

Interpreting LLMs with Geometry

Created by: Lauren Alvarez Ph.D.
Senior Applied AI Researcher
TELUS Digital Fuel iX

How can we understand GenAI?

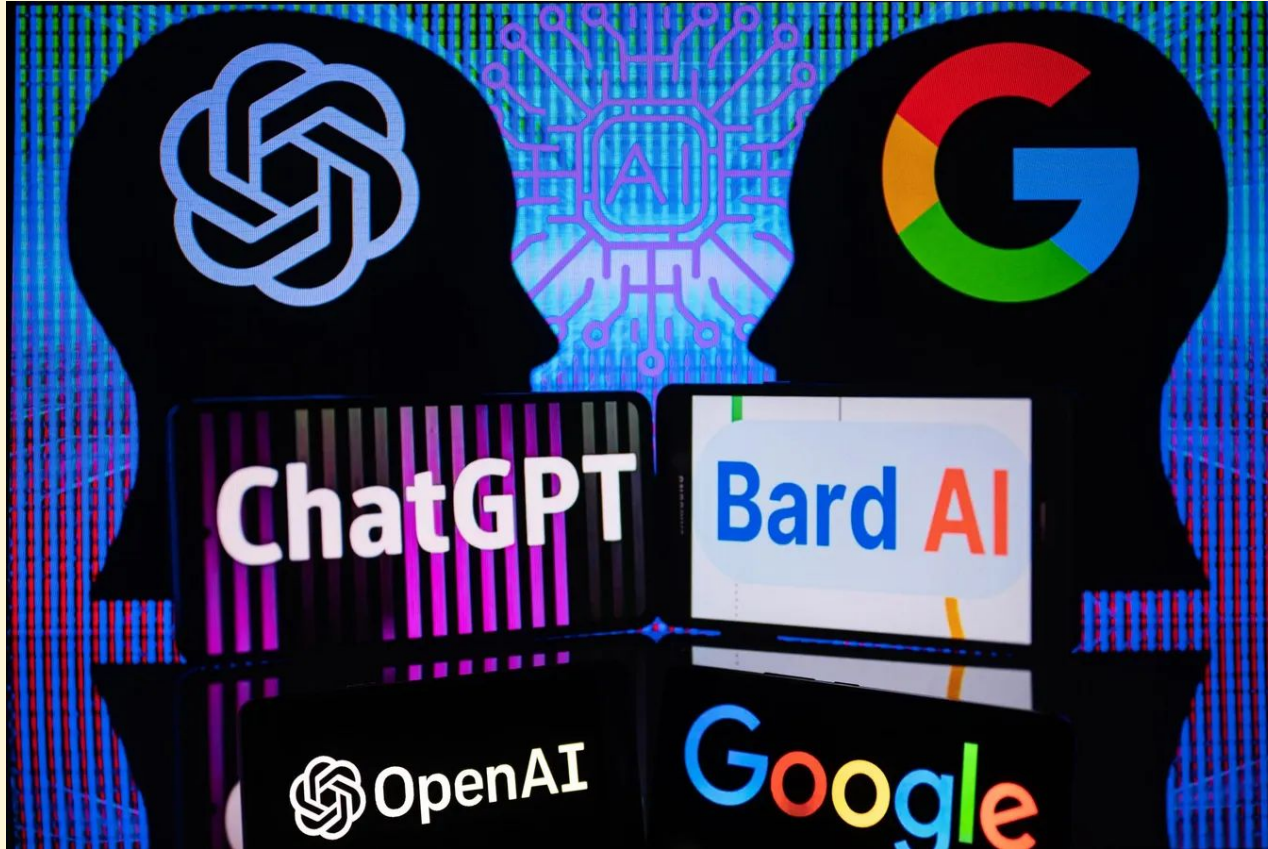


Image Source: [Generative AI Tools Like ChatGPT And Bard Heralding Generational Shift In Job Roles. Adapt Or Risk Obsolescence](#)

Lecture Sections

01

Present key terms

02

Geometry!

03

Summary

04

Advanced Content





01

Key Terms



Tokenization

represent discrete units of text (i.e., letters, subwords, words)

GPT-4o & GPT-4o mini

GPT-3.5 & GPT-4

GPT-3 (Legacy)

This is an example of tokenization. Each word is one token except tokenization. Tokenization is an example of having subword tokens.

Clear

Show example

Tokenization

represent discrete units of text (i.e., letters, subwords, words)

Tokens	Characters
28	132

This is an example of tokenization. Each word is one token except tokenization. Tokenization is an example of having subword tokens.

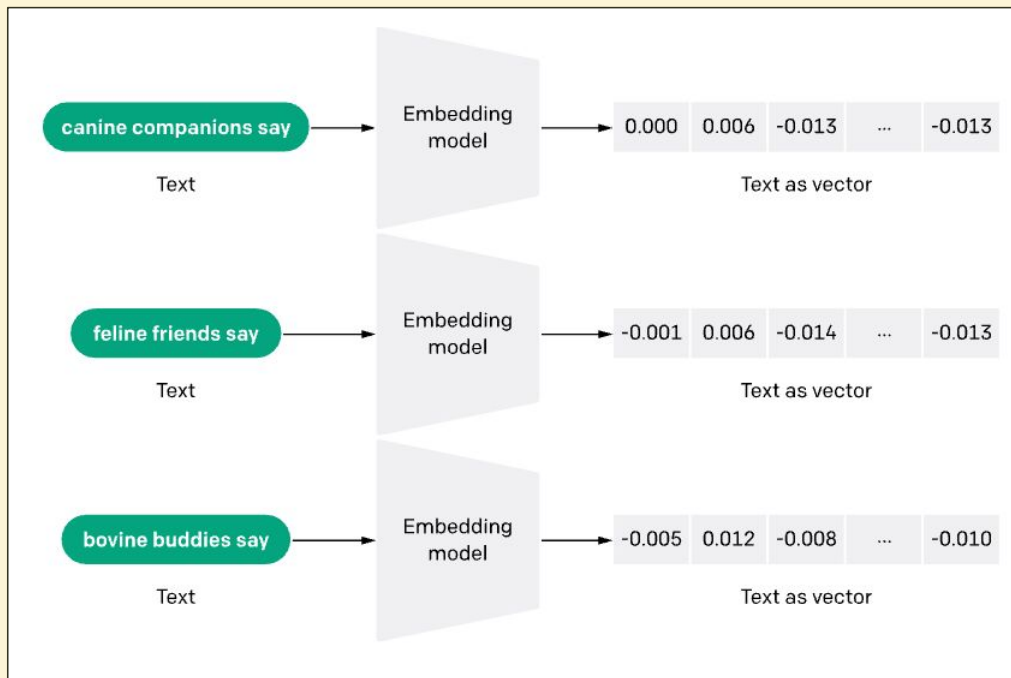
Text

Token IDs

A helpful rule of thumb is that one token generally corresponds to ~4 characters of text for common English text. This translates to roughly $\frac{3}{4}$ of a word (so 100 tokens \approx 75 words).

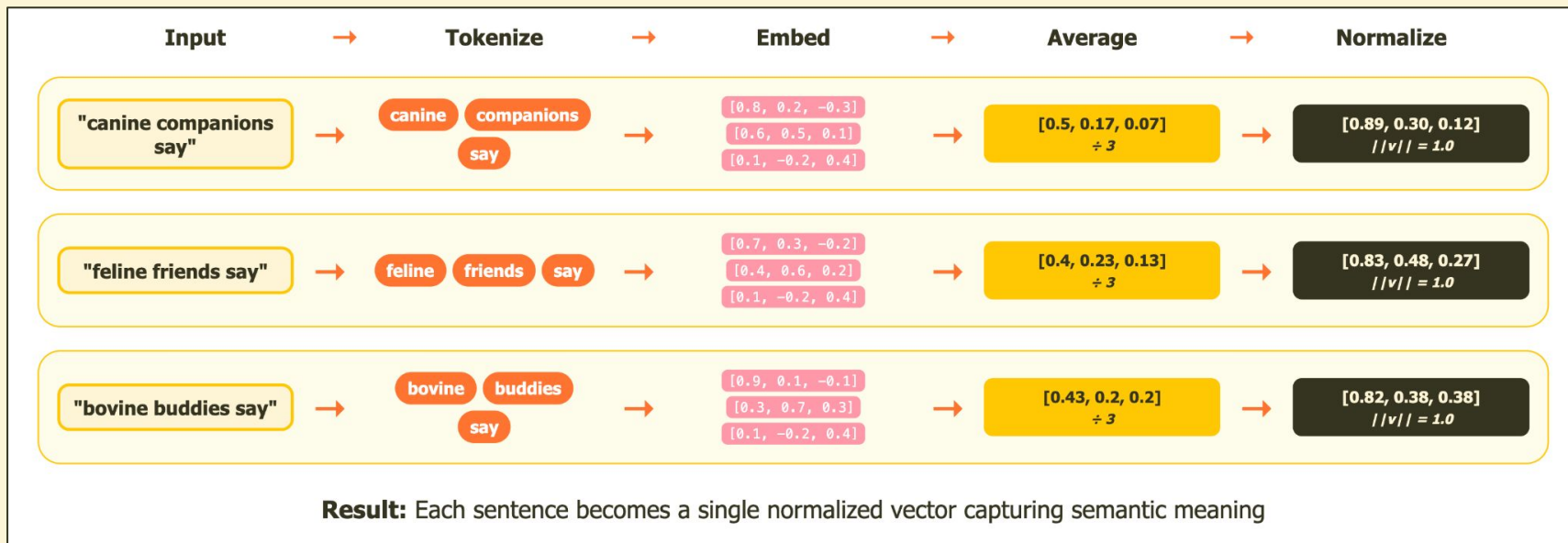
Embedding Vector

units of text/tokens represented by numbers/vectors



Embedding Model

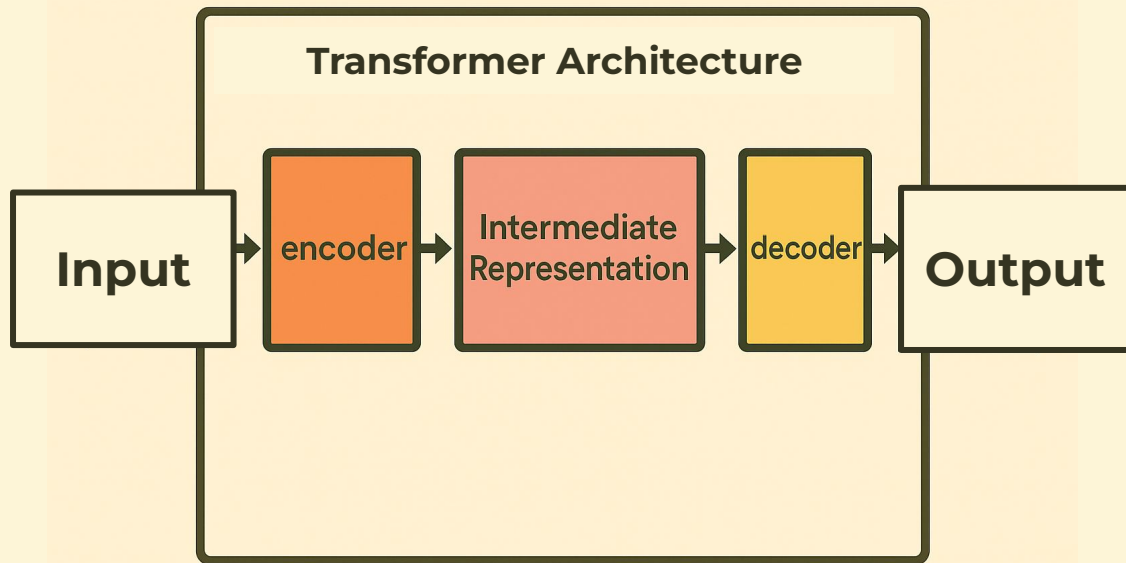
encodes text/tokens into a vector embedding



Large Language Model (LLM)

****large**** neural networks, specifically transformers, designed to process and generate natural language

Examples: GPT-4, Claude, Llama, etc.



Embedding Model vs LLMs

share architectural roots with LLMs but are trained for fundamentally different tasks

Output Type

Embedding: number vectors

LLM: human-readable text

Use Cases

Embedding: search, recommendations

LLM: chatbots, Q&A, summarization

Main Purpose

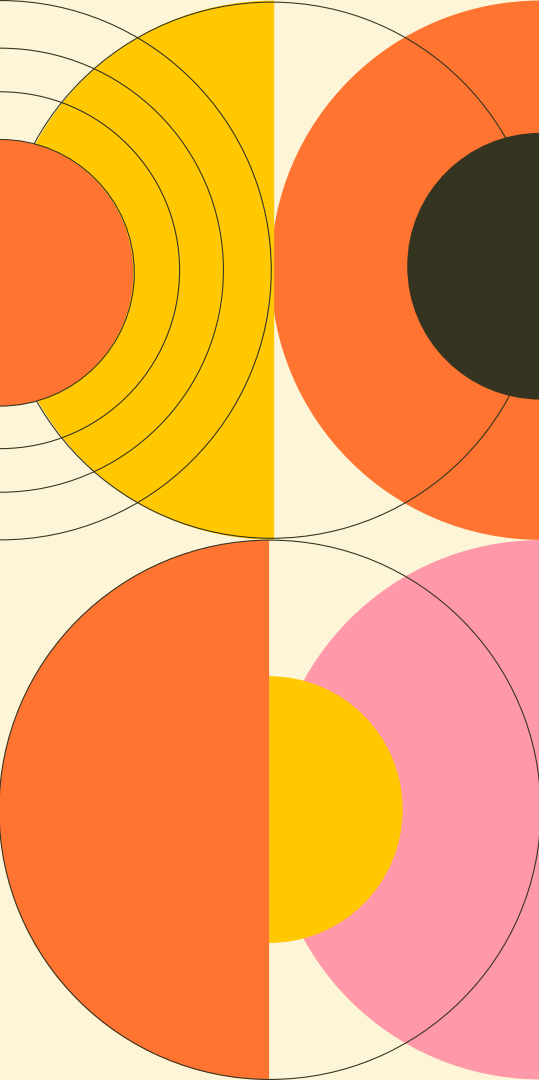
Embedding: understand & compare

LLM: generate & mimic conversation

Processing

Embedding: faster, more efficient

LLM: slower, more complex

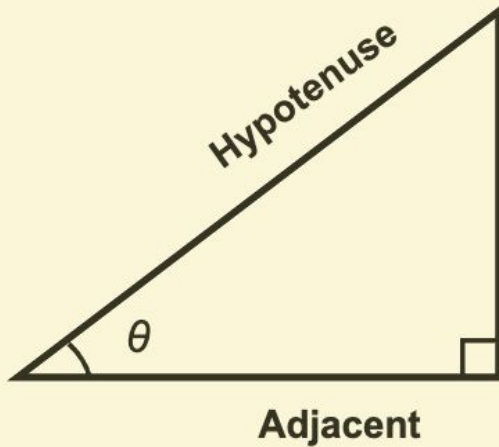


02

The Geometry of Language

How math represents meaning

INTRO TO SOH CAH TOA



$$\sin \theta = \frac{\text{opposite}}{\text{hypotenuse}}$$

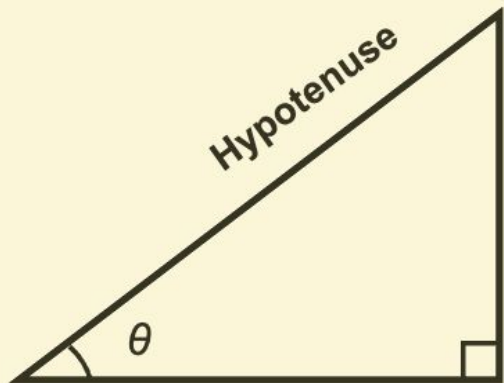
$$\cos \theta = \frac{\text{adjacent}}{\text{hypotenuse}}$$

$$\tan \theta = \frac{\text{opposite}}{\text{adjacent}}$$

Image Source: Claude

SOH **CAH** TOA \rightarrow **Cosine** Similarity

2D Triangle

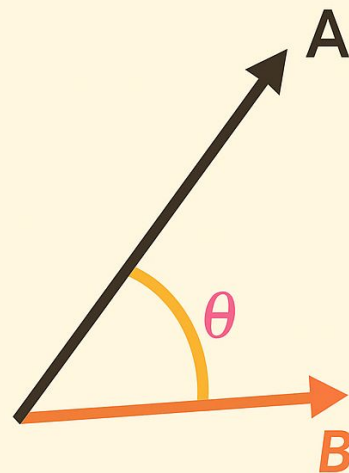


Adjacent

Opposite

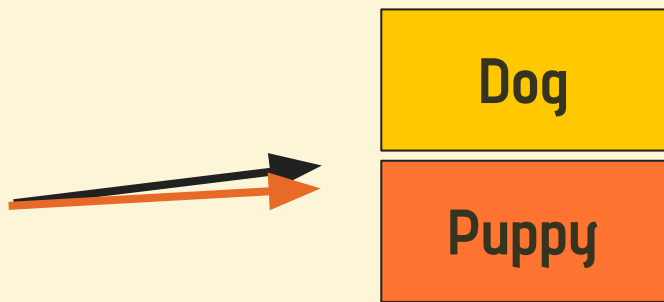


High-dimensional Vectors

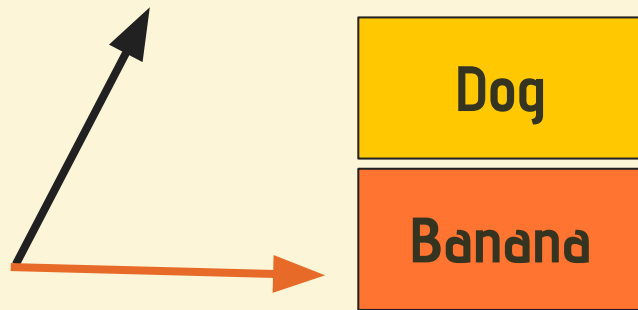


Similarity captures meaning

In LLMs, vectors that represent **similar concepts** end up pointing in almost the same direction:



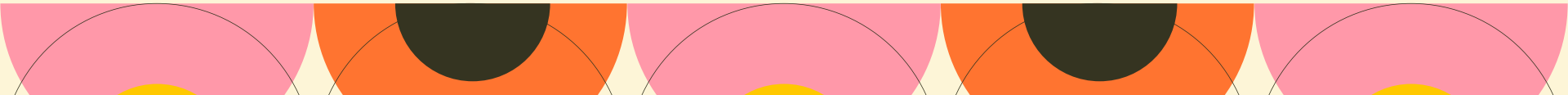
High Cosine Similarity



Low Cosine Similarity

Takeaway

Cosine similarity (the high-dimensional soh-cah-toa) is a useful **measure of semantic similarity**.

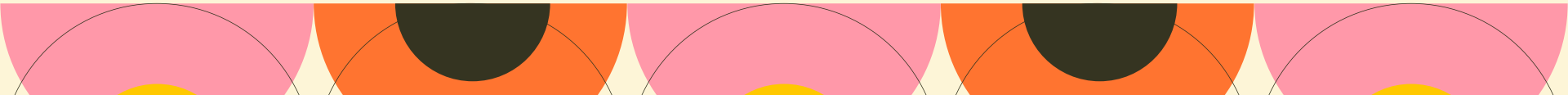


03

Summary

LLMs & Similarity

- Embeddings = numerical representation of text.
- Cosine similarity measures the angle between embeddings.
- High similarity means close meaning.
- LLMs use it for search, recommendations, and understanding context.





Who cares?

¬(ツ)¬



Understanding the geometry helps us:

Search & Recommend Better

Detect Meaning & Semantic Understanding

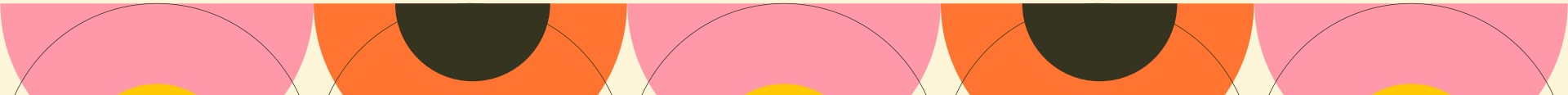
Expose & Prevent Bias



Understanding the geometry helps us:

Search & Recommend Better

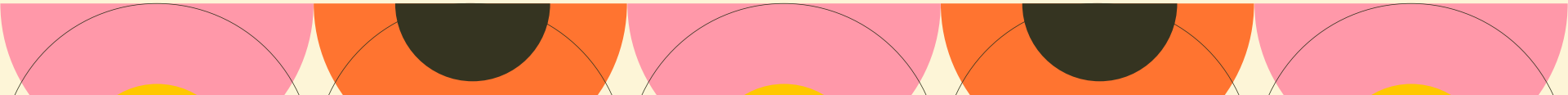
RAG (Retrieval-Augmented Generation) systems or search engines can use **cosine similarity** to measure and rank documents by relevance to **ensure closely related documents** are returned for a user's search query.



Understanding the geometry helps us:

Detect Meaning & Semantic Understanding

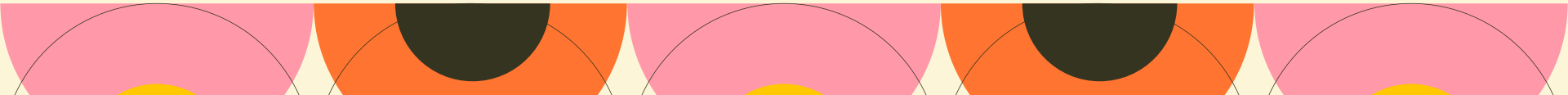
With **cosine similarity**, LLM-generated product descriptions can be compared to human-written versions. A high cosine similarity indicates the model's output is **similar to human-written text**.

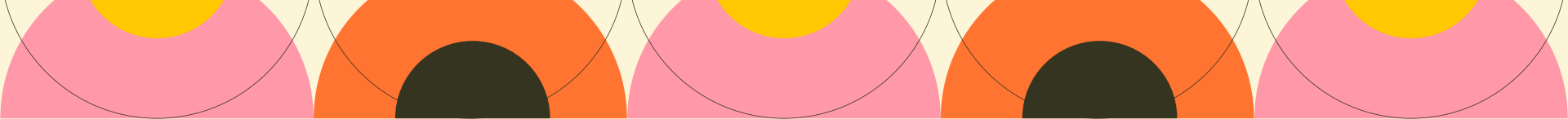


Understanding the geometry helps us:

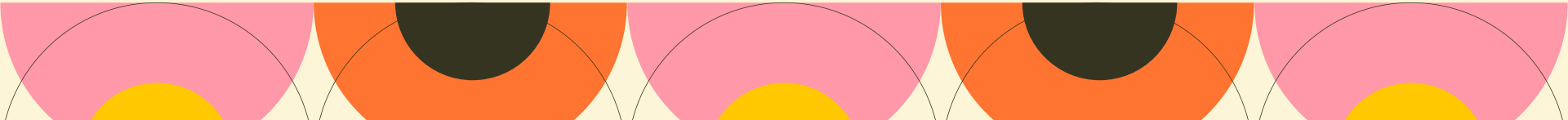
Expose & Prevent Bias

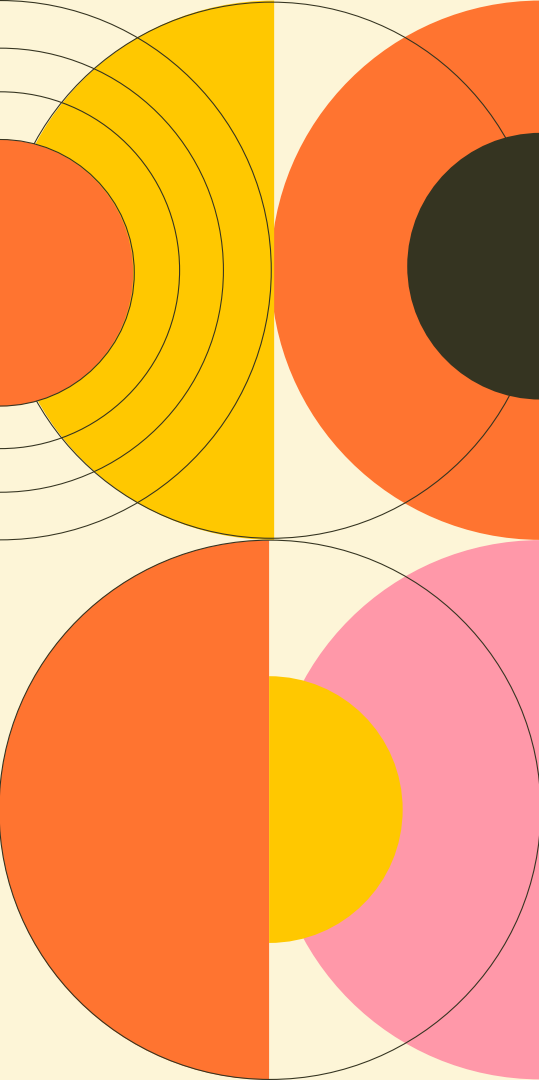
A 2016 NeurIPS publication, “Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings” by Bolukbasi et. al. used **cosine similarity** to explain how embeddings can **amplify gender bias**.





In short: Geometry is the invisible map inside an LLM and cosine similarity is the compass that helps us navigate meaning and interpret LLMs.





04

Advanced Content

Unit Circle

all points lie exactly one unit away from the origin.

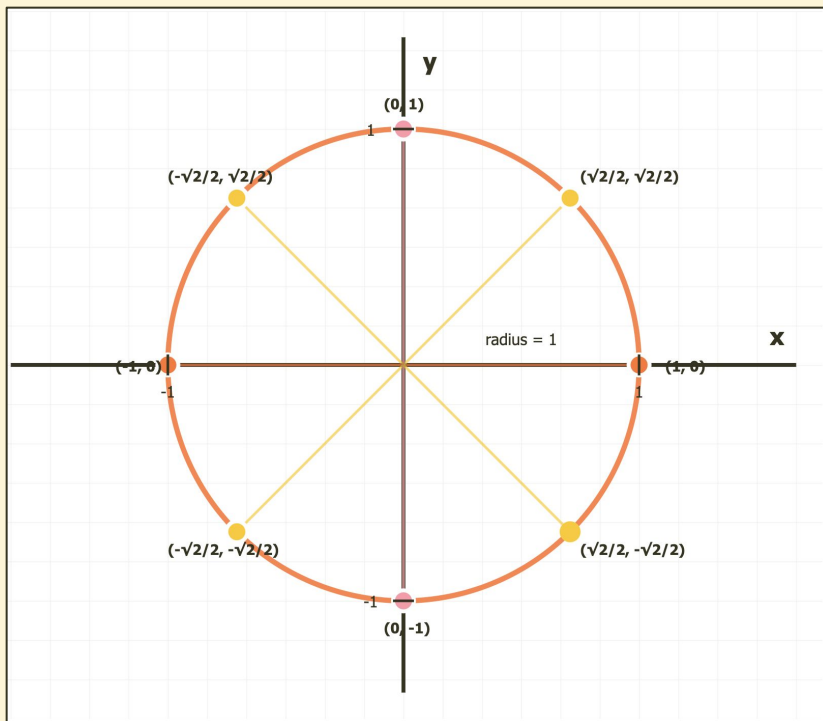


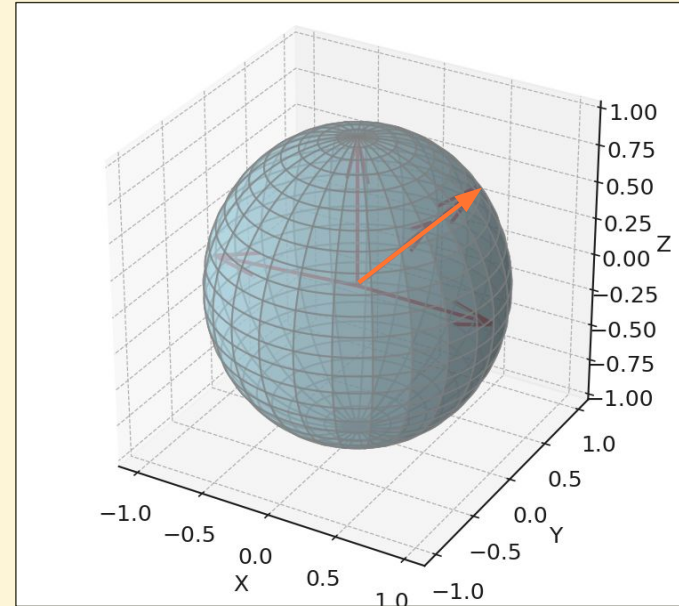
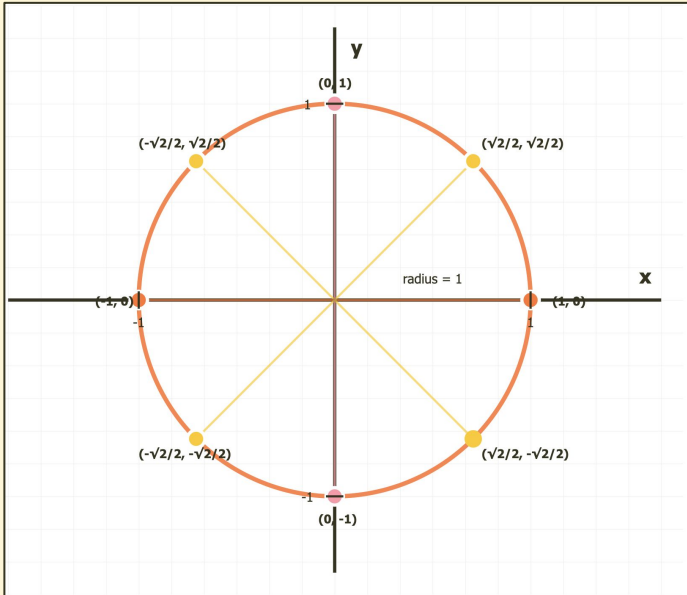
Image Source: Claude

Unit Circle \rightarrow Unit Sphere

2D Unit Circle

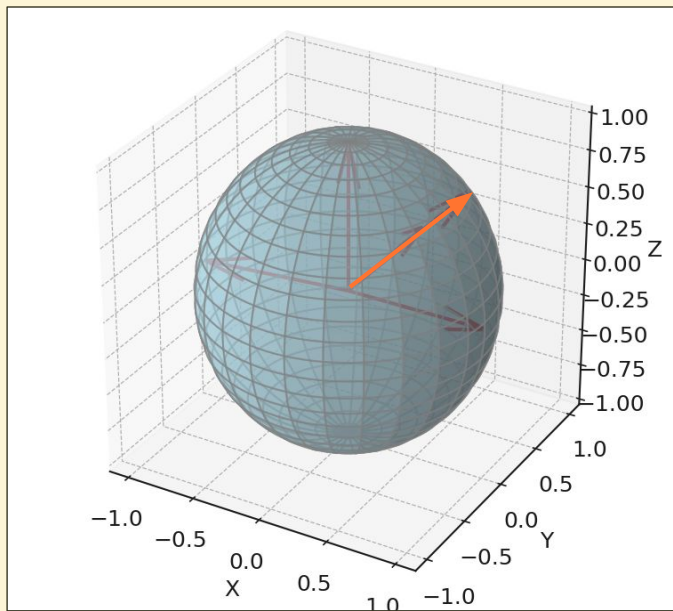


3D Unit Sphere



Unit Circle → Unit Hypersphere

3D Unit Sphere

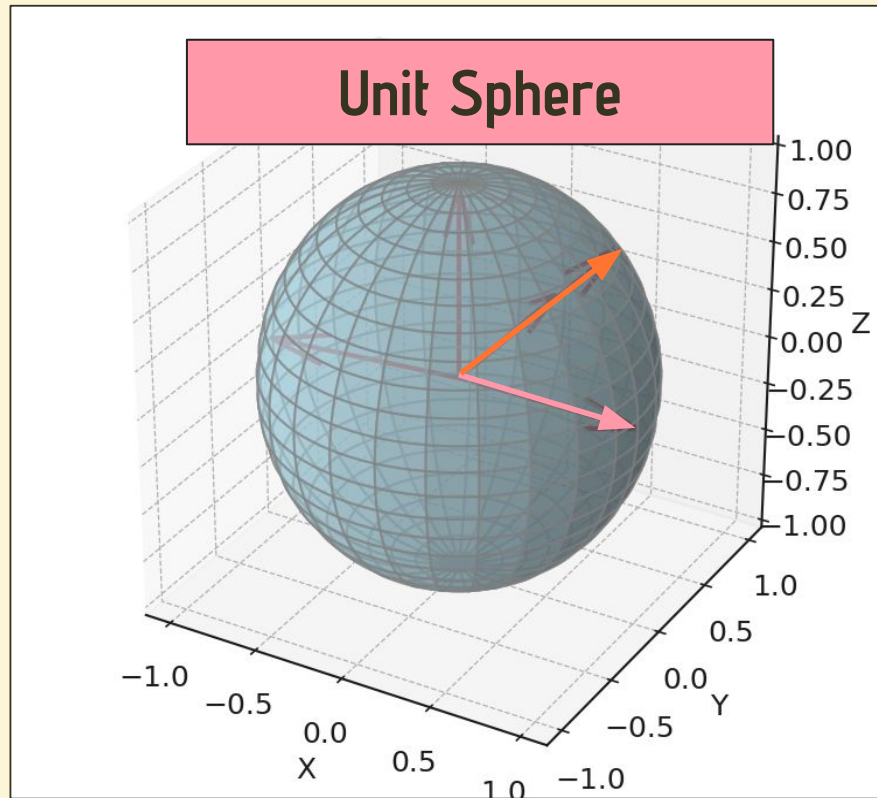
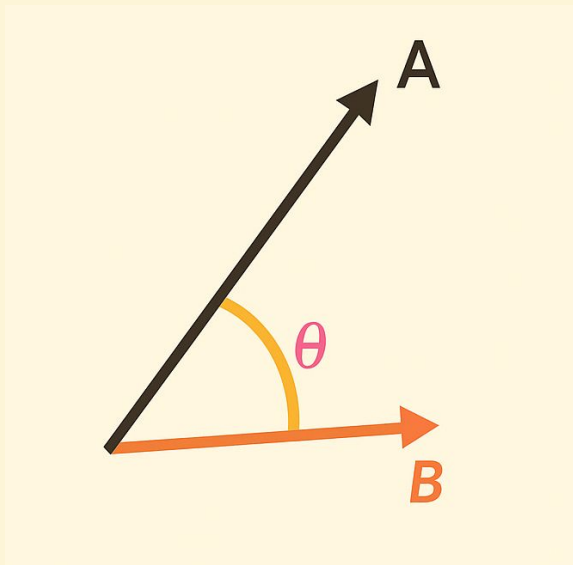


n-D Unit Hypersphere

While we cannot visualize a 4D or n-D hypersphere, but we understand it mathematically through the same constraint principle.

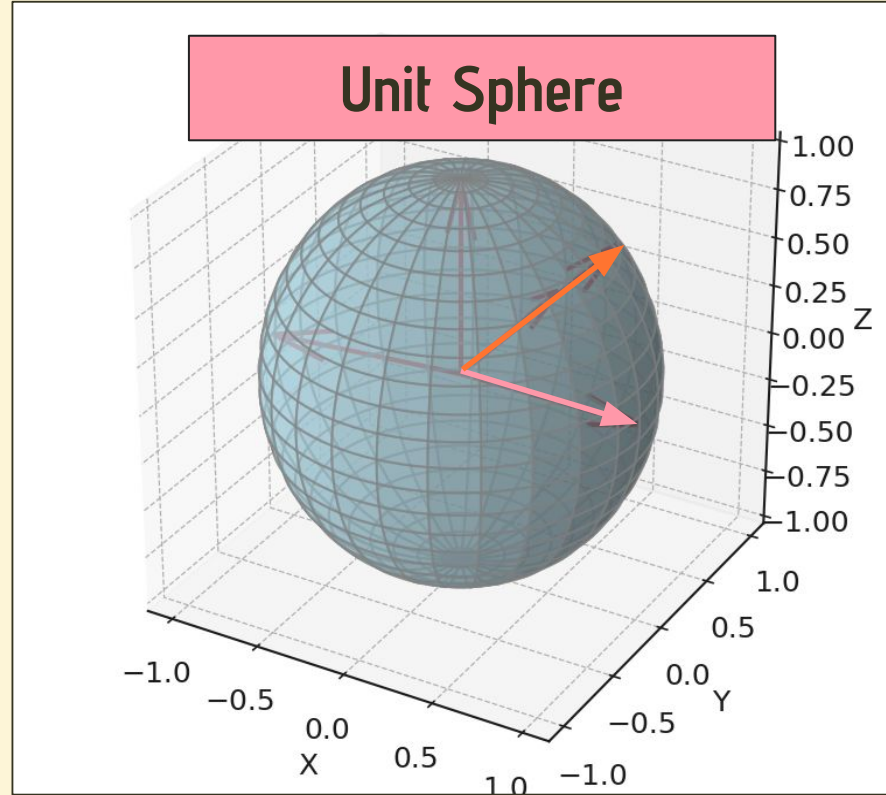
All points lie exactly one unit away from the origin.

Cosine Similarity



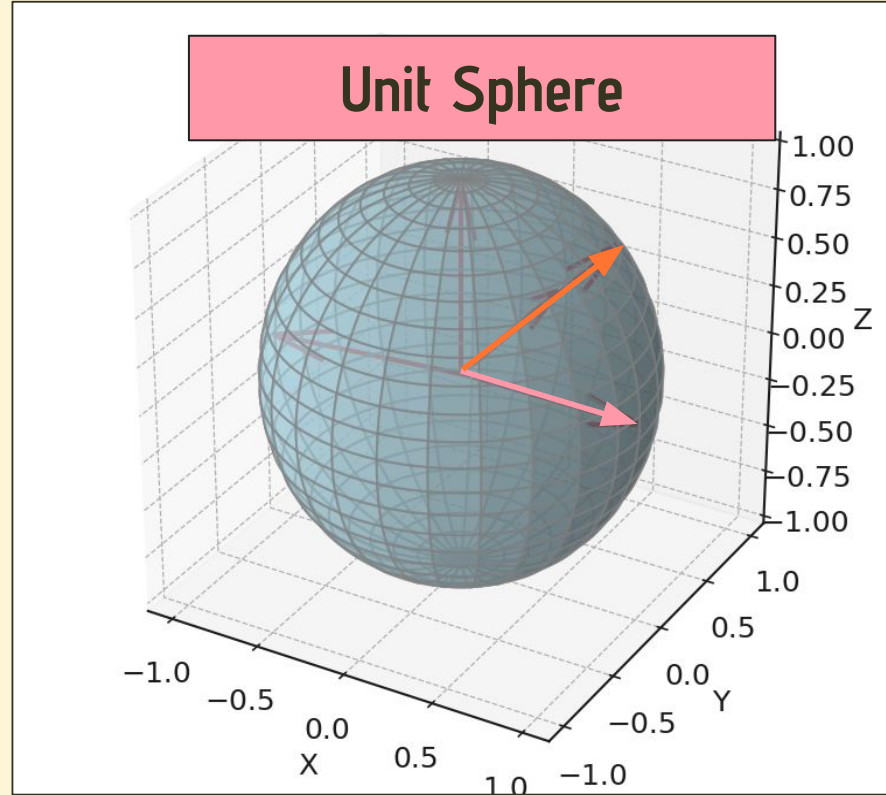
Cosine Similarity

$$\text{CosineSimilarity}(A, B) = \frac{A \cdot B}{|A||B|}$$



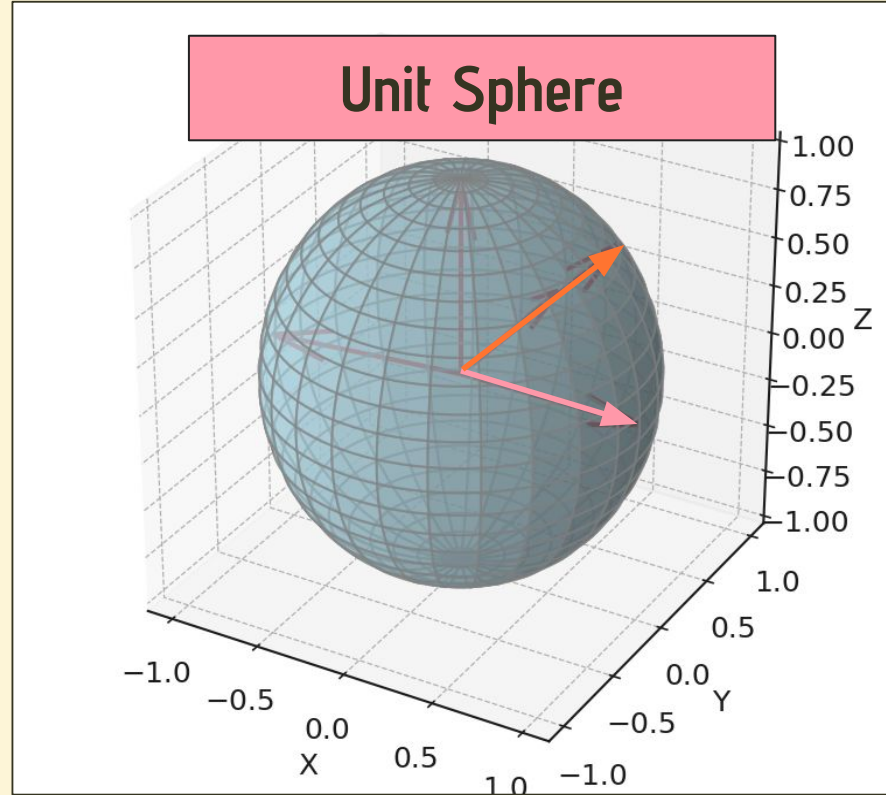
Cosine Similarity

$$\text{CosineSimilarity}(A, B) = \frac{\text{Dot Product}}{|A| |B|}$$



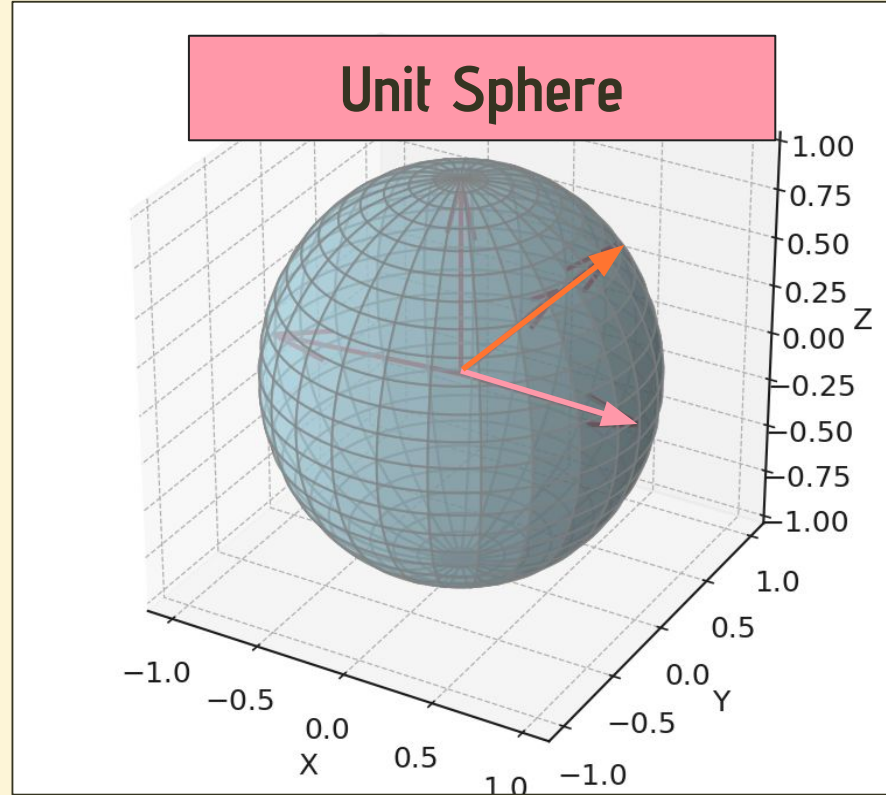
Cosine Similarity

$$\text{CosineSimilarity}(A, B) = \frac{\text{Dot Product}}{|A| |B|}$$



Cosine Similarity

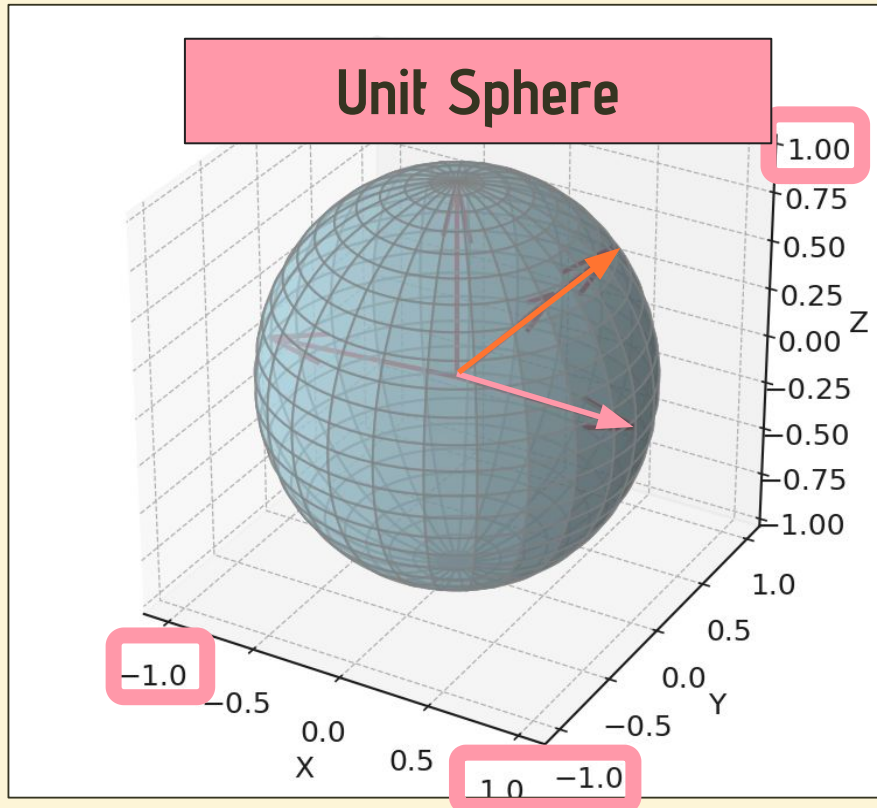
$$\text{CosineSimilarity}(A, B) = \frac{\text{Dot Product}}{A \text{ \& B's Length}}$$



Cosine Similarity

$$\text{CosineSimilarity}(A, B) = \frac{\text{Dot Product}}{A \text{ \& B's Length}}$$

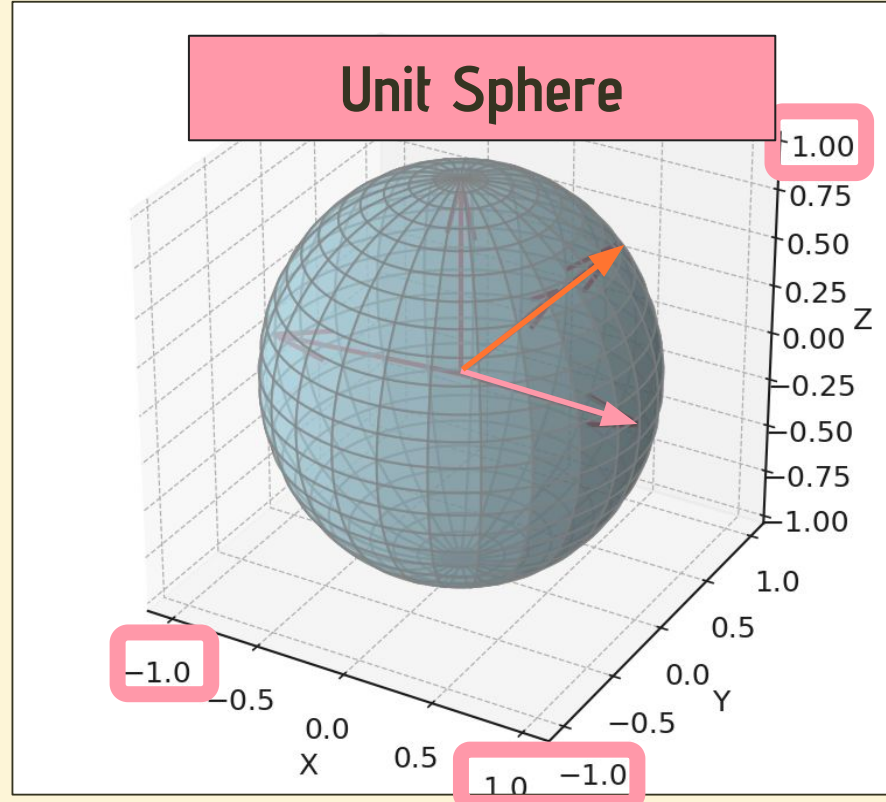
* only with normalized vectors or vectors of length 1



Cosine Similarity

$$\text{CosineSimilarity}(A, B) = \frac{\text{Dot Product}}{1}$$

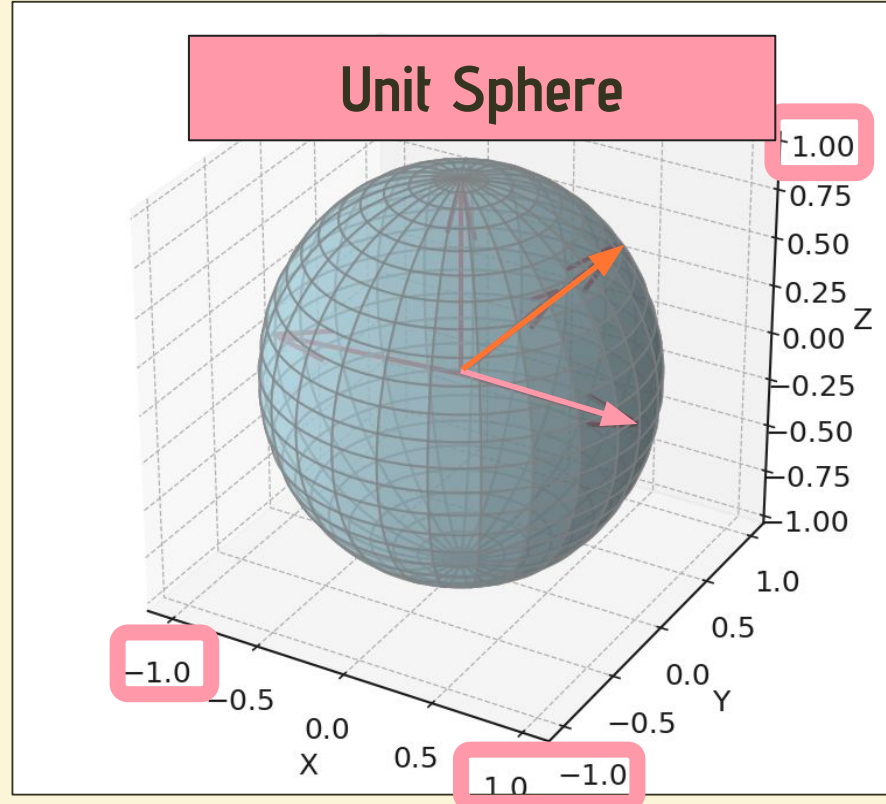
* only with normalized vectors or vectors of length 1



Cosine Similarity

$$\text{CosineSimilarity}(A, B) = \text{Dot Product}$$

* only with normalized vectors or vectors of length 1



Takeaway

Cosine Similarity and the Dot Product (Euclidean Standard Product) are interchangeable when operating within a **unit hypersphere**.



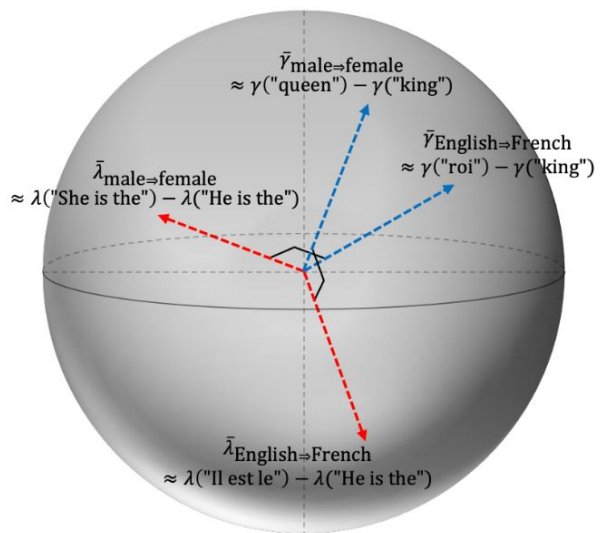


State of the Art Research

The Linear Representation Hypothesis and the Geometry of Large Language Models

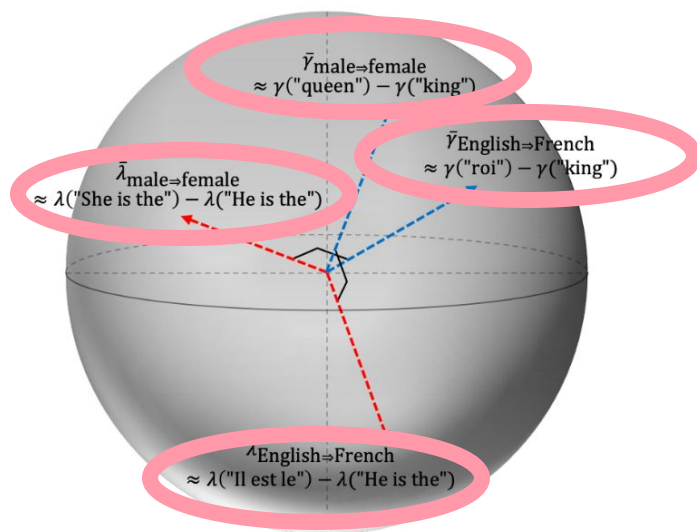
Kiho Park¹ Yo Joong Choe¹ Victor Veitch¹

The Linear Representation Hypothesis and the Geometry of Large Language Models



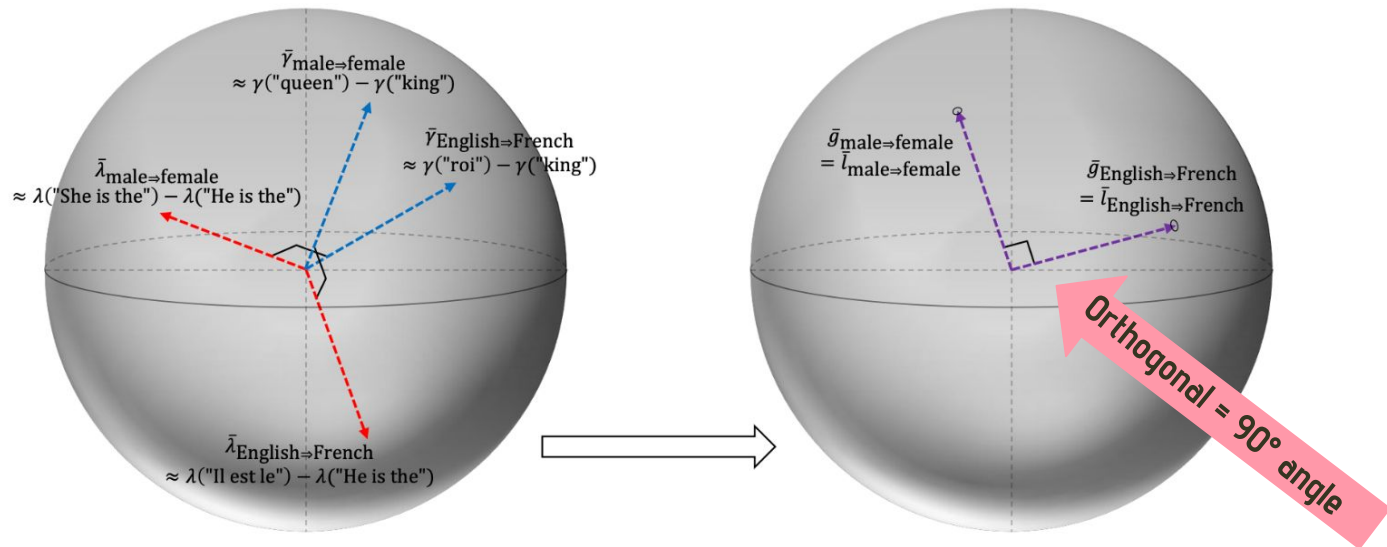
1. Counterfactual pairs define **concepts**

The Linear Representation Hypothesis and the Geometry of Large Language Models



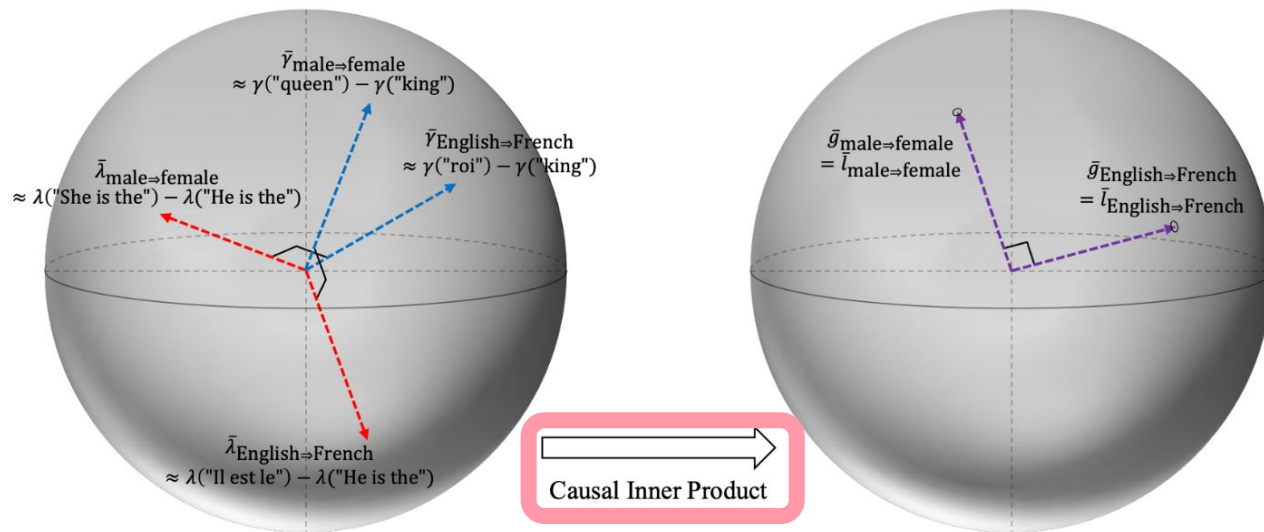
1. Counterfactual pairs define **concepts**

The Linear Representation Hypothesis and the Geometry of Large Language Models



1. Counterfactual pairs define **concepts**
2. Concepts that vary independently should be **orthogonal** (causal separability)

The Linear Representation Hypothesis and the Geometry of Large Language Models



1. Counterfactual pairs define **concepts**
2. Concepts that vary independently should be **orthogonal** (causal separability)
3. Requires **custom inner product** to respect semantics

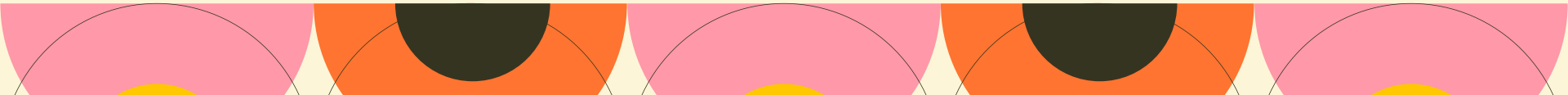
Takeaway

Using the right inner product, practitioners can manipulate vector embeddings **with** simple vector addition.



Linear Representation Hypothesis

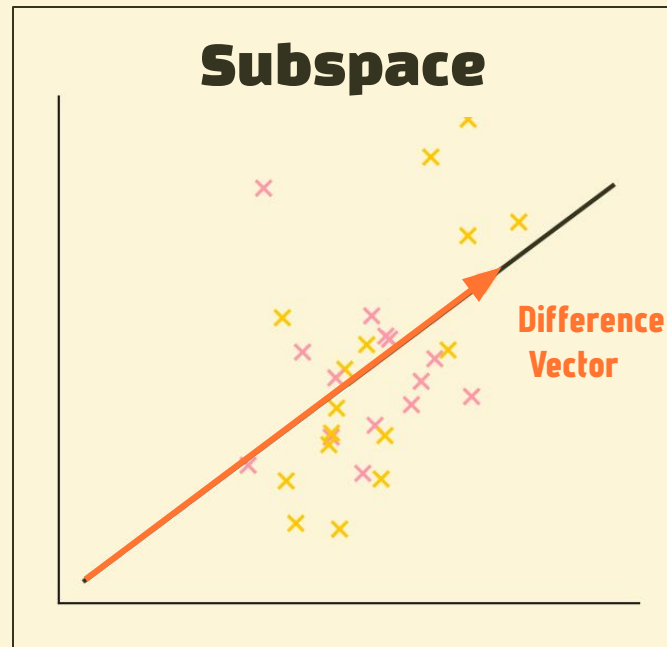
Linear Representation Hypothesis posits that high-level **concepts** encoded as **directions** (one-dimensional subspaces) and simple **linear** algebraic operations can be used to **interpret** and **control** the model (Mikolov et al., 2013c; Arora et al., 2016; Elhage et al., 2022).



3 Views of Linear Representation

1. Subspace (direction) view

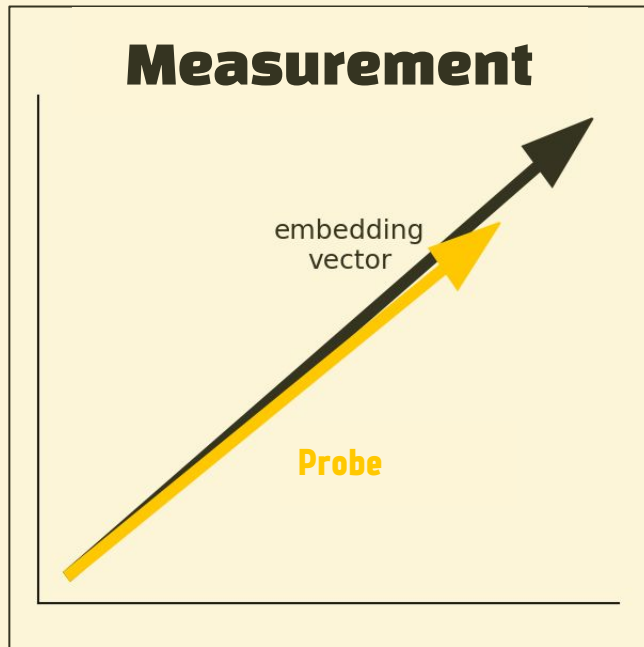
→ captures a concept as a **single direction**



3 Views of Linear Representation

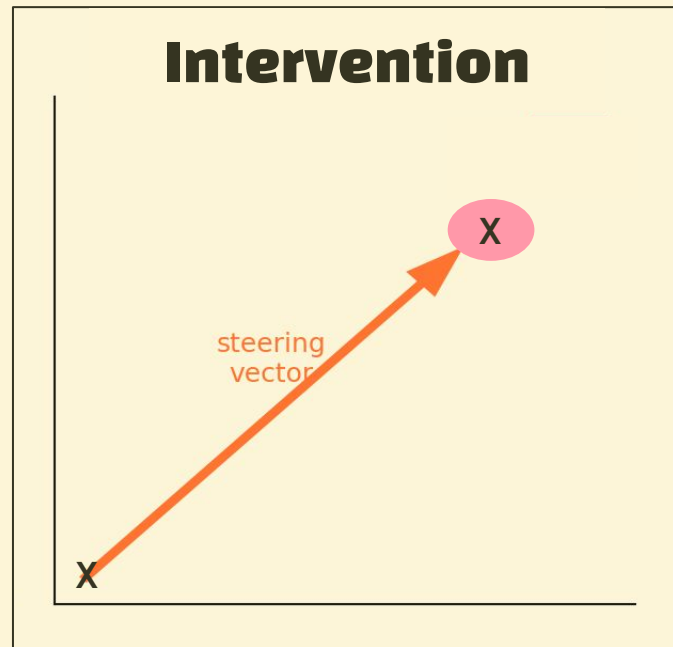
2. Measurement (probe) view

→ the same direction serves as a **probe** to read out how likely the model is to express that vector (“extract the scalar/logit”)



3 Views of Linear Representation

3. **Intervention (steer)** view → the same direction can **steer** the model's output toward the concept

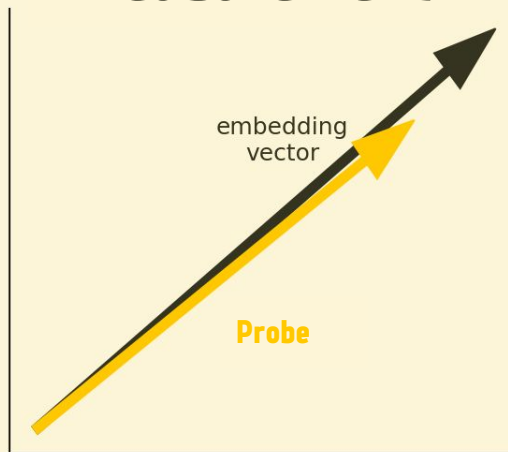


3 Views of Linear Representation

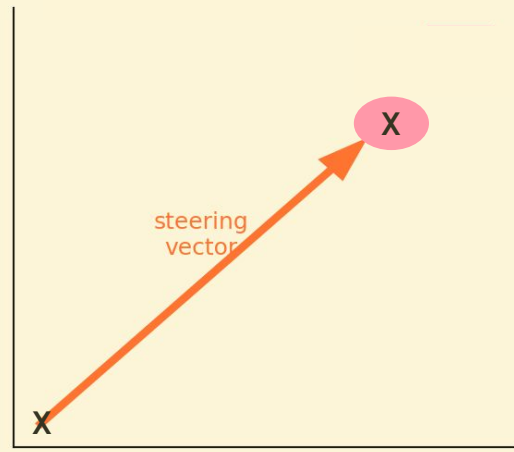
Subspace



Measurement



Intervention



Conclusion

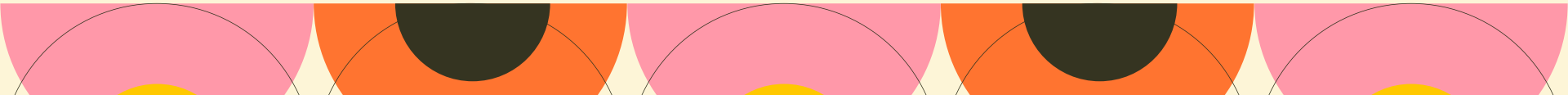
All three views:

- (1) seeing a concept as a **subspace**,
- (2) as something you can measure via a linear **probe (interpretation)**, and
- (3) as something you can **steer (control)** —are just different lenses on **the same underlying direction** in the model's **embedding space**.



Takeaway

- Using the right inner product, all three Linear Representation views are related
- Linear representations can be used to **interpret** (probe) and **control** (steer) LLM outcomes — improving the **interpretability** of LLMs.



References

- Arora, S., Li, Y., Liang, Y., Ma, T., and Risteski, A. A latent variable model approach to PMI-based word embeddings. Transactions of the Association for Computational Linguistics, 4:385–399, 2016.
- Bolukbasi, T., Chang, K. W., Zou, J. Y., Saligrama, V., & Kalai, A. T. (2016). Man is to computer programmer as woman is to homemaker? debiasing word embeddings. Advances in neural information processing systems, 29.
- Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., et al. Toy models of superposition. arXiv preprint arXiv:2209.10652, 2022.
- Mikolov, T., Yih, W.-T., and Zweig, G. Linguistic regularities in continuous space word representations. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 746–751, 2013c.
- Park, K., Choe, Y. J., & Veitch, V. (2024) The Linear Representation Hypothesis and the Geometry of Large Language Models. In the Proceedings of the 41st International Conference on Machine Learning (ICML) 235:39643–39666, 2024.



Thanks!

lauren.alvarez@fueilx.ai
linkedin.com/laurenalvarez1

CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon**, and infographics & images by **Freepik**

