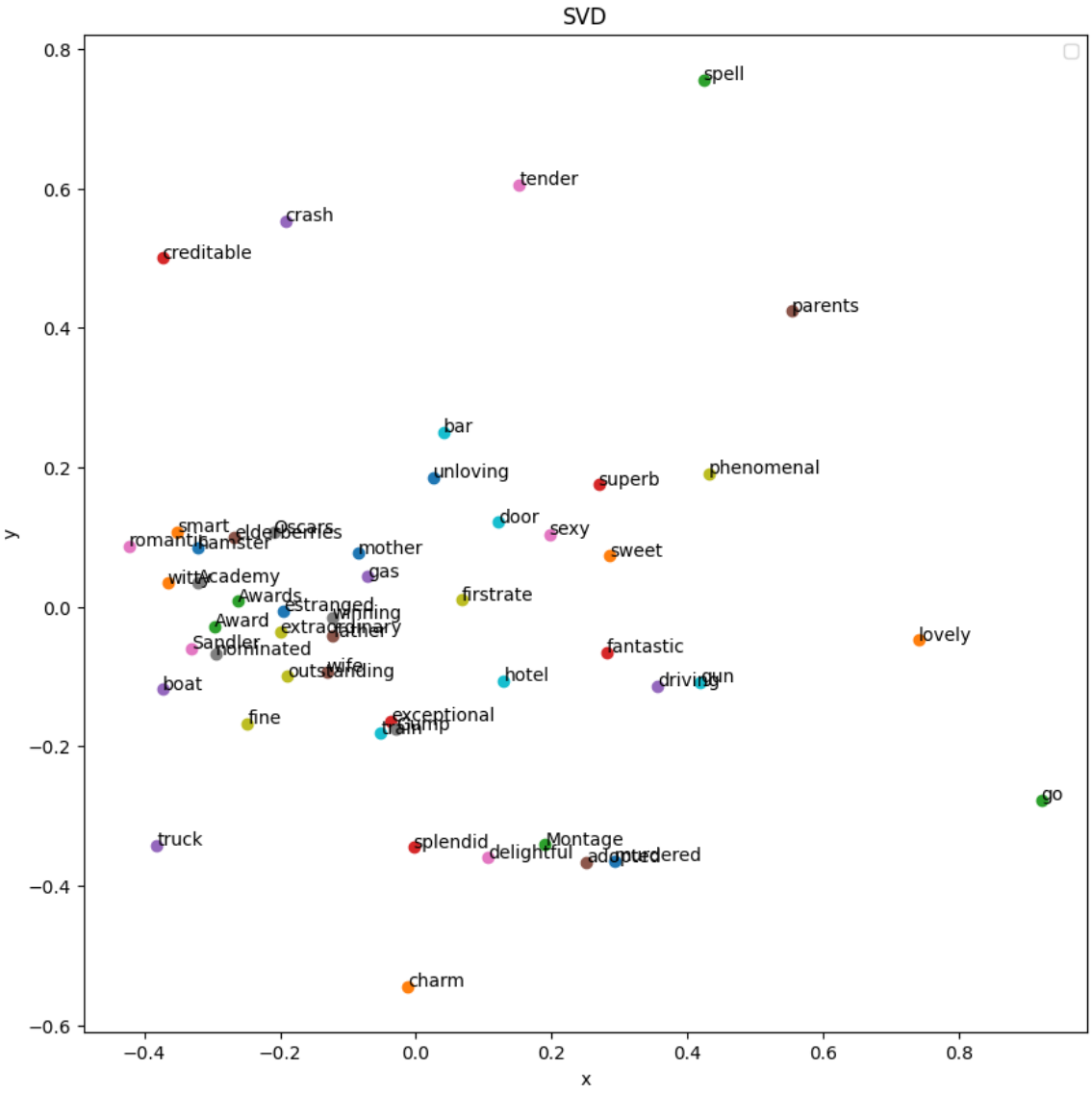
SVD-based word embeddings have been shown to perform well on tasks such as word similarity and analogy detection. However, they are often outperformed by more recent techniques such as Word2Vec and GloVe, which use neural networks to learn word embeddings directly from the corpus.

Report

The words selected are

1. child
2. charming
3. win
4. excellent
5. car

ten words closest to the above words shown below



Top ten vectors closest to Titanic in

Note

- This model has been trained using Singular Value Decomposition on the Co-occurrence matrix.
- The saved models are in the directory: ./svd_vectors.txt.
- The graph above is the result of TSNE the top 10 closest words to selected words

- the original vectors had 97 dimensions but for the sake of representation they were compressed to 2 using PCA
- **Hyper-parameters**
    1. Window size = 4
    2. Number of Sentences used = 80,000
    3. Min frequency of tokens = 5
    4. Number of dimensions of vectors = 97

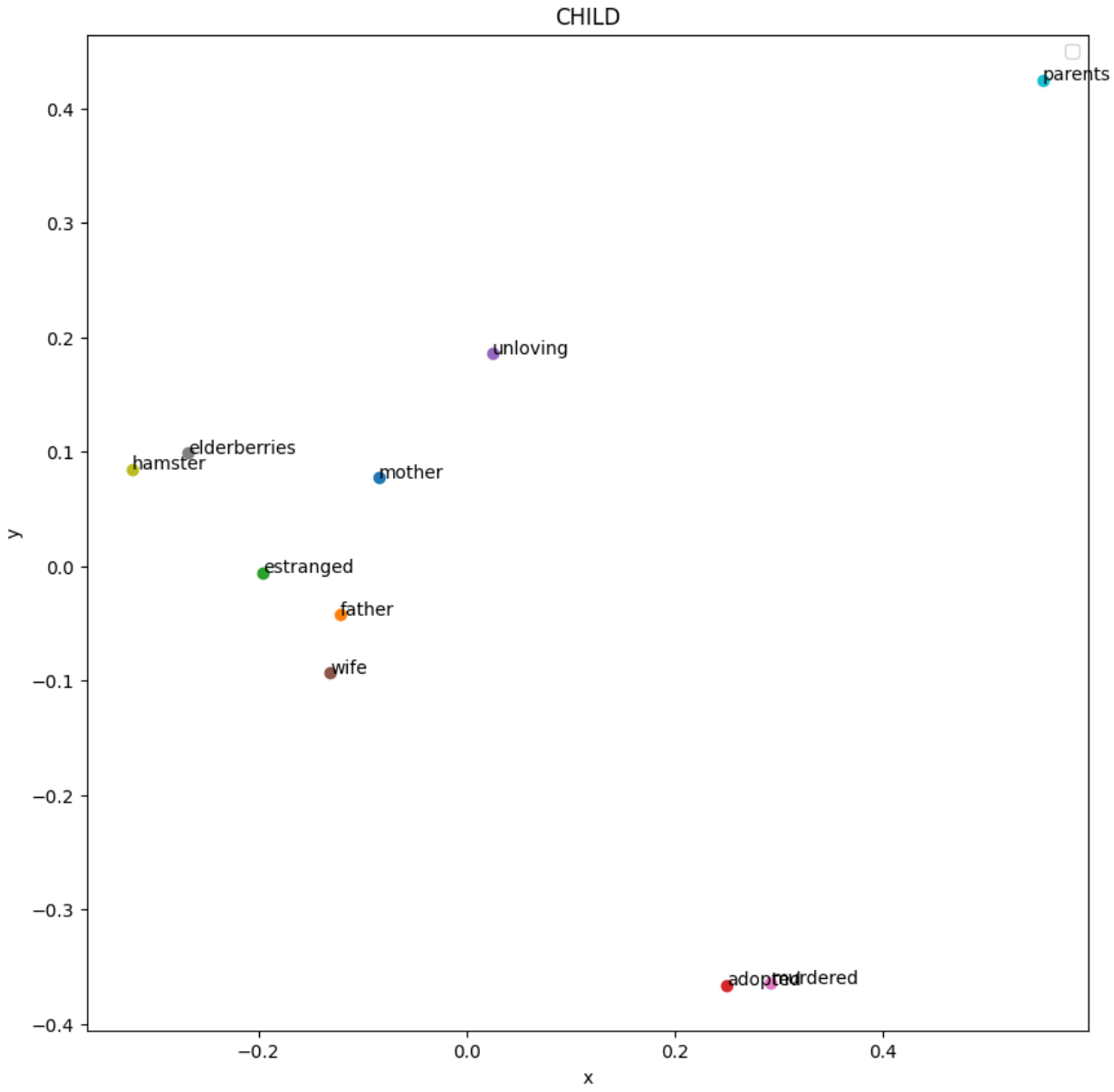
- Files
    - SVD folder
        - svd.py :- contains the code
        - run using

```
python3 svd.py
```

        - svd_vectors.txt :- contains all the vectors formed by running the code
        - sentences.json :- sentences extracted from reviews_Movies_and_TV.json
        - reviews_Movies_and_TV.json :- original data
        -

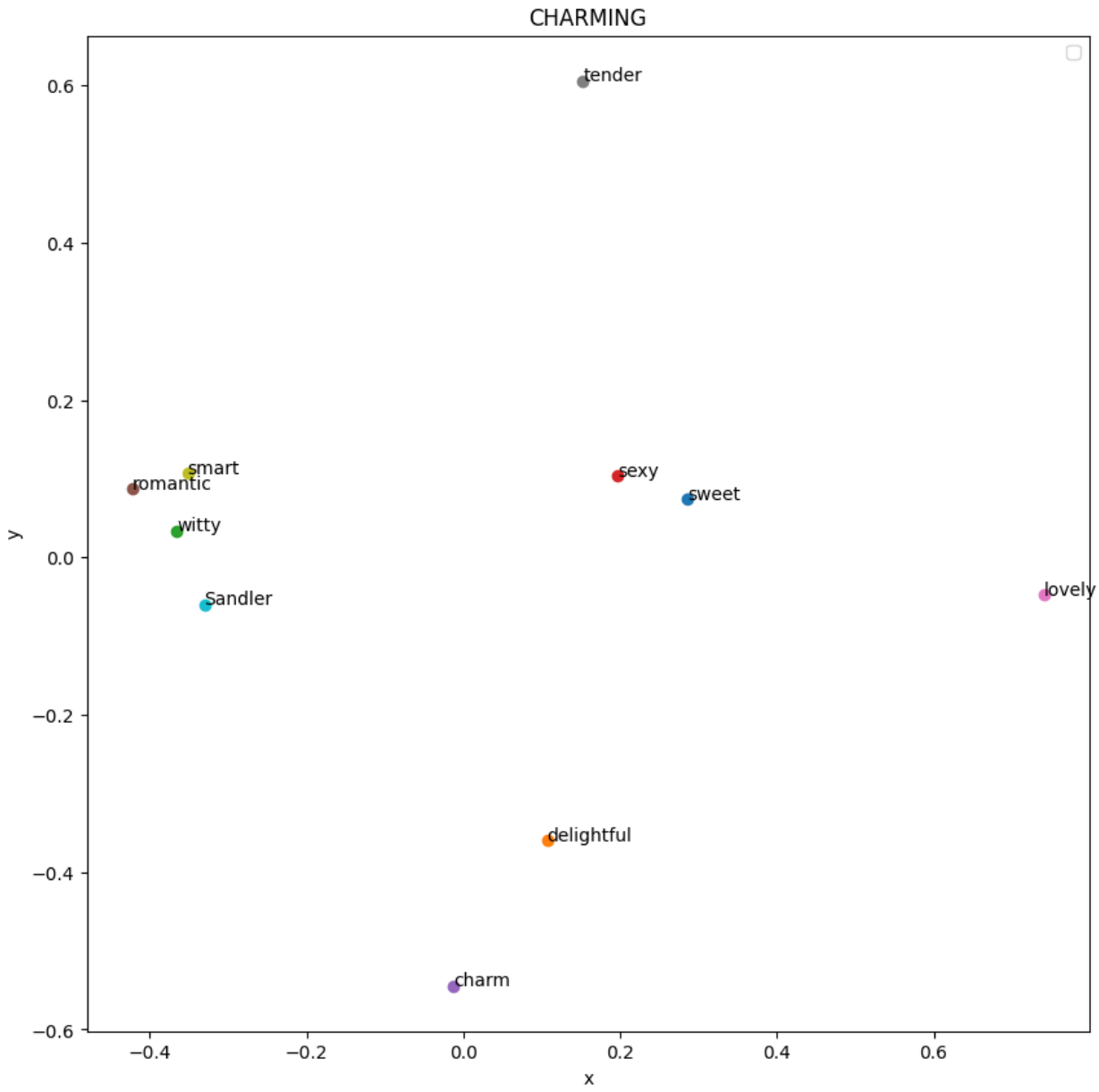
Closest words for Child

	word	Cosine similarity
0	mother	0.766916
1	father	0.752409
2	estranged	0.689101
3	adopted	0.683729
4	unloving	0.677120
5	wife	0.674606
6	murdered	0.674258
7	elderberries	0.657486
8	hamster	0.648585
9	parents	0.639331



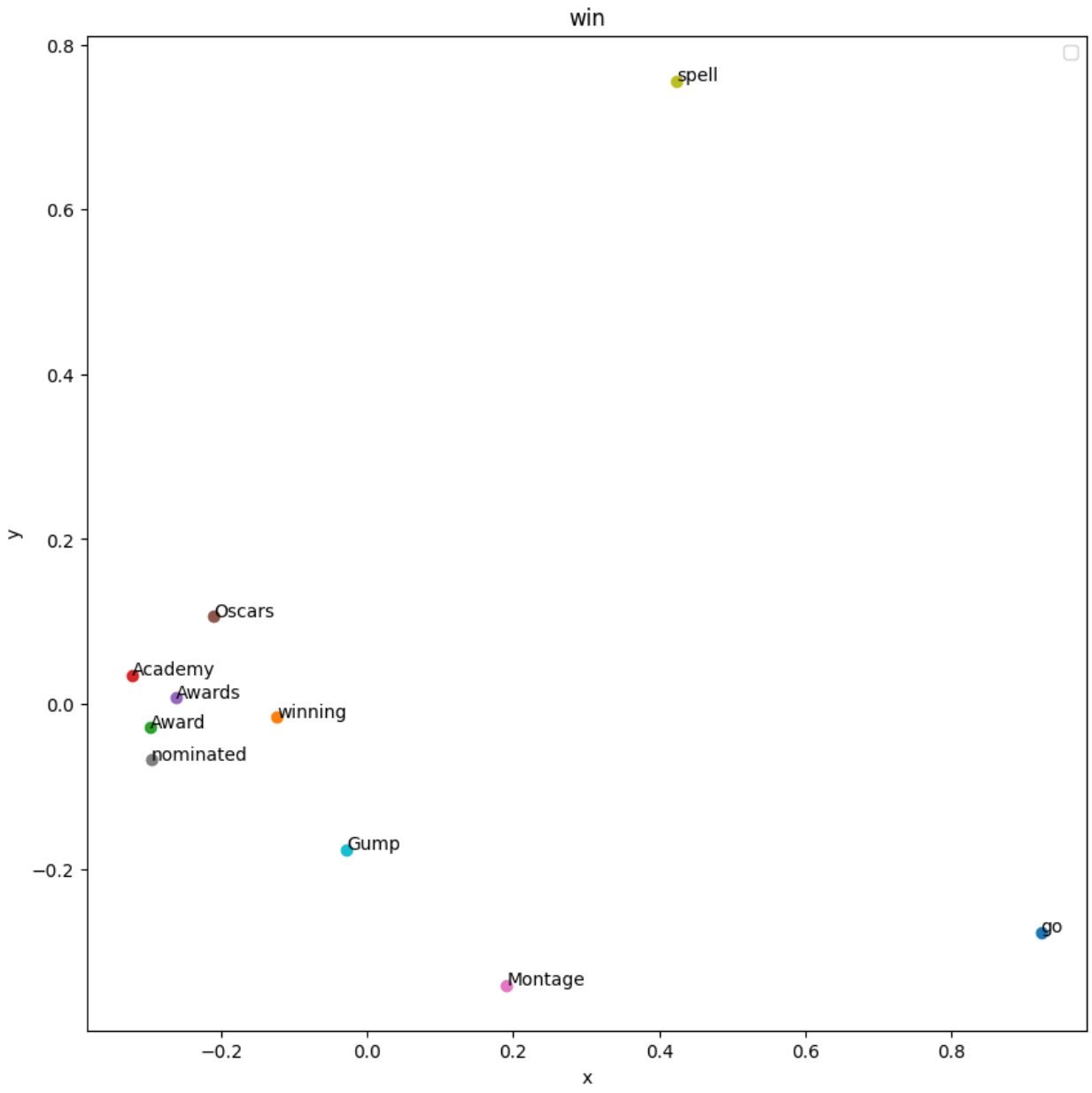
Closest words for charming

	word	Cosine similarity
0	sweet	0.693972
1	delightful	0.680442
2	witty	0.665052
3	sexy	0.649164
4	charm	0.649083
5	romantic	0.642438
6	lovely	0.618233
7	tender	0.616535
8	smart	0.595139
9	Sandler	0.594738



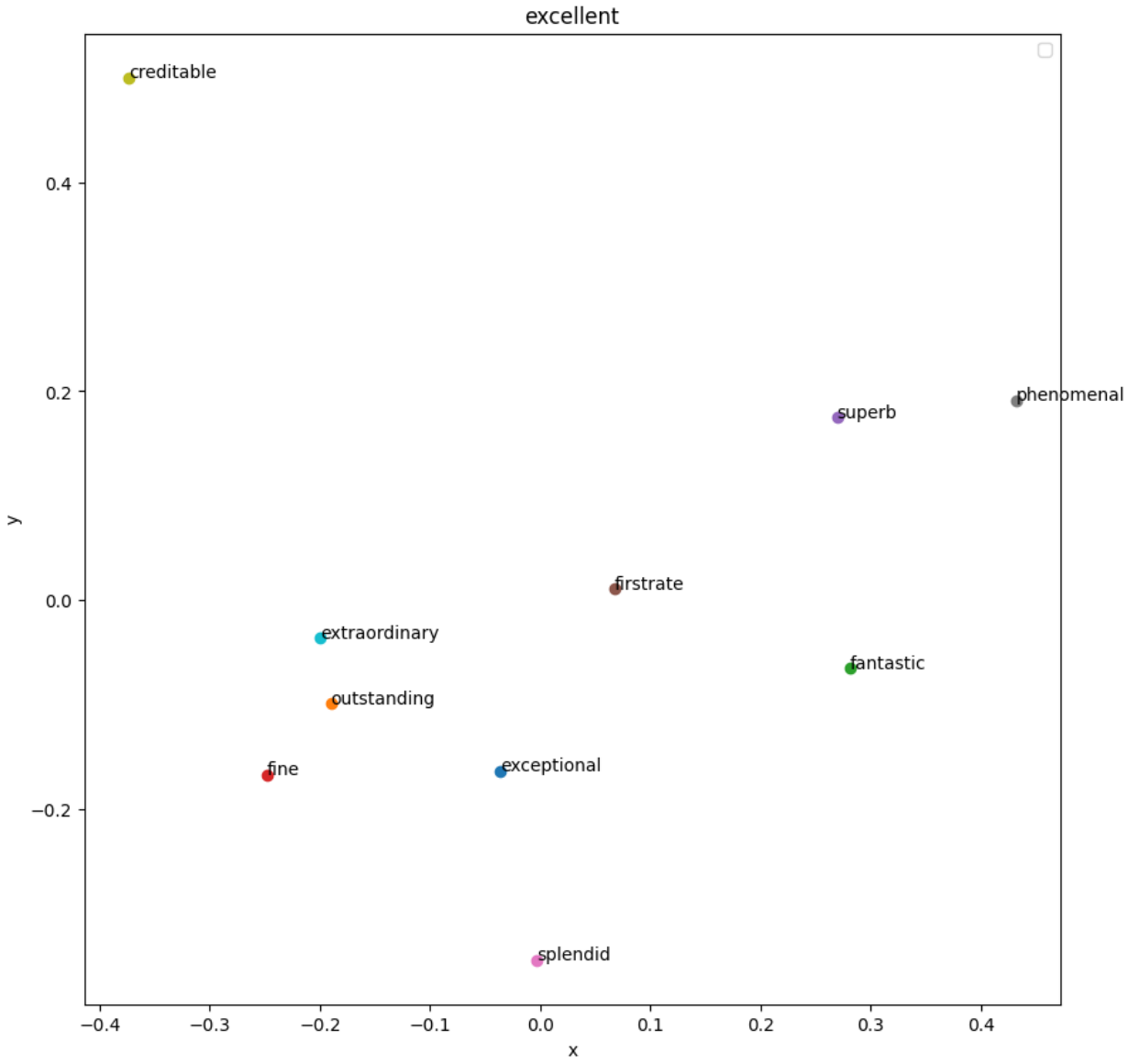
Closest words for win

	word	Cosine similarity
0	go	0.578261
1	winning	0.559174
2	Award	0.552178
3	Academy	0.548510
4	Awards	0.535574
5	Oscars	0.530353
6	Montage	0.528429
7	nominated	0.525669
8	spell	0.523016
9	Gump	0.521578



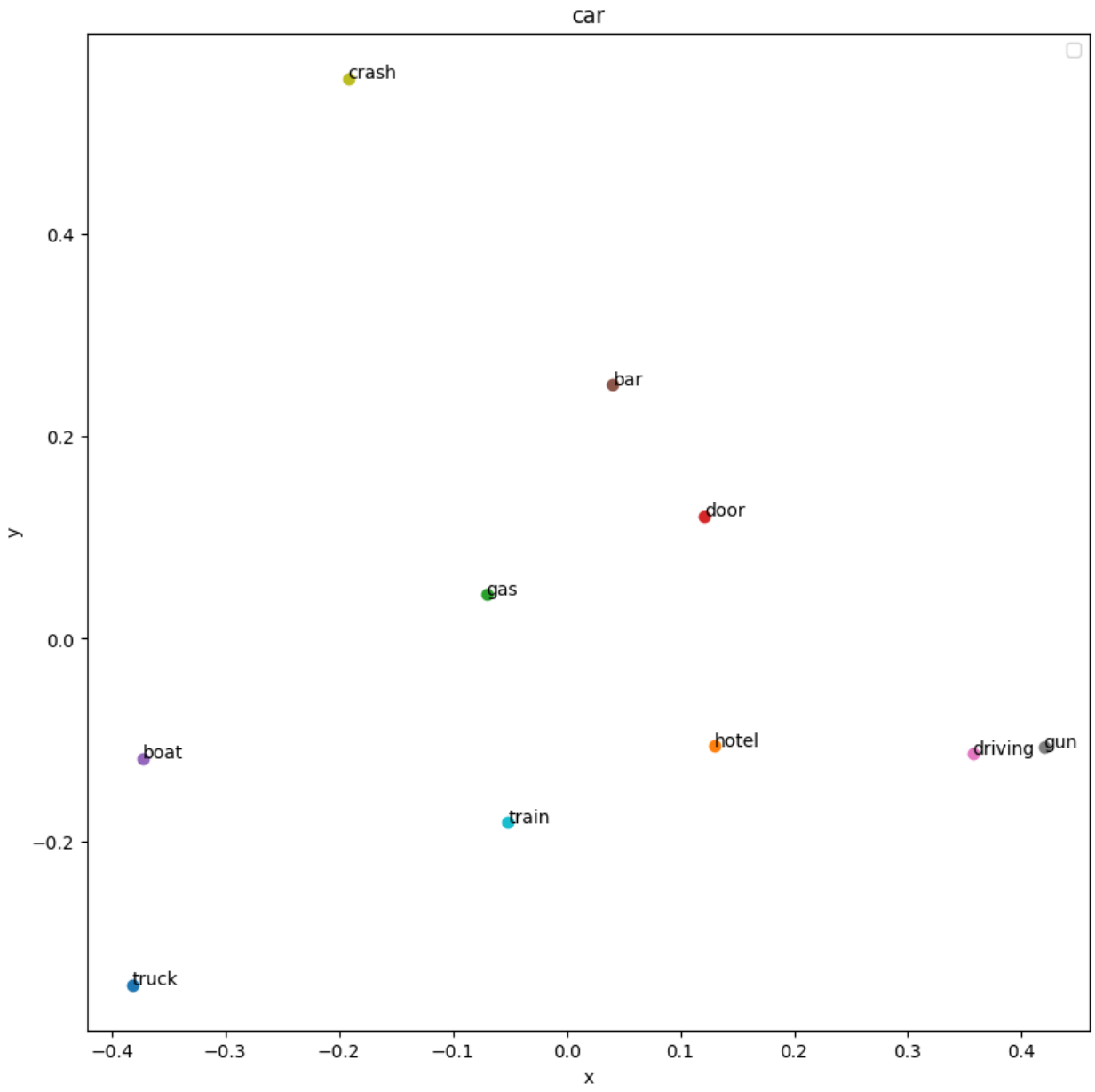
Closest words for excellent

	word	Cosine similarity
0	exceptional	0.838643
1	outstanding	0.809171
2	fantastic	0.787451
3	fine	0.772042
4	superb	0.733758
5	firstrate	0.720622
6	splendid	0.696360
7	phenomenal	0.668560
8	creditable	0.666718
9	extraordinary	0.658123



Closest words for car

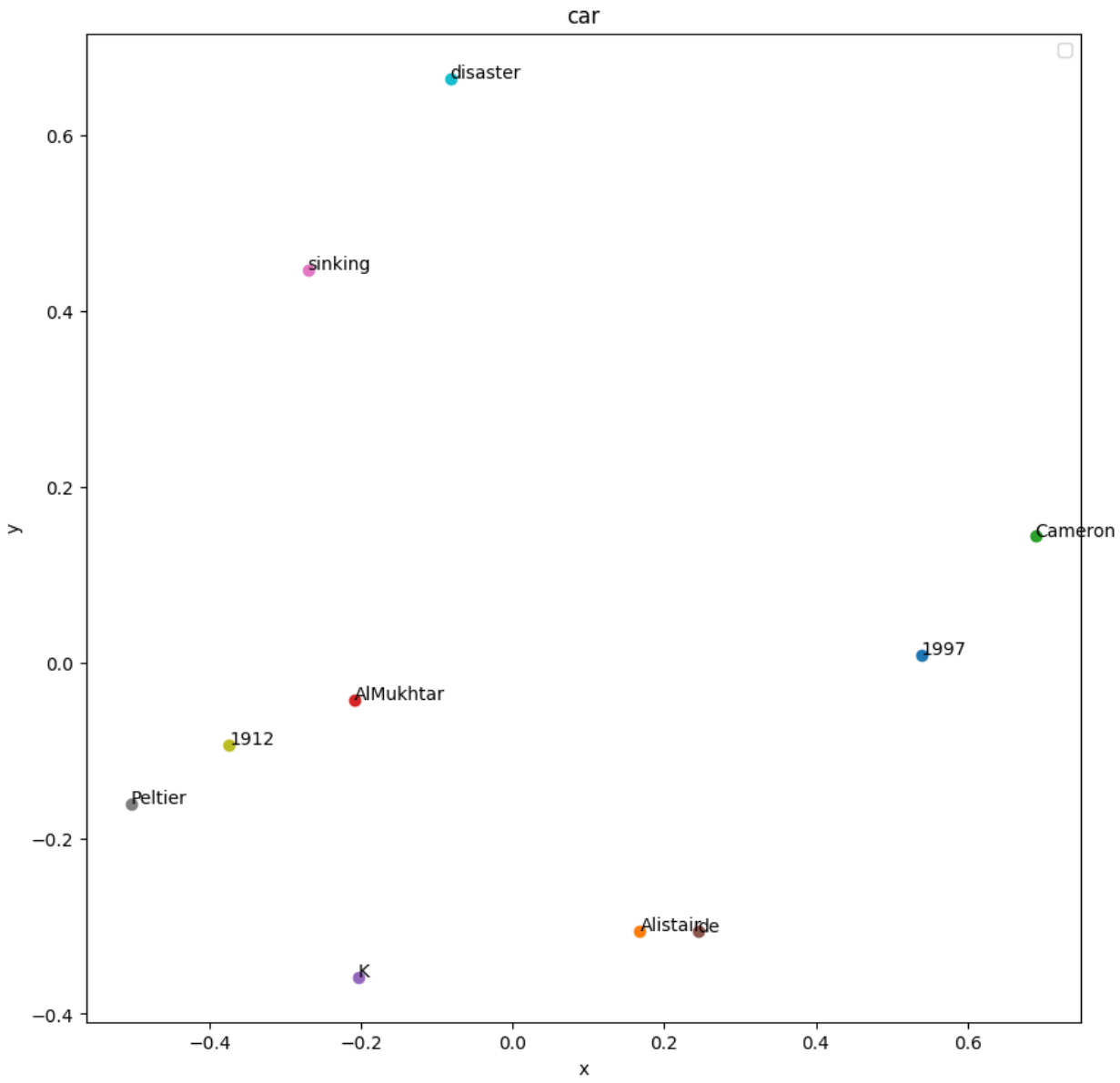
	word	Cosine similarity
0	truck	0.736210
1	hotel	0.726858
2	gas	0.714591
3	door	0.711838
4	boat	0.703883
5	bar	0.699623
6	driving	0.689875
7	gun	0.688375
8	crash	0.686675
9	train	0.679839



Titanic

The 10 closest words given by SVD embeddings are

	word	Cosine similarity
0	1997	0.534644
1	Alistair	0.501210
2	Cameron	0.496096
3	AlMukhtar	0.494600
4	K	0.492968
5	de	0.490246
6	sinking	0.489657
7	Peltier	0.478327
8	1912	0.477442
9	disaster	0.474922



The Result given is quite satisfactory given the fact that this is a movie review corpus so the results are related to both Titanic Ship and the Movie

- Sinking ,Disaster :- represents the Sinking of Titanic
- Cameron :- Director of the Titanic movie

- 1997 :- release year of the movie titanic
- alistiar :- richest man who died in titanic
- 1919 :- year of sinking of the Titanic

I chose the gensim's glove vectors trained on the `glove-wiki-gigaword-50` corpus. The top 10 closest words for the gensim model were :

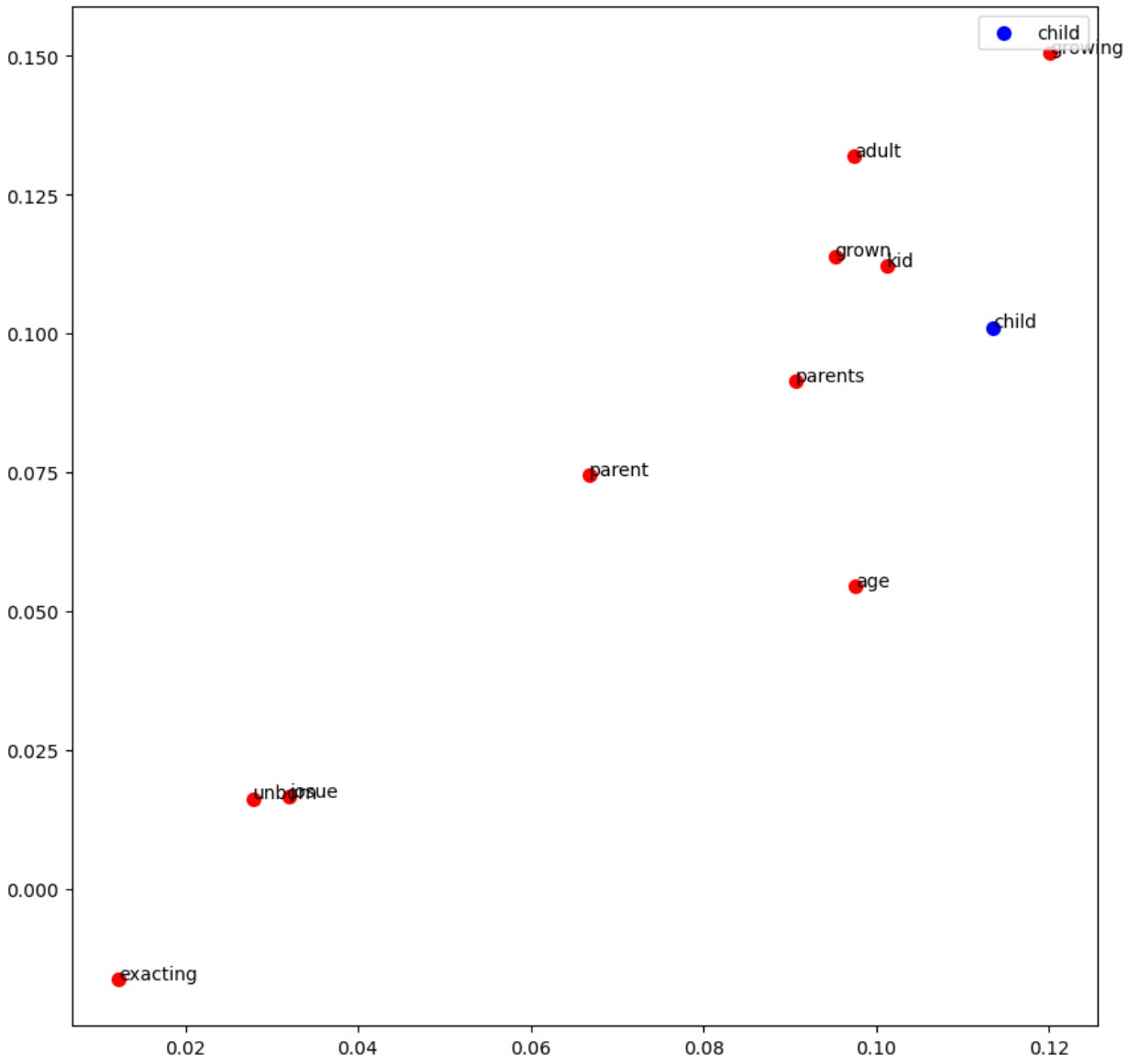
'odyssey', 0.65303
'phantom', 0.65100
'doomed', 0.64136
'r.m.s.', 0.63026
'cinderella', 0.62629
'voyager', 0.62269
'wreck', 0.60453
'ghost', 0.59909
'horror', 0.59604
'tragedy', 0.59548

Continuous bag of words with negative Sampling

Files

Closest words for child

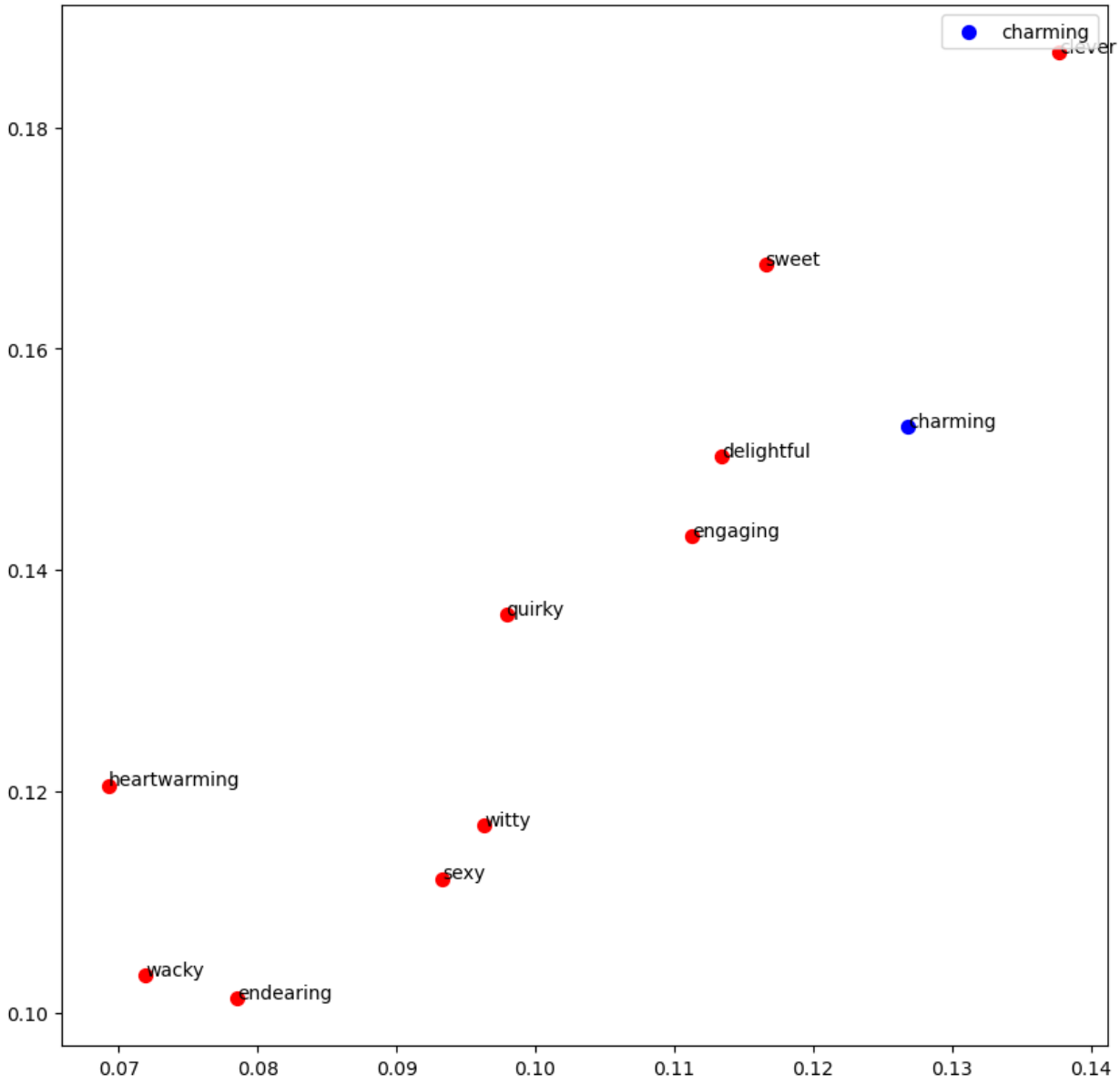
	word	similarity
0	parent	0.6206430025747933
1	kid	0.6016792141279749
2	adult	0.6015611219947066
3	parents	0.5438939369642553
4	growing	0.5320784415561689
5	grown	0.5078449438001665
6	josue	0.477609076728625
7	age	0.47569880532326964
8	exacting	0.47348621761033016
9	unborn	0.4659809906757655



Charming

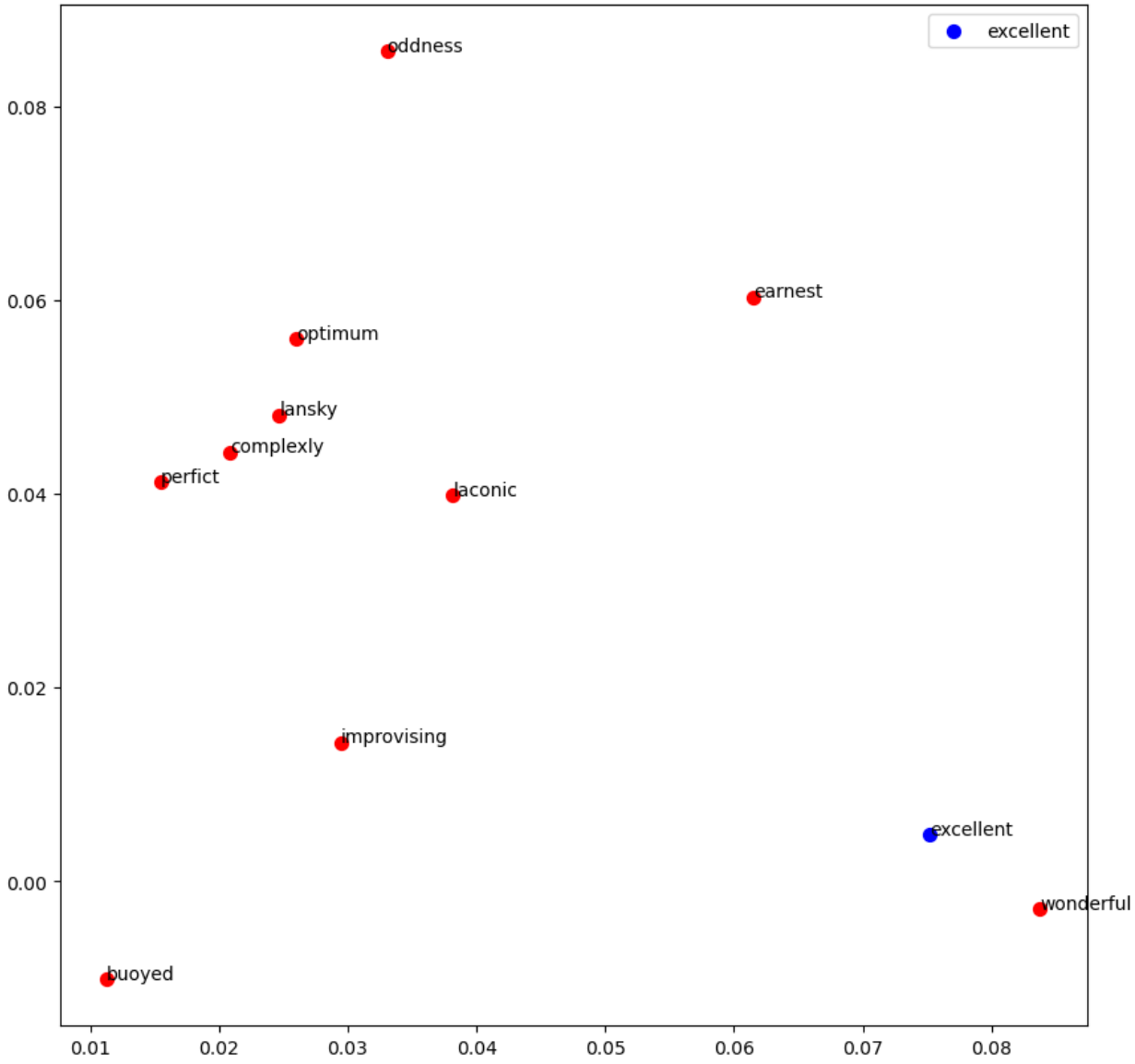
word	similarity
0 witty	0.6257693656740175
1 delightful	0.6067004653955097
2 sweet	0.5221672480871558
3 sexy	0.5211201008495645
4 quirky	0.5093167354040316
5 wacky	0.5081041579494683

6 heartwarming 0.4933603111523638
7 engaging 0.48672213257763486
8 clever 0.4788328678410152
9 endearing 0.47482741451726873



Excellent

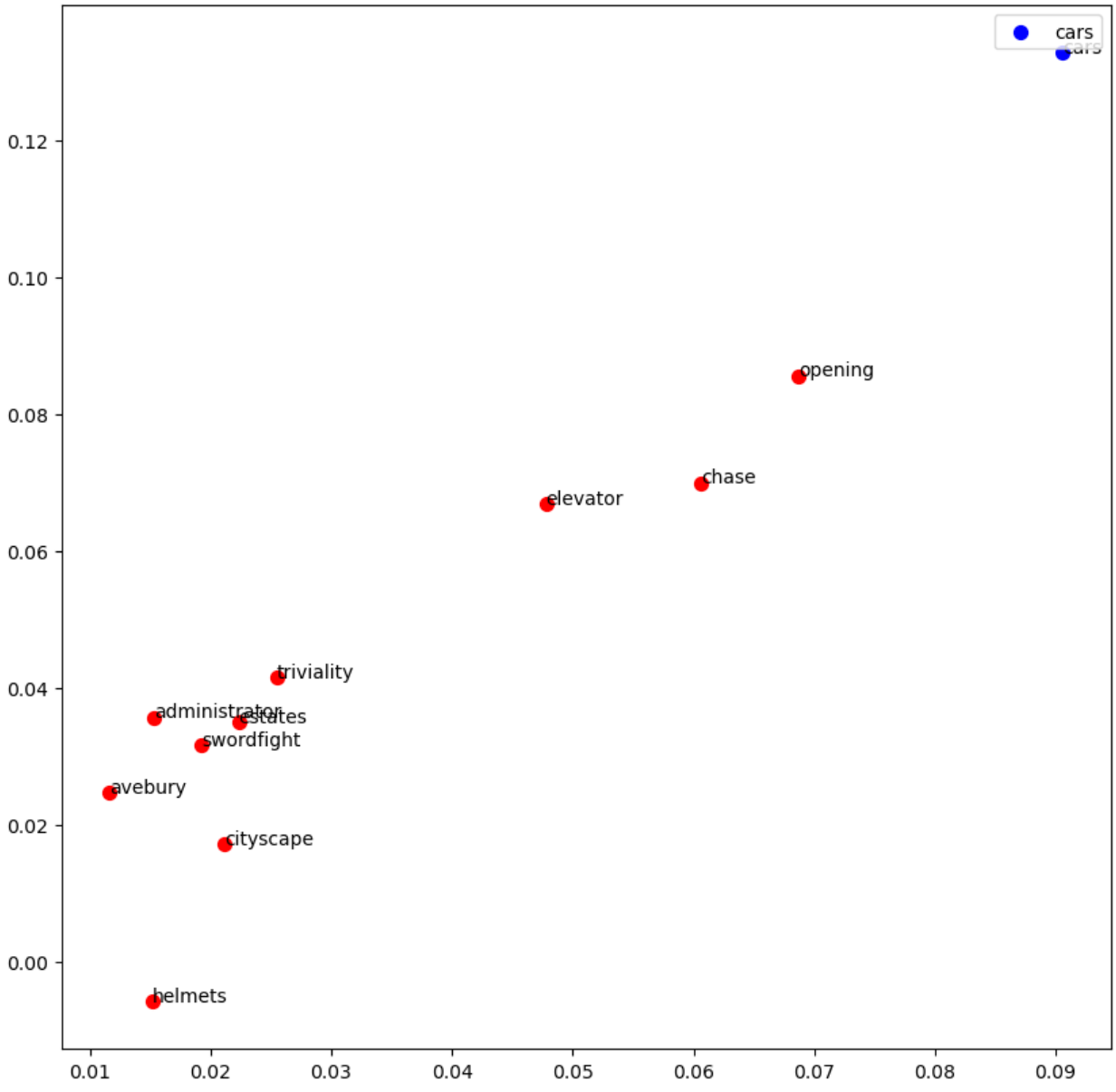
	word	similarity
0	wonderful	0.35335085391821247
1	buoyed	0.314055797860157
2	complexly	0.31078500505767626
3	earnest	0.30929803108750503
4	optimum	0.2934337304355848
5	lansky	0.29312556911323434
6	perfict	0.2921003757885086
7	laconic	0.2919102038504406
8	oddness	0.29114723736738596
9	improvising	0.2908489933535261



CARS

0	avebury	0.5257515889389852
1	swordfight	0.5214027666482685
2	cityscape	0.5164964869323908
3	elevator	0.5102079255742378
4	helmets	0.496899901480639
5	triviality	0.4894300078035356
6	chase	0.48897280221610867
7	estates	0.48518521091024197

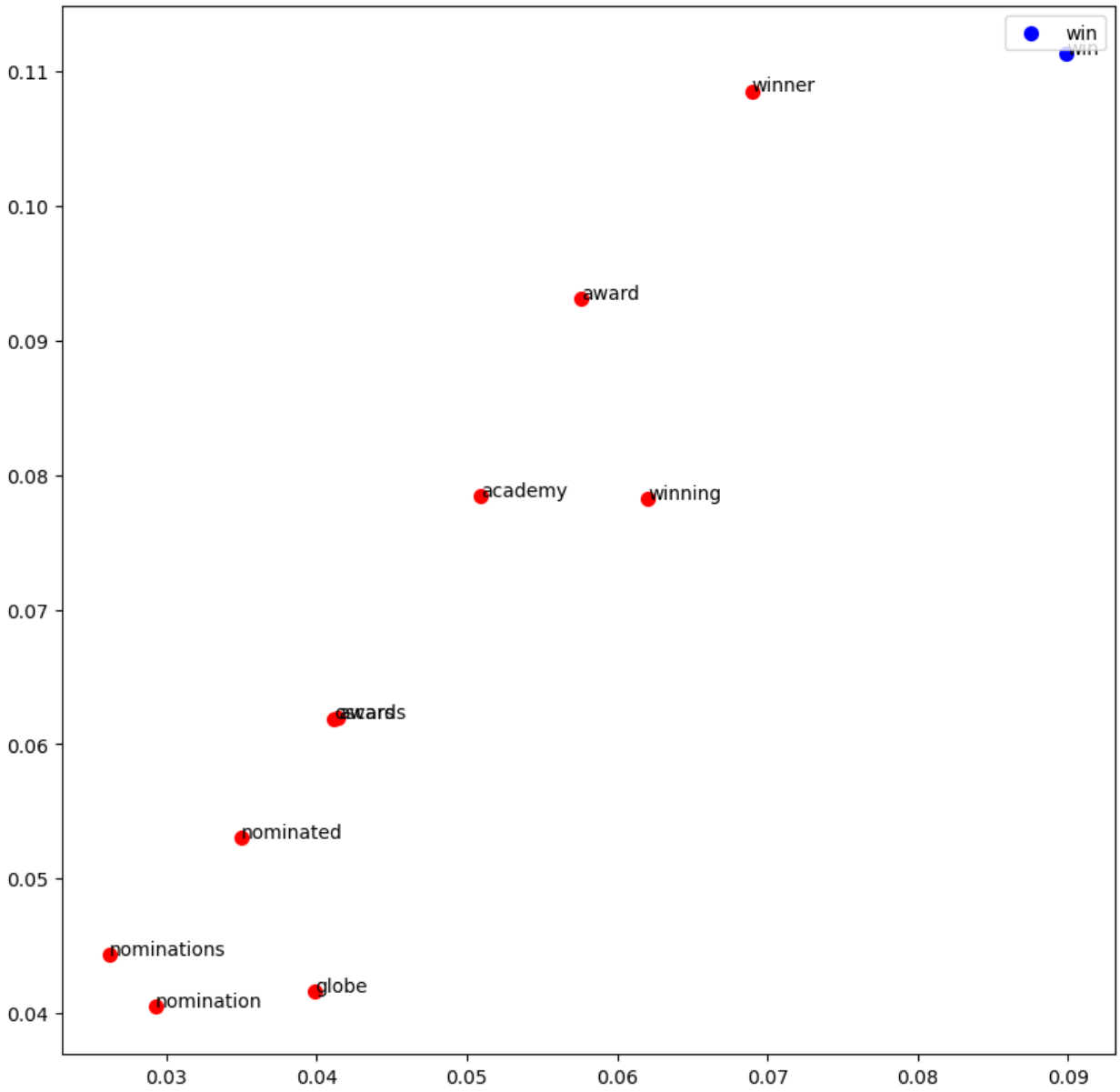
8 administrator 0.482178388888763537
9 opening 0.48169709084071455



WIN

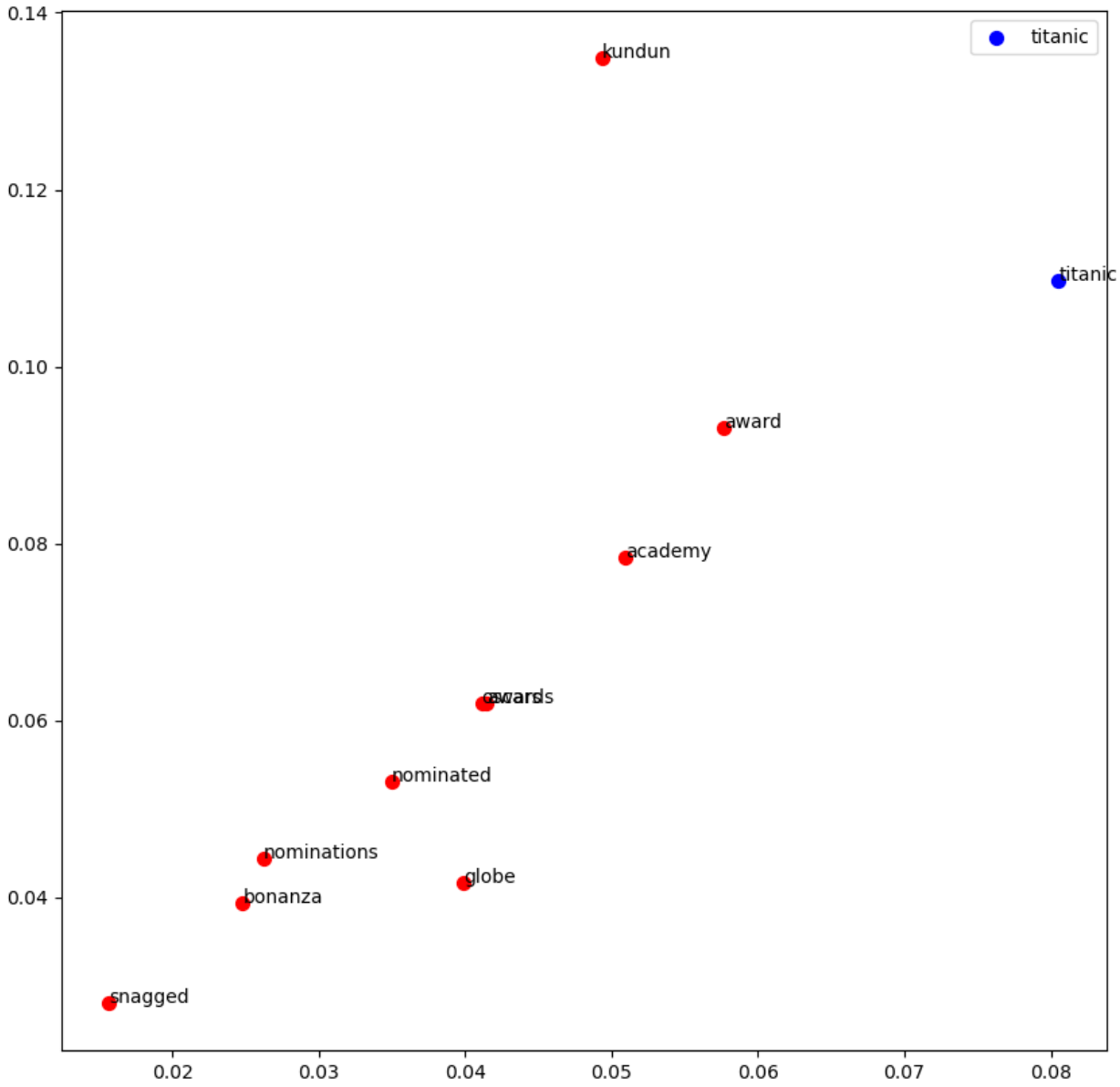
word similarity
0 awards 0.4966860492050777
1 nominated 0.4917653525065562
2 Oscars 0.4849682835113751

3	academy	0.46631767352452314
4	snagged	0.4601805468537133
5	bonanza	0.45725114624622953
6	kundun	0.44574260139370964
7	nominations	0.445199025251897
8	globe	0.4406920069570572
9	award	0.43104422809111287



Titanic

	word	similarity
0	awards	0.4966860492050777
1	nominated	0.4917653525065562
2	Oscars	0.4849682835113751
3	academy	0.46631767352452314
4	snagged	0.4601805468537133
5	bonanza	0.45725114624622953
6	kundun	0.44574260139370964
7	nominations	0.445199025251897
8	globe	0.4406920069570572
9	award	0.43104422809111287



The Result given is quite satisfactory given the fact that this is a movie review corpus so the results are related to both Titanic Ship and the Movie and also the awards related to the movie , as it is a very popular film

like the results Academy, Nominations , Awards , Oscar is related to the success of film .

I chose the gensim's glove vectors trained on the `glove-wiki-gigaword-50` corpus. The top 10 closest words for the gensim model were :

'odyssey', 0.65303
'phantom', 0.65100
'doomed', 0.64136
'r.m.s.', 0.63026
'cinderella', 0.62629
'voyager', 0.62269
'wreck', 0.60453
'ghost', 0.59909
'horror', 0.59604
'tragedy', 0.59548