ANLP Assignment 3 2018

s1449692, s1885778

January 10, 2019

1 Research questions

We define several areas of exploration, which are by no means independent. In presenting our results we try to relate the answers to questions we pose in this section.

Sensitivity to word frequency In the ideal case, our embeddings should carry no information about word frequency. However, it is known that many methods for embeddings suffer from the so-called *hubness problem* [Bakarov, 2018] – more frequent words tend to have a smaller distance to a disproportionately large number of words. [Schnabel et al., 2015] find a very strong positive log-linear relationship between the average rank of a word w in the nearest neighbours for a set of words and the frequency rank of w in the dataset the embeddings were produced on. In other words, more frequent words tend to be closer to any given word than less frequent words.

We aim to investigate whether this is true for our embeddings, see if some metric are less sensitive to this problem than others and examine the effect of different hyperparameters on this relationship.

Impact of eigenvalue weighting for SVD Performing Singular Value Decomposition (SVD) (Section 3) on the PMI or PPMI matrix is a popular method of producing dense word embeddings. We can use truncated SVD to obtain two matrices W_{SVD} and C_{SVD}, the rows of which are dense embeddings of words and contexts, respectively:

$$W_{SVD} = U_d \Sigma_d^p \quad C_{SVD} = V_d \tag{1}$$

where Σ_d^p is the diagonal matrix with the first d singular values raised to the power of p. Normally, p is set to either 0 or 1. However, [Levy et al., 2015] argue that p is a hyperparameter that can and should be tuned. They call this approach eigenvalue weighting and the p parameter – `pow`. The authors find that values lower than 1 yield better performance in a word similarity ranking task.

We define two questions here. First, we aim to investigate whether this will be true for the embeddings produced by our dataset. Second, we want to see how different metrics are affected by this hyperparameter.

Impact of different metrics on the performance on a semantic similarity task A popular method of evaluating word embeddings is to take a list of word pairs that are associated with similarity scores given by humans, compute the similarity between all pairs of words based on your word embeddings and then report the correlation between your predictions and the scores given by humans.

We choose a popular word similarity dataset (Section 2) and explore the effect of different similarity metrics and hyperparameters on the correlation between predictions and human scores.

Visualizing analogies Word analogy is another popular method of word embedding evaluation. It is based on the idea that arithmetic operations in a word vector space could be predicted by humans: given a set of three words, a, a^* and b the task is to identify such a word b^* such that the relation between b and b^* is the same as the relation a and a^*. Our goal was to see how does reducing the dimensionality of sparse vectors, effect the analogy properties of the word pairs.

2 Choice of query inventory

In order to see the difference between the proposed semantic similarity metrics and the difference between the sparse PPMI context vectors and the dense context vectors that we created with SVD, we needed to choose words for evaluation. We made our evaluation on three tasks: semantic similarity of words, analogy and sensitivity to word frequency, and for the first two tasks we choose the words from online lists. From the online lists, we stemmed all words and removed the words that weren't in the Twitter dataset.

Semantic similarity For semantic similarity we used WordSim-353 [Finkelstein, 2002], which is a dataset of 353 word pairs along with their similarity scores. The scores ranged from 0 to 10 and were assigned by 13 subjects who had a near-native command of English.

Frequency impact Inspired by [Schnabel et al., 2015], we randomly choose S_1 – a set of 500 words from our dataset. Then, we sample a disjoint set of words S_2 of size $50,000$. We compute the distances of all words in S_1 to all words in S_2 and take the 1000 nearest neighbours for every word in S_1 amongst the words in S_2. Then, we average the frequency rank in our dataset of the k-th nearest neighbours for all words in S_1.

Analogy The state-of-the-art data set for analogy is the Google Analogy data set, which has 19 544 questions ($a : a^* \to b : b^*$) divided into 10 smaller subclasses (8 869 semantic questions and 10 675 morphological questions).

3 Methods

While exploring the research questions that we stated, we used 3 different word similarity metrics and used Singular Value Decomposition for creating dense vectors from the sparse PPMI vectors.

Metrics Aside from using the cosine similarity, we used these following similarity metrics:
Bray-Curtis distance: $d(u, v) = \frac{\sum_i |u_i - v_i|}{\sum_i |u_i + v_i|}$
Canberra distance: $d(u, v) = \sum_i \frac{|u_i - v_i|}{|u_i| + |v_i|}$

Singular Value Decomposition (SVD) In our work, we perform truncated SVD on the PPMI matrix using the top $d = 100$ singular values. Some experiments we conducted have shown that our findings are consistent for smaller values but we omit an deeper investigation into the effects of d for the sake of brevity.

4 Results and discussion

Word frequency, similarity metrics and the `pow` hyperparameter We first investigate how the sensitivity to word frequency changes with the hyperparameter `pow` for different metrics. Arguably the most interesting finding is that increasing the value of `pow` smoothly changes the sensitivity of the cosine metric(Figure 2a) – for $p = 1$, it is biased towards frequent words and by decreasing the value, we tend to rank less frequent words near the top. We notice a similar effect on the Bray-Curtis distance. However, even for higher values of p, that remains biased towards high-frequency words.

The above might explain why [Levy et al., 2015] found the *pow* parameter to be important. This finding is also slightly disconcerting – probably the best value for `pow` depends heavily on the frequencies of the words in the dataset the hyperparameter is tuned on and does not change the quality of the embeddings in any other sense. This hypothesis is supported by the evidence in Figure 2d – For both Bray-Curtis and Cosine distances, we get the best performance on the WordSim353 dataset for the values of `pow` for which the metric is biased towards frequent words. And the average frequency rank of the words in this dataset is around 10,000 – it contains very frequent words.

From Figure 2c we can see that Canberra is not affected by the powers. Looking at the formula for how it is computed, this makes perfect sense: $d_p(u, v) = \sum_i \frac{|\sigma_i^p u_i - \sigma_i^p v_i|}{|\sigma_i^p u_i| + |\sigma_i^p v_i|} = \sum_i \frac{|u_i - v_i|}{|u_i| + |v_i|} = d(u, v)$ where σ_i is the ith singular value. That being said, the Canberra distance is heavily biased towards infrequent words. We acknowledge that this is a flaw in our dataset but we omit further investigation for the sake of brevity.

Word similarity It is worth noting that using the dense vectors produced by SVD improved the Spearman correlation between our predictions and the human rankings for WordSim353. This seems to agree with the findings in [Levy et al., 2015].

Visualizing analogies In order to visualise the analogies, we reduced the dimensionality of the PPMI vectors using truncated SVD (taking the top-2 singular vectors) and plot the word pairs from the Google Analogy set. In Figure 1a we can see that by using SVD we are able to create two nearly-separable clusters of adjectives and their comparative counterparts. There are also visible clusters in Figure 1b and Figure 3b.

In Figure 3b we can see a very pronounced city-in-state relationship. There is one pair that clearly does not fit there – *(Georgia – Atlanta)*. We hypothesize that the reason for this is that Georgia is also the name of a country. In

Figure 3a we can see that for some frequent words (like children, women, men), the analogy task might be feasible. But for other less frequent word, we would fail to make an analogy like *child* : *children* → *goose* : *geese*.

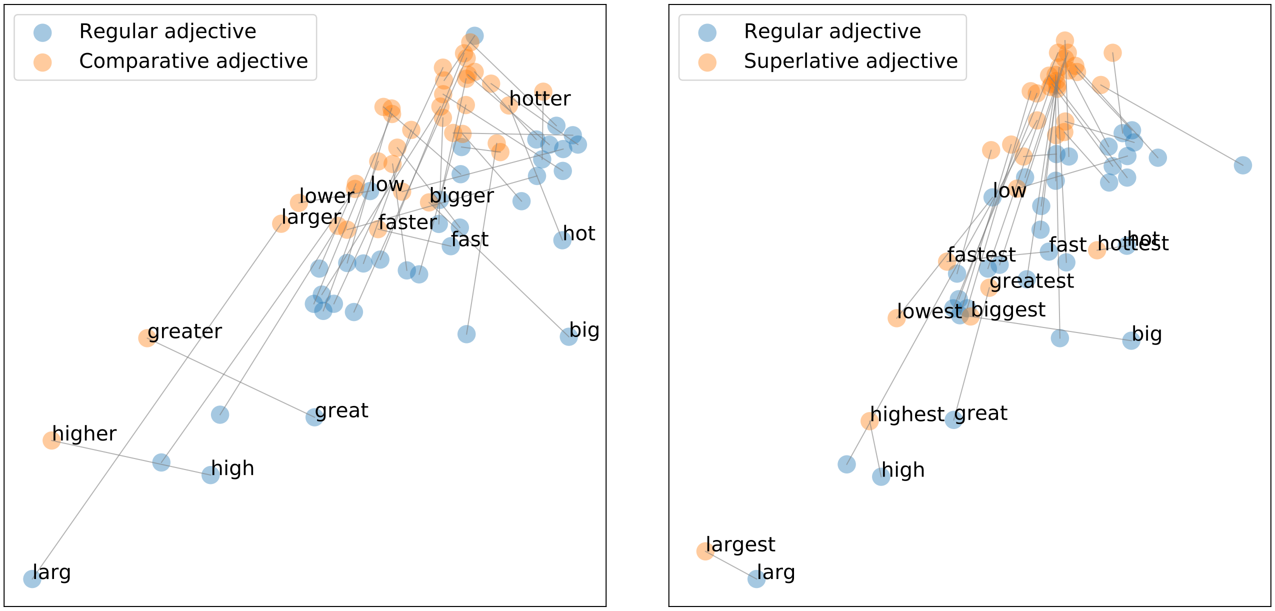
5 Additional material

Preliminary task output The output of the preliminary task is given in Listing 1

```
cos     word pair                          count1   count2

0.36    ('cat', 'dog')                     169733   287114
0.17    ('comput', 'mous')                 160828   22265
0.12    ('cat', 'mous')                    169733   22265
0.09    ('mous', 'dog')                    22265    287114
0.07    ('cat', 'comput')                  169733   160828
0.06    ('comput', 'dog')                  160828   287114
0.02    ('@justinbieber', 'dog')           703307   287114
0.01    ('cat', '@justinbieber')           169733   703307
0.01    ('@justinbieber', 'comput')        703307   160828
0.01    ('@justinbieber', 'mous')          703307   22265
```

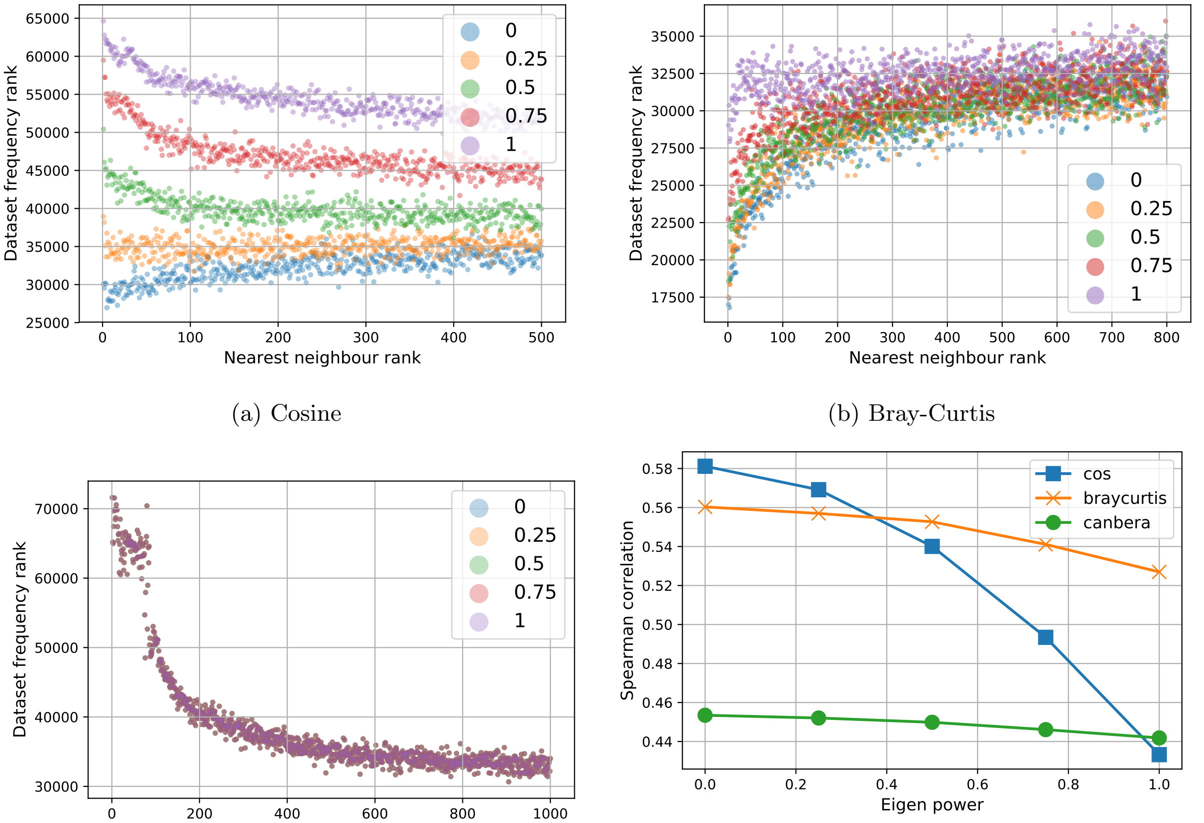
Listing 1: Output of the preliminary task



(a) Comparative adjectives

(b) Superlative adjectives

Figure 1: Plots of adjective pairs from the Google analogy task



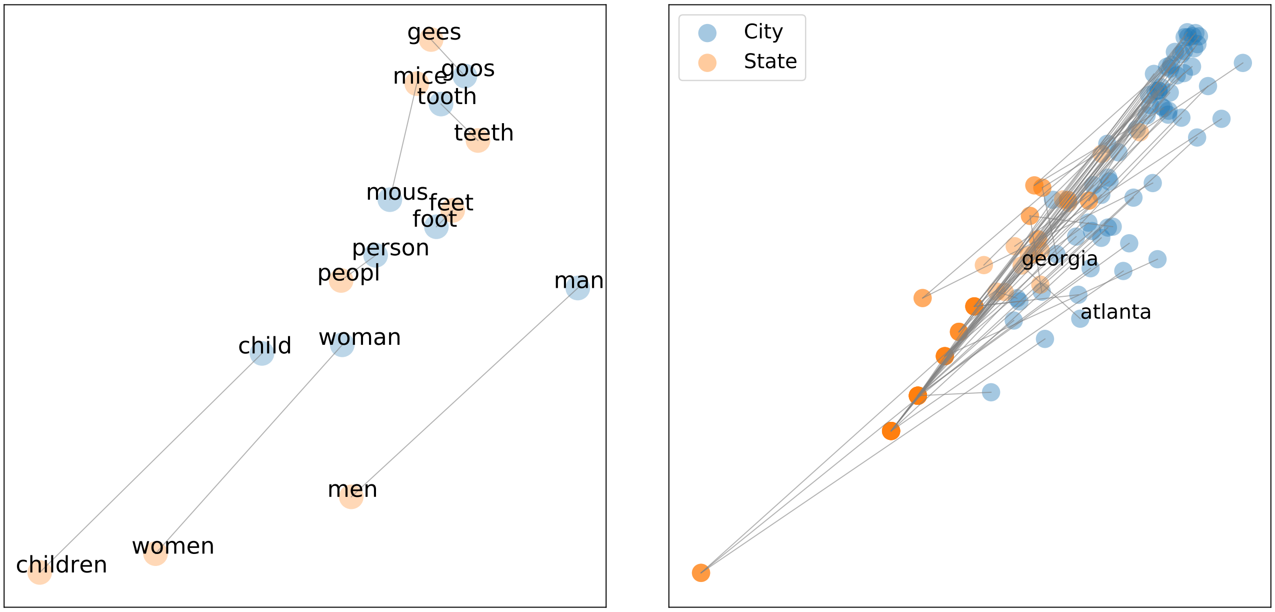
(a) Cosine

(b) Bray-Curtis

(c) Canberra

(d) Spearman correlation coefficient between WordSim human scores and predictions based on SVD embeddings for different metrics and different values of `pow`

Figure 2: Nearest neighbour rank v Frequency rank plots for different distance metrics and eigen weightings. The different colors correspond to different settings of the `pow` parameter



(a) Irregular plurals

(b) City-in-State

Figure 3: Singular-plural and City-State pairs

References

[Bakarov, 2018] Bakarov, A. (2018). A survey of word embeddings evaluation methods. *CoRR*, abs/1801.09536.

[Finkelstein, 2002] Finkelstein, L. (2002). Placing search in context: The concept revisited. *ACM Trans. Inf. Syst.*, 20(1):116–131.

[Levy et al., 2015] Levy, O., Goldberg, Y., and Dagan, I. (2015). Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

[Schnabel et al., 2015] Schnabel, T., Labutov, I., Mimno, D., and Joachims, T. (2015). Evaluation methods for unsupervised word embeddings. pages 298–307.