GloVe Dataset Analysis Using High-Dim Tools

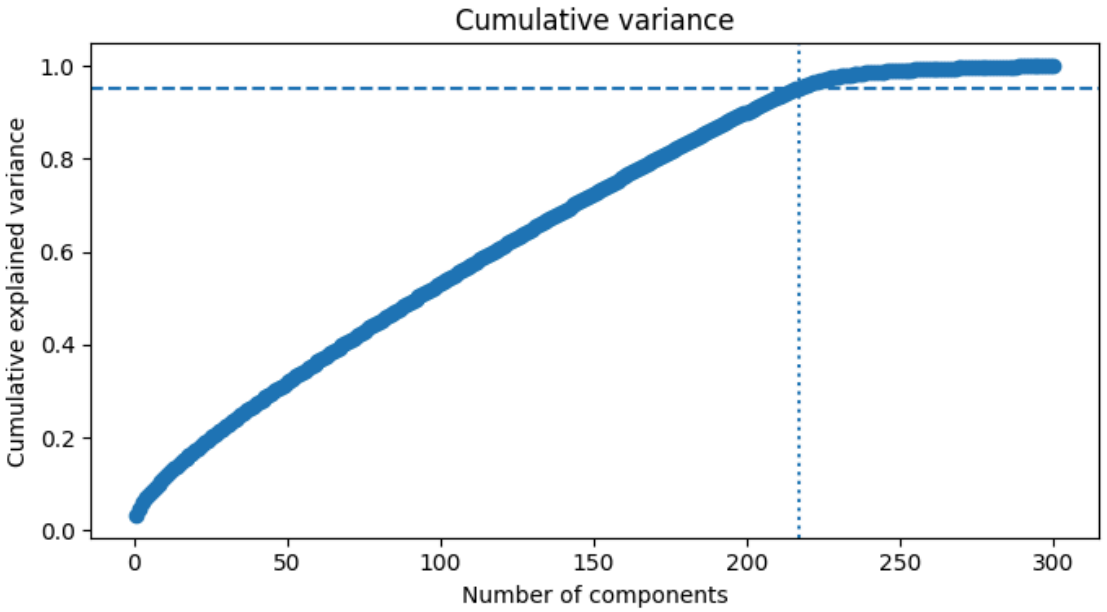
Roy Harel

1. Introduction

I chose a dataset of pre-trained GloVe word embeddings[1], which consists of 400,000 English words represented as 300-dimensional vectors trained on Wikipedia and the Gigaword News Corpus. One would assume that semantic similarity between words should manifest as clustering behavior in the embedding space. The goal of this project is to verify and explore this assumption using high-dimensional data analysis methods.

2. Intrinsic Dimension Estimation

We are interested in intrinsic dimension estimation as a way to better understand the structure of word embeddings: although they live in a 300-dimensional space, the true number of meaningful degrees of freedom is usually smaller. To test this, we first applied PCA. The scree plot and cumulative variance plot show that to preserve 95% of the variance we need 217 components, with an elbow estimate of 224 (the elbow estimate being a rough indication of where adding further components becomes inconsequential). This suggests that, at the global scale, the embeddings do span a very high-dimensional space.



However, PCA is fundamentally a global, variance based approach. To complement it, we also examined KNN based methods that estimate the intrinsic dimension of local neighborhoods. On a large portion of the dataset (25%), this yielded an intrinsic dimension

[1] GloVe: Global Vectors for Word Representation

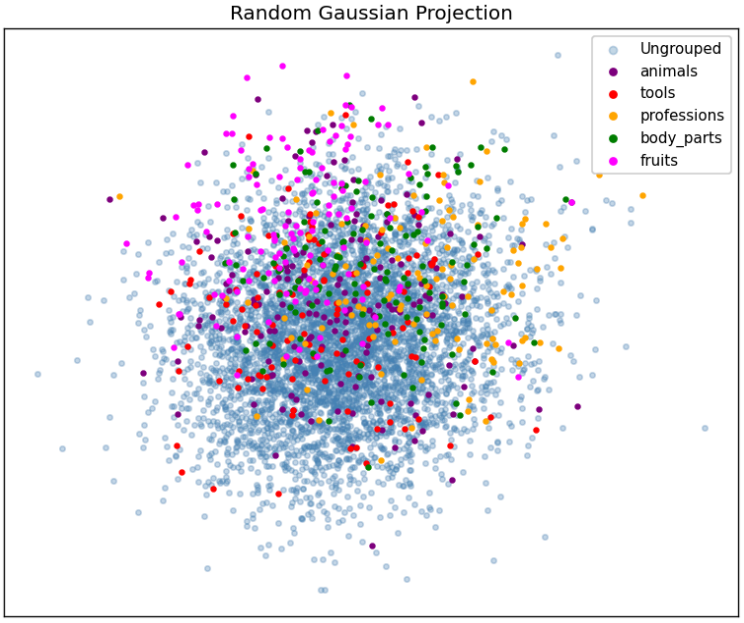
of 56. This indicates that while the global space is very high-dimensional, local neighborhoods, where semantically related words cluster, are far less complex.

To push this further, we examined subgroups of words such as animals, countries, tools, professions, body parts and fruits. Since each subgroup (100–200 words) is smaller than the embedding dimension, PCA would be unstable, so we did not apply it there. Using the KNN based method within these categories, the estimated intrinsic dimension dropped to around 20–25. This is expected, restricting to a semantically focused group reduces the intrinsic complexity, while still preserving the relationships between the words.

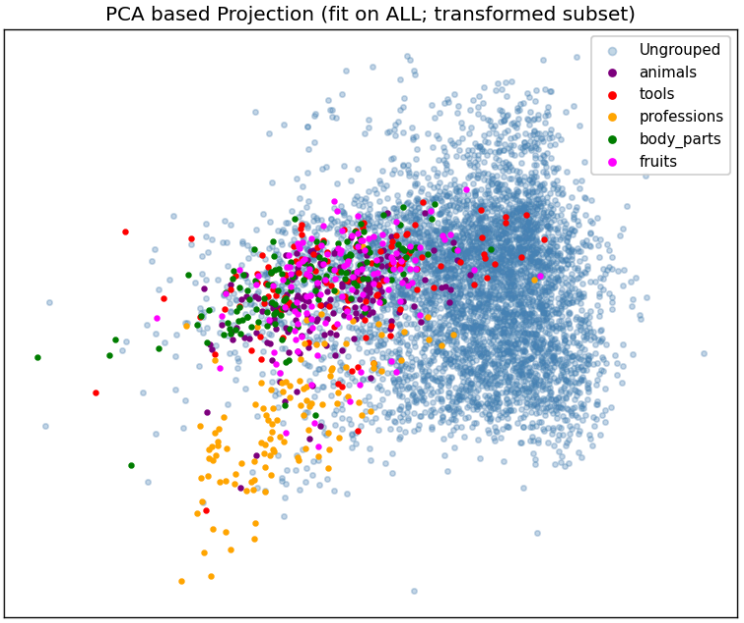
3. Low-Dimensional Projections

To follow up, we projected the embeddings into two dimensions to test whether local neighborhoods form tight groups and to examine the overall geometry of the dataset. I will note that the images attached were generated using ~1% of the data for clarity.

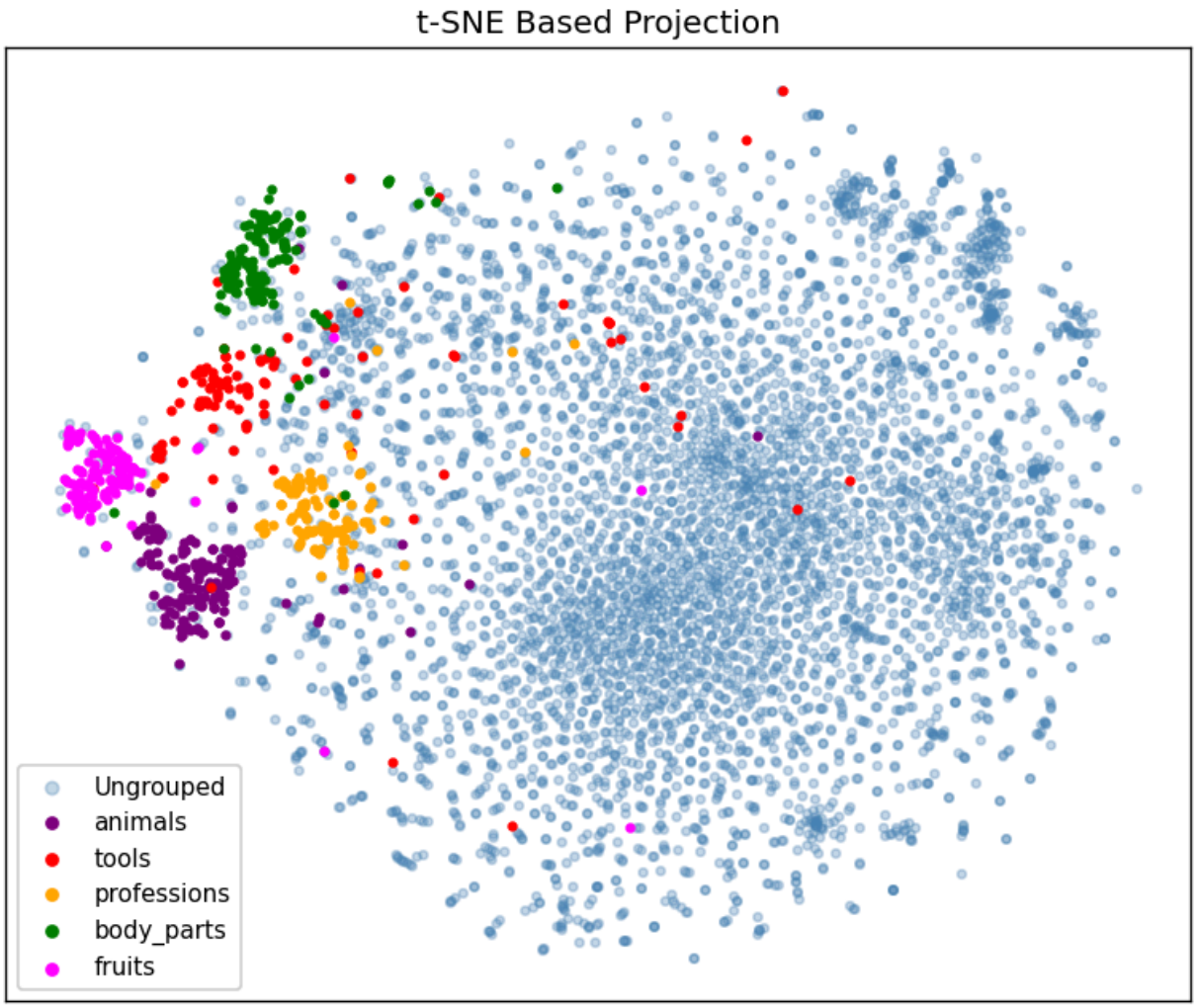
As a baseline, we began with a random Gaussian projection. As expected, the result was a uniform scatter, with no visible grouping or separation. This confirmed that the embeddings do not trivially collapse into two dimensions and indeed has no inherent low-dimensional geometry, and thus more informed projection methods are required.



We then used a PCA based projection. It produced a spread where our semantic control groups (animals, fruits, etc.) were roughly in the same general area, but not sharply distinguished. This reflects the fact PCA optimizes for variance at the global scale and thus our axes are not necessarily aligned with semantic categories.



In contrast, t-SNE provided a much clearer picture. By preserving local neighborhood structure, t-SNE revealed clear clusters for our semantic control groups, which all appeared as tight groups, distinctly separated from one another. Beyond these labeled categories, the remaining “ungrouped” samples also displayed visible clustering behavior, with patches of words forming local neighborhoods rather than dispersing uniformly. This suggests that semantic structure is indeed encoded in the embeddings and becomes apparent once we use a method that emphasizes local relationships.



4. Conclusion

In conclusion, our analysis shows that while GloVe embeddings span a high-dimensional space globally, local neighborhoods are much simpler and form clear semantic clusters. This confirms our initial assumption, and we saw that the structure is most clearly revealed when using methods that emphasize local relationships.

5. AI Code Generation

Most of the code in this project was generated with the assistance of ChatGPT, with only minor adjustments and fixes made on my part.
A record of the conversation can be found in this link[2].

[2] https://chatgpt.com/share/68a05241-eca4-8008-a73d-b0e5c8e8e26c