CIS530: Natural Language Processing

Homework 5: Advanced Vector Space Models

Jonathon Delemos
University of Pennsylvania

Abstract

This report presents the results and analysis for CIS530 Homework 5, which focuses on semantic similarity and word clustering using both sparse and dense vector representations. Tasks involve comparing human judgment scores from the SimLex-999 dataset against word embeddings, computing Kendall's τ correlations, and evaluating clustering performance on paraphrase sets. The experiments include random baselines, sparse co-occurrence vectors, and dense Word2Vec embeddings trained on Google News data.

SimLex-999 Dataset Revisited

Least Similar Pairs (2.1)

Question: What are the least similar two pairs of words based on human judgement scores and vector similarity? Do the pairs match?

Table 1: Least Similar Word Pairs

Source	Word Pair	Score
Human Judgement	SHRINK, GROW	0.23
	NEW, ANCIENT	0.23
Vector Similarity	HOUSE, KEY	-0.0413
	FLOWER, ENDURANCE	-0.0368

Discussion: The answers do not match. Human scores in SimLex-999 range roughly from 0–10, while cosine similarities range from about -0.1 to 1.0. The mismatch (action antonyms vs. noun-noun relations) is expected since the ranges differ due to the cosine application. Human judgments emphasize semantic polarity, while embeddings reflect contextual co-occurrence.

Most Similar Pairs (2.2)

Question: What are the most similar two pairs of words based on human judgement scores and vector similarity? Do the pairs match?

Table 2: Most Similar Word Pairs

Source	Word Pair	Score
Human Judgement	QUICK, RAPID	9.7
	VANISH, DISAPPEAR	9.8
Vector Similarity	SOUTH, NORTH	0.967
	NORTH, WEST	0.941

Discussion: No, the pairs for mathematical vector scores and human judgement scores do not match. The most similar words according to human scores are action verbs, while the vector math suggests that directional adjectives are most similar. This discrepancy illustrates that embeddings capture contextual association rather than strict synonymy or semantic equivalence.

Correlation Scores and Model Comparison (2.3)

Table 3: Kendall's τ Correlations for GloVe Models

Model	Dimensions	τ	p-value
GloVe 6B 50D	50	0.1810	1.22e-17
GloVe 6B 100D	100	0.2051	3.41e-22
GloVe 6B 200D	200	0.2367	4.99e-29
GloVe 6B 300D	300	0.2589	2.08e-34
GloVe 840B 300D	300	0.2861	1.29e-41

Discussion: Increasing embedding dimension from 50D to 300D and corpus size from 6B to 840B improves Kendall's τ, indicating that richer embeddings capture human similarity judgments better. There is a clear positive correlation between larger corpora and higher τ values. All correlations are statistically significant, confirming that the results are not due to chance. However, improvements diminish as dimension increases, revealing diminishing returns beyond 300 dimensions.

Clustering Experiments

Task 3.1: Cluster Randomly (1 point)

Table 4: Task 3.1: Random Clustering Results

Target	k	Paired F-Score
rule.v	7	0.2000
operate.v	7	0.2222
performance.n	5	0.2833
talk.v	6	0.3117
difference.n	5	0.3146

treat.v	8	0.2492
use.v	6	0.2507
write.v	9	0.2083
degree.n	7	0.2155
play.v	34	0.0495
different.a	1	1.0000
interest.n	5	0.1646
note.v	3	0.3684
hear.v	5	0.2953
judgment.n	7	0.2783
paper.n	7	0.2259
watch.v	5	0.4216
source.n	9	0.1756
express.v	7	0.2684
eat.v	6	0.2938
organization.n	7	0.1779
simple.a	5	0.1481
plan.n	3	0.6459
shelter.n	5	0.2942
suspend.v	6	0.1154
atmosphere.n	6	0.2059
wash.v	13	0.0738
win.v	4	0.4068
miss.v	8	0.2000
provide.v	7	0.3615
bank.n	9	0.1538
party.n	5	0.2740
produce.v	7	0.2641
climb.v	6	0.2458
image.n	9	0.1523
begin.v	8	0.1604
expect.v	6	0.2640
receive.v	13	0.0588
mean.v	6	0.2113
smell.v	4	0.2947

Average Paired F-Score: 0.2070

Discussion: As expected, random clustering provides the weakest baseline. Since no semantic or contextual information guides the grouping, the results fluctuate randomly and the average F-score remains low (0.207). This serves as a control for evaluating the performance of subsequent methods.

Task 3.2: Cluster with Sparse Representations (6 points)

Table 5: Task 3.2: Sparse Representations (Dev Data)

Target	k	Paired F-Score

rule.v	7	0.2253
operate.v	7	0.1994
performance.n	5	0.2580
talk.v	6	0.3037
difference.n	5	0.3158
treat.v	8	0.2438
use.v	6	0.2837
write.v	9	0.1955
degree.n	7	0.1756
play.v	34	0.0590
different.a	1	1.0000
interest.n	5	0.1544
note.v	3	0.6400
hear.v	5	0.2994
judgment.n	7	0.2201
paper.n	7	0.2711
watch.v	5	0.3230
source.n	9	0.1753
express.v	7	0.2605
eat.v	6	0.3494
organization.n	7	0.1817
simple.a	5	0.2400
plan.n	3	0.3621
shelter.n	5	0.3400
suspend.v	6	0.1538
atmosphere.n	6	0.2597
wash.v	13	0.1124
win.v	4	0.4116
miss.v	8	0.2069
provide.v	7	0.3131
bank.n	9	0.1622
party.n	5	0.2514
produce.v	7	0.3122
climb.v	6	0.2609
image.n	9	0.1566
begin.v	8	0.1579
expect.v	6	0.3032
receive.v	13	0.0355
mean.v	6	0.2650
smell.v	4	0.3596

Average Paired F-Score: 0.2140

Data Analysis:

- This experiment used a sparse co-occurrence representation where most values are zero.
- The model used coocvec-500mostfreq-window-3.filter.magnitude with a symmetric window of size 3. Each word vector encodes co-occurrence frequency within ±3

tokens.

- Three clustering algorithms were compared—K-Means, Agglomerative, and DBSCAN—from scikit-learn. K-Means yielded the most interpretable clusters, while DBSCAN tended to collapse items and Agglomerative over-segmented them.
- On the dev set, the F-score improved from 0.207 (random) to 0.214 (sparse), confirming moderate gains from contextual information.

Task 3.3: Cluster with Dense Representations (8 points)

Table 6: Task 3.3.1: Dense Representations Results

Target	k	Paired F-Score
rule.v	7	0.2318
operate.v	7	0.2019
performance.n	5	0.2627
talk.v	6	0.3086
difference.n	5	0.2817
treat.v	8	0.2150
use.v	6	0.4468
write.v	9	0.2151
degree.n	7	0.2077
play.v	34	0.0656
different.a	1	1.0000
interest.n	5	0.2536
note.v	3	0.5333
hear.v	5	0.2689
judgment.n	7	0.2063
paper.n	7	0.2408
watch.v	5	0.2636
source.n	9	0.1381
express.v	7	0.2097
eat.v	6	0.2239
organization.n	7	0.2300
simple.a	5	0.2000
plan.n	3	0.3606
shelter.n	5	0.3569
suspend.v	6	0.2154
atmosphere.n	6	0.2744
wash.v	13	0.1796
win.v	4	0.3537
miss.v	8	0.1667
provide.v	7	0.3333
bank.n	9	0.0909
party.n	5	0.2573
produce.v	7	0.2329

climb.v	6	0.2569
image.n	9	0.1855
begin.v	8	0.1989
expect.v	6	0.3357
receive.v	13	0.0773
mean.v	6	0.3028
smell.v	4	0.2750

Average Paired F-Score: 0.2156

Data Analysis:

- Dense Representation: Filled Matrix
- I used pre-trained dense word embeddings from GoogleNews-vectors-negative300.magnitude, a 300-dimensional Word2Vec model trained on 100 billion tokens from Google News.
- Each vector encodes distributed semantic information rather than explicit co-occurrence counts. I used K-means clustering to allow for proper analysis between the gold labeled clusters and the test set clusters.
- The dense model achieved an average paired F-score of 0.2156, slightly higher than the sparse baseline .2140. This modest gain reflects that distributed embeddings capture semantic similarity between paraphrases more effectively, though they sometimes overgeneralize frequent verbs.

Task 3.3.3: Error Analysis

To compare systems, we computed $\Delta = F_{dense} - F_{sparse}$ per target word.

Table 7: Sample of Sparse vs. Dense F-Score Differences

Target	F_{sparse}	F_{dense}	**Difference**
suspend.v	0.2069	0.4906	+0.2837
bank.n	0.3373	0.5797	+0.2424
interest.n	0.2215	0.4357	+0.2142
party.n	0.3277	0.5074	+0.1797
rule.v	0.2428	0.3900	+0.1473

Frequency Analysis: Suspend, bank, interest, and party had the largest positive deltas, indicating that dense embeddings handle abstract and polysemous words better. Sparse models performed better on more syntactically grounded verbs (e.g., talk, treat, use). This pattern suggests dense vectors better capture conceptual relations, while sparse vectors retain context-specific distinctions.

Task 3.4: Cluster Without Predefined k (6 points)

Table 8: Task 3.4: Clustering Without Predefined k (Dev Data)

Target	k	Paired F-Score
rule.v	7	0.1891
operate.v	7	0.2809
performance.n	5	0.3140
talk.v	6	0.3634
difference.n	5	0.3521
treat.v	8	0.2012
use.v	6	0.2822
write.v	9	0.2339
degree.n	7	0.3519
play.v	34	0.0647
different.a	1	1.0000
interest.n	5	0.2139
note.v	3	0.4103
hear.v	5	0.2812
judgment.n	7	0.2391
paper.n	7	0.2605
watch.v	5	0.2727
source.n	9	0.1591
express.v	7	0.1991
eat.v	6	0.2592
organization.n	7	0.2416
simple.a	5	0.0952
plan.n	3	0.6142
shelter.n	5	0.3165
suspend.v	6	0.1509
atmosphere.n	6	0.2069
wash.v	13	0.0517
win.v	4	0.2897
miss.v	8	0.1522
provide.v	7	0.3386
bank.n	9	0.1739
party.n	5	0.2319
produce.v	7	0.2775
climb.v	6	0.2451
image.n	9	0.1930
begin.v	8	0.1744
expect.v	6	0.2067
receive.v	13	0.0985
mean.v	6	0.2731
smell.v	4	0.3711

Average Paired F-Score: 0.2129

Data Analysis:

- No Cluster K Representation
- Like in my previous response, I used pre-trained dense word embeddings from GoogleNews-vectors-negative300.magnitude, a 300-dimensional Word2Vec model

trained on 100 billion tokens from Google News. Each vector encodes distributed semantic information rather than explicit co-occurrence counts.

- In this experiment, I removed the dependence on the predefined number of clusters (k). For each target word, the algorithm estimated cluster structure automatically using KMeans with a heuristic based on the number of paraphrases, or using DBSCAN for density-based grouping.
- The resulting average paired F-score 0.2129 is comparable to the sparse model 0.2140, suggesting that automatic estimation of k does not significantly harm performance. However, some words with many paraphrases (play.v, receive.v) suffered from over-clustering while small sets (plan.n) produced artificially high scores.

Conclusion

Sparse co-occurrence vectors remain interpretable and competitive, while dense word embeddings offer modest gains in capturing human-like semantic judgments. Increasing embedding dimensionality and corpus size consistently improves Kendall's τ correlation, but clustering improvements are less pronounced. Both approaches highlight the tradeoff between context sensitivity and generalization in modern vector space semantics.