Word Embeddings and the Brain

Dmitry Kremiansky, Yossi Gisser

Students for data science and engineering at The Technion

Abstract

In this work we based on Pereira, F., Lou, B., Pritchett, B., Ritter, S., Gershman, S. J., Kanwisher, N., Botvinick, M., & Fedorenko, E. (2018). Toward a universal decoder of linguistic meaning from brain activation. Nature communications, 9 (1), 1–13. We tried to extend the work and compare our results to the original results described at the paper.

The project consists of three parts. The first part is about sentence decoding, the second part is about brain encoding and the and the third part is about concept clustering.

1 Introduction & Related work

In the Periera et al. (2018) the try presented a new approach for building a brain decoding system in which words and sentences are represented as vectors in a semantic space constructed from massive text corpora. By efficiently sampling this space to select training stimuli shown to subjects, we maximize the ability to generalize to new meanings from limited imaging data. To validate this approach, we train the system on imaging data of individual concepts and show it can decode semantic vector representations from imaging data of sentences about a wide variety of both concrete and abstract topics from two separate datasets. These decoded representations are sufficiently detailed to distinguish even semantically similar sentences, and to capture the similarity structure of meaning relationships between sentences.

In our work we added additional analysis on the decoding representation they made and in addition we checked the performance using other embeddings.

For the second part of our work, we based on Huth, A. G., De Heer, W. A., Griffiths, T. L., Theunissen, F. E., & Gallant, J. L. (2016). Natural speech reveals the semantic maps that tile human cerebral cortex. Nature, 532 (7600), 453–458. Instead of predicting sentence identities using neural signals (i.e., neural decoding), we predict human neural signals from the embedding vectors representations of the sentences (neural encoding).

In the Huth et al., 2016’s paper they systematically mapped semantic selectivity across the cortex using voxel-wise modelling of functional MRI (fMRI) data collected while subjects listened to hours of narrative stories. They show that the semantic system is organized into intricate patterns that seem to be consistent across individuals. They then use a novel generative model to create a detailed semantic atlas. Their results suggested that most areas within the semantic system represent information about specific semantic domains, or groups of related concepts, and their atlas shows which domains are represented in each area. Their study demonstrated that data-driven methods—commonplace in studies of human neuroanatomy and functional connectivity—provide a powerful and efficient means for mapping functional representations in the brain.

In our work we fit a separate linear model for each voxel in the dataset (384 sentences). For each model we calculated the R^2 score and examine how many voxels are significantly associated with the information embedded in the word vectors. We use 2 different embedding vectors for this task.

2 Structured Part

We started by load the data and run the same analysis that we did in HW 3 Q 3. The difference in this approach from the previous is that we use fasttext embedding (instead of glove). The embedding that we used is 'fasttext-wiki-news-subwords-300' from genism library. We saw that all our words are already in the pretrained model, so we didn’t have the problem of out-of-

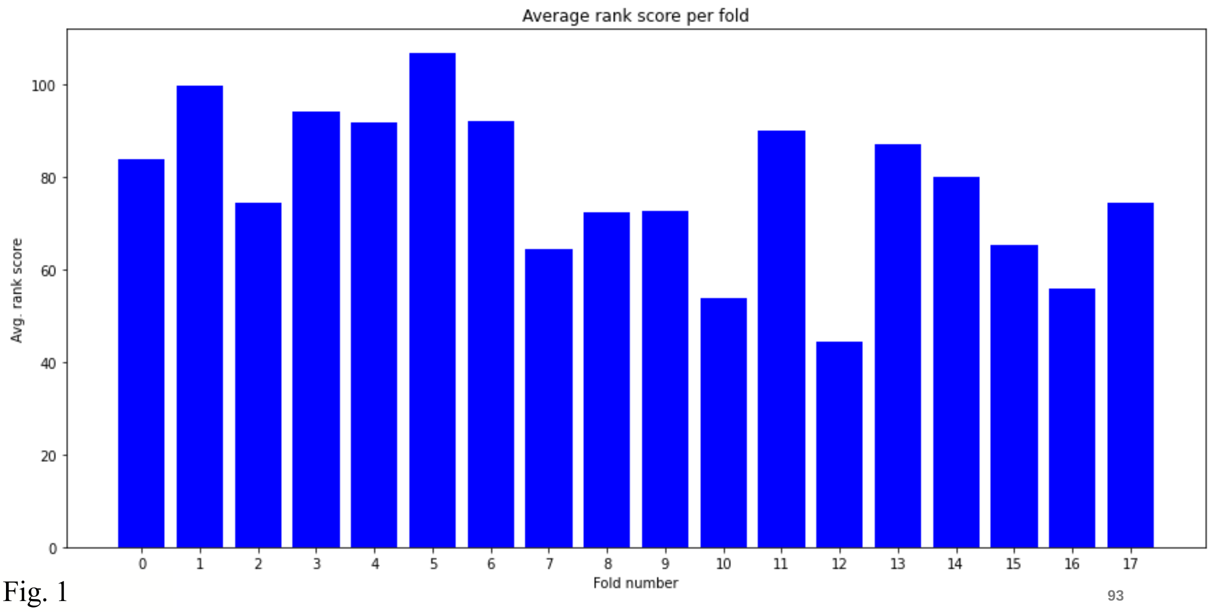

Fig. 1

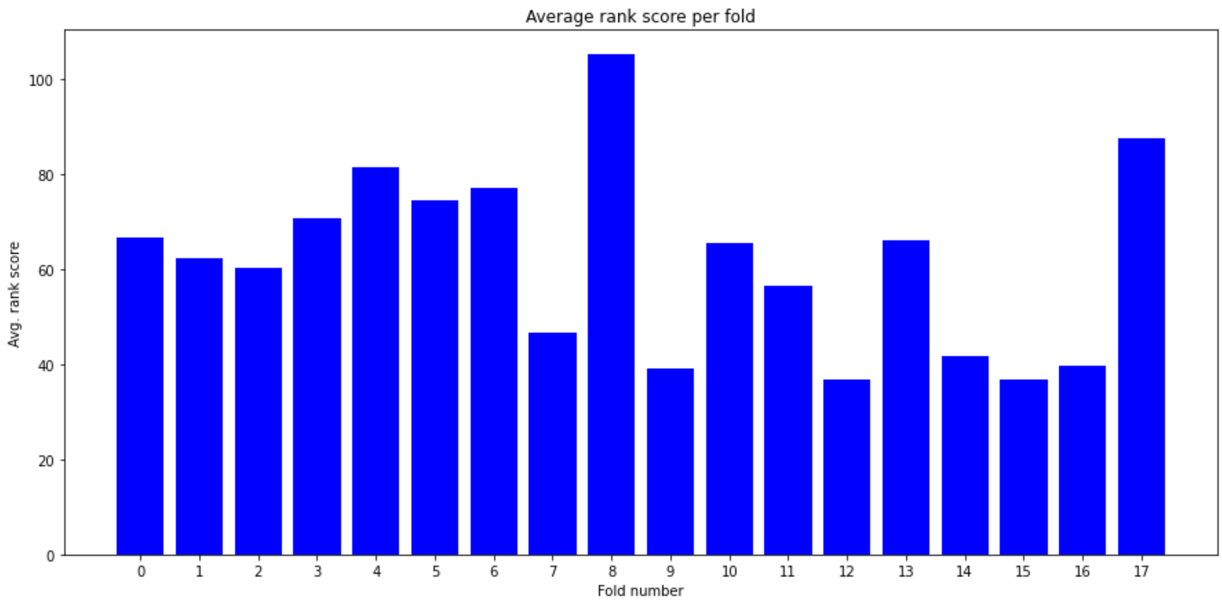

Fig. 2

vocabulary. The results that we obtained are similar to the results that we obtained using glove, but not identical. In the graph we can see that the fold with highest rank score (the worse fold) is 8 in both cases. The total average rank score is lower using fasttext (55.76) than using glove (61.9). The 10 best and then worst concepts according to the rank score have overlap in both methods but they are not the same.

The results using fasttext (Fig. 1). The results using Glove (Fig. 2).

After reading the Periera et al. (2018) we assumed that the similarity of all 3 experiments that are described in the paper are that in all of them the representation was sentences. The difference between experiment 1 and experiments 2 and 3 that in experiment 1 they have also 2 other paradigms: with an image, or with five related words. The difference between experiment 2 and 3 is that in experiment 2 every topic has subtopics and in experiment 3 every topic has 3 passages (not subtopics).

We use the same model we trained in Homework 3 using Glove embedding (after tiny change for matching the dimension and retrain on all dataset

1) and test it on dataset 2 (384 sentences) and dataset 3 (243 sentences).

For each dataset, we used the learned decoder model to decode sentence representations and evaluate the performance via the rank accuracy method (as we did in HW3).

The decoding process is evaluated in the following way. A decoder is trained on imaging data from the training set, and then used to decode the imaging data from the test set. The decoded vectors are evaluated according to the "average rank" metric: - Average rank: Given a decoded vector $\hat{v}$ for a concept whose true semantic vector is v, rank all the semantic vectors in order of their closeness (cosine similarity in our case) to $\hat{v}$. Now calculate the position of v in this ranking: for example, if the vector for v is the 10th closest vector to $\hat{v}$, then the rank is 10. Then we can get the average ranking for all the decoded vectors. The average ranking is an accuracy score where the optimal score would be 1 and the worst possible score would be number of sentences. If the decoder is outputting random noise, then the resulting average rank should be number of sentences / 2.

The average rank score on dataset 2 is 156.9 and the average rank score on dataset 3 is 100.7. In both cases we got average rank score less than half of number of sentences. About 40% of number of sentences for each dataset.

Now, after getting the results we analyzed them. The analysis that we apply is sorting them according to the rank score and identify the best and the worst performing topics.

The top 5 topics from dataset 2:

	topic_id	rank	topic_name
3	4	70.4375	body_part
13	14	92.7500	human
8	9	98.9375	drink_non_alcoholic
9	10	113.6875	dwelling
1	2	113.7500	appliance

The bottom 5 topics from dataset 2:

17	18	185.0000	music
22	23	186.6875	vehicles_transport
0	1	196.2500	animal
21	22	237.6250	vegetable
19	20	249.1250	profession

The top 5 topics from dataset 3:

	topic_id	rank	topic_name
6	7	52.800000	dreams
21	22	58.200000	stress
4	5	59.100000	castle
13	14	63.636364	opera
3	4	68.090909	bone_fracture

The bottom 5 topics for dataset 3:

16	17	136.600000	pharmacist
18	19	147.200000	pyramid
12	13	148.100000	lawn_mower
14	15	148.600000	owl
1	2	169.818182	beekeeping

3 Semi-Structured Part

Decoder Model

In this part we train the model on dataset 2, using K-Fold Cross-Validation with K=10. We repeated this twice, for two different embedding vectors: Glove (the original embedding from the paper) and Bert (specific 'sentence-transformers/bert-base-nli-mean-tokens'.

The average rank score for the Glove embedding is 193.3. This is worse result than the result we obtained trained on dataset 1 and test on dataset 2. This result is like random results, and even little bit worse than random results. Fig. 3 represents the average rank score per fold for Glove embedding.

For using Bert consistently with our dimension of the embedding vectors, we apply PCA (Principal Components Analysis) on the embedding vectors from Bert output and reduce the dimension from 768 to 300.

The average rank score for the Bert embedding is 194.7. This is worse result than the result we obtained with Glove. It could have happened because of the PCA that we apply that loses

B”H

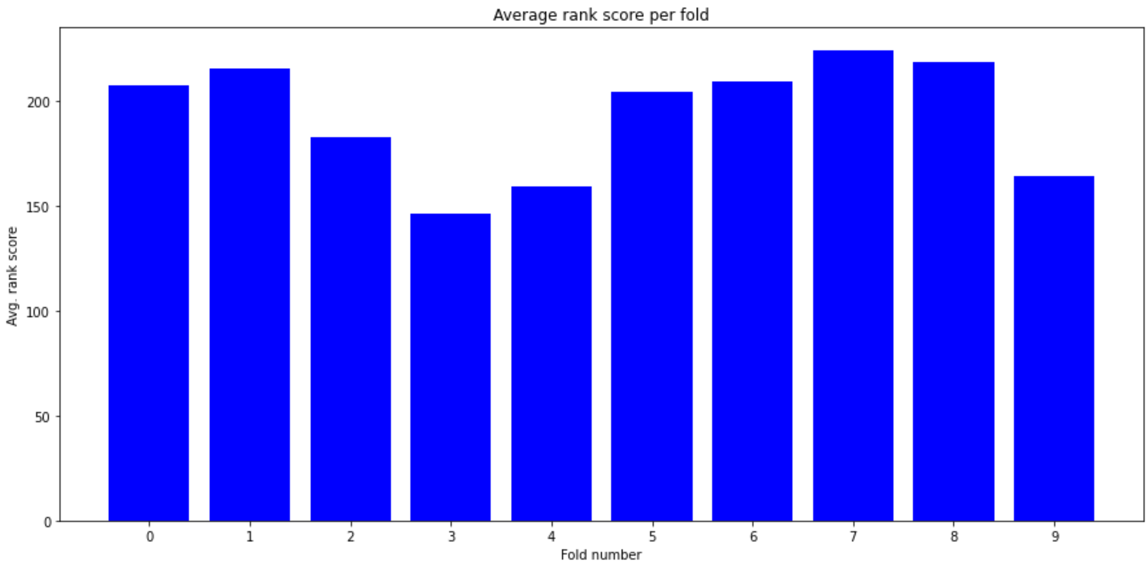

Fig. 3

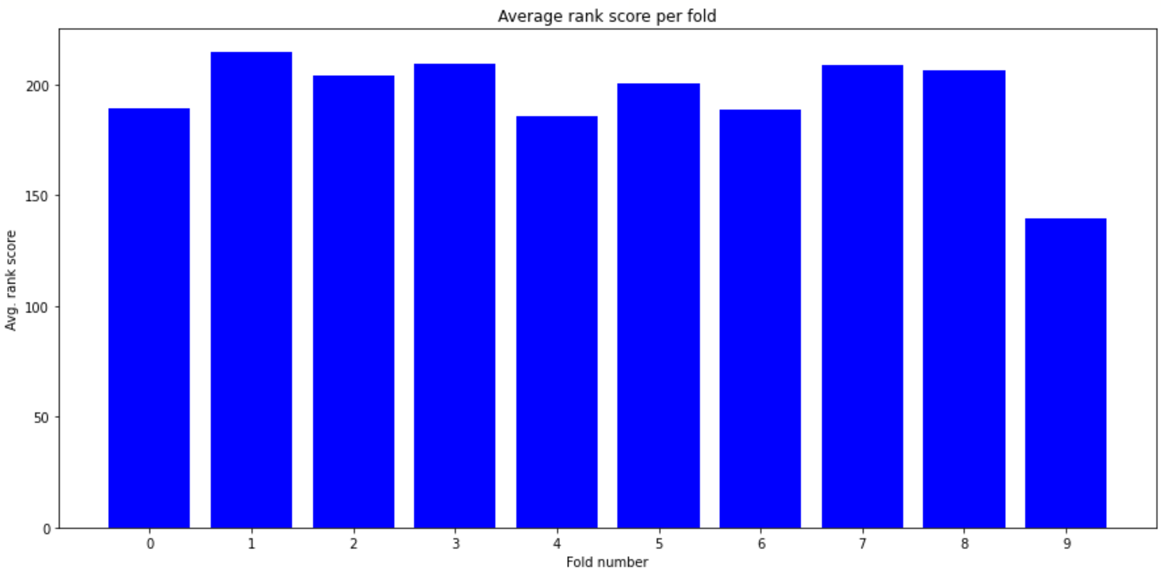

Fig. 4

information. Fig. 4 represents the average rank score per fold for Bert embedding.

Brain encoder

Instead of predicting sentence identities using neural signals (i.e., neural decoding), we try to predict human neural signals from the embedding vectors representations of the sentences (neural encoding).

We fit a separate linear-regression model for each voxel in the dataset 2 (384 sentences). For each voxel/model, calculate the R^2 score and examine how many voxels are significantly associated with the information embedded in the word vectors.

We repeat This twice: using Glove embedding and using Bert embedding. We checked the average R^2 score, the average adjusted R^2 score and the proportion of significant voxels every 1000 voxels. The average is moving average.

The results for the Glove embedding: After the first 1000 voxels the results were - The average R^2 is: 0.808, The average R^2 adjusted is: 0.113, The proportion of significant voxels is: 0.169. The final results were - The average R^2 is: 0.803, The average R^2 adjusted is: 0.09, The proportion of significant voxels is: 0.16.

The results for the Bert embedding: After the first 1000 voxels the results were - The average R^2 is:

0.801, The average R^2 adjusted is: 0.0831, The proportion of significant voxels is: 0.126. The final results were - The average R^2 is: 0.809, The average R^2 adjusted is: 0.12, The proportion of significant voxels is: 0.21.

In this task we can see that in all measures the Bert is better than Glove. Specifically, if we focused on the proportion of the significant voxels, we could see that on Glove embedding the proportion start with increasing but then after about 70000 voxels it start decreasing to similar value that we had in the beginning but on Bert embedding it start with lower value, but it keeps increasing and in the end the proportion is higher than on Glove embedding. We could see a similar trend with adjusted R^2. The R^2 keeps the same value more or less along all iterations.

4 Open-ended Task

In the beginning we decided to apply clustering method on dataset 1 for better understanding of the performance that we obtained on dataset 1 using fasttext in the structured part (it could also be used for checking the results that we obtained in HW3 using Glove, but here we focused on the project results).

The clustering algorithm that we used is K-Means with K=18 (as the number of folds that we use in K-Fold Cross-Validation). We tried two variations of K-Means: without constrain on the clustering and with constrain on cluster size that each cluster will be with size 10. Each one of the clustering algorithms we applied on Glove embedding vectors and on Bert embedding vectors (without using PCA for reducing dimension to 300 because num of samples is 180 and that is less than 300 and because of that we couldn't using PCA).

The evaluate method that we use is silhouette score with distance metric 'cosine similarity'. This method computes the mean Silhouette Coefficient of all samples. The Silhouette Coefficient is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample. The Silhouette Coefficient for a sample is $(b - a) / max(a, b)$. To clarify, b is the distance between a sample and the nearest cluster that the sample is not a part of. The best value is 1 and the worst value is -1. Values near 0 indicate overlapping clusters. Negative values generally indicate that a sample has been assigned to the wrong cluster, as a different cluster is more similar.

The results of the Silhouette score are:

	No constrain	With constrain
Glove	0.027	0.016
Bert	-0.11	0.03

For no constrain K-Means algorithm on Glove embedding we find out that 3 of the 10 best preforming concepts using fasttext are in the same cluster. The concepts are: laugh, smiling and emotionally. On Bert embedding this algorithm put just laugh and smiling together. The K-Means constrain algorithm put in few clusters just two words together from best or worst 10 concepts. Therefore, our try for explanation of the rank score results wasn't successful enough.

In addition, another analysis that we did is that we checked the performance of our decoder trained on one dataset and tested on other, like what we did in the structured part.

We trained the decoder on dataset 2 using Glove and Bert embeddings and tested it on dataset 1 and 3 when in both cases dataset 1 and 3 where embedded using Glove (there are problem to use Bert embedding on them because the number of sentences and concepts are less than 300).

	Dataset 1	Dataset 3
Glove	80.04	94.4
Bert	89.4	118.1

The surprise in this that Bert achieve worse results than Glove. Possible explanation that it is because of the PCA.

Top 5 topics from dataset 3 (using Glove embedding on dataset 2):

	topic_id	rank	topic_name
17	18	34.100000	polar_bear
22	23	50.444444	taste
9	10	57.500000	ice_cream
6	7	68.400000	dreams
11	12	71.700000	law_school

Bottom 5 topics from dataset 3 (using Glove embedding on dataset 2):

16	17	118.500000	pharmacist
13	14	121.000000	opera
23	24	131.300000	tuxedo
2	3	138.300000	blindness
1	2	143.363636	beekeeping

Top 5 topics from dataset 3 (using Bert embedding on dataset 2):

	topic_id	rank	topic_name
1	2	65.727273	beekeeping
2	3	83.200000	blindness
16	17	86.300000	pharmacist
14	15	87.200000	owl
15	16	88.600000	painter

Bottom 5 topics from dataset 3 (using Bert embedding on dataset 2):

12	13	139.100000	lawn_mower
17	18	150.800000	polar_bear
0	1	152.000000	astronaut
18	19	160.100000	pyramid
9	10	168.700000	ice_cream

Also, we trained the decoder on dataset 3 using Glove embeddings and tested it on datasets 1 & 2 using Glove embedding.

The average rank score on dataset 1 is 83.45 and the average rank score on dataset 2 is 151.8.

The top 5 topics from dataset 2:

	topic_id	rank	topic_name
9	10	88.6875	dwelling
13	14	93.5000	human
3	4	93.6250	body_part
8	9	100.5000	drink_non_alcoholic
12	13	105.1875	furniture

The bottom 5 topics from dataset 2:

22	23	187.3125	vehicles_transport
11	12	187.3750	fruit
17	18	218.9375	music
19	20	220.5000	profession
21	22	238.8750	vegetable

From comparing the top and bottom topics for each dataset 2 & 3 that we got in this part to the top and bottom topics that we got in the structured part we can see that there is a resemblance between the results.

5 Link to the code

The whole code the reproduced this work can be found in the following link: Word-Embeddings-and-the-Brain.