Semantic Autoencoder for Memory and Recall

Ari Beller

March 2021

Introduction

From early in the course, I've been interested in using neural networks to model memory. Our investigations into the McClelland and Rogers (2003) semantic network highlighted the capacity of neural nets to help us understand the underlying memory representations that give rise to behavior we observe in humans. I liked this modeling direction, and I was interested to try and expand on it in my final project. One idea that caught my interest was to think about a semantic neural network as a model of a memory representation in the Semantic Fluency Task (Bousfield & Sedgewick, 1944; Thurstone & Thurstone, 1938). The Semantic Fluency Task (SFT) is a memory assessment often used to assess memory deficits, where the participants freely recall as many elements of a category they can think of (usually the category is animals). One of the most commonly observed empirical patterns in this task is that participants tend to recall items in semantically related clusters (Troyer, Moscovitch, & Winocur, 1997). Thus if a participant is tasked with recalling as many animals as they can think of, they might start by recalling a sequence of farm animals, followed by a sequence of sea creatures, and then a sequence of birds, and so on. These semantic clusters reminded me of the semantic clusters we observed in the hidden representations of the McClelland and Rogers network, and got me thinking that there could be an interesting connection here.

In this course project, I set out to model aspects of the Semantic Fluency Task using a variant of the semantic network in McClelland and Rogers, a semantic autoencoder. My investigation proceeded in two parts. In the first part I explored the cluster structure in an animal dataset provided to me by Professor McClelland. Once I had a sense of the structure that existed in the data, I trained a semantic autoencoder to learn a representation of the data, and then compared the structure in that learned representation to the structure in the original data. Finding that the semantic autoencoder did a good job of capturing cluster structure in the data, I went on to design a model to explore the process of recall in the SFT. In the second part of my investigation, I designed a random walk in the memory space learned by the semantic autoencoder to simulate recall. With tuning, the random walks were able to capture response time characteristics of human behavior observed in prior research.

Part A: Cluster Structure in Model and Data

In the first part of the project I compared cluster structure in an animals dataset to cluster structure in the learned representations of the model.

Dataset

Throughout the project I worked with a dataset provided to me by Professor McClelland. The Mammals Dataset consists of 50 mammals represented by feature vectors of 85 dimensions, where feature values are collected from human raters. I concatenated a 50 dimensional one-hot vector to each of the feature vectors to represent the label. In order to perform the reconstruction task, the semantic autoencoder would need to reconstruct both the features and the class label from the lower dimensional representation space.

I performed a cluster analysis to uncover structure in the dataset. I used the affinity propagation algorithm (Frey & Dueck, 2007) which has the advantage of not requiring a pre-specified number of clusters, instead discovering an appropriate number in the computation. The results of this analysis are depicted in Figure-1. Figure-1A visualizes the discovered clusters in a low dimensional projection space. Notably certain clusters are closer together than others suggesting their may be hierarchical cluster structure in the data. Figure-1B depicts the cosine similarity between each pair of items in the dataset. The clusters discovered by affinity propagation are apparent along the diagonal. The large bounded box of above average similarities in middle supports the intuition of hierarchical clustering. All clustering and dimensionality routines were performed using Scikit-Learn's builtin libraries (Pedregosa et al., 2011).

Model Design

I implemented a variant of a semantic network, a semantic autoencoder, as a model of human memory for the mammals dataset. The autoencoder is a neural network architecture that learns to reconstruct input data in an unsupervised fashion. Critically, the network must reconstruct the input from a lower dimensional projection, learning an efficient low dimensional representation of the data in the process. The low dimensional representation offers a means to investigate the memory representation of the model. After training the network on the data, we can investigate these low dimensional representations in order to get a deeper sense of what structure the network has extracted.

Figure-2A depicts a schematic of the autoencoder architecture. The model is fairly simple. The 135-dimensional input vector projects through a single layer, the encoder, to the lower dimensional 50-unit hidden layer. The hidden representation then projects back out to the input space through a single layer decoder. The encoder uses a hyperbolic tangent activation function, while the decoder uses a sigmoid activation function. I trained the model for 2000 epochs using mean squared error reconstruction loss and the Adam optimizer. Figure-2B depicts the loss trajectory across training. The model quickly learns the

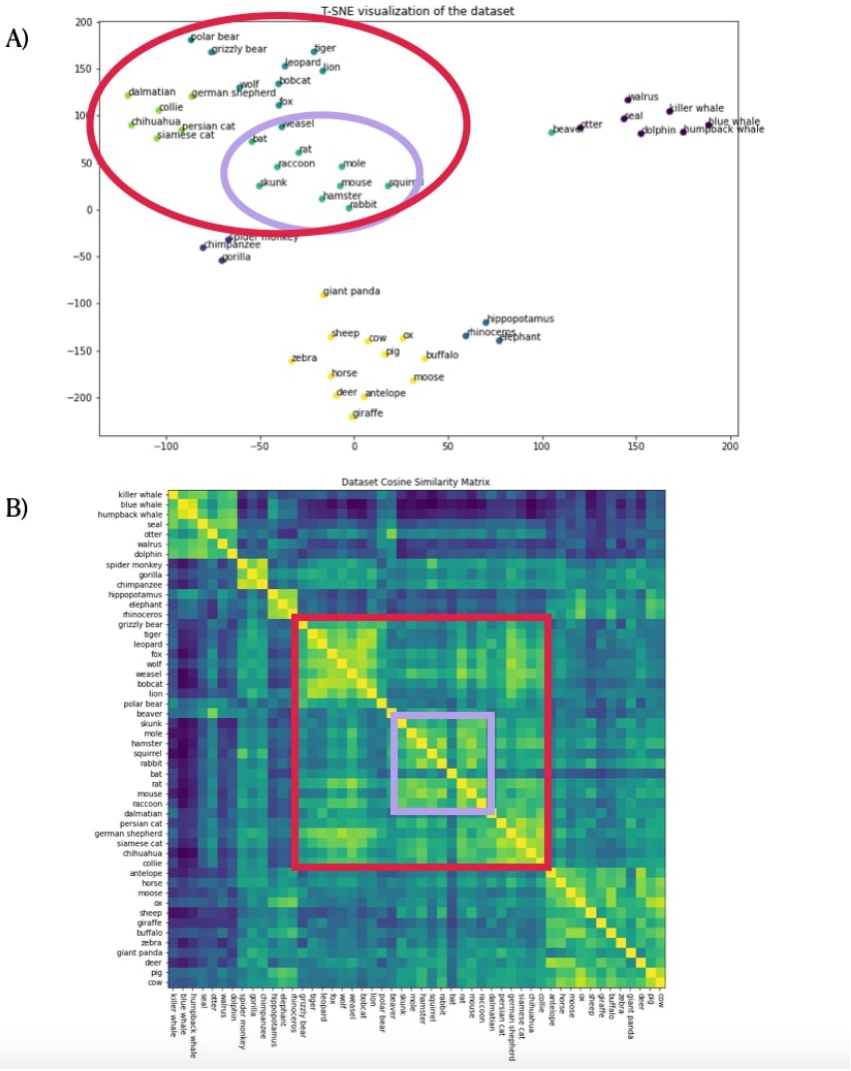

Figure 1: Cluster Analysis of the dataset. 1A depicts a low-dimensional projection of the dataset computed using the T-SNE dimensionality reduction technique (Van der Maaten & Hinton, 2008). The colors of the points represent the different clusters computed using the affinity propagation algorithm. Certain clusters are closer to each other than others, suggesting there may be hierarchical cluster structure in the data. 1B depicts the pairwise cosine similarity between every pair of points in the dataset. The clusters discovered by affinity propagation are apparent along the diagonal. The similarity matrix further supports the presence of hierarchical cluster structure as can be seen by the wide high similarity box in the middle of the matrix, with internal subregions of greater cluster coherence. The purple bounded regions highlight a lower level cluster and the red bounded regions highlight a higher level cluster in both the low-dimensional projection and the similarity matrix.

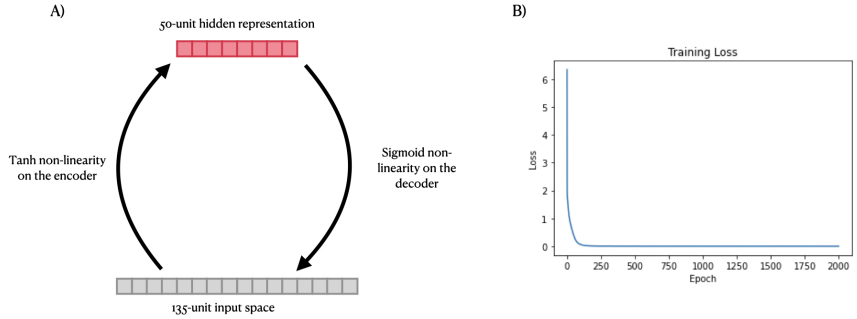

Figure 2: **A)** A schematic of the semantic autoencoder architecture. A single layer encoder projects from the input space to the lower dimensional hidden representation. The encoder uses a hyperbolic tangent activation function. The decoder projects back from the hidden representation to the input space. The decoder uses a sigmoid activation function. The model was trained for 2000 epochs using mean squared error reconstruction loss and the Adam optimizer. The model was implemented using the neural network library Pytorch (Paszke et al., 2019). **B)** Training loss of the model across epochs. The model quickly masters the reconstruction task and remains at peak performance throughout most of training.

Hidden Units	10	20	30	40	50
Correlation Coefficient (r)	0.51	0.56	0.66	0.80	0.93

Table 1: Correlation between the pairwise similarity matrix of the data and the corresponding matrix of the learned representation vectors, for semantic autoencoders with different sized hidden layers. With 50 hidden units, the semantic network captures much of the similarity structure evident in the network. Correlation declines substantially with smaller network size. All correlation values are averaged over networks initialized with five different random seeds.

structure in the data and performs at ceiling for the majority of the training process.

Model Analysis

Once we have trained the network, we can use the learned representations to investigate structure in the memory of the model. A natural question that we might be interested to ask is how well does the model preserve structure in the data in its hidden representations? Given my interest in the SFT, I was particularly interested to investigate how well the model preserved the cluster structure that we observed in the data in the original cluster analysis.

I performed a cluster analysis on the learned representations of the semantic autoencoder, analogous to the one I performed previously on the dataset. The results are depicted in Figure-3. 3A depicts the low-dimensional projection of the learned representations of the datapoints. The coloring of the points reflects the output of the affinity propagation algorithm performed on the learned representations. Broadly the structure is similar to that observed in the original dataset with some notable differences. Most notably, affinity propagation has split the sea creatures cluster into two subclusters, one more fish like (whales and the dolphin), and the other more land mammal like (seal, walrus, otter, and beaver). On the whole though the clusters are largely similar, most animals staying within the same clusters with a few other switches.

Figure-3B depicts the pairwise cosine similarity matrix for the learned representations of the semantic autoencoder. The cluster structure is more faint here, but still largely preserved on the whole. In particular the seven clusters identified in the original cluster analysis of the data are apparent in the learned representation as well, with the arguable exception of the rodent cluster in the middle of the matrix, which is far more faint. The hierarchical cluster structure noted in the original cluster analysis is also apparent, though again, more faint.

In order to quantify the degree of similarity structure from the data preserved in the learned representation, I computed the element-wise correlation between the similarity matrix of the data and the similarity matrix for the learned representations of the model. Curious about how the effectiveness of the network in preserving structure in the data would decline with smaller hidden representations, I performed the same assessment for networks with successively smaller

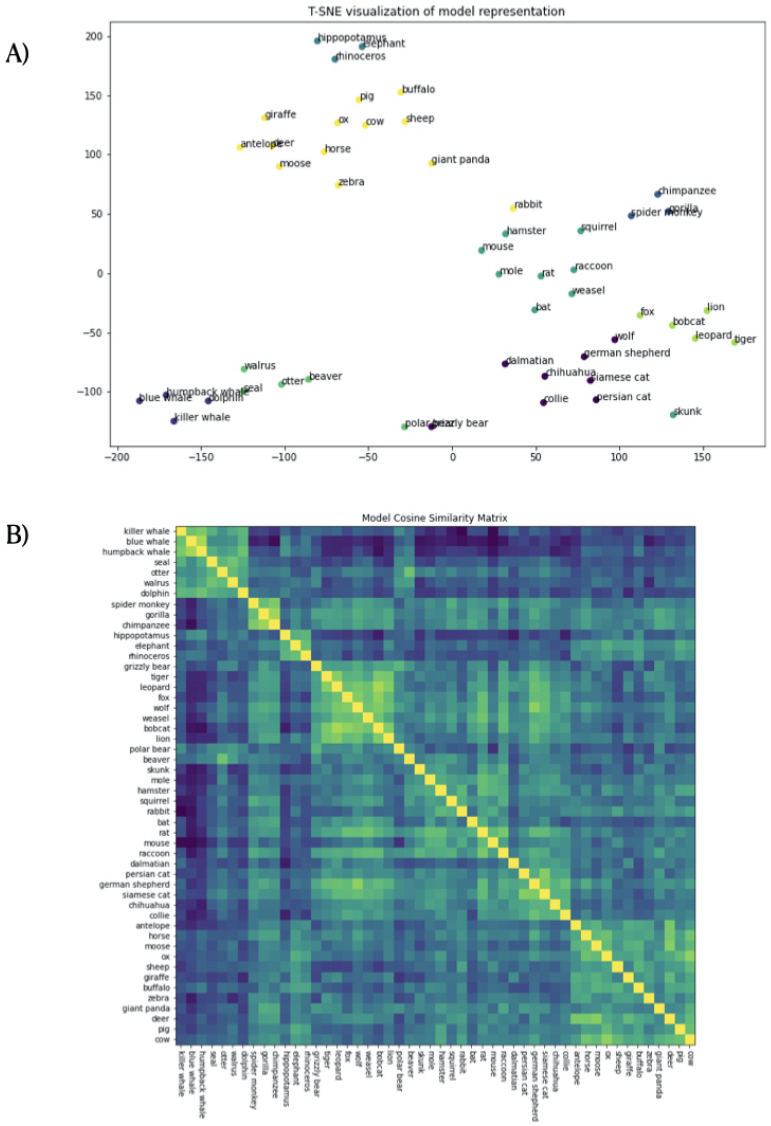

Figure 3: Results of the cluster analysis performed on the hidden representations of the semantic autoencoder. **3A** depicts the low-dimensional projection of the hidden representation vectors computed as before using T-SNE. The colors represent the clustering assignments from the affinity propagation algorithm on the hidden vectors. While the clustering is largely similar, there are some notable differences including the bifurcation of the sea creatures cluster into two sub-clusters (bottom left). **3B** depicts the cosine similarity matrix of the hidden representations, ordered according to the cluster structure of the original data in order to facilitate comparison. The cluster structure is largely preserved, though in general more faint than in the original dataset.

hidden unit representations (40 units, 30 units, 20 units, and 10 units). The results are depicted in Table-1. The full model does a fairly good job of preserving similarity structure in the original dataset, with a correlation coefficient of 0.93. Correlation falls substantially with each successive decrement in hidden unit size, calling into question whether lower dimensional models can adequately capture the structure in human memory.

Part A: Summary

With a sufficient number of hidden units, the semantic autoencoder does a good job of capturing cluster structure in Mammals Dataset. This suggests that learned representations in a semantic autoencoder could be a useful tool for modeling memory representations in the Semantic Fluency Task. In order to address this question more deeply, we need to think more about the process of recall, to which I turn next.

Part B: Modeling Recall as a Random Walk

As I noted in the introduction, a consistently observed and theoretically interesting finding in the SFT is that participants tend to recall items from a category in semantically related clusters. From this we've inferred that the underlying memory representations are analogously structured, spurring our interest in the semantic autoencoder as a candidate model for memory in the setting. In order to explain the phenomenon, we need a process model which operates on this representation in such a way that the underlying cluster structure is respected.

A natural candidate for this process is a random walk. A random walk is a stochastic process where the value at each successive step in the process is dependent on the value of the previous steps in the sequence. Random walks have been utilized to model recall in the SFT before. Hills, Jones, and Todd (2012) develop a random walk model that operates on a vector space language model, and Abbott, Austerweil, and Griffiths (2015) develop a similar approach where the walk operates on an associative graph[1]. Here I was interested to design a similar approach, where the underlying representation that the walk traverses is the memory space defined by the hidden representations of the semantic autoencoder.

Designing the Random Walk

The random walk is defined over the points that represent the animals in the memory space of the semantic autoencoder. The walk specifies the probability of transitioning to any animal in the memory space given the current animal.

[1]Note, Abbott et al. refer to the underlying representation in their model as a semantic network. The network is not an artificial neural network, but instead an empirically defined graph where the connections between nodes are based on the associative frequency between items collected from human free-association experiments.

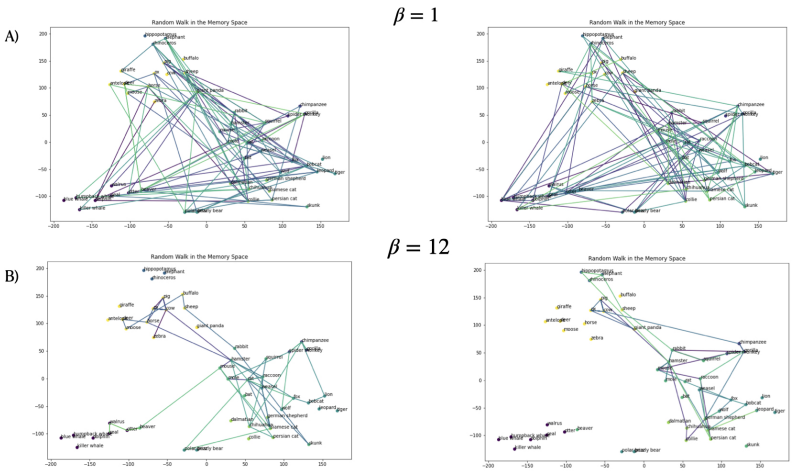

Figure 4: Visualizations of 100 step random walks in the low dimensional projection space of the model hidden representations. Darker color in the walk line indicates an earlier step in the walk, where lighter colors signify later. **A** depicts two untuned walks ($\beta = 1$) in the memory space. The walks are highly diffuse throughout the space, moving from one point to another without respect for the underlying cluster structure. **B** depicts two tuned walks ($\beta = 12$). The respect for cluster structure is more clearly discernable in these examples, with mostly local steps interspersed with large jumps between clusters.

Following Hills et al. I define the probability of transitioning from one animal to another in terms of their similarity in the memory space. Formally, the probability of transitioning from point i in the memory space to point j is as follows:

$$P(X_t = j | X_{t-1} = i) = \begin{cases} \dfrac{S(i,j)^\beta}{\sum_{k=1}^{N} S(i,k)^\beta} & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Here N is equal to the number of animals in the dataset, β is a parameter controlling the entropy of the conditional probability distributions, and S is a similarity function. In my experiments I defined S as inverse Euclidean distance. Note that walks can return to the same points as before, though they cannot stay where they are on any given step. For the reported experiments, each walk ran for 100 steps.

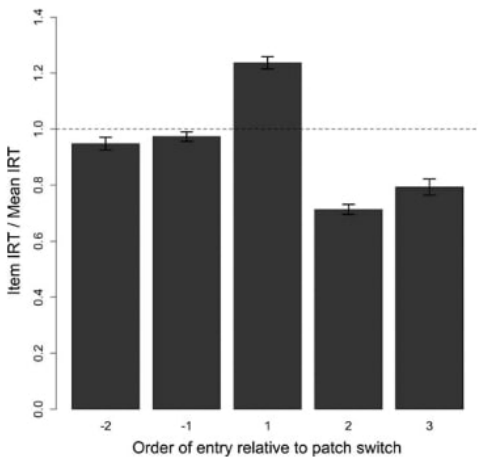

Figure 5: Empirical pattern of inter-item response time in SFT from Hills et al. (2012). The switch between semantic clusters of animals (represented as the transition on the x axis that switches from -1 to 1) on average takes substantially longer than transitions within cluster.

Analysis

I performed a qualitative exploration of the behavior of the random walks to better understand whether they capture characteristics of human behavior in the SFT. One thing that became quickly apparent is that the walks needed to be tuned using the beta parameter in order to resemble patterns of human behavior. Figure-4A depicts two examples of the untuned random walk, visualized in the low-dimensional projection space of the model hidden representations. The walks are highly diffuse throughout the space, hopping about without any respect for the underlying cluster structure. However, with tuning, the walks begin to conform to the types of patterns we see in the SFT. Figure-4B depicts two examples of random walks with beta equal to 12. The walks largely obey cluster structure in the data, making mostly short hops from point to point within cluster, and occasionally making a larger jump to switch clusters.

One interesting pattern that Hills et al. observe in their human data for the SFT is that response times between recollected animals show spikes when participants switch between semantic clusters of animals. Figure-5, taken from Hills et al. (2012), demonstrates this pattern. Inter-item response time from the last item in the previous cluster (-1 on the x-axis) to the first item in the current cluster (1 on the x-axis) is in general substantially longer than within cluster steps.

Our random walk model of human recall in the SFT can capture this pattern of human responses. Using distance in the memory space as a stand-in for inter-item response time, we can plot the distance between successive steps of

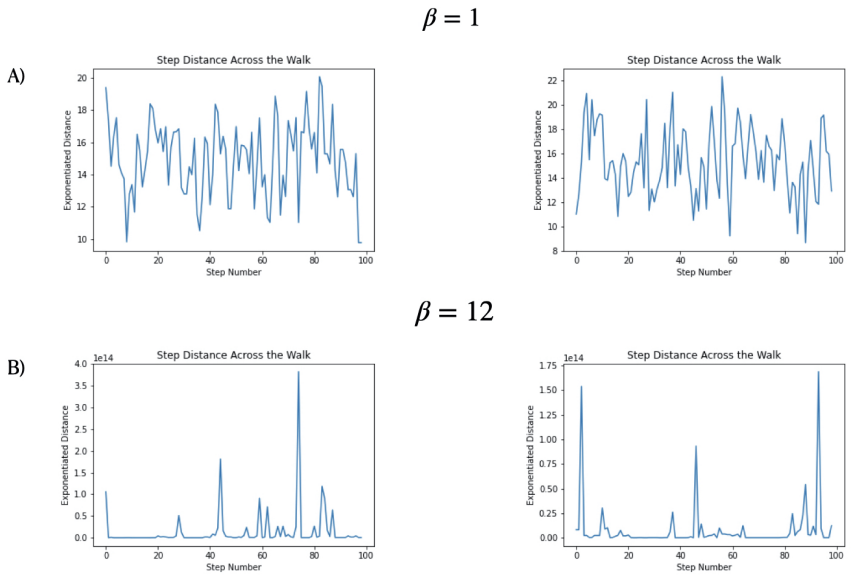

Figure 6: Distance across time graphs for the random walks depicted in Figure-4. Each graph plots the distance traveled at each step in the random walk (magnified by the beta parameter). Untuned walks display a highly noisy pattern without human-like characterists. Tuned walks on the other hand evidence a pattern of cluster-hopping. Long periods of short distance within-cluster hops are interspersed with large distance jumps representing cluster switches. This pattern mirrors the iter-item response time characteristics from Hills et al.

the walk (magnified by the β parameter) across the time course of the walk. Figure-6 plots the distance traveled at each timepoint for the random walks depicted in Figure-4. For the untuned walks in Figure-6A, the patterns are highly noisy and not human like. However for the tuned walks in 6B, we see the human like pattern of long periods of small distance hops, interspersed with spikes representing cross cluster jumps. This suggests that our random walks can capture certain aspects of human recall in the SFT.

Part B: Summary

This investigation suggests that a random walk in the memory space of a semantic autoencoder can capture certain characteristics of human behavior in the Semantic Fluency Task. Going forward it would be interesting to collect data with human participants and put the model to the test.

Conclusion

In this project, I developed an approach to model human cognition in the Semantic Fluency Task. Noting the semantic cluster structure in human responses in the SFT, and drawing a parallel to the semantic cluster structure we observed the McClelland and Rogers semantic network, I set out to design a neural network that could model human memory representations in the task. I built a semantic autoencoder, trained on a dataset of animals, and used the memory representations in the hidden layer of the autoencoder as a stand-in for human memory representations in the task. Comparing the cluster structure in the autoencoder's learned representation to the original cluster structure in the data, I found there was a good correspondence between the two in networks with large enough memory capacity.

With these memory representations in hand, I designed a model of sequential recall in the learned network representations to mimic the process of human recall in the SFT. I implemented a random walk in the memory space that transitioned between items in the dataset based on their distance in the learned space. I found that, with tuning, these walks were able to capture certain patterns of human response in the SFT, namely the inter-cluster spikes in response time that humans exhibit in the task.

Going forward, I'd be interested to collect actual human data and attempt to model human thinking in this domain with greater rigor. There is a lot of work that could be done refining the nature of the memory representation, experimenting with different types of random walks, and thinking about more precise quantitative ways to characterize the patterns of human responses. I think this is a really interesting area that gets at the heart of questions about representation and process in human cognition. I think this would be a very fun space to engage with going forward.

Coda

Thanks so much to Jay and Effie for all your guidance this quarter! Code for the project is available at: `https://github.com/aribeller/psych209_finalproject`

References

Abbott, J. T., Austerweil, J. L., & Griffiths, T. L. (2015). Random walks on semantic networks can resemble optimal foraging. In *Neural information processing systems conference; a preliminary version of this work was presented at the aforementined conference.* (Vol. 122, p. 558).

Bousfield, W. A., & Sedgewick, C. H. W. (1944). An analysis of sequences of restricted associative responses. *The Journal of General Psychology*, *30*(2), 149–165.

Frey, B. J., & Dueck, D. (2007). Clustering by passing messages between data points. *science*, *315*(5814), 972–976.

Hills, T. T., Jones, M. N., & Todd, P. M. (2012). Optimal foraging in semantic memory. *Psychological review*, *119*(2), 431.

McClelland, J. L., & Rogers, T. T. (2003). The parallel distributed processing approach to semantic cognition. *Nature reviews neuroscience*, *4*(4), 310–322.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems 32* (pp. 8024–8035). Curran Associates, Inc. Retrieved from `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Thurstone, L. L., & Thurstone, T. G. (1938). *Primary mental abilities* (Vol. 119). University of Chicago Press Chicago.

Troyer, A. K., Moscovitch, M., & Winocur, G. (1997). Clustering and switching as two components of verbal fluency: evidence from younger and older healthy adults. *neuropsychology*, *11*(1), 138.

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, *9*(11).