Visualizing Word Embeddings: Semantic Relationships in TensorFlow's Embedding Projector

Robert McCoy
Indiana Wesleyan University
Model Development (AIML-501-01A)
Dr. Lora Lounge
June 17, 2025

Visualizing Word Embeddings with TensorFlow's Embedding Projector

Introduction

Embeddings simplify complex data by visually mapping them into multi-dimensional spaces, enabling clear identification of patterns and relationships. These vector-based representations encode semantic meaning by capturing the statistical co-occurrence of words within a corpus. By positioning similar concepts close to one another, embeddings facilitate efficient learning and generalization in machine learning models.

Embeddings are widely used in diverse applications including text classification, image recognition, recommendation systems, and natural language processing (NLP) tasks such as machine translation and sentiment analysis (Vu,2017). They offer a scalable and intuitive method to represent symbolic data in a format that algorithms can process efficiently.

In this assignment, I utilized the TensorFlow Embedding Projector (https://projector.tensorflow.org/), a robust visualization tool for exploring static word embeddings. By entering specific words into the interface, the tool allows real-time exploration of semantic proximity in high-dimensional space. This report includes structured documentation of these findings with visual examples, detailed observations, and a synthesized reflection.

Visual Exploration and Observations

1. Dense Semantic Clusters — Example: "Jesus"

Searching for the term "Jesus" resulted in a rich semantic neighborhood with dense clustering. Nearby vectors included "Christ," "apostles," "resurrection," and "salvation," all tightly associated within theological contexts. The high frequency and contextual consistency of the term in religious texts enabled the model to develop a robust and specific vector space representation.

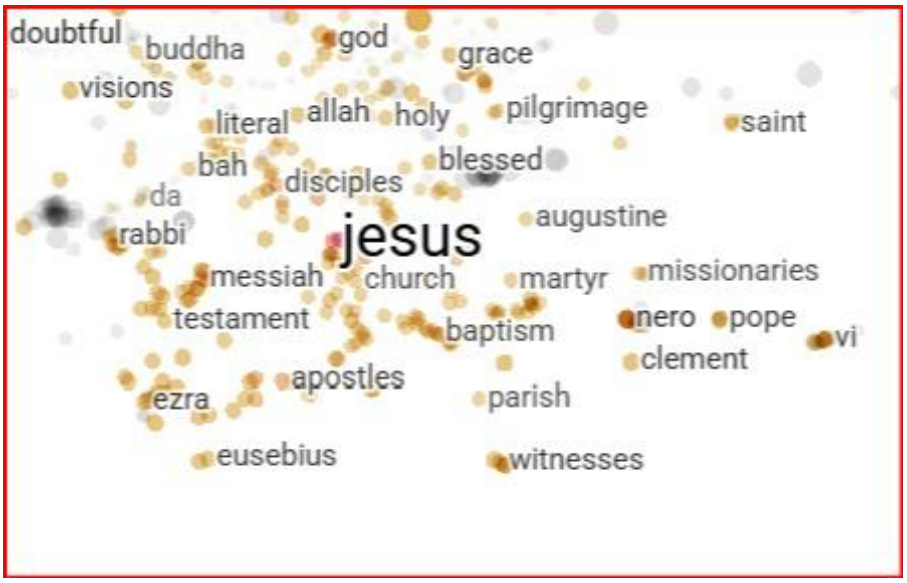

Figure 1. Tensorflow Embedding Projector Custom View for the Term "Jesus"

2. **Diffuse Contextual Space — Example: “President”**

   The term “President” exhibited a more scattered and ambiguous embedding space. While loosely connected to terms such as “leader,” “government,” and “executive,” there was no tight cluster of meaning. This reflects the polysemous nature of the term across political, corporate, and academic domains. The lower contextual consistency reduced the embedding's clarity.

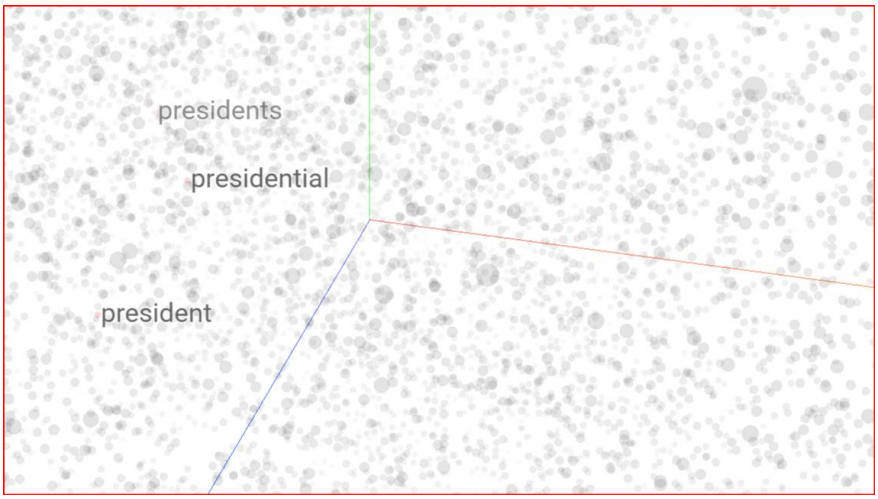

Figure 2. Tensorflow Embedding Projector Custom View for the Term “President”

3. Dimensionality Reduction and Outlier Behavior

Word embeddings originate in 200+ dimensions, which must be compressed for visualization via techniques like PCA or t-SNE. These reductions preserve local clusters but can distort absolute distance or global structure (Van der Maaten & Hinton, 2008). An example of this effect is observed when semantically unrelated words appear close due to projection artifacts.

Advanced Discussion and Technical Insights

Why Are Some Words Absent?

Words may be excluded from the embedding set due to low frequency, irregular formatting, or being beyond the top N most common tokens used during training. Preprocessing steps often filter out stopwords, rare symbols, or domain-specific jargon unless explicitly retained (Mikolov et al., 2013).

Static vs. Contextual Embeddings

The TensorFlow Embedding Projector displays static embeddings, where each word has a single, unchanging vector. This contrasts with contextual embeddings (e.g., from BERT or GPT models), where word meaning varies by sentence context. Thus, polysemous words like “bank” are ambiguous in static models, limiting their utility for certain applications (Liang, 2019).

Impact of Dimensionality Reduction

While PCA emphasizes global structure, t-SNE is better for preserving local neighbor relationships. However, both methods involve information loss, which can misrepresent true relationships in high-dimensional space. Users should interpret 2D or 3D projections cautiously and not assume linear meaning from proximity alone.

Embeddings in Real-World Applications and LLM Design

Word embeddings are fundamental to many of today’s most powerful AI systems. Major technology companies utilize embeddings to personalize user experiences and improve service accuracy. For example, Google employs embeddings in search ranking and

query understanding, while Netflix uses them to power content recommendation systems based on user behavior patterns and viewing history (Vu et al., 2017). In each case, embeddings enable systems to relate new input to known patterns in a mathematically meaningful way.

These embeddings also underpin large language models (LLMs) like GPT, Gemini, Claude, and Grok. However, it is crucial to recognize that embeddings reflect the data and contexts they are trained on. If an LLM is deployed without a clear understanding of its training base, users may unknowingly receive skewed or incomplete outputs—particularly when querying specialized or nuanced topics.

This underscores the need for transparent model documentation, and user education about model limitations and contextual scope. When embedding-based systems are designed or applied in environments such as healthcare, law, or engineering, bias and knowledge gaps can significantly impact decision quality if not mitigated by proper oversight or integration with domain-specific data (Bhattacharya et al., 2024).

Visualizing Word Embeddings: Semantic Relationships

Reflection

Exploring word embeddings visually has reinforced my understanding of how AI models represent and interpret language. The TensorFlow Embedding Projector made tangible, the often-abstract concept of vector semantics. Words like “Jesus” demonstrated how contextually rich terms form dense clusters, revealing strong thematic coherence. In contrast, words like “President” illustrated the challenges of polysemy, as they sprawl across multiple domains with inconsistent neighboring vectors.

This exercise also highlighted the limitations of static embeddings. Without context, the model cannot differentiate between meanings of ambiguous words, and dimensionality reduction adds another layer of distortion. These constraints make clear why modern AI systems increasingly rely on contextual embeddings that adapt based on usage.

Overall, this project provided an insightful, hands-on understanding of how semantic relationships are modeled in NLP systems. It also underlined the importance of data diversity and interpretability in embedding-based applications. Embedding visualization not only demystifies internal model behavior but can also be a diagnostic tool in AI system design.

References

Bhattacharya, A., Stumpf, S., De Croon, R., & Verbert, K. (2024). *Explanatory Debiasing: Involving Domain Experts in the Data Generation Process to Mitigate Representation Bias in AI Systems*. https://doi.org/10.48550/arxiv.2501.01441

Liang, J. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient estimation of word representations in vector space*. arXiv preprint arXiv:1301.3781.

Van der Maaten, L., & Hinton, G. (2008). *Visualizing data using t-SNE*. Journal of Machine Learning Research, 9, 2579–2605.

Vu, T., Nguyen, D. Q., Johnson, M., Song, D., Song, D., & Willis, A. (2017). *Search personalization with embeddings* (pp. 598–604). Springer, Cham. https://doi.org/10.1007/978-3-319-56608-5_54