Exploring the Embedding Space for Enhanced RAG System Performance

A Statistical Approach to Understanding and Improving Retrieval-Augmented Generation

Gaurav Pooniwala

December 5, 2023

1 Introduction

1.1 Objective

The objective of this project is to explore and understand the embedding space of a Retrieval-Augmented Generation (RAG) system to enhance its performance using statistical methods.

1.2 Research Question

How can visualizing and analyzing the embedding space help us understand the RAG algorithm and use this understanding to improve its performance?

1.3 Context and Motivation

As large language models (LLMs) and retrieval-augmented generation (RAG) systems become increasingly prevalent, understanding and enhancing their performance is crucial. This project aims to analyze and understand the embedding space of a RAG system using statistical methods learned in STAT 7010: Modern Data Mining. By visualizing and analyzing the embedding space, we aim to gain insights that can inform future improvements in RAG performance. The main focus of this project is on dimension reduction and clustering of word tokens and sentence embeddings to understand their structure and relationships.

1.4 Background

I am working with a startup called Mongo App that uses RAG to act as a Financial Coach. RAG systems combine the retrieval of relevant documents with the generation of responses using a language model. In our system, we use embeddings of documents in a database to match with embeddings of the query. The process works as follows:

- **Query Embedding:** When a user asks a question, the query is converted into an embedding, which is a dense vector representation of the query.
- **Document Embedding:** All documents in the database are also converted into embeddings.
- **Similarity Search:** The query embedding is compared with the document embeddings using cosine similarity to find the most relevant documents.

- **Response Generation:** The most relevant pieces of data from the database are then used by a language model to generate a coherent and informative response for the user.

Understanding the embedding space is critical to improve the search algorithm in RAG, which in turn enhances the quality and relevance of the coaching provided to users.

2 Data Description

2.1 Data Source

The data used in this analysis is synthetically generated based on typical datasets. Internal datasets were also used but results are not shared due to GDPR and proprietary constraints.

2.2 Data Types

To better understand the embedding space, we start with small datasets and smaller "chunks". A chunk refers to a segment of text, which can range from a single word to a sentence or a paragraph. We can create one embedding per chunk regardless of the size. For this analysis, we are using a progression from words to sentences to paragraph chunks to first understand the embedding space with simple data before moving towards more complex and real-world data. Words and sentences are synthetically generated to have enough similarities and differences so that we can plot and visualize them to get a better understanding of the space.

Our goal is to analyze these generated embeddings. The embeddings are theoretically supposed to contain semantic information about the text they represent.

- **Word Embeddings:** Common words, nouns, verbs, and adjectives.
- **Sentence Embeddings:** Question-answer pairs.
- **Paragraph Embeddings:** Actual chunks from the database.

The paragraph chunks from the database information are not included in this report because of GDPR and proprietary information.

2.3 Data Size

The analysis was run repeatedly in groups of up to 40 words and 16-20 question-answer pairs, totaling about 100 questions.

2.4 Word List

We generated the embeddings for the following set of words using the text-embedding-ada-002 model:

- **Common Words:** the, be, to, of, and, a, in, that, have, I
- **Nouns:** cat, dog, house, car, tree, book, phone, computer, city, ocean
- **Verbs:** run, jump, eat, sleep, write, read, swim, dance, sing, think
- **Adjectives:** happy, sad, big, small, fast, slow, hot, cold, new, old

2.5 Question-Answer Pairs

For sentence embeddings, we used question-answer pairs as a simplification of the RAG process where we want to find data that best corresponds with a user query. The question-answer pairs span various topics such as literature, physics, geopolitics, etc.

3 Methods

We will be using OpenAI's text-embedding-ada-002 model to generate all the embeddings in this report.

The distances in the embedding space are calculated using cosine similarity. As a result, the space is nonlinear and normal statistical methods may not be sufficient to fully explain the space. Nonetheless, we can still plot the data in PCA and t-SNE plots along with clustering algorithms to get a better understanding of the space.

PCA is a linear method that will only represent part of the variance. t-SNE is a nonlinear method that preserves local neighborhood structure but warps overall distances.

3.1 Word Embeddings

We generated embeddings for common words. To understand the structure of the embedding space, we visualized the embeddings using PCA and t-SNE. We also tested clustering methods (K-means and hierarchical clustering) to identify patterns in the embeddings.

3.2 Sentence Embeddings

We used a set of question-answer pairs to generate sentence embeddings. We visualized the embeddings using t-SNE and analyzed cosine distances to evaluate the accuracy of identifying relevant answers. We also performed K-means and hierarchical clustering to group similar sentences.

4 Results

4.1 Word Embedding Analysis

PCA Visualization: The first two components explain only 13% of the variance. Regardless, word embeddings of similar types are generally clustered together. We also performed PCA analysis on the 3rd and 4th components, which can be seen in the appendix.

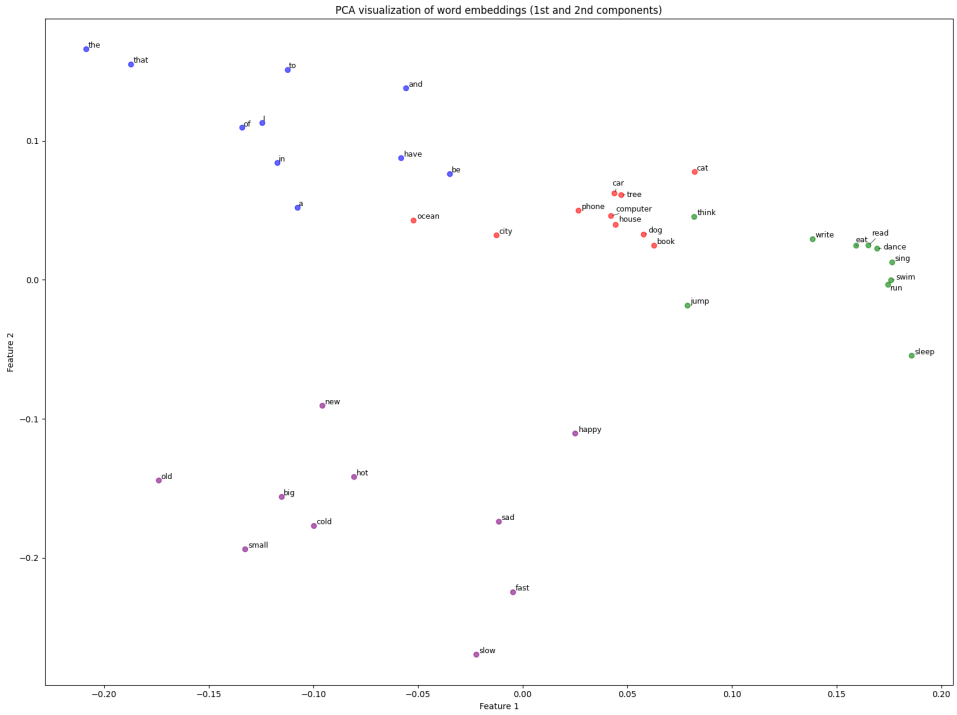

Figure 1: PCA visualization of word embeddings (1st and 2nd components)

t-SNE Visualization: Word embeddings of similar types are generally clustered together. This can be seen in the 2D t-SNE visualizations. We also performed 3D t-SNE visualizations, and while there is clustering, it is much harder to see in the 3D space. The 3D plots are included in the appendix for reference.

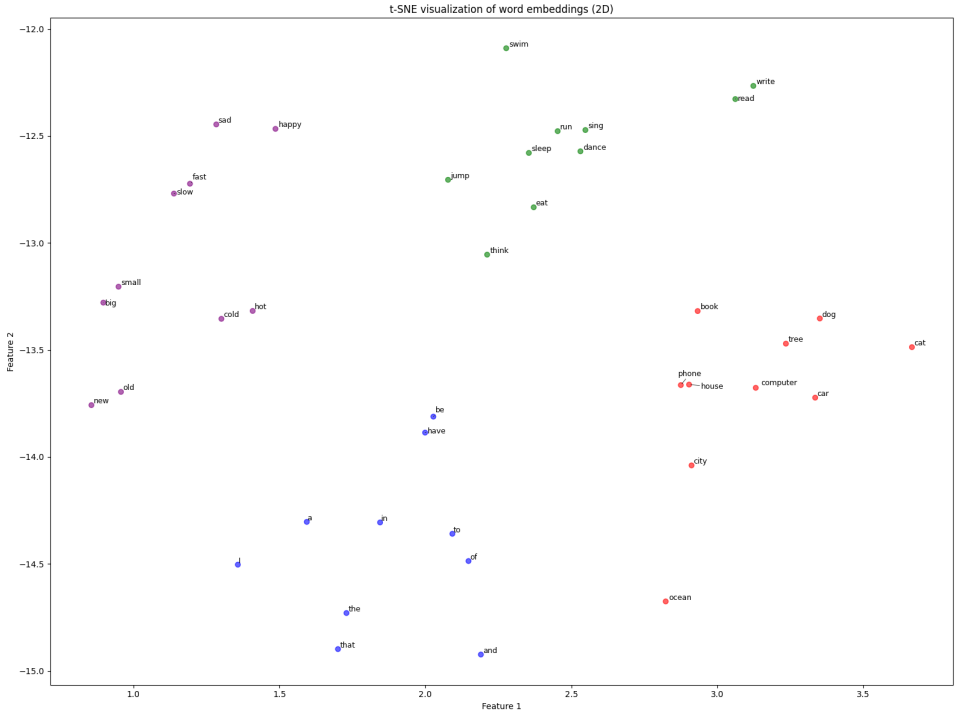

Figure 2: t-SNE visualization of word embeddings (2D)

K-means Clustering: K-means clustering results show some clustering patterns. However,

K-means assumes clusters are spherical and equally sized, which may not be suitable when using cosine distances instead of Euclidean distances.

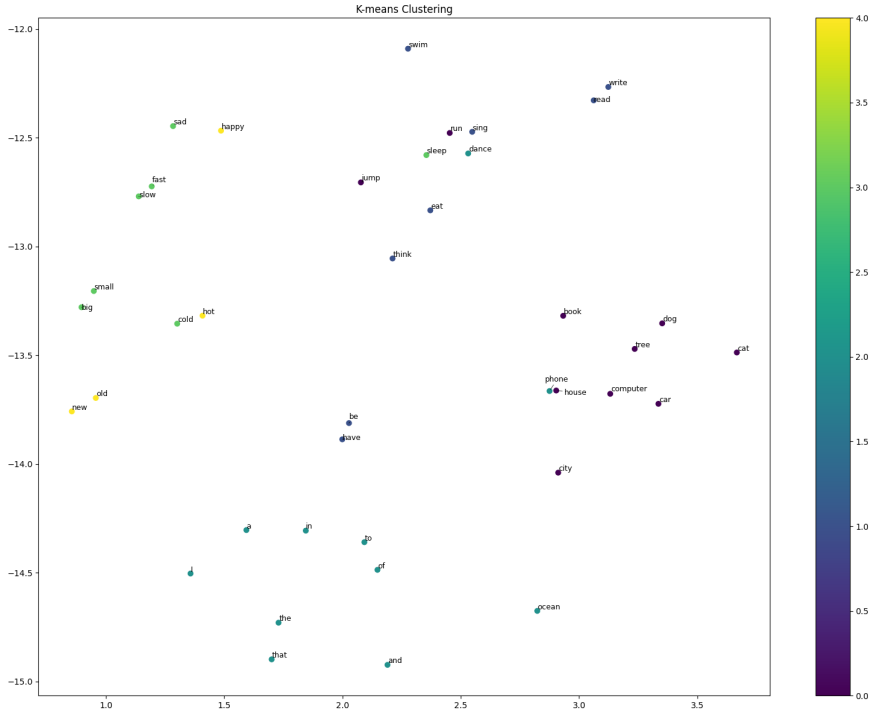

Figure 3: K-means Clustering

Hierarchical Clustering: Hierarchical clustering seems to perform marginally better than K-means.

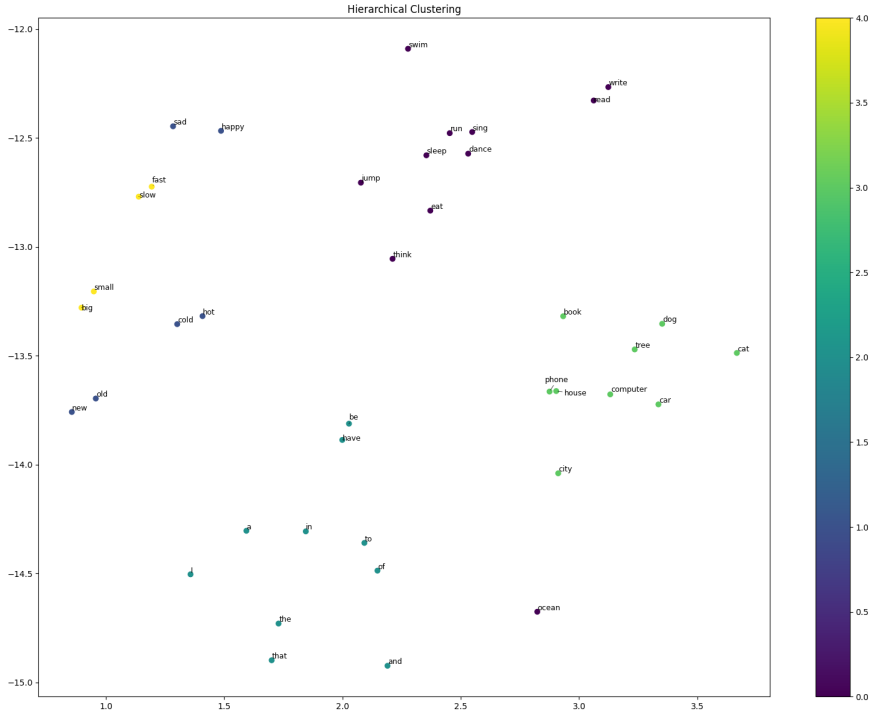

Figure 4: Hierarchical Clustering

Dendrogram: The dendrogram provides a visual representation of the hierarchical clustering.

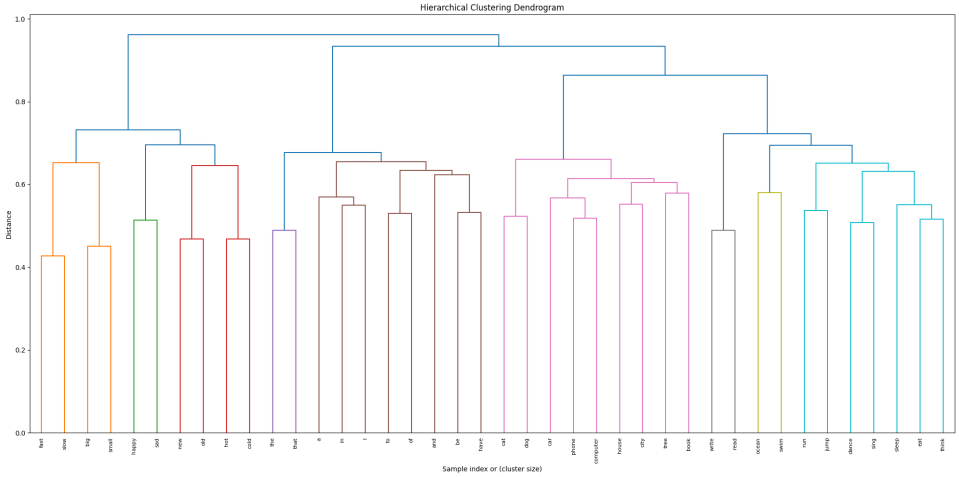

Figure 5: Dendrogram

4.2 Sentence Embedding Analysis

Cosine Similarity: The accuracy of finding the closest answer based on cosine similarity is 100% for our small dataset of 16 questions.

K-means Clustering: K-means clustering accuracy is 94%.

Hierarchical Clustering: Hierarchical clustering accuracy is 88%.

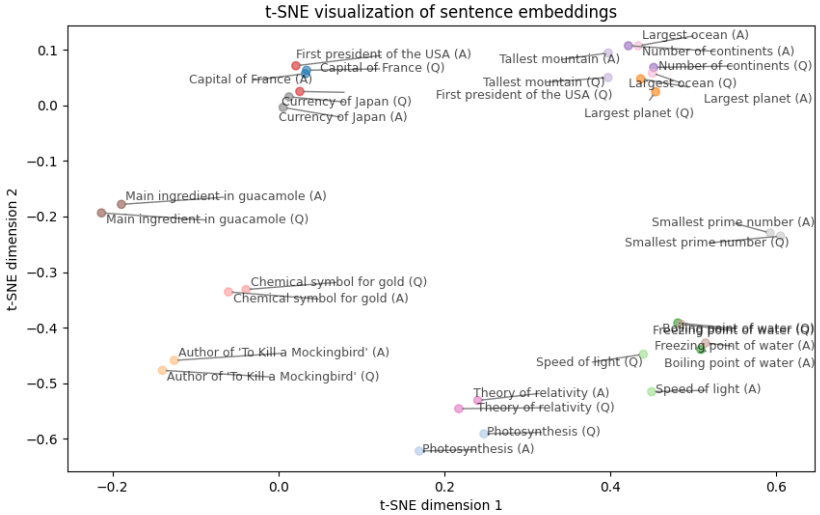

Figure 6: t-SNE visualization of sentence embeddings

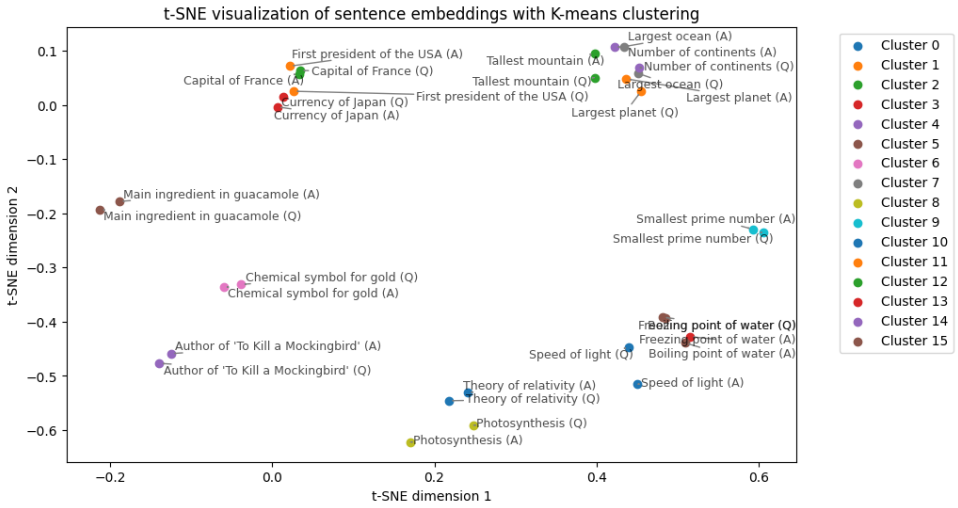

Figure 7: K-means Clustering of Sentence Embeddings

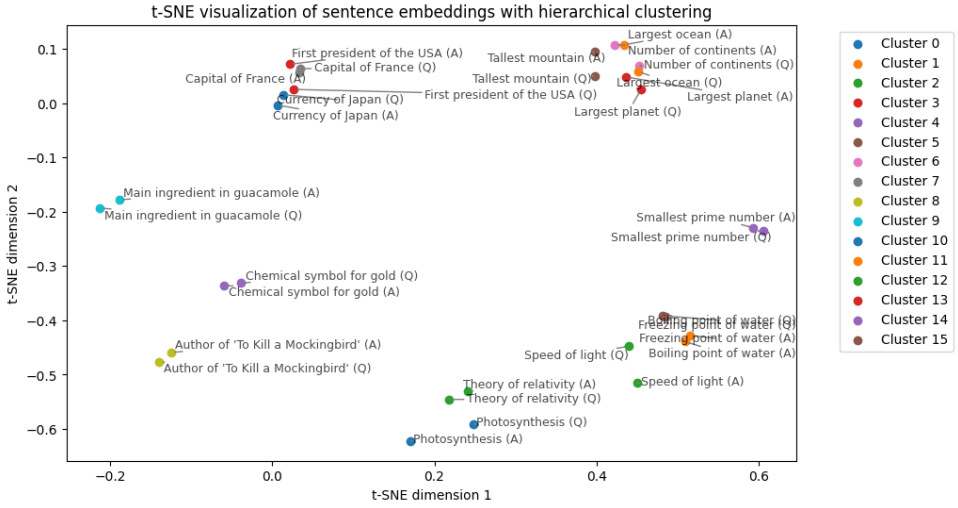

Figure 8: Hierarchical Clustering of Sentence Embeddings

5 Interpretations

5.1 Embedding Space

The embedding space prioritizes grouping words and sentences of similar types together, such as verbs, nouns, or topics like geography and physics. Opposite sentiments, emotions, or even opposite words tend to be very close to each other in the embedding space.

5.2 Methods

PCA, being a linear technique and the first 2 dimensions only accounting for 13% of the variance, is not ideal but still shows promising results. t-SNE works well to visualize the data as a whole on the macro level, but individual distances are too warped to be interpreted directly. The assumptions for both K-means and hierarchical clustering do not work very well for cosine distances. While K-means failed at the word level, hierarchical clustering partially failed at the sentence level.

6 Limitations

- **Synthetic Data:** The data used in this analysis is synthetically generated, which makes it easier to identify the best question-answer pair. Real-world data may not have direct answers or may have multiple candidates, making the analysis more complex.
- **Ground Truth:** The ground truth for real-world data may be unknown, requiring human validation to ensure the accuracy of the analysis.
- **Qualitative Testing:** Current testing for larger datasets is qualitative, meaning it relies on subjective judgment. Quantitative analysis is needed for real-world applications to provide objective and measurable results.
- **Scalability:** The high accuracy achieved with simple question-answer pairs may not extend well to larger datasets. As the number of candidate answers increases, the probability of finding the correct answer decreases. Additionally, multiple correct or relevant answers may exist, and qualitative analysis is not feasible for large datasets.

7 Potential Next Steps

- **Extensions:**
  - Use an LLM to automatically analyze the relevance of selected answers. This can help in scaling the analysis to larger datasets and provide more accurate results.
  - Test various strategies such as different embeddings and chunk sizes. Experimenting with different models and chunk sizes can help identify the most effective approach for improving RAG performance.
- **Improvements:**
  - Develop methods for quantitative analysis to enhance the current qualitative approach. Quantitative analysis can provide objective and measurable results, making it easier to evaluate the effectiveness of different strategies. One option could be to use multiple independent emebddings and use consensus to determine the best answer.
  - Implement more advanced clustering algorithms that are better suited for cosine distances. This can help improve the accuracy of clustering results and provide more meaningful insights into the structure of the embedding space.

8 Conclusion

8.1 Summary

We explored the embedding space of a RAG system using statistical methods. Visualizing and analyzing the embedding space helped us understand the RAG algorithm and identify ways to improve its performance. By using PCA and t-SNE for dimensionality reduction and clustering methods like K-means and hierarchical clustering, we were able to gain insights into the structure and relationships within the embedding space.

8.2 Final Thoughts

The embedding space is non-linear, making classical techniques less effective. Future work should focus on quantitative analysis and real-world data applications. By addressing the limitations of synthetic data and qualitative testing, we can develop more robust methods for analyzing and improving RAG systems.

8.3 Future Work

This project integrates statistical learning techniques with practical applications in deep learning and natural language processing. By understanding the embedding space, we aim to inform future improvements in RAG performance. The findings will contribute to the optimization of RAG systems, bridging the gap between statistical methods and deep learning practices. Future work should focus on:

- Applying the analysis to real-world data to validate the findings and ensure their applicability in practical scenarios.
- Developing more advanced methods for quantitative analysis to provide objective and measurable results.
- Exploring different models and chunk sizes to identify the most effective approach for improving RAG performance.
- Implementing more advanced clustering algorithms that are better suited for cosine distances.

9 Appendix

9.1 3D t-SNE Visualizations

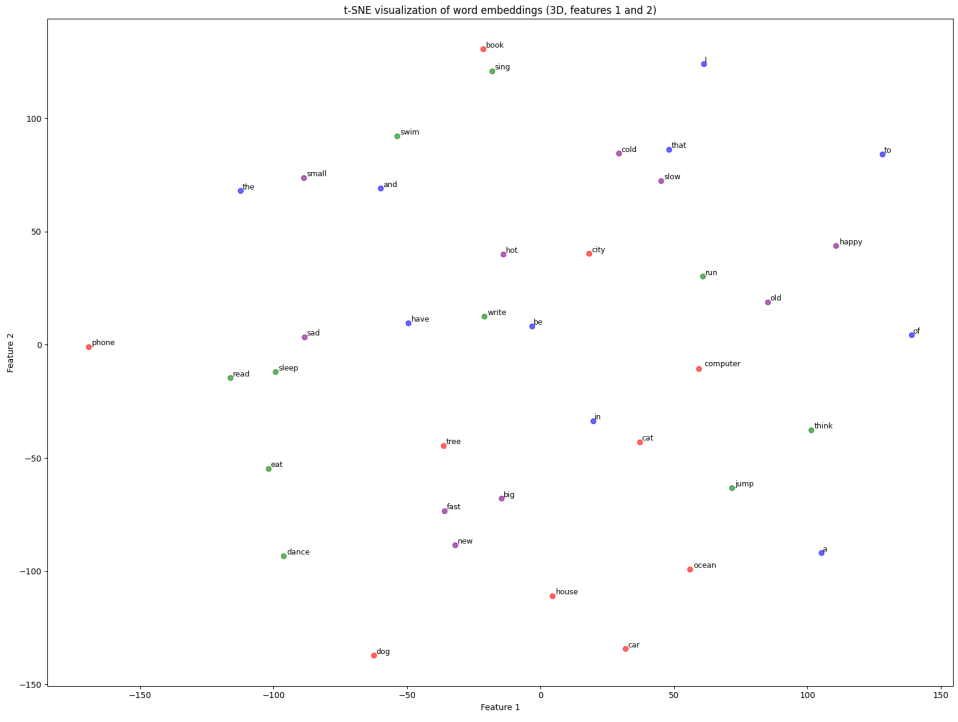

Figure 9: t-SNE visualization of word embeddings (3D, features 1 and 2)

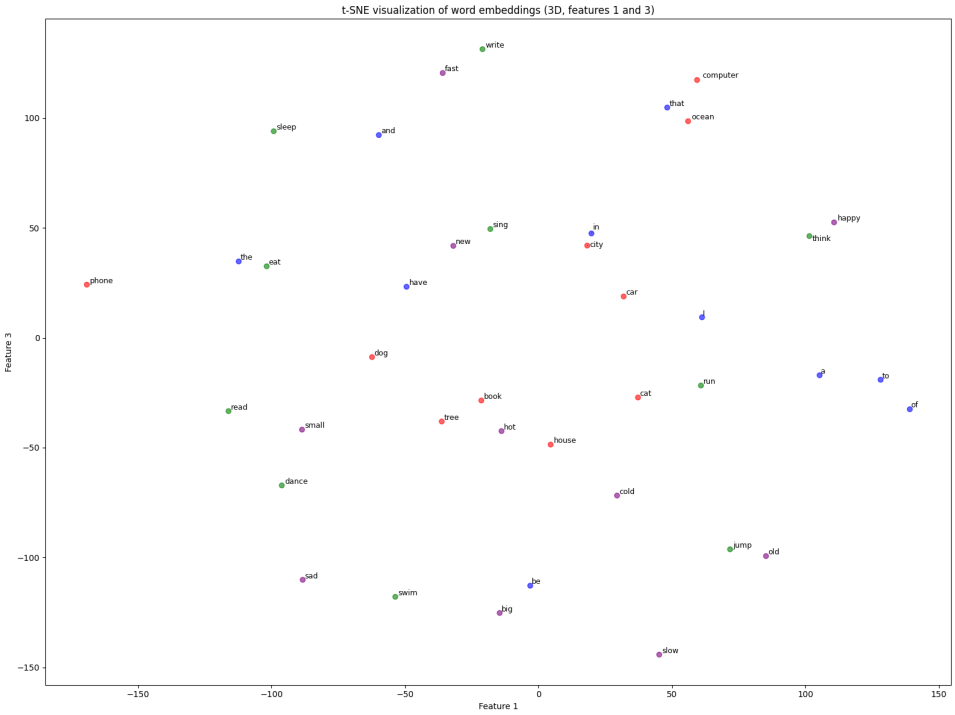

Figure 10: t-SNE visualization of word embeddings (3D, features 1 and 3)

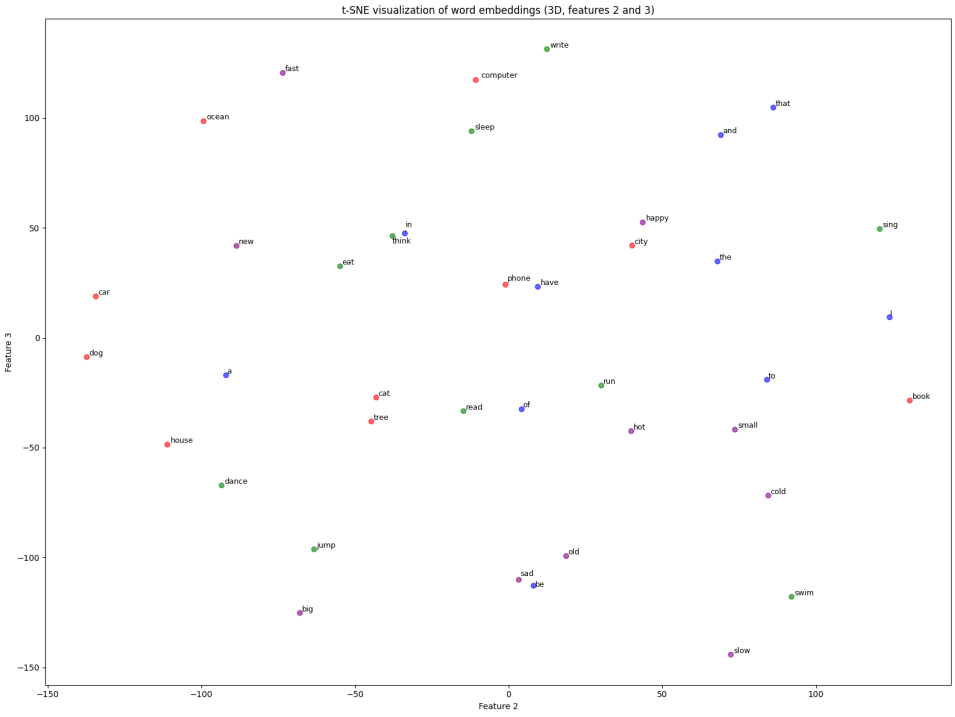

Figure 11: t-SNE visualization of word embeddings (3D, features 2 and 3)

9.2 PCA Visualization (3rd and 4th components)

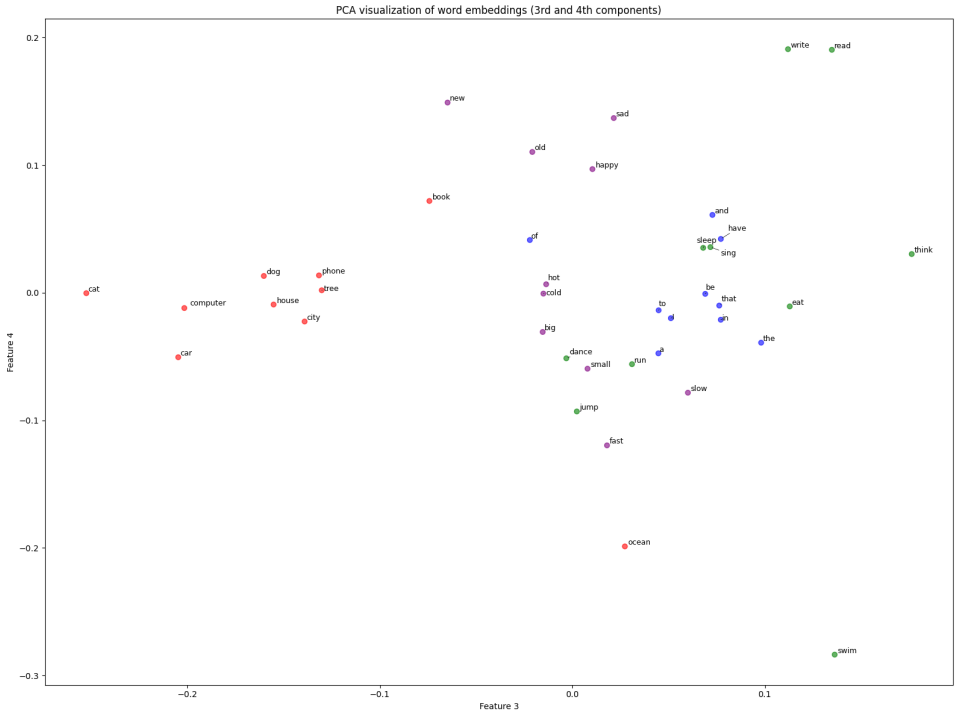

Figure 12: PCA visualization of word embeddings (3rd and 4th components)