VSM Evaluation

Pablo de Vicente Abad 22/04/2025

Introduction

In this report, I'll go over the performance of different word embedding techniques—Word2Vec, GloVe, and FastText—on text classification. I'll compare how fine-tuning affects Word2Vec, testing both a pre-trained model and a finetuned version. To make the comparison more meaningful, I'll look at datasets with three and four classes, each with 15 documents per class.

For GloVe, I'll explore why it performs well and whether fine-tuning adds any value. FastText will also be tested, with a focus on how its subword information impacts classification.

To support the analysis, I'll also create a new demo dataset with three classes and 50 documents per class. Additionally, I'll present a Word2Vec demo using a three-class dataset to highlight its behavior in practice.

15/03/2025

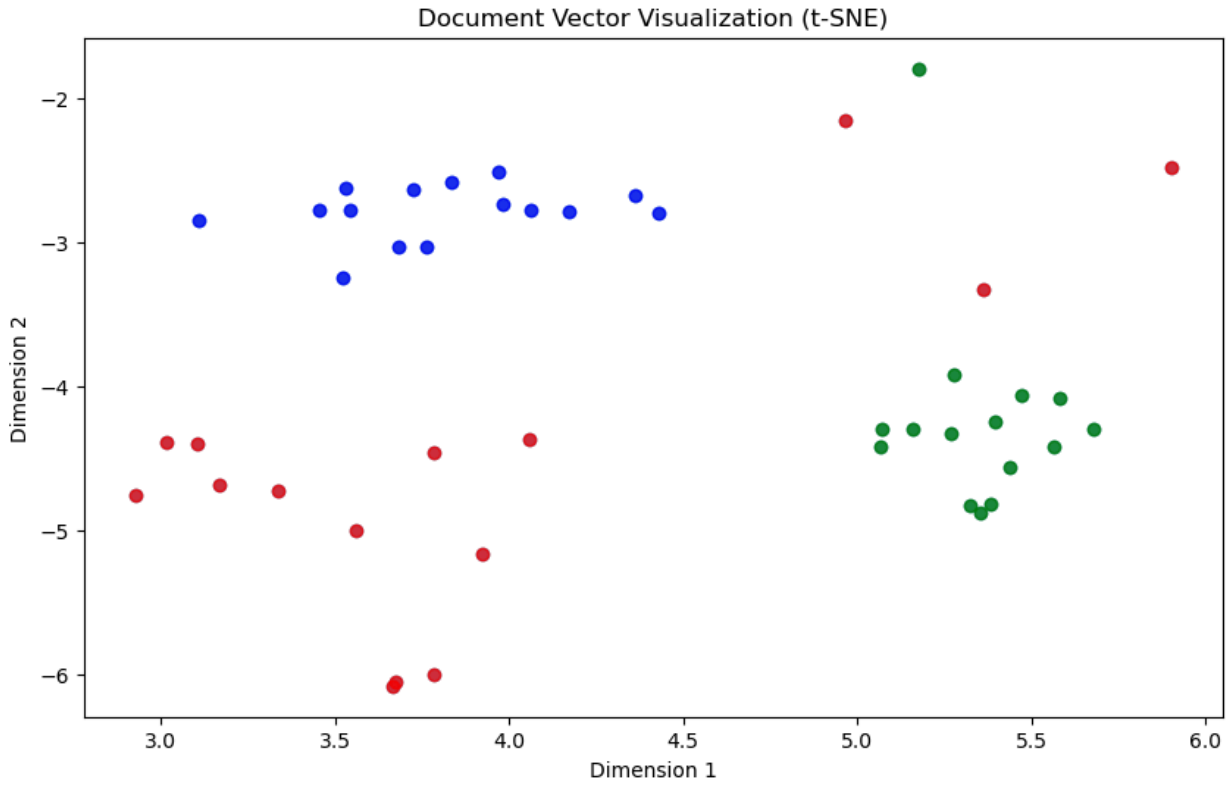
Outline

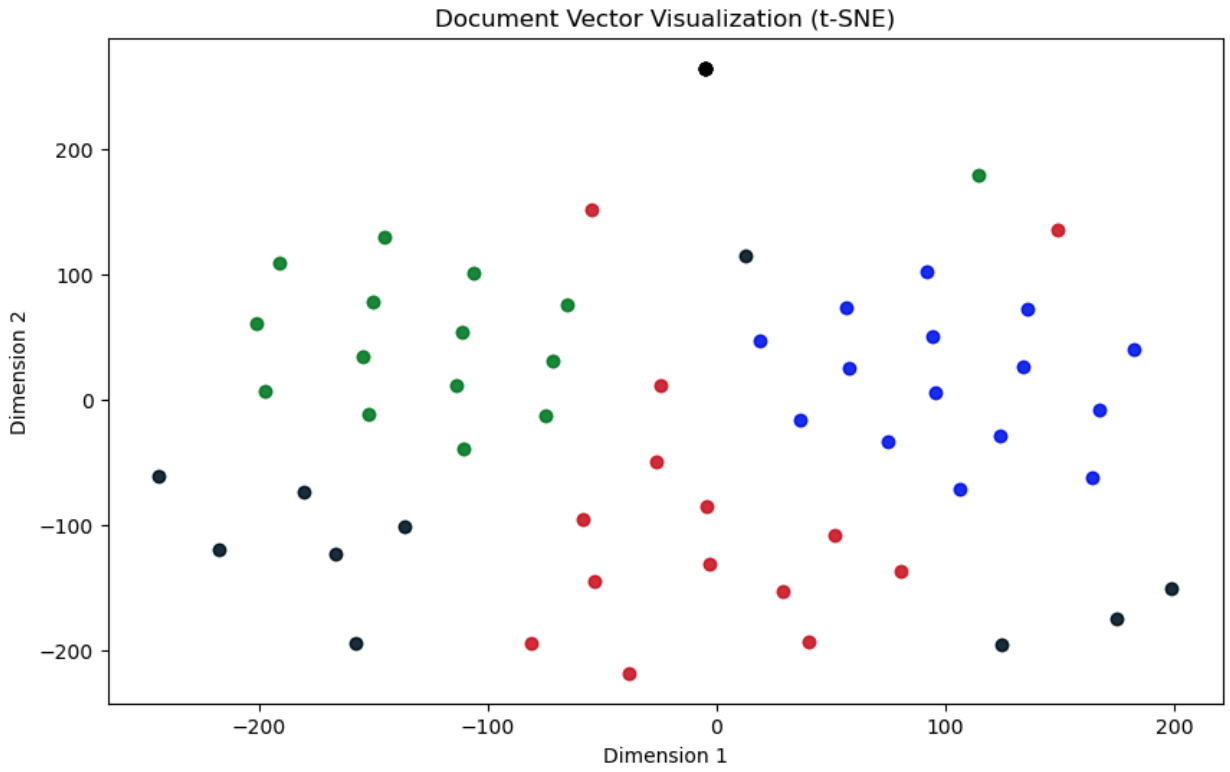
1. **Word2Vec**
    - Comparing pre-trained vs. fine-tuned versions
    - Performance with 3 vs. 4 classes (15 docs per class)
2. **GloVe**
    - Why it performs well
    - Is fine-tuning worth it?
3. **FastText**
    - How it handles subword information
    - Performance examples
4. **New Dataset for Testing**
    - 3-class dataset with 50 documents per class
    - Insights from using a larger dataset

Word2Vec

Examples of pre-trained Word2Vec with three and four classes. The representation for the fourth class appears distorted or inconsistent.

Pretrained model used: word2vec-google-news-300.bin



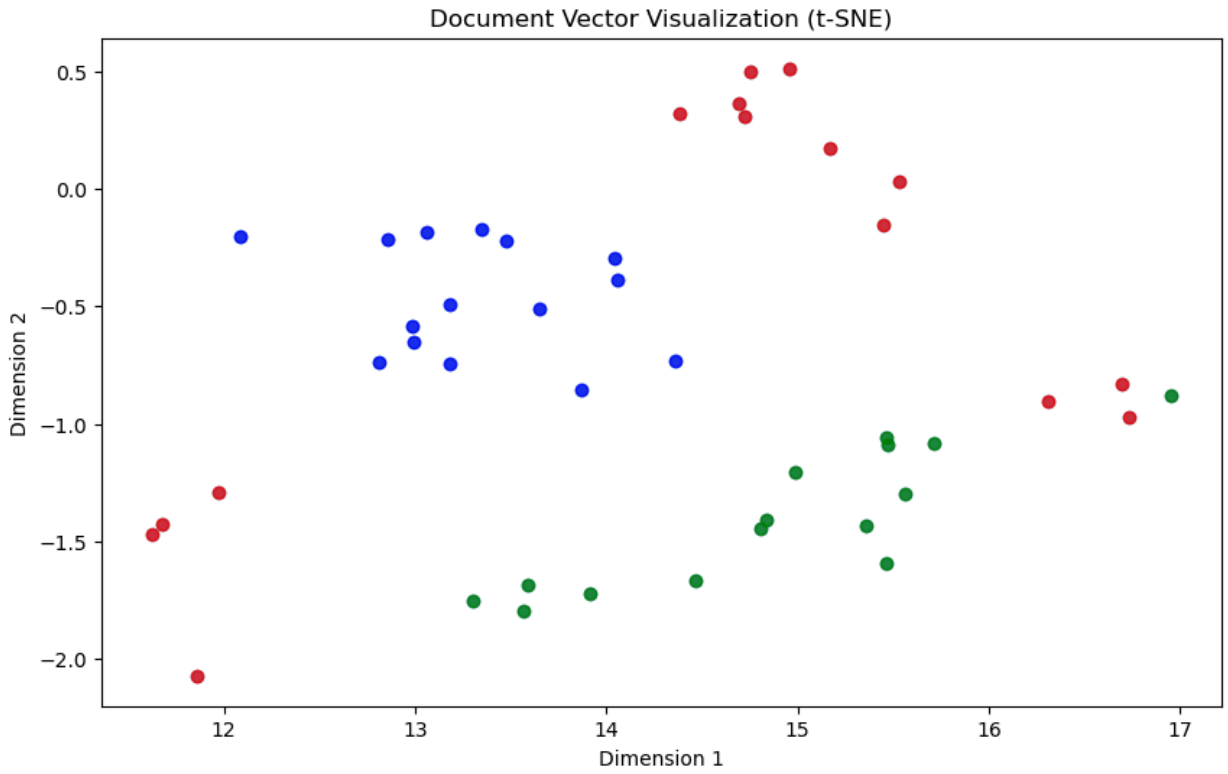


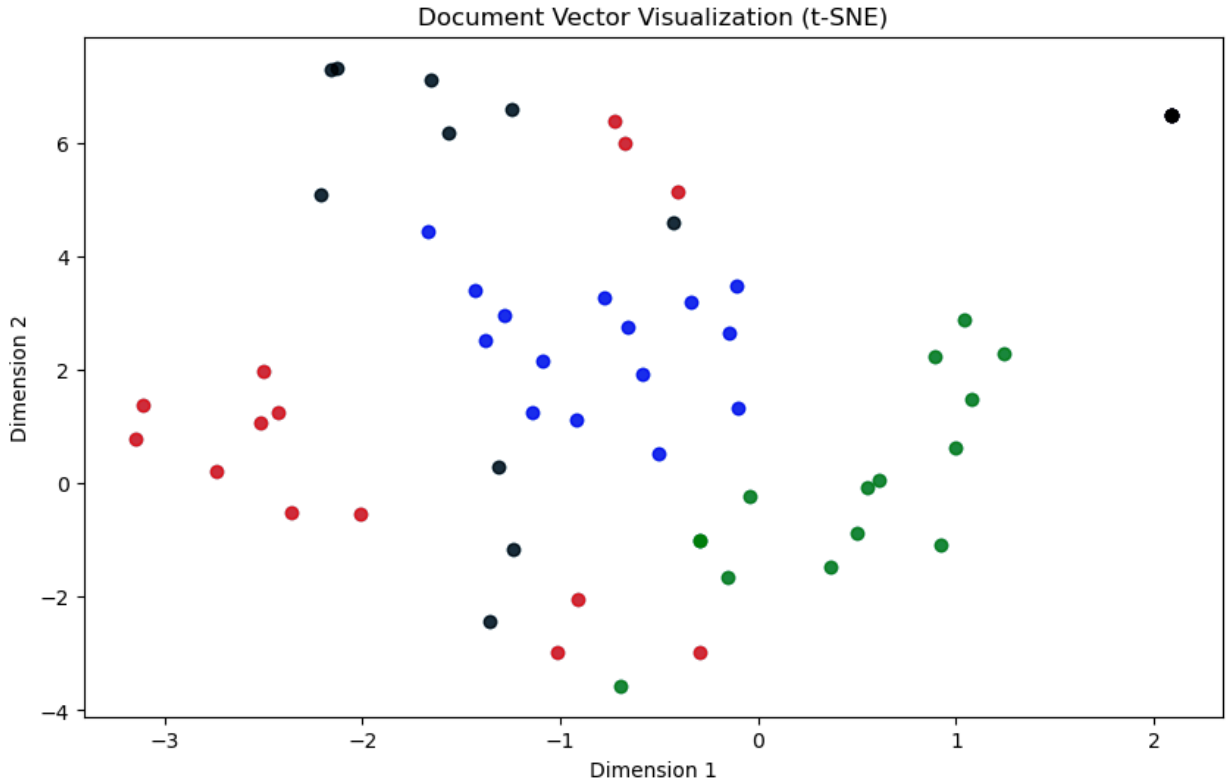
Word2Vec Finetuned

Examples of the fine-tuned Word2Vec model. A key concern is that fine-tuning did not introduce any new vocabulary words. Literature also suggests that continuing pretraining on Word2Vec often provides little to no benefit.

Several sources suggest that fine-tuning Word2Vec embeddings may offer limited benefits, particularly when working with small datasets. Fine-tuning on small datasets can increase the risk of overfitting, leading to poor performance when the model encounters new data. Additionally, fine-tuning may not introduce new vocabulary words, limiting its effectiveness. Furthermore, it has been observed that in some cases, fine-tuning embeddings can hurt generalization performance, especially on small training datasets.

[1] https://telnyx.com/resources/embedding-vs-fine-tuning
[2] https://stackoverflow.com/questions/56166089/wor2vec-fine-tuning
[3] https://www.reddit.com/r/MachineLearning/comments/84r7ws/d_to_fine_tune_word_embeddings_or_not/


Document Vector Visualization (t-SNE)
Dimension 2
Dimension 1
0.5
0.0
−0.5
−1.0
−1.5
−2.0
12
13
14
15
16
17

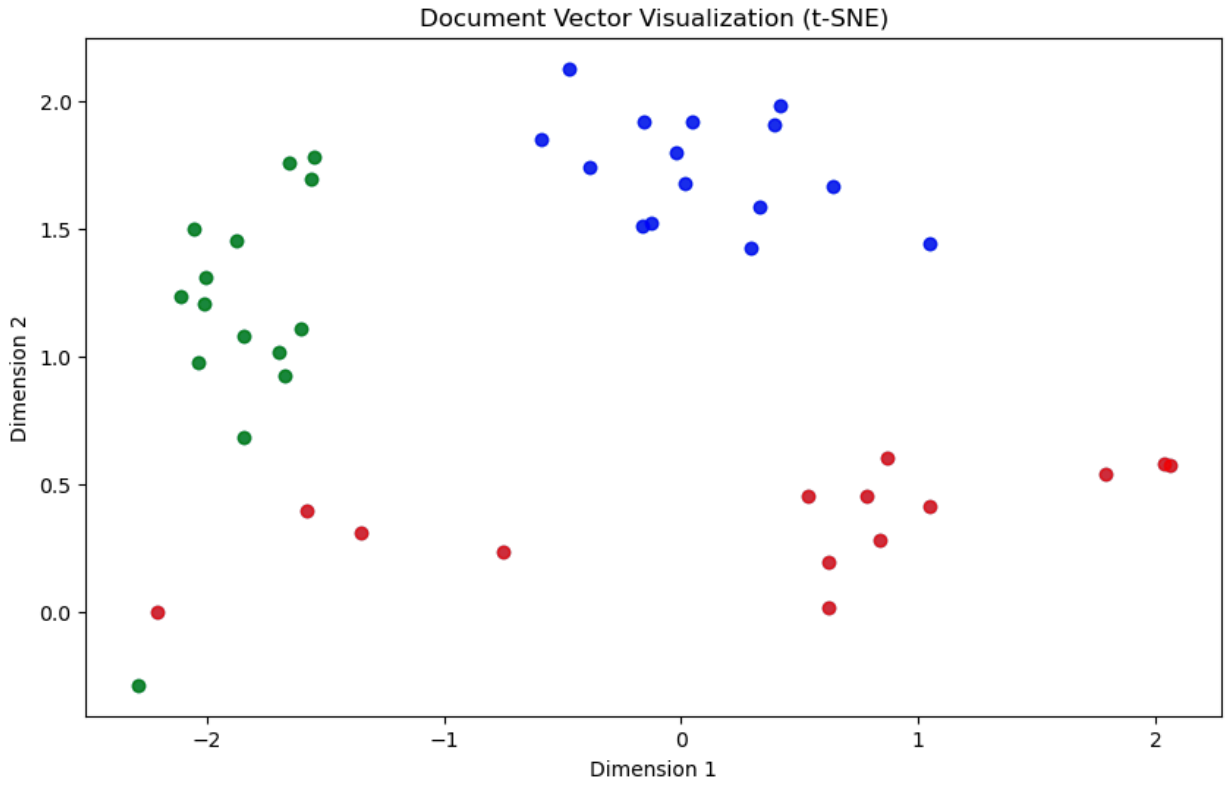

Document Vector Visualization (t-SNE)
Dimension 2
Dimension 1
6
4
2
0
−2
−4
−3
−2
−1
0
1
2

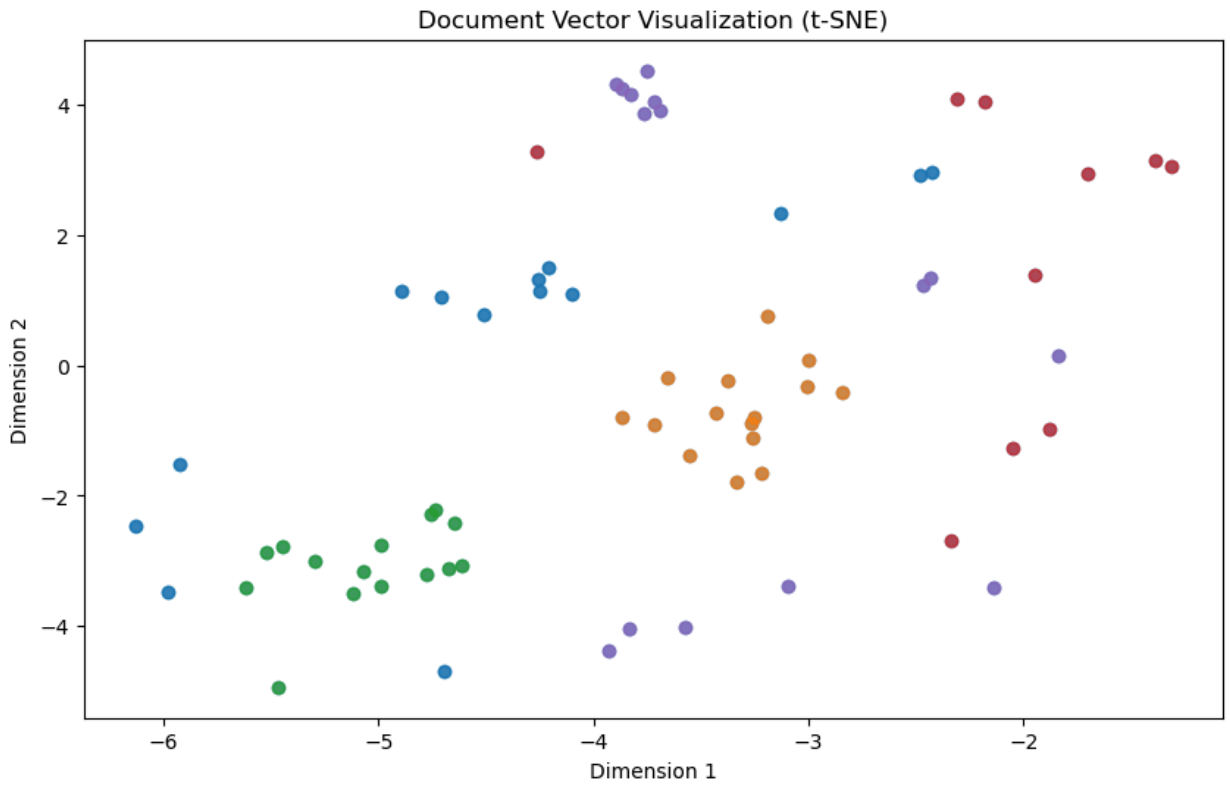
Glove

GloVe has several advantages over Word2Vec, mainly due to its **global co-occurrence approach** rather than relying on local context windows. This leads to:

- **Better semantic relationships**: Captures word associations across the entire corpus, improving performance on analogy tasks.
- **More efficient training**: Uses a word co-occurrence matrix, extracting more information with fewer data.
- **Greater stability**: Produces more consistent embeddings, whereas Word2Vec can be sensitive to training data.

While Word2Vec is useful for dynamic updates, GloVe generally provides **richer, more reliable word representations** for NLP tasks.

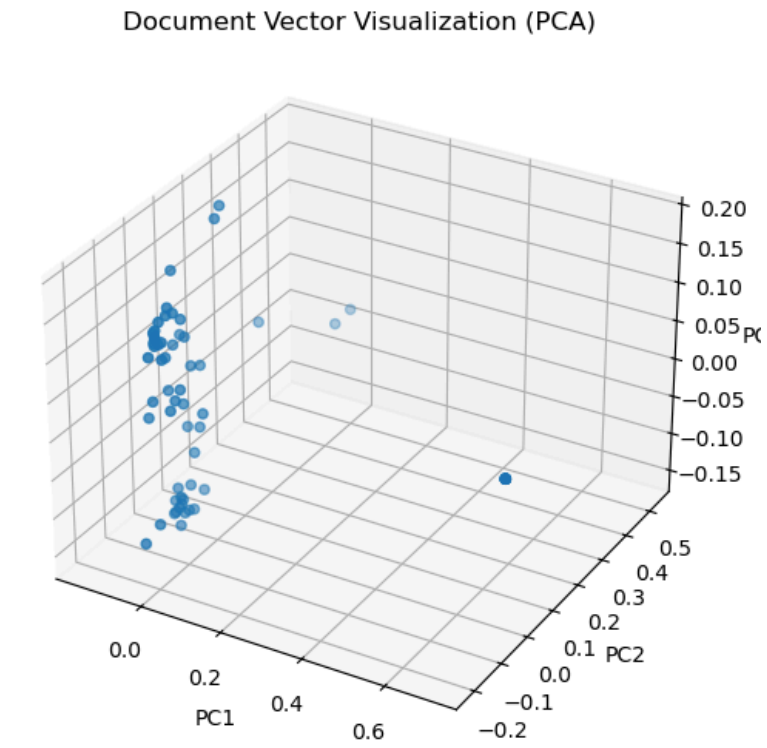




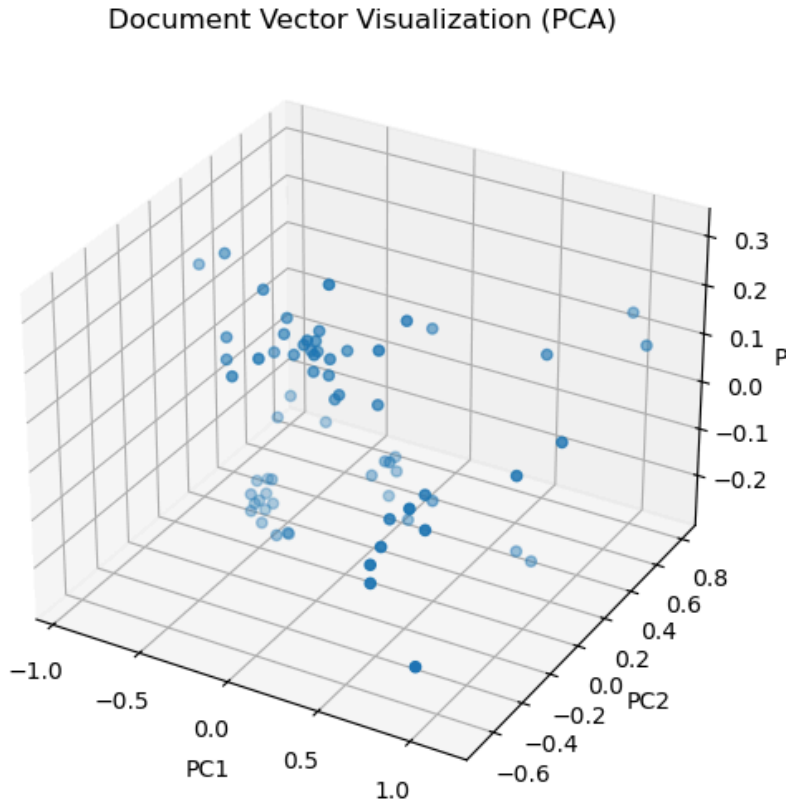
How much to trust this representations?

Principal Component Analysis (PCA) is a dimensionality reduction technique used to identify patterns in data, highlighting the most important features that explain the variance. By transforming data into a new set of axes (principal components), PCA helps to reduce the complexity while retaining as much variability as possible.

Additionally, PCA assumes linearity, which means it may not capture complex non-linear patterns in the data. Therefore, combining PCA with other methods, like **t-SNE or UMAP**, might provide a more comprehensive understanding of your data.



word2vec - 4 classes (weird example above)



glove - 5 classes

What evaluation methods can be used, and how reliable are they in representing the data?

generate vsm

test on glove

What about the search engine? Encode sentences and create a “fictional document embedding”?
Sentence Embedding? S-Bert → limitations?