# Dense Word Embeddings from Leipzig-300K: PPMI+SVD vs fastText

## What I did

* Built **symmetric, distance-weighted (1/d)** co-occurrence matrices on **Leipzig eng\_news\_2024\_300K** for **W ∈ {2, 5, 8, 10}** using **CSR sparse** storage.
* Converted counts → **PPMI** (computed only for observed pairs), then applied **Truncated SVD** to obtain **N×d** dense word embeddings; rows L2-normalized so cosine = dot.
* **Chose d without external gold** using four intrinsic diagnostics: **Explained Variance Ratio (EVR)**, **PPMI reconstruction RMSE**, **Neighborhood Stability**, and **Silhouette**.
* Selected **window size W** via **seed-set clustering** (purity & NMI) over {animals, countries, colors, sports, professions, emotions}.
* Evaluated the chosen configuration on **SimLex-999** and **WordSim-353**, and compared against **pretrained fastText (wiki-news-300d-1M)** with the *same* metrics.
* Added qualitative checks (labeled PCA/t-SNE and anchor-neighborhood maps).

## Data & preprocessing

* Tokenization: lowercase regex.
* Vocabulary pruning: **min\_count = 100** → **|V| = 5,314**; in-vocab tokens processed ≈ **5.22M**.

## Co-occurrence matrix & window analysis

| W | nnz | density |
| --- | --- | --- |
| 2 | 2,727,946 | 0.0966 |
| 5 | 5,292,484 | 0.1874 |
| 8 | 6,759,486 | 0.2394 |
| 10 | 7,416,895 | 0.2627 |

**What different windows capture (Jaccard overlap of top-10 neighbors):**

* **india:** (2,5)=0.67; (5,8)=1.00; (8,10)=1.00 → neighborhoods **stabilize** by W≥5 (topical/geo co-mentions).
* **government:** (2,5)=1.00; (2,8)=0.82; (8,10)=1.00 → W=2 is tighter/syntactic; W≥8 broadens to institutions/policy terms.
* **football:** mostly 1.00; (2,10)=0.82 → very large windows pull broader sports/business context.
* **market:** mostly 0.82; (8,10)=1.00 → stabilizes at larger W.

**Why Jaccard here?** It measures **set overlap** between neighbor lists, letting us quantify how the *content* of a word’s neighborhood drifts with W (no labels required).

## Dimensionality reduction & choosing d (intrinsic only)

**EVR (Explained Variance Ratio).** With TruncatedSVD on sparse PPMI, cumulative EVR is the fraction of the matrix’s (uncentered) energy captured by the first *d* components.  
**Observed:** EVR increases with *d* for all W:

* W=8 EVR **0.1387 → 0.3005** (d=50→300)
* W=10 EVR **0.1428 → 0.3057** (d=50→300)

**PPMI-RMSE.** RMSE between true PPMI(i,j) and the low-rank dot product on sampled observed pairs.  
**Observed:** Near-best by **d ≈ 200–300**:

* W=10 best **1.1449** at d=200; **1.1517** at d=300 (+0.6%)
* W=8 **~1.2025–1.2067** across d∈{50,100,200,300}

**Neighborhood stability.** Median **Spearman ρ** between similarity vectors at *d* and at a high-capacity reference (*d*=300).  
**Observed:** Monotonic ↑ (examples):

* W=8: **0.882 → 0.917 → 0.947 → 1.000** for d=50→100→200→300
* W=10: **0.888 → 0.923 → 0.951 → 1.000**

**Silhouette (cosine).** Clusterability of a random subset under k-means; often **drops slightly** as *d* grows (typical high-dim effect).  
Examples (cosine silhouette):

* W=8: **0.105, 0.095, 0.067, 0.055** (d=50,100,200,300)
* W=10: **0.133, 0.094, 0.070, 0.061**

**Rule used:** choose the **smallest d** with **EVR ≥ 90% of max** and **RMSE within 3% of best**; tie-break by **silhouette**, then **stability**.  
**Chosen:** **d = 300** (meets EVR/RMSE thresholds and yields most stable neighborhoods across W).

## Picking the window size W (seed-set clustering)

| W | d | Purity | NMI |
| --- | --- | --- | --- |
| 2 | 300 | 0.979 | 0.962 |
| 5 | 300 | 0.979 | 0.968 |
| **8** | **300** | **1.000** | **1.000** |
| 10 | 300 | 0.917 | 0.898 |

**Chosen configuration:** **W=8, d=300** (perfect separation on the six seed categories).

## Intrinsic evaluation (chosen config)

**PPMI+SVD (W=8, d=300)**

* **SimLex-999:** ρ **0.210** (pairs=437)
* **WordSim-353:** ρ **0.581** (pairs=188)

**fastText (wiki-news-300d-1M)**

* **Seed clustering:** purity **0.938**, NMI **0.907**
* **SimLex-999:** ρ **0.379** (pairs=437)
* **WordSim-353:** ρ **0.740** (pairs=188)

**Qualitative neighbors (PPMI+SVD → fastText)**

* **india:** topical geo set {*africa, europe, nations*} → subword/NE heavy {*indian, delhi, kashmir*}.
* **government:** shared {*governments, officials, federal*}; fastText adds {*authorities, legislature, gov*}.
* **football:** PPMI links {*league, team, rugby*}; fastText emphasizes {*soccer, sport/sports, stadium*}.
* **market:** PPMI leans {*stocks, prices, demand*}; fastText brings {*industry, products, economy*}.

## Figures (render straight from GitHub)

**PCA-2D of seed words — PPMI+SVD**  
PCA seed — PPMI+SVD

**PCA-2D of seed words — fastText**  
PCA seed — fastText

**Anchor neighborhoods (k=10) — PPMI+SVD**  
Anchors — PPMI+SVD

**Anchor neighborhoods (k=10) — fastText**  
Anchors — fastText

**t-SNE of 200 frequent content words — PPMI+SVD**  
t-SNE — PPMI+SVD

**t-SNE of 200 frequent content words — fastText**  
t-SNE — fastText

**Quick read of the plots.**

* **Seed PCA:** both models separate categories well; **countries** and **sports** are especially tight. PPMI+SVD also yields compact **colors/animals**; fastText spreads **professions/emotions** slightly more—consistent with subword effects influencing geometry.
* **Anchors:** PPMI+SVD highlights **topical co-mentions**; fastText adds **subword/named-entity** neighbors (e.g., *indian, delhi*).
* **t-SNE:** both show coherent micro-clusters (days of week, finance terms). fastText often bunches **morphologically related** forms more tightly; PPMI+SVD groups **topically co-used** words.

## Takeaways

* **Best configuration:** **W=8, d=300** — perfect seed clustering (purity & NMI = 1.0) and stable neighborhoods.
* **Neural vs co-occurrence:** fastText scores higher on **SimLex/WS353** and reflects **subword/named-entity** info; PPMI+SVD is simple/transparent and excels at **topical co-mention structure**, producing clean category clusters on news text.
* **Effect of window size:** neighborhoods stabilize around **W≈5–8**; too small is overly syntactic, too large drifts toward noisy topical associations.

## Plan & hypotheses

### A) Dimensionality reduction (method + how d is chosen; no SimLex/WS353 here)

**Method choice (simple & scalable).**

1. Build **PPMI** from the sparse co-occurrence matrix to remove raw frequency bias and highlight association strength.
2. Run **Truncated SVD** (aka LSA) **directly on sparse PPMI** to get **N×d** embeddings; L2-normalize rows so cosine = dot.

**Why not PCA?** PCA requires centering and typically dense ops; **TruncatedSVD** gives the same linear-subspace idea but **works on sparse matrices**, so it’s faster and lighter in Colab.

**Alternatives considered (not used):**

* **NMF** (nonnegative, interpretable) → slower and often lower similarity accuracy at small d.
* **Random Projections** (very fast) → lower fidelity and unstable neighbors for small d.  
  We keep **PPMI+SVD** for the best balance of **simplicity, speed, and quality**.

**How we pick d (intrinsic only):** sweep **d ∈ {50,100,200,300}** and compute four signals:

* **EVR (Explained Variance Ratio).** Fraction of (uncentered) matrix energy captured by the first *d* SVD components.  
  *Rule:* choose the smallest **d** that reaches ≈**90%** of the EVR at *d*=300 (or where marginal gain < 1% per +50 dims).
* **PPMI-RMSE (held-out reconstruction).** Sample observed (i,j), compare true PPMI(i,j) vs low-rank dot product; compute RMSE.  
  *Rule:* choose the smallest **d** whose RMSE is within **2–3%** of the RMSE at *d*=300.
* **Neighborhood stability.** For a probe set, compare cosine-similarity vectors at *d* vs *d*=300 using **Spearman ρ** (or top-k Jaccard).  
  *Rule:* prefer **d** where median ρ is very high (e.g., ≥0.99) or top-10 Jaccard ≥0.8.
* **Silhouette (cosine).** Clusterability on a random subset under k-means; use as a tie-breaker to prefer compact clusters.

**Final decision rule.** Pick the **smallest d** that satisfies EVR and RMSE, and is stable by the neighbor metric; if tied, use the better silhouette.

**Hypotheses.**

* EVR shows **diminishing returns**; expect useful d around **200–300**.
* Stability increases with d; the “knee” near **200** should already be close to *d*=300.
* Silhouette may drop slightly with d (common in high-dim), so we don’t over-penalize it.

### B) Window size (W ∈ {2,5,8,10}): how we pick it (still no SimLex/WS353)

**Counting once for all windows.** Build **four** co-occurrence matrices in **one streaming pass**, with **distance weighting 1/d**, stored as sparse CSR.

**Diagnostics we use to compare windows:**

* **Jaccard drift of neighbors across W.** Jaccard(A,B) = |A∩B|/|A∪B| on top-k neighbor sets; tells us **how neighborhood content changes** as W grows.
* **Seed-set clustering (purity & NMI).** Tiny human categories (animals, countries, colors, sports, professions, emotions) clustered via k-means; higher is better.

**Decision.** Choose the **W** with highest **seed purity**; if tied, prefer the one with more stable neighborhoods (higher Jaccard, balancing topicality with some syntactic sharpness).

**Hypotheses.**

* Small **W** → more **syntactic** neighbors; large **W** → more **topical** neighbors.
* Expect best W around **5–8** on news text (stable topical context without too much noise).

### C) Full embedding evaluation (now we use SimLex & WordSim)

We evaluate **both** the chosen **PPMI+SVD** (best W,d) **and** **fastText (wiki-news-300d-1M)** with the same protocol:

* **Word similarity benchmarks.** **SimLex-999** and **WordSim-353**: compute **Spearman ρ** between human scores and cosine similarities; report #pairs covered.
* **Seed-set clustering.** Same seed categories; k-means (k=6); report **purity** and **NMI**.
* **Neighborhood inspection (qualitative).** Top-10 neighbors for anchors (e.g., *india, government, football, market*) to compare **topical vs subword** behavior.
* **Quick visuals.** PCA-2D of seed words (labeled) and t-SNE-2D of ~200 frequent content words (labeled).
* **(Optional) Analogies.** A small smoke-test set; report counts (illustrative only).

**Hypotheses.**

* **fastText** should score higher on similarity (SimLex/WS353) due to **subword** information and large-scale training.
* **PPMI+SVD** should show **clean topical clusters** and intuitive co-mention neighborhoods on news text.

### D) Efficiency & practicality (Colab)

* **min\_count=100** to keep |V| manageable.
* **One-pass** multi-window counting; **sparse CSR** everywhere.
* PPMI computed **only on nonzeros**.
* **Randomized TruncatedSVD** with few iterations.
* Cache artifacts (**.npz/.npy**) so later sections run fast.

### E) Deliverables in the repo/report

* **Stats per W:** |V|, nnz, and neighbor drift snapshots.
* **Curves/tables for d-selection:** EVR, **PPMI-RMSE**, neighborhood stability, silhouette.
* **Final choice:** best **(W, d)** with one-line justification.
* **Side-by-side results table:** PPMI+SVD vs fastText → **SimLex ρ**, **WordSim ρ**, **purity/NMI** (and analogies if used).
* **Figures:** labeled **PCA**, labeled **t-SNE**, and **anchor-neighborhood** maps for both methods.

## Abbreviations, formulas, and why each matters

### Co-occurrence (with distance weighting)

* **What:** For each target word *w*, count context words *c* within a symmetric window **W**; weight by **1/d** where *d* is token distance.
* **Matrix:** **C[w,c]** = weighted count.
* **Why:** Captures **which words occur together**, the basis of distributional semantics.

### PMI / PPMI (Pointwise Mutual Information / Positive PMI)

* **PMI:**  
  [ \text{PMI}(w,c) = \log \frac{p(w,c)}{p(w),p(c)} \approx \log \frac{C[w,c]\cdot T}{(\sum\_{c’} C[w,c’])(\sum\_{w’} C[w’,c])} ] with (T=\sum\_{w,c}C[w,c]).
* **PPMI:** (\max(\text{PMI}(w,c),0)).
* **Why:** Highlights **association strength** beyond frequency; reduces function-word dominance.

### SVD / TruncatedSVD (Singular Value Decomposition)

* **Low-rank model:**  
  [ X \approx U\Sigma V^\top \quad (\text{keep top } d) ]
* **Embeddings:** rows of (U\Sigma) (L2-normalized).
* **Why:** Extracts **latent factors** from PPMI; **TruncatedSVD** works **sparse** (unlike centered PCA).

### PCA (Principal Component Analysis) — visualization only

* **What:** Orthogonal directions of maximal variance in centered data.
* **Why:** 2D plots of seed words.

### t-SNE (t-Distributed Stochastic Neighbor Embedding) — visualization

* **What:** Non-linear 2D method preserving **local neighborhoods**.
* **Why:** Qualitative views of micro-clusters.

### Cosine similarity

* (\cos(\theta)=\frac{x\cdot y}{|x||y|}) → with L2 rows, cosine=dot.
* **Why:** Nearest neighbors, similarity benchmarks, stability.

### Jaccard index (neighbor sets)

* (J(A,B)=\frac{|A\cap B|}{|A\cup B|}) on top-k lists.
* **Why:** Quantifies **neighborhood drift** across windows/dimensions.

### EVR (Explained Variance Ratio)

* **What:** Fraction of matrix “energy” captured by first *d* SVD components (uncentered).
* **Why:** Intrinsic guide for **choosing d** (diminishing returns).

### **PPMI-RMSE (Reconstruction error)**

* [ \text{RMSE}=\sqrt{\frac{1}{|\mathcal{S}|}\sum\_{(w,c)\in\mathcal{S}}\big(\text{PPMI}(w,c)-x\_w\top y\_c\big)2} ] where (\mathcal{S}) is a sample of observed pairs and (x\_w,y\_c) are d-dim vectors.
* **Why:** Ensures the low-rank model **faithfully reconstructs** association strengths; a second, independent criterion for **choosing d**.

### Neighborhood stability (Spearman ρ vs reference d)

* **Spearman ρ:** rank correlation between similarity vectors.
* **Why:** Checks that **nearest-neighbor geometry** at chosen d aligns with a **high-capacity** reference (d=300).

### Silhouette (cosine)

* (a)=intra-cluster dist, (b)=nearest other-cluster dist;  
  (s=\frac{b-a}{\max(a,b)}\in[-1,1]).
* **Why:** Tie-breaker favoring **clean clusterability**.

### Purity (clustering)

* (\text{Purity}=\frac{1}{N}\sum\_k \max\_y |C\_k\cap Y\_y|).
* **Why:** Simple measure to compare windows with seed labels.

### NMI (Normalized Mutual Information)

* (\text{NMI}(Y,K)=\frac{I(Y;K)}{\sqrt{H(Y)H(K)}}).
* **Why:** Information-theoretic agreement; robust to cluster size.

### CSR (Compressed Sparse Row)

* Stores only non-zeros with row pointers.
* **Why:** Makes **co-occ/PPMI/SVD** feasible at this scale.

### Window size **W** and distance weighting **1/d**

* **W:** tokens per side; small W → **syntactic**, large W → **topical**.
* **1/d:** closer words weigh more.
* **Why:** Control and diagnose the **type of context** captured.

### fastText (wiki-news-300d-1M)

* **What:** Pretrained **subword** embeddings (character n-grams).
* **Why:** Strong **neural baseline**; better for morphology/rare forms; we evaluate with the **same metrics**.