Advanced Methods in Natural Language Processing

A bit more on embeddings

Arnault Gombert

May 2026

Barcelona School of Economics

Why Cosine over L2?

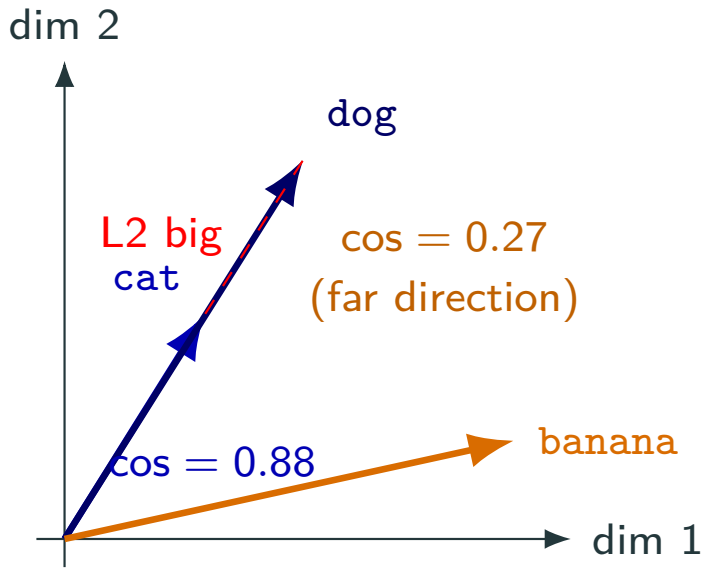
Why Not Euclidean Distance? The Magnitude Problem

L2 confuses *length* with *meaning*. Norm grows with surface artefacts (word frequency, document length), not semantics. **Cosine** keeps only **direction**.

1. Same meaning, longer text (distilBERT).
"*Great film.*" vs. "*Great film. I loved every single minute of it.*": norm 7.5 → 8.3, so L2 = 4.9 (far), yet cos = 0.81 (close).

2. Close in direction, different norm (GloVe).
`cat` vs. `dog`: cos = 0.88, but norms differ (5.0 vs 5.6), inflating L2 = 2.7.

3. Close in norm, far in direction (GloVe).
`cat` vs. `banana`: nearly equal norms (5.0 vs 5.4, small norm gap), yet cos = 0.27. L2 = 6.3 hides how unrelated they are.



`cat`/`dog`: same direction, different norm.

`cat`/`banana`: similar norm, far direction.

Cosine *is* L2, After Normalization

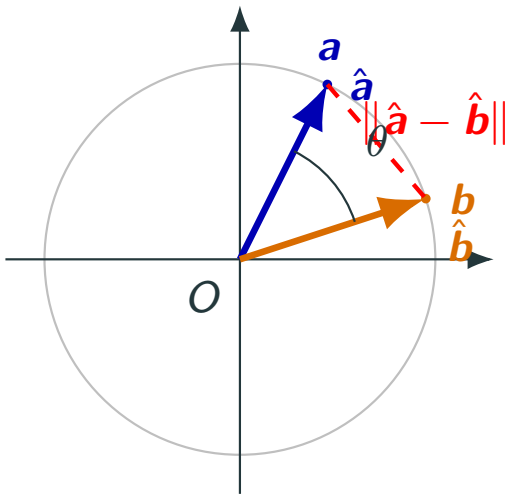
Embeddings don't live on the unit sphere (norms vary), so L2 $\neq$ cosine. Project them onto it ($\hat{\boldsymbol{a}} = \boldsymbol{a}/\|\boldsymbol{a}\|$) and the two coincide:

$$\begin{aligned}\|\hat{\boldsymbol{a}} - \hat{\boldsymbol{b}}\|^2 &= \hat{\boldsymbol{a}} \cdot \hat{\boldsymbol{a}} - 2\,\hat{\boldsymbol{a}} \cdot \hat{\boldsymbol{b}} + \hat{\boldsymbol{b}} \cdot \hat{\boldsymbol{b}} \\ &= 1 - 2\cos(\boldsymbol{a}, \boldsymbol{b}) + 1 \\ &= 2\big(1 - \cos(\boldsymbol{a}, \boldsymbol{b})\big).\end{aligned}$$

$$\boxed{\cos(\boldsymbol{a}, \boldsymbol{b}) = 1 - \tfrac{1}{2}\,\|\hat{\boldsymbol{a}} - \hat{\boldsymbol{b}}\|^2}$$

On normalized vectors, cosine and (negative) L2 give the **same ranking** (Manning et al. 2008; used by FAISS / pgvector). So if you fine-tune word2vec and skip normalization, compare with **cosine**, not L2.

Take-away: **cosine** reads semantics directly; L2 only matches it *after* normalization.



Chord length on the unit circle

$$= \sqrt{2(1 - \cos\theta)}.$$

Static Embeddings: the Norm Tracks Frequency

word2vec / GloVe: the Norm Encodes *Frequency*

The objective scores with a dot product.
Skip-gram (negative sampling) maximises

$$\log \sigma(\boldsymbol{v}_c^\top \boldsymbol{v}_w) + \sum_k \log \sigma(-\boldsymbol{v}_{n_k}^\top \boldsymbol{v}_w)$$

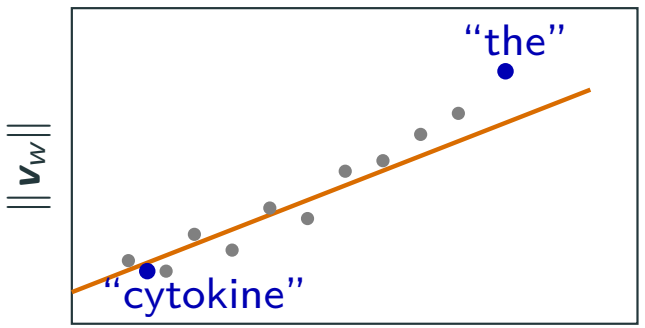
(Mikolov 2013). The norm $\|\boldsymbol{v}_w\|$ scales with how *often* a word is updated, not its meaning.

Schakel & Wilson (2015). Norm grows with **frequency**: function words ("the", "of") get the *largest* norms, rare content words the smallest.

Mu & Viswanath (2018). The top PC of GloVe *is* frequency. Removing it lifts similarity: SimLex-999 0.37 → 0.42, WordSim-353 0.60 → 0.64.

Nuance: the frequency PC lives in the *direction* too, so cosine doesn't remove it. Cosine > L2, but still strip the top PC.

Norm vs. word frequency



Frequent words → large norm.

Same idea works for GloVe.

When Both Metrics Break: Contextual Embeddings

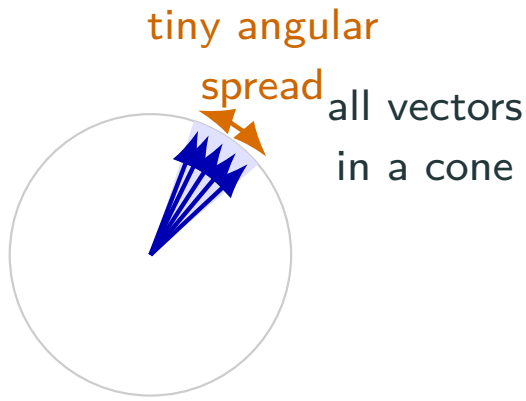
BERT / GPT: Why *Both* Metrics Struggle

1. Curse of dimensionality ⇒ L2 fails. In high d (768 for BERT), L2 distances *concentrate*: random vectors become nearly equidistant, max / min L2 $\to 1$ (Beyer 1999). The near/far gap collapses, so L2 can't rank neighbours.

2. Anisotropy ⇒ cosine loses contrast. Vectors pile into a narrow **cone** (Ethayarajh 2019). Real distilBERT cosines:

- `cat/dog`: 0.80 (related)
- `cat/car`: 0.50 (unrelated)
- `movie/weather`: 0.49

Even unrelated words score ≈ 0.5, not 0: the usable range is squeezed, so cosine barely separates similar from dissimilar.



Small angles ⇒ all cosines crowd near 1, little **contrast** to rank by.

3. Rogue dimensions (Timkey 2021). A few dims dominate the dot product (distilBERT: top std ≈ 3.8 vs. median ≈ 0.28). Toy case $\boldsymbol{u} = [50, 1, 0]$, $\boldsymbol{v} = [50, 0, 1]$: the "50" forces cos $\approx \frac{2500}{2501} = 0.9996$, though dims 2–3 are orthogonal.

On raw contextual vectors, **both** cosine and L2 lose contrast. Next: how to fix the geometry.

Fixing the Geometry

Fixing the Geometry: Make Embeddings Comparable

Which vector? *Token-level* (raw hidden states) or *sentence-level* (mean-pool tokens, or `[CLS]`). Both inherit the cone; pooling alone does not fix it.

A. Post-hoc (repair at inference, no retraining):

Drop / standardize rogue dims (Timkey 2021): $v_i' = (v_i - \mu_i)/\sigma_i$ per dim, so high-variance dims stop dominating.

All-but-the-top (Mu & Viswanath 2018, words): subtract mean, project out the top $\sim d/100$ PCs (frequency).

Whitening (Su 2021): map to zero mean, identity covariance,

$$\boldsymbol{x}' = (\boldsymbol{x} - \boldsymbol{\mu})\, W, \quad W = U\Lambda^{-1/2},$$

($\Sigma = U\Lambda U^\top$). All directions equal variance (*isotropic*); top-k cols of W also cut dimension.

B. Train it right (fix it *by construction*):

Sentence-BERT (Reimers & Gurevych 2019): siamese fine-tuning on NLI/STS so **output cosine matches semantic similarity** directly.

SimCSE (Gao 2021): contrastive InfoNCE over cosine; pulls paraphrases together, pushes negatives apart. The canonical contrastive recipe.

Rule: fix the geometry first (post-hoc), or train so cosine is meaningful by design.