DSE 220: Homework #4 - Vector Embedding Analysis of Words from Brown Corpus

Professor: Volkan Vural

Teaching Assistants: Chetan Gandotra and Tushar Bansal

Joshua Wilson **A53228518**

1 Objective

The Brown Corpus was the first million-word electronic corpus of English, created in 1961 at Brown University. It contains text from 500 sources, categorized into genres such as news, editorial, government, fiction, and humor.

In this analysis, we investigate creation of a clustering or embedding of words from the Brown corpus, so that words with similar meanings are clustered together, or have embedding that are close to one another. That is, words that tend to appear in similar contexts are likely to be related. We investigate this idea by coming up with an embedding of words that is based on co-occurrence statistics.

Specifically, we utilize Pointwise Mutual Information (PMI) by following the general process described in Chapter 15: Vector Semantics from *Speech and Language Processing* by Daniel Jurafsky & James H. Martin.

All code associated with this analysis can be found on GitHub:
https://github.com/mas-dse/jsw037/tree/master/DSE220.

2 Data Processing

The Brown corpus is downloaded using Python's nltk.corpus package. Once downloaded, the corpus text is processed by removing stopwords (per the nltk.corpus list of English stopwords), punctuation (per the string package's punctuation list), and converting all letters to lowercase. We then count the number of occurrences of each word.

Next, we identify a vocabulary (V) of the 5,000 most commonly occurring words, and a list of the most common 1,000 words as context words (C). For each word $w \in V$, and each occurrence of it in the Brown corpus, we look at the surrounding window of four words (two before and two after w) and count how often each context word in C appears around each word w:

```
c_count_mtx:
[[ 67.  71.  41. ...,   1.   1.   4.]
 [ 71.  42.  62. ...,   2.   1.   3.]
 [ 39.  60.  19. ...,   1.   3.   1.]
 ...,
 [  0.   0.   0. ...,   0.   0.   0.]
 [  1.   1.   0. ...,   0.   0.   0.]
 [  0.   0.   1. ...,   0.   0.   0.]]

c_count_mtx shape: (5000, 1000)
```

Using these counts, which are stored in a variable named c_count_mtx in the code above, and which we denote as $n(w, c)$ in this report, we construct the probability distribution $P(c|w)$ of context words around w (for each $w \in V$), as well as the overall distribution $P(c)$ of context words:

$$P(c|w) = \frac{n(w,c)}{\sum_{c'} n(w,c')}, P(c) = \frac{\sum_x n(c,w)}{\sum_{c'} \sum_w n(w,c')}$$

We then represent each vocabulary item w by a $|C|$-dimensional vector $\Phi(w)$, whose c^{th} coordinate is:

$$\Phi(w) = max\left(0, log\frac{P(c|w)}{P(c)}\right)$$

Finally, we perform Principal Component Analysis (PCA) to reduce the dimensionality of each word vector in $\Phi(w)$ to 100 from $|C|=$ 1,000, creating a 100-dimensional embedding for each word $w \in V$.

3 Nearest Neighbor Analysis

For each word $w \in V$, we calculate the nearest neighbor $w' \neq w$ in V. These results are based on the *cosine distance* measure:

$$1 - \frac{\Psi(w) \bullet \Psi(w')}{||\Psi(w)|| \times ||\Psi(w')||}$$

Other distance measures were also used, and result in similar but not identical results. In general, the nearest neighboring word is something with a very similar meaning. Below are a few selected words and nearest neighbor distances:

```
communism / utopian / distance = 0.549
autumn / summer / distance = 0.493
cigarette / lighted / distance = 0.480
pulmonary / artery / distance = 0.260
africa / asia / distance = 0.348
chemical / drugs / distance = 0.438
```

We can also determine the pairs of words that are closest together. The closest words in our vocabulary are clearly words that are very likely to appear in similar contexts, validating our vector embedding results:

```
closest pairs of words:
puerto / rico / distance = 0.070
per / cent / distance = 0.123
1 / 2 / distance = 0.137
know / think / distance = 0.140
f / l / distance = 0.145
```

4 Clustering Analysis

Using the 100-dimensional vectorial representation of our words, we cluster the words in V into 100 groups, using k-means++ to initialize the cluster centers and ”full” as the algorithm. In general, the clusters share a common theme, which validates the utility of embedding words as vectors based on context similarity. Words from a few selected clusters are depicted below:

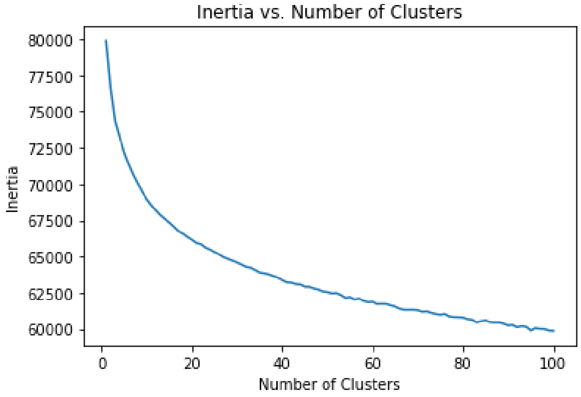
```
cluster 33:
 ['c', 'b', 'p', 'j', 'e', 'r', 'f', 'l', 'n', 'g']

cluster 49:
 ['two', 'three', 'several', 'four', 'five', 'ago', 'six',
 'minutes', 'hundred', 'ten', 'couple', 'seven', 'eight',
 'dollars', 'thousand', 'nine', 'twenty', 'fifty', 'thirty',
 'fifteen', 'twelve', 'eleven', '300', 'forty', 'fourteen',
 'twentyfive', 'seventeen', 'sixty']

cluster 54:
 ['mr', 'mrs', 'john', 'dr', 'brown', 'william', 'george',
 'thomas', 'charles', 'james', 'director', 'h', 'w',
 'henry', 'robert', 'jr', 'richard', 'chairman', 'joseph', 'smith']

cluster 64:
 ['president', 'party', 'washington', 'report',
 'meeting', 'kennedy', 'press', 'recently', 'reported',
 'plans', 'news', 'conference', 'announced', 'reports',
 'presented', 'khrushchev', 'laos']
```

In order to evaluate whether using 100 clusters was appropriate, we can look at the inertia graph:



While there is no obvious elbow to the graph, we can see that in terms of additional clusters reducing the distance, we reach a point of diminishing returns beyond approximately ten or fifteen clusters. At the very least, the graph confirms that we do not have too *few* clusters.

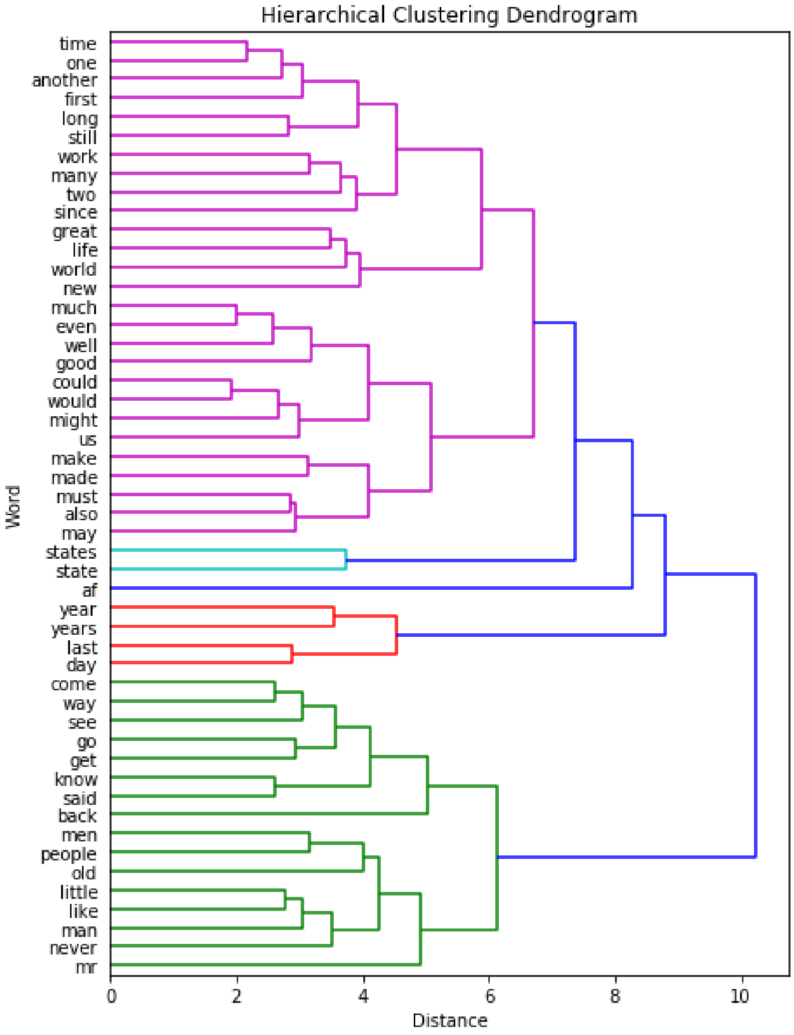
Finally, we perform an agglomerative clustering using Ward's method, and create a dendrogram to depict the clustering hierarchy for the first 100 words in our vocabulary. Again, the words whose vector embeddings are nearest generally make intuitive sense, as the smallest clusters (i.e. nearest words) are clearly related. For example, the smallest non-singleton cluster consists of the two words *state* and *states*, and the next smallest consists of the four words *year*, *years*, *last*, and *day*.

5 Conclusions and Future Work

Creating a vectorial representation of words clearly results in meaningful information. Distance measures between words align with our intuition about word meanings, as shown via nearest neighbor analysis results. In terms of clustering results and inertia graph, 100 clusters seems to be sufficient to place our vocabulary into meaningful groups.

Several opportunities for additional work are suggested:

- Investigate the impact of using different context window sizes
- Investigate the impact of applying Laplace smoothing
- Investigate whether farthest neighbors have some intuitive meaning (are they antonyms, for example?)
- Try to determine minimum dimensional embedding (via PCA dimensionality reduction) that still yields meaningful results
- Try to determine minimum number of clusters that still yields meaningful results
- Try to identify semantic themes of clusters of various sizes
- Investigate impact of using different distance metrics along with different methods of agglomeration (i.e. single-linkage vs. complete-linkage)
- Investigate whether clustering using Expectation Maximization performs better than k-means (as k-means assumes spherical clusters)


Hierarchical Clustering Dendrogram
time
one
another
first
long
still
work
many
two
since
great
life
world
new
much
even
well
good
could
would
might
us
make
made
must
also
may
states
state
af
year
years
last
day
come
way
see
go
get
know
said
back
men
people
old
little
like
man
never
mr
Word
0
2
4
6
8
10
Distance