CIS 530 – HW4 - Spring 2019

Authors:
Ilayda Ipek Onur
Srinivas Suri

Part 2:

1. Least similar 2 pairs of words?

   Human Judgement

   word1: new
   word2: ancient

   Vector Similarity

   word1: house
   word2: key

   **They don't match

2. Most similar 2 pairs of words?

   Human Judgement

   word1: vanish
   word2: disappear

   Vector Similarity

   word1: south
   word2: north

   **They don't match

3. .
   - glove.6B.50d.magnitude
     Correlation = 0.18100126067449063, P Value = 1.2242211264976945e-17
   - glove.6B.100d.magnitude
     Correlation = 0.20506409092608713, P Value = 3.4122866339517884e-22
   - glove.6B.200d.magnitude
     Correlation = 0.23670323199262908, P Value = 4.9936324557834e-29
   - glove.6B.300d.magnitude
     Correlation = 0.25894302181101986, P Value = 2.080389068003349e-34
   - glove.840B.300d.magnitude
     Correlation = 0.2860664813618063, P Value = 1.2933356133610945e-41

We can observe that as dimension and size of vector increases, Kendall's Tau correlation based on the similarity increases. Also p values get lower, this means our results are less due to chance. This is meaningful because we take human judgment as a reliable source and the more data that matrices have, the more important features vectors are able to capture.

PART 3:

3.2:

- Description of the model:

  The model uses the matrix with 500 dim on window size = 3

- Clustering algorithm

  PCA n n_components=200

  Agglomerative Clustering with linkage='average'

- Results of any preliminary experiments you might have run on the dev set

  We tried variety of clustering algorithms with and without PCA:

KMeans	0.2644
KMeans with PCA 200	0.2550
DBSCAN	0.3873
Spectral Clustering	0.2217
Agglomerative Clustering with linkage='ward'	0.2640
Agglomerative Clustering with linkage='average' with PCA=200	0.3481

Although DBSCAN gave the best score on Dev, Agglomerative gave the best score for the test

3.3:

- Description of the model:

  GoogleNews-vectors-negative300.filter.magnitude

- Clustering algorithm

Agglomerative Clustering with linkage='average'

- Results of any preliminary experiments you might have run on the dev set

We tried variety of clustering algorithms:

KMeans
DBSCAN
Spectral Clustering
Agglomerative Clustering with linkage='ward'

Agglomerative Clustering with linkage='average' gave the best result on our dev set

3.4:

- Description of the model:

GoogleNews-vectors-negative300.filter.magnitude

- Clustering algorithm

KMeans with PCA fit=300

- Results of any preliminary experiments you might have run on the dev set

In our dev set, we tried cluster sizes of 1,2 and 7 and calculated kmeans.inertia, took the cluster size that gave the best performance. This method gave the best performance for our dev set and we used the same method for our test set