Exploring distributional similarity in Twitter

Federico Arenas, Dominic Phillips

1. Introduction

Seen as a form of dimensionality reduction, word embedding models derive their use from the potential to distill the most pertinent syntacic and semantic properties of a word into vector form. Such vector models have become an integral part of many natural language processing systems, with widespread applications in machine translation, text classification, and sentiment analysis, amongst others (Bahdanau et al., 2015; Maas et al., 2011; Li and Yang, 2018).

Most modern pipelines employ embeddings from neural language models, yet these are often difficult and time-consuming to train (Li et al., 2019). In this paper we compare and evaluate two simple embedding models which can be constructed directly from the co-occurrence matrix alone; Positive Pointwise Mutual Information (PPMI), and Hellinger Principal Component Analysis (H-PCA).

For each embedding model we consider three alternative metrics for word similarity: cosine, euclidean and manhattan distance. Then, taking each combination of embedding model and similarity measure, we report results of two intrinsic evaluation measures, word similarity and concept categorization, on gold-standard datasets. We then qualitatively compare hierarchical-clustering dendrograms produced by the two most promising methods on sets of concept-categorized words, finding that the resulting dendrograms reproduce sensible semantic segmentations under both embedding types.

Hellinger Principal Component Analysis (H-PCA)

H-PCA is an embedding method which leverages word co-occurrence data. H-PCA has the advantages of being both computationally fast whilst also showing comparable extrinsic evaluation performance to neural language models trained for weeks (Lebret and Collobert, 2017).

The intuition behind H-PCA is that word similarity can be quantified by computing the 'distance' between their corresponding co-occurrence probability distributions. Since probability distributions are normalized, it can be argued that the natural distance metric to use is not the Eucledian distance but rather the Hellinger distance (Hellinger, 1909), which is defined for two distributions $P = (p_1, ..., p_k)$, $Q = (q_1, ..., q_k)$ as:

$$H(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{k} (\sqrt{p_i} - \sqrt{q_i})^2}$$

One might consider embedding by using the discrete probability distributions directly, utilizing Hellinger distance as a similarity measure, however this yields a highly-sparse, vocabulary-dependent embedding. To avoid this, H-PCA instead first square-roots and re-scales the probability distributions, then performs a dimensionality reduction using truncated SVD. The algorithmic steps are summarised in Algorithm 1 in *Annex A*.

Constructing Vector Embedding Models

We construct vector embedding based on word co-occurrence matrix data which is derived from a 2011 Twitter dataset of approximately 100 million tweets. Details of the dataset and pre-processing steps can be found in (ANLP Lab 8 Notes). From this co-occurence data we are able to construct three distinct vector embedding models as follows:

Sparse PPMI embedding: From the co-occurrence matrix we compute the PPMI of a given word with all other words in the dataset. The resulting vector of PPMI values defines a sparse PPMI embedding of the given word.

Dense PPMI embedding: We consider the $50,000$[1] most common words in the corpus and compute the PPMI matrix whose i, j entry is the PPMI between the i^{th} and j^{th} most common words. We then perform a 250-dim truncated SVD dimensionality reduction on the PPMI matrix. The rows of the $50,000 \times 250$ U matrix so obtained constitute our dense PPMI embedding vectors.

H-PCA embedding: We first obtain the co-occurrence matrix for the $50,000$ most common words in the corpus. We then construct a 250-dim H-PCA embedding as described in Algorithm 1.

2. Evaluating Embedding Models and Similarity Metrics

Given a set of embedding models, how do we quantify their relative performance? This is no simple task, since what constitutes better performance might be task-specific or difficult to quantify. The traditional solution is to assess performance according to a set of evaluation methods. In this section we evaluate the relative performance of our embedding models and similarity measures using two types of intrinsic evaluation: word similarity, and concept categorization.

2.1. Intrinsic Evaluation on word similarity Gold Standards

A word similarity evaluator measures the degree of correlation between human perceived semantic similarity and the word-vector similarity according to a given metric[2]. To implement this evaluation method we use four gold standard word similarity datasets: the MEN dataset (Bruni et al., 2013), RG65 (Ruben-

[1] The entire corpus size consists of approximately $200,000$ distinct words. Using spare matrix routines we are limited to computing a SVD decomposition for up to $50,000$ words with a desktop PC.

[2] The metrics we consider in this paper are the Euclidean distance $\sum_i (v_i - u_i)^2$, weighted Euclidean distance $\sum_i w_i (v_i - u_i)^2$, Manhattan Distance $\sum_i |v_i - u_i|$, weighted Manhattan Distance $\sum_i w_i |v_i - u_i|$, Cosine Similarity $\sum_i \frac{u_i v_i}{|u||v|}$, and weighted Cosine Similarity $\sum_i w_i \frac{u_i v_i}{|u||v|}$

stein and Goodenough, 1965), WordSim Relatedness, WordSim Similarity (Finkelstein et al., 2001). Each of these datasets consists of word pairs[3] accompanied by a human-perceived similarity score such as[4]

autograph	shore	0.06
bird	woodland	1.24
gem	jewel	3.94

To evaluate word-similarity performance on each dataset we rank the word pairs in order of human-perceived similarity, and separately in order of similarity according to the similarity metric. We then compute the Spearman Rank Correlation Coefficient (SRCC) between the two sets of ranked data.

In *Annex B* we tabulate the SRCCs so obtained for each of the different vector embedding models, and for three different measures of similarity: Cosine, Euclidean, and Manhattan. Note that the dense vector embedding models are split into two rows - the first row shows the SRCCs when using an unweighted similarity metric, the second row shows the SRCCs when weighting by the singular values of the SVD decomposition.

There are a few general remarks we can draw from the results in *Annex B*. Most importantly we observe that all embeddings consistently perform best under a variant of Cosine similarity, rather than Euclidean or Manhattan. We also observe that a variant of PPMI embedding (whether sparse or dense) always outperforms H-PCA under cosine similarity. Finally we note that, whereas the dense embeddings have satisfactory performance under Eulidean and Manhattan similarities, sparse PPMI is essentially useless for assessing word similarity under these metrics.

2.2. Concept Categorization Evaluation using the BLESS data set

For this part of the study we focus on finding which of the word embedding and word similarity methods are best fit to regroup semantically related words. However, a word can be semantically related in many ways to another word. Therefore, we use the BLESS dataset (Baroni and Lenci, 2011), which allows us to define a framework where a single *source word* can be related to a single superset of *target words*, further divided in subsets of *relationships* between the source word and the target word[5]. There are 6 types of relationships in the dataset: *attribute, coordinate, hypernym, meronym, event,* and *random*. The *random* relationship contains subsets of target words that are randomly selected and attached to a source word. The dataset is composed of 200 source words, and 725 target words.

In order to measure the capacity of a given word embedding method to represent semantic relationships between words, we calculate the per-relationship average distance, measured according to our similarity metrics, between all source words and their corresponding target words for each of the six relationship categories. From this measurement, we expect to find that a well-performing word embedding will assign the highest average distance to the random relationship. Similarly, the smallest average distance indicates the relationship that is best represented by a given word embedding method.

We perform these distance measurements for the H-PCA embedding and dense PPMI embeddings for each of the similarity metrics. The results are visualized in *Annex D*.

From these measurements we can see that under all similarity metrics H-PCA consistently assigns the highest average distance to the random relationship, with the overall best performing metric being unweighted cosine. It seems that H-PCA's best represented relationship is *hypernym* under this highest performing metric, although the uncertainties are too large to be conclusive.

Similarly to H-PCA, PPMI dense embeddings also perform best with the cosine metric, and it also seems that the best represented relationship is *hypernym*.

Comparing the two embedding methods with cosine similarity, we see that, on the whole, PPMI better discriminates the non-random relationships from the random relationship.

2.3. Visualizing word similarity through Hierarchical Clustering

We now proceed to visualize semantic segmentation through hierarchical clustering, which we perform on a set of 84 words divided into 8 categories. We perform the clustering for H-PCA, and dense PPMI embeddings, using cosine similarity since it was the metric that consistently performed best in §2.1 and §2.2. The results can be seen in *Annex E*. From the dendrograms, we see that word categories are being successfully regrouped for both embeddings. However, the PPMI dendrogram is forming fewer clusters, each with a more homogeneous set of words that belong to a specific category.

We conclude that the PPMI dense embedding better represents semantic categories.

3. Conclusion

In this report we evaluated several word embedding models using quantitative and qualitative intrinsic evaluation methods, testing their performance according to a set of similarity metrics.

Of the metrics we tested, we consistently found that for each embedding model, the best performing was cosine similarity. Additionally we found that PPMI embeddings outperformed H-PCA embeddings under cosine similarity in both the word similarity and concept categorization quantitative tests. And furthermore, in hierarchical clustering tests, we found that PPMI embeddings produced more semantically homogeneous clusters.

Although we found that the PPMI word embedding outperformed H-PCA in our results, the differences are marginal which is understandable given that both embeddings were drawn from the same numerical representation (a co-occurrence matrix).

Finally we note that the embedding methods we selected were chosen because of their computational simplicity, however there are more complex embeddings models exist which use the same co-occurrence data to obtain better performance[6].

[3] 3000, 65, 252, and 203 such pairs respectively

[4] From the RG65 dataset

[5] See *Annex C* for an example of these source word - target word relationships for 'alligator'

[6] For example, GloVe (Pennington et al., 2014) has been shown to achieve a SRCC of 0.685 under cosine similarity for the MEN dataset (Wang et al., 2019).

Source word	Target word	relationship
alligator	green	attribute
alligator	heavy	attribute
.	.	.
alligator	toad	coordinate
alligator	turtle	coordinate
.	.	.
alligator	swim	event
alligator	walk	event
.	.	.
alligator	reptile	hypernym
alligator	vertebrate	hypernym
.	.	.
alligator	tail	meronym
alligator	tooth	meronym

Algorithm 1 H-PCA

Input: Co-occurrence matrix, C_{ij}, size $N \times N$
Dimensionality of desired vector embedding, K
Output: Matrix of word embeddings U_{ij}, size $N \times K$
1) Normalize rows, $C_{ij} \longleftarrow \frac{C_{ij}}{\sum_j C_{ij}}$
2) Square-root and scale, $C_{ij} \longleftarrow \frac{1}{\sqrt{2}} \sqrt{C_{ij}}$
3) Run K-dim truncated SVD on C_{ij}, obtain U_{ij}, S_{ij}, V_{ij}
return U_{ij}

Preliminary Results

Word Pair	Cosine Similarity
('cat', 'dog')	0.36
('comput', 'mous')	0.17
('cat', 'mous')	0.12
('mous', 'dog')	0.09
('cat', 'comput')	0.07
('comput', 'dog')	0.06
('@justinbieber', 'dog')	0.02
('cat', '@justinbieber')	0.01
('@justinbieber', 'comput')	0.01
('@justinbieber', 'mous')	0.01

Annex B

		MAN	RG65	WordSimR	WordSimS
Cosine Sim.					
PPMI-sparse		**0.608**	**0.704**	0.425	0.475
PPMI-dense	*unweighted*	0.563	0.580	0.491	**0.514**
	weighted	0.571	0.575	**0.492**	0.510
H-PCA	*unweighted*	0.515	0.611	0.420	0.505
	weighted	0.520	0.538	0.450	0.491
Euclidean Sim.					
		MAN	RG65	WordSimR	WordSimS
PPMI-sparse		0.084	0.011	0.006	0.062
PPMI-dense	*unweighted*	0.150	0.270	0.118	0.140
	weighted	0.149	0.274	0.112	0.123
H-PCA	*unweighted*	**0.295**	**0.361**	**0.234**	**0.281**
	weighted	0.268	0.340	0.208	0.218
Manhattan Sim.					
		MAN	RG65	WordSimR	WordSimS
PPMI-sparse		-0.075	0.037	0.007	-0.026
PPMI-dense	*unweighted*	0.152	0.287	0.117	0.140
	weighted	0.150	0.291	0.114	0.128
H-PCA	*unweighted*	**0.293**	**0.360**	**0.231**	**0.278**
	weighted	0.266	0.348	0.196	0.216

Annex D

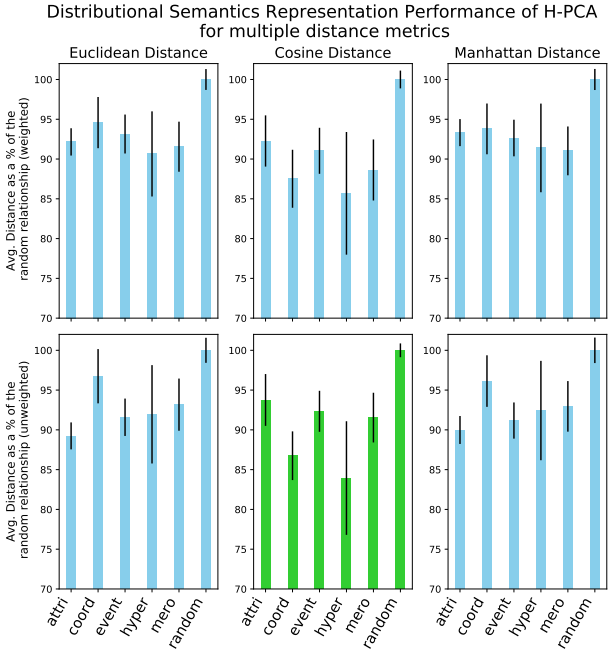

Figure 1. Average distances as a percentage of source words to target words H-PCA vectors per category, using multiple distance metrics

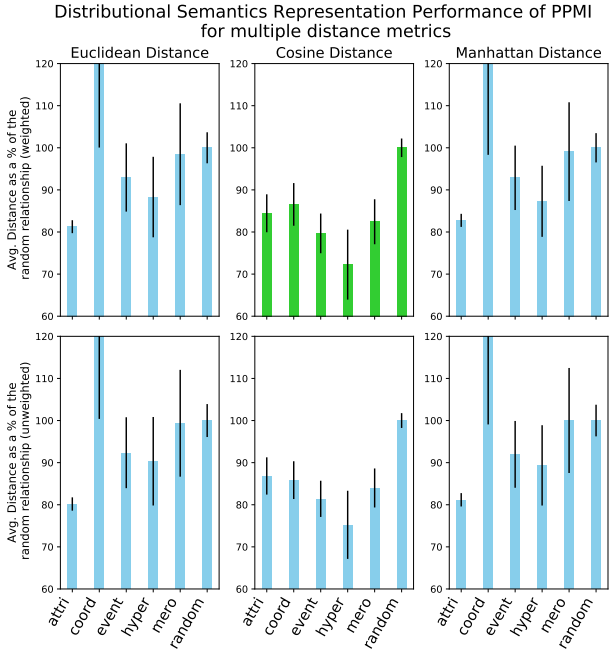

Figure 2. Average distances as a percentage of source words to target words PPMI vectors per category, using multiple distance metrics

Annex C

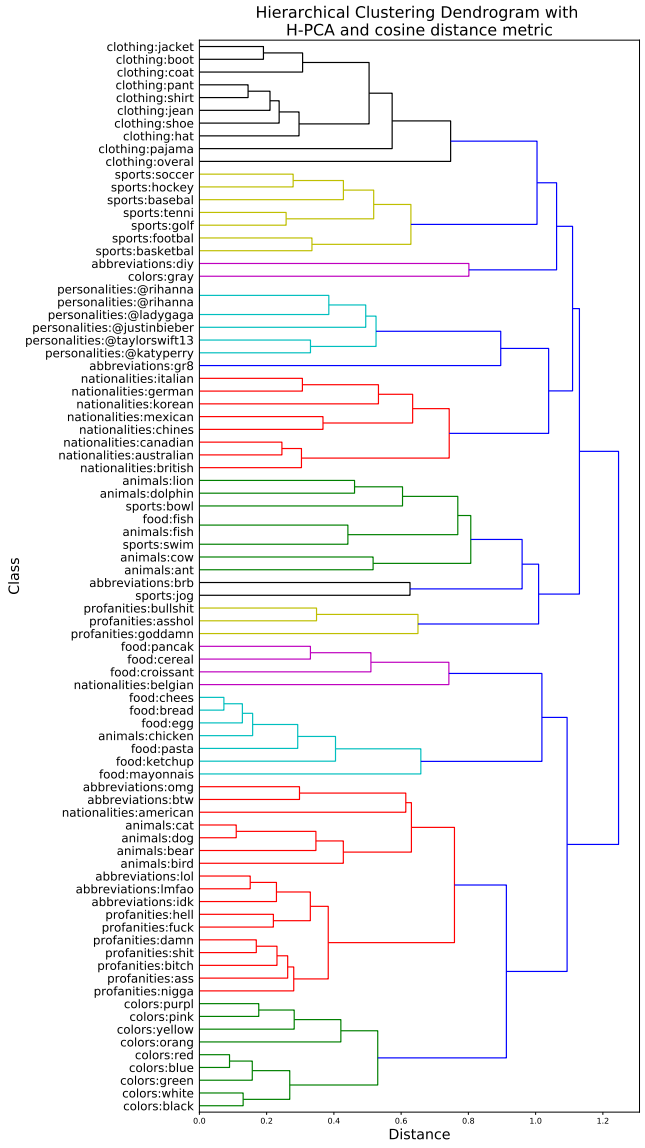

Figure 3. Dendrogram performed on categories of text, using H-PCA word embeddings

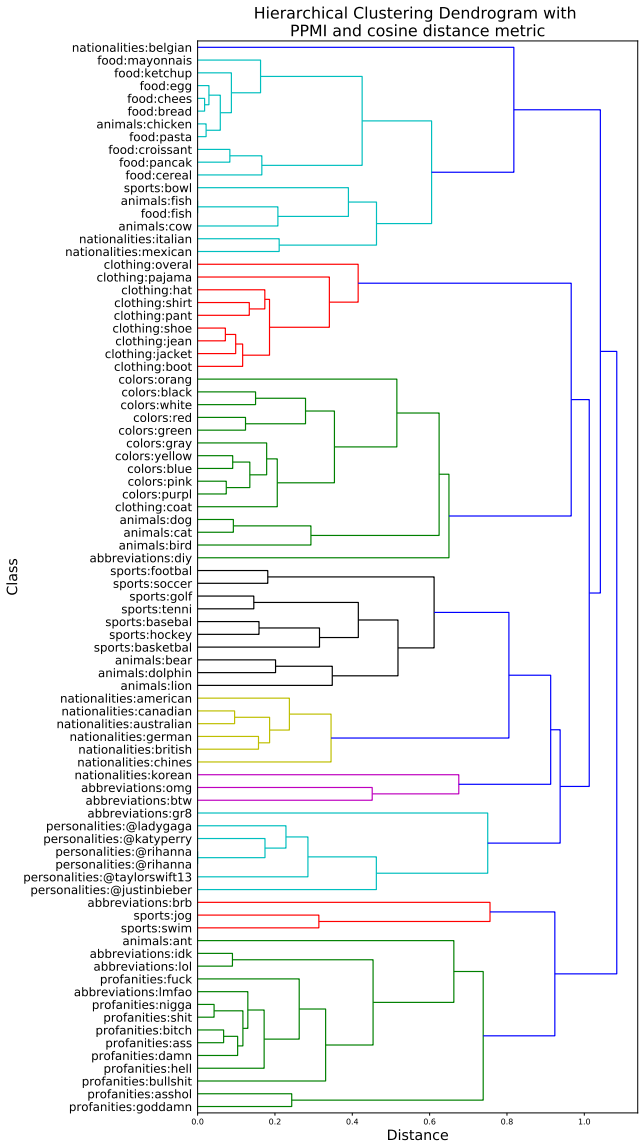

Figure 4. Dendrogram performed on categories of text, using dense PPMI word embeddings

References

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR, 9 2015. URL https://arxiv.org/abs/1409.0473v7.

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning Word Vectors for Sentiment Analysis. Technical report, 2011. URL https://www.aclweb.org/anthology/P11-1015.

Yang Li and Tao Yang. Word Embedding for Understanding Natural Language: A Survey. pages 83–104. Springer, Cham, 2018. doi: 10.1007/978-3-319-53817-4{_}4. URL https://link.springer.com/chapter/10.1007/978-3-319-53817-4_4.

Bofang Li, Aleksandr Drozd, Yuhe Guo, Tao Liu, Satoshi Matsuoka, and Xiaoyong Du. Scaling Word2Vec on Big Corpus. *Data Science and Engineering*, 4(2):157–175, 6 2019. ISSN 23641541. doi: 10.1007/s41019-019-0096-6. URL https://doi.org/10.1007/s41019-019-0096-6.

Rémi Lebret and Ronan Collobert. Word Emdeddings through Hellinger PCA. *arXiv:1312.5542 [cs]*, 1 2017. URL http://arxiv.org/abs/1312.5542.

E. Hellinger. Neue Begründung der Theorie quadratischer Formen von unendlichvielen Veränderlichen. *Journal fur die Reine und Angewandte Mathematik*, 1909(136):210–271, 1 1909. ISSN 14355345. doi: 10.1515/crll.1909.136.210. URL https://www.degruyter.com/view/journals/crll/1909/136/article-p210.xml.

ANLP Lab 8 Notes. Lab 8: Sentiment on Twitter, and working with large files. URL https://www.inf.ed.ac.uk/teaching/courses/anlp/labs2020/lab8.html.

Elia Bruni, Nam Khanh Tran, and Marco Baroni. Multimodal Distributional Semantics. Technical report, 2013.

Herbert Rubenstein and John B. Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633, 10 1965. ISSN 15577317. doi: 10.1145/365628.365657. URL https://dl.acm.org/doi/10.1145/365628.365657.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. Placing Search in Context: The Concept Revisited †. Technical report, 2001.

M. Baroni and A. Lenci. How we BLESSed distributional semantic evaluation. *Association of Computational Linguistics*, pages 1–10, 2011.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global vectors for word representation. In *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pages 1532–1543. Association for Computational Linguistics (ACL), 2014. ISBN 9781937284961. doi: 10.3115/v1/d14-1162.

Bin Wang, Angela Wang, Fenxiao Chen, Yuncheng Wang, and C.-C. Jay Kuo. Evaluating Word Embedding Models: Methods and Experimental Results. *arXiv:1901.09785 [cs]*, 1 2019. doi: 10.1017/ATSIP.2019.12. URL http://arxiv.org/abs/1901.09785.