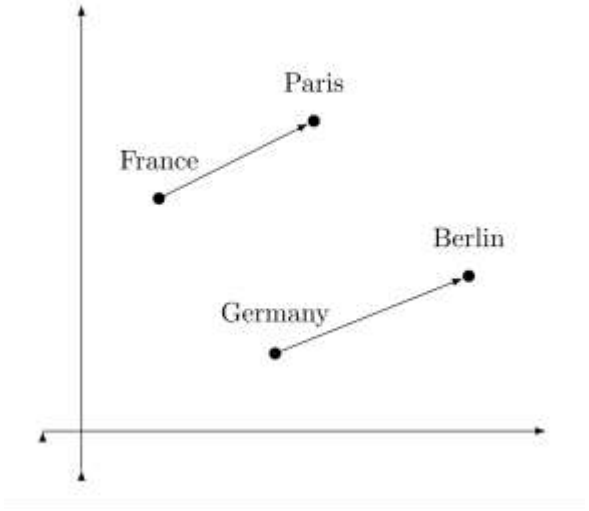
Word Embeddings as Statistical Estimators

Understanding Word Embeddings in NLP

- **What are Embeddings?**
  - Map words/phrases to vectors in Euclidean space
  - Enable statistical & ML methods in NLP
- **Challenge**
  - Evaluated via downstream tasks
  - Lack formal understanding of what embeddings capture



Paper Goals

- **Problem & Proposal**
  - Embeddings lack theoretical grounding
  - Current tests don’t check if real language features are captured
- **Proposal:**
  - Define embeddings in a statistically meaningful way
  - Use generative models with known features
  - Test if embeddings consistently recover them
- **Main Contributions:**
  - Theoretical link: Skip-gram ≈ PMI
  - SVD+EM estimator for infinite PMI
  - SVD + DD copula-based model inspired by Zipf’s law
  - Core question: Do embeddings capture meaning beyond performance?

Word2Vec and Pointwise Mutual Information

- **Key insight (Levy & Goldberg, 2014):**
  - Word2Vec with skip-gram and 1 negative sample implicitly factorizes the **PMI matrix**:
    - $PMI(w,c) = log(^{P(w,c)}/_{P(w)P(c)})$
- **Generalization:**
  - With k negative samples:
    - $SPMI = PMI - \log(k) \cdot J$
    - where J is the all-ones matrix
- **Challenge:**
  - PMI matrices from real corpora often contain **infinite entries** due to zero co-occurrence:

Two Statistical Settings for Language

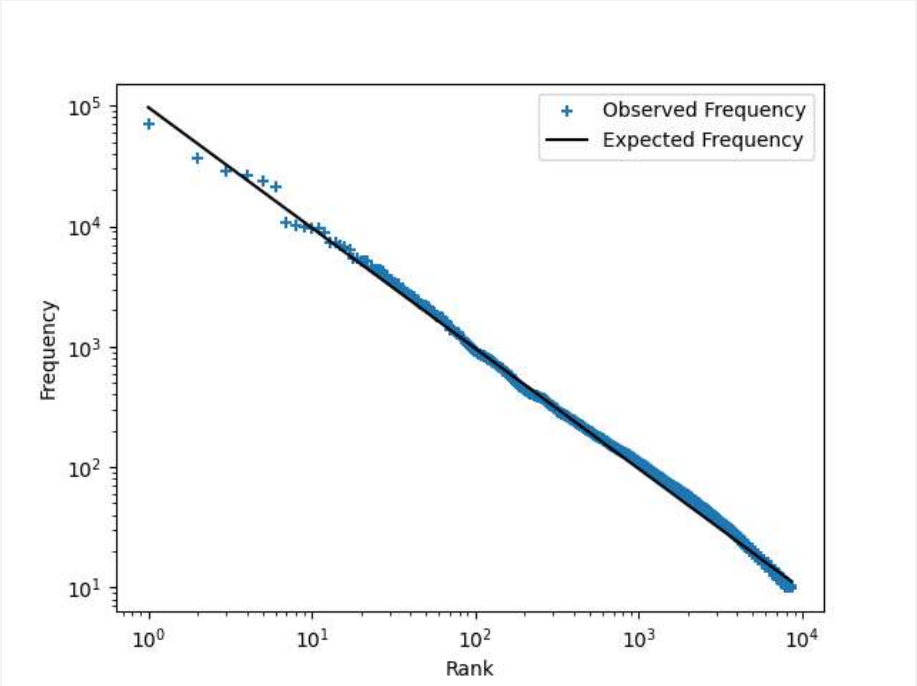
- **Sparse Setting:**
  - Some word pairs truly never co-occur
  - Infinite PMI = real absence of relation
- **Dense Setting:**
  - All word pairs have **non-zero probability** of co-occurrence
  - Observed zeros are due to **finite sample size**

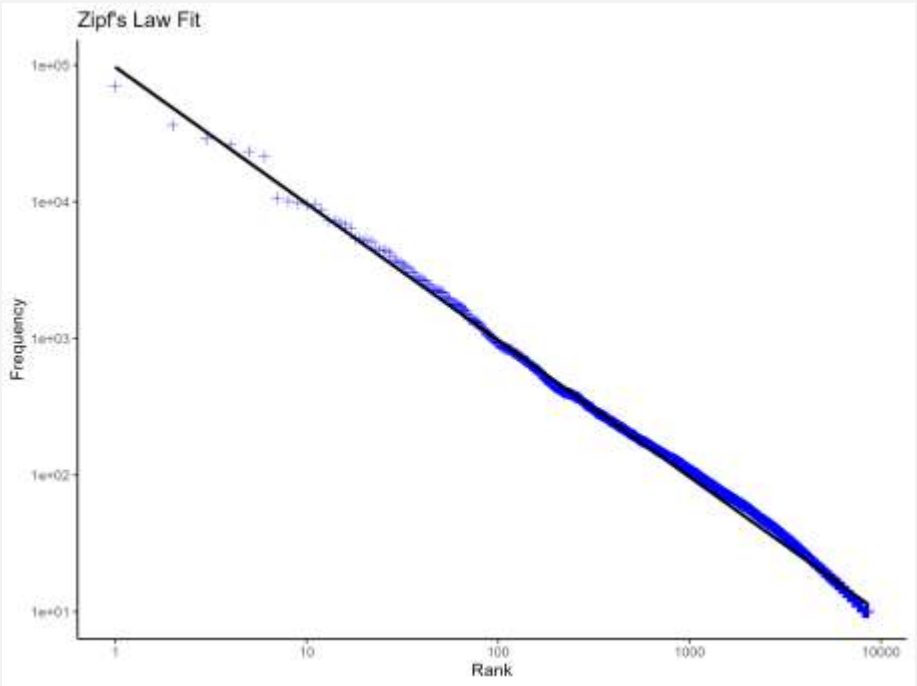
Building a Statistical Text Generation Model

- **Assumption:**
  - Corpus = stationary token sequence
- **Model:**
  - First-order Markov chain with Zipfian marginals
  - Full co-occurrence via **copulas**
- **Key Result:**
  - Sklar’s Theorem: Combines marginals into joint
- **Empirical Choice:**
  - Gaussian copula fits Zipfian data well
    → Enables dense corpus generation for PMI estimation

Empirical vs. Zipfian Word Distributions

- **Data:**
  - Brown Corpus (Francis & Kucera, 1979)
  - Filtered to remove words with <10 occurrences
- **Visual takeaway:**
  - The power-law behavior of word frequencies supports Zipfian assumptions in modeling





Embedding Algorithms: From Word2Vec to Statistical Estimators

- **Key Idea:**
  - Word2Vec ≈ empirical SPMI (from finite corpus)
- **Theoretical Goal:**
  - Factor **population** SPMI → generalizable embeddings
- **Challenge:**
  - Methods must:
    - Approximate Word2Vec
    - Be analyzable
    - Retain task performance
- **Two Strategies:**
  - Truncated SVD
  - Missing Value SVD (MVSVD)

SPPMI: A Truncated Approximation

- **Approach**:
  - Apply SVD to:

  $SPPMI(w, c) = \max(SPMI(w, c), 0)$
- **Why**:
  - Avoids $-\infty$ in the matrix
- **Limitations**:
  - Loses negative associations (semantic info)
  - Weaker performance with more negative samples (Levy & Goldberg, 2014)
- **Conclusion**:
  - Practical, but statistically inefficient for full language modeling

EM-MVSVD: Filling Gaps in the SPMI Matrix

- **Goal:** Factorize sparse SPMI matrices by imputing missing values
- **Procedure:**
  1. **Initial Guess:** Fill missing entries with a placeholder value
  2. **Expectation Step:** Compute truncated $SVD \rightarrow W_c = U\sum V^T$
  3. **Maximization Step:** Replace missing values using SVD reconstruction
  4. **Iterate:** Repeat until convergence using exponential smoothing:

$$W_c^{(t)} = \lambda \cdot W_c + (1-\lambda) \cdot W_c^{(t-1)}$$

- **Limitation:**
  - Optimizes only over **observed values**
  - Ignores structure of **missing entries**, potentially inconsistent with underlying distribution
- **Use Case:** Best suited for data missing **completely at random**—not ideal for PMI matrices

Algorithm 1 EM-MVSVD Algorithm from Kurucz et al. (2007)

Require: W, a matrix with missing values
Require: d, the number of singular values to keep

Initialization

$R \leftarrow \{(i,j) \mid W_{ij} \text{ is missing}\}$
for $(i,j) \in R$ **do**
  $W_{ij} \leftarrow$ initial guess
end for
$U, \Sigma, V^\top \leftarrow \text{SVD}(W, d)$
$\widehat{W}^{(0)} \leftarrow U\Sigma V^\top$
for $(i,j) \in R$ **do**
  $W_{ij} \leftarrow \widehat{W}_{ij}^{(0)}$
end for

E-step

for $t = 1, 2, 3, \ldots$ **do**
  if converged **then**
    return U, Σ, $V^\top$
  end if
  $U, \Sigma, V^\top \leftarrow \text{SVD}(W, d)$
  $\widehat{W} \leftarrow U\Sigma V^\top$

M-step

  $\lambda \leftarrow \arg\min \sum_{(i,j) \notin R} \left[W_{ij} - (\lambda \cdot \widehat{W}_{ij} + (1-\lambda) \cdot \widehat{W}_{ij}^{(t-1)})\right]^2$
  $\widehat{W}^{(t)} \leftarrow \lambda \cdot \widehat{W} + (1-\lambda) \cdot \widehat{W}^{(t-1)}$
  for $(i,j) \in R$ **do**
    $W_{ij} \leftarrow W_{ij}^{(t)}$
  end for
end for

DD-MVSVD: Incorporating Linguistic Structure into Matrix Completion

- **Key Idea:** Leverage **distributional estimates** (from copulas) to guide matrix completion
- **Inputs:**
  - W: Empirical matrix with missing values
  - d: Target rank for SVD
  - $\widehat{W}$: Estimated "true" population SPMI (via Gaussian copula)
- **Procedure Enhancements:**
  - **Objective:** Minimize **Chi-square distance** between reconstructed matrix and $\widehat{W}$, not just observed entries using the normalization when finding lambda
- **Advantages:**
  - Matches structure of **Zipfian-distributed** language data
  - Better alignment with population-level semantics
- **Bottom Line:** Statistically grounded, interpretable alternative to Word2Vec with improved consistency

Algorithm 2 DD-MVSVD Algorithm

Require: W, a matrix with missing values
Require: $\widetilde{W}$, a matrix approximating the "true" matrix from a distribution
Require: d, the number of singular values to keep

Initialization

$R \leftarrow \{(i,j) \mid W_{ij} \text{ is missing}\}$
for $(i,j) \in R$ **do**
  $W_{ij} \leftarrow$ initial guess
end for
$U, \Sigma, V^\top \leftarrow \text{SVD}(W, d)$
$\widehat{W}^{(0)} \leftarrow U\Sigma V^\top$
for $(i,j) \in R$ **do**
  $W_{ij} \leftarrow \widehat{W}^{(0)}_{ij}$
end for

E-step

for $t = 1, 2, 3, \ldots$ **do**
  if converged **then**
    return U, Σ, $V^\top$
  end if
  $U, \Sigma, V^\top \leftarrow \text{SVD}(W, d)$
  $\widehat{W} \leftarrow U\Sigma V^\top$

M-step

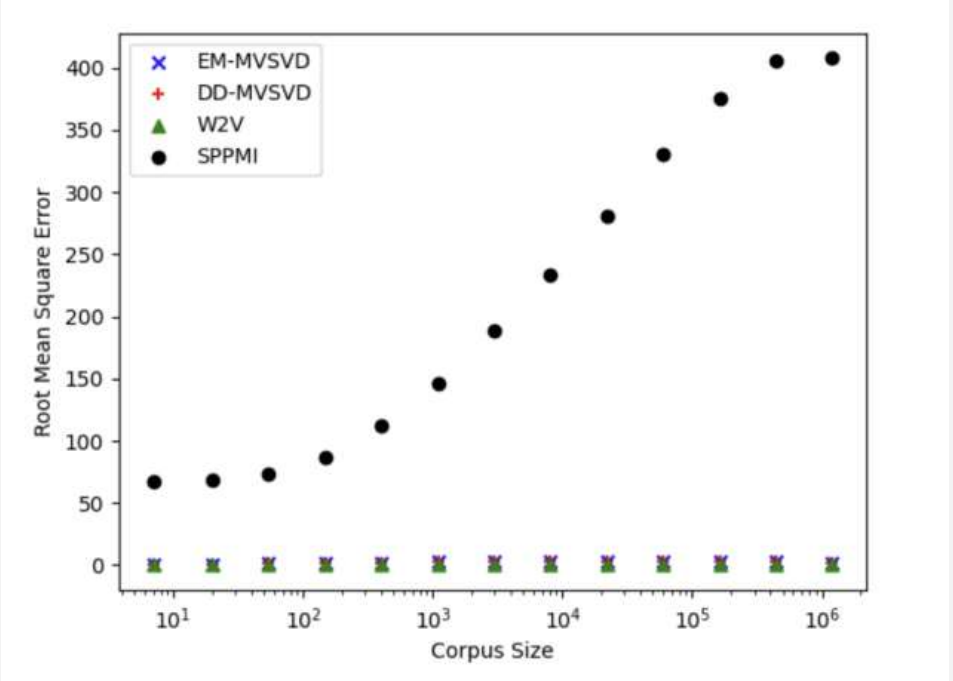
  $\lambda \leftarrow \arg\min \sum_{(i,j) \notin R} \frac{\left[\widetilde{W}_{ij} - (\lambda \cdot \widehat{W} + (1-\lambda) \cdot \widehat{W}^{(t-1)})_{ij}\right]^2}{\widetilde{W}_{ij}}$
  $\widehat{W}^{(t)} \leftarrow \lambda \cdot \widehat{W} + (1-\lambda) \cdot \widehat{W}^{(t-1)}$
  for $(i,j) \in R$ **do**
    $W_{ij} \leftarrow W^{(t)}_{ij}$
  end for
end for

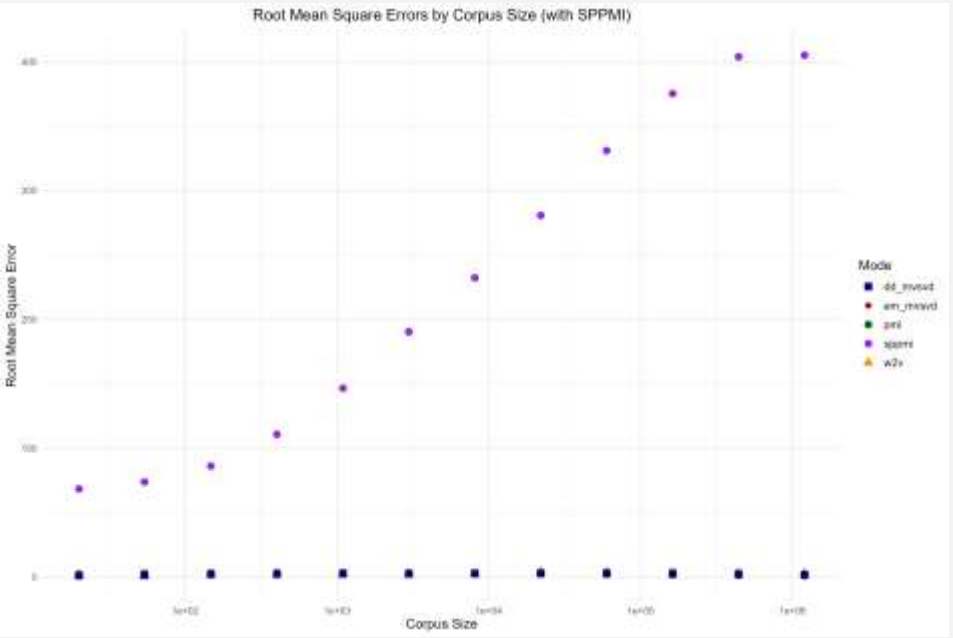
Comparing Methods Against the Population SPMI Matrix

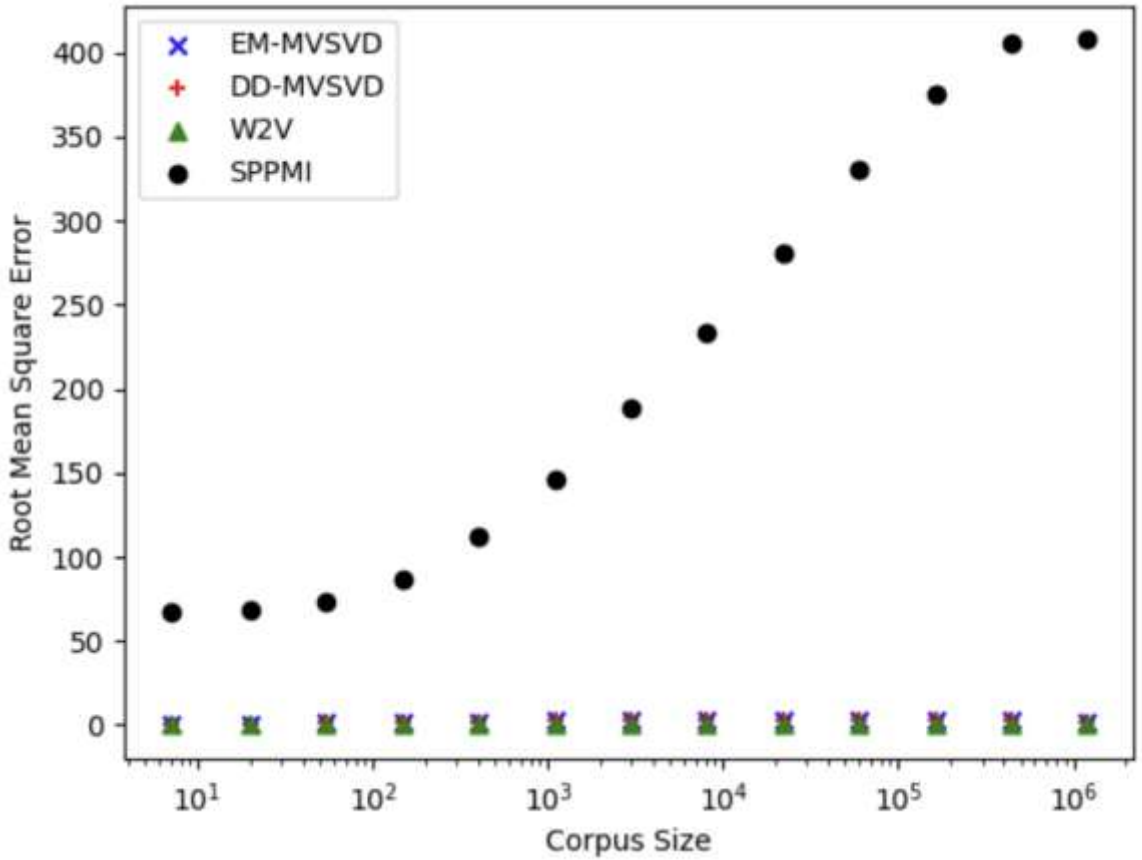
- **Goal**:
  - Compare methods by how well they recover **population SPMI**
- **Setup**:
  - 500 words from Brown Corpus
  - Synthetic text via Gaussian copula
  - True dense SPMI matrix as reference
- **Methods Compared:**
  - Word2Vec
  - Truncated SVD (SPPMI)
  - EM-MVSVD (Alg. 1)
  - DD-MVSVD (Alg. 2)
    (All: 100-dim, 10 negative samples)
- **Metric**:
  - RMSE vs. population SPMI
- **Finding**:
  - **Word2Vec ≈ MVSVD** (low RMSE)
  - **SPPMI SVD** performs worse as data grows

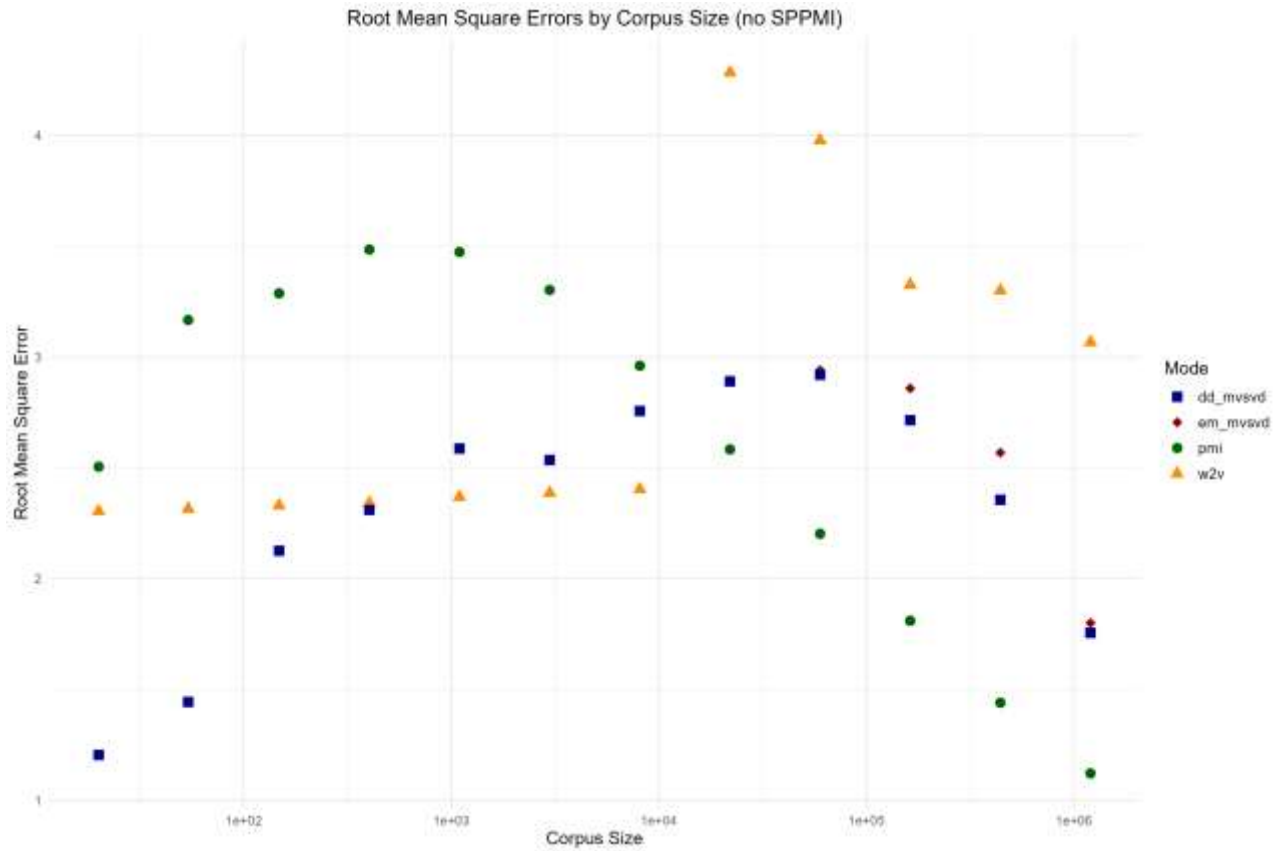
RMSE Performance Across Methods (20 Trials Avg.)

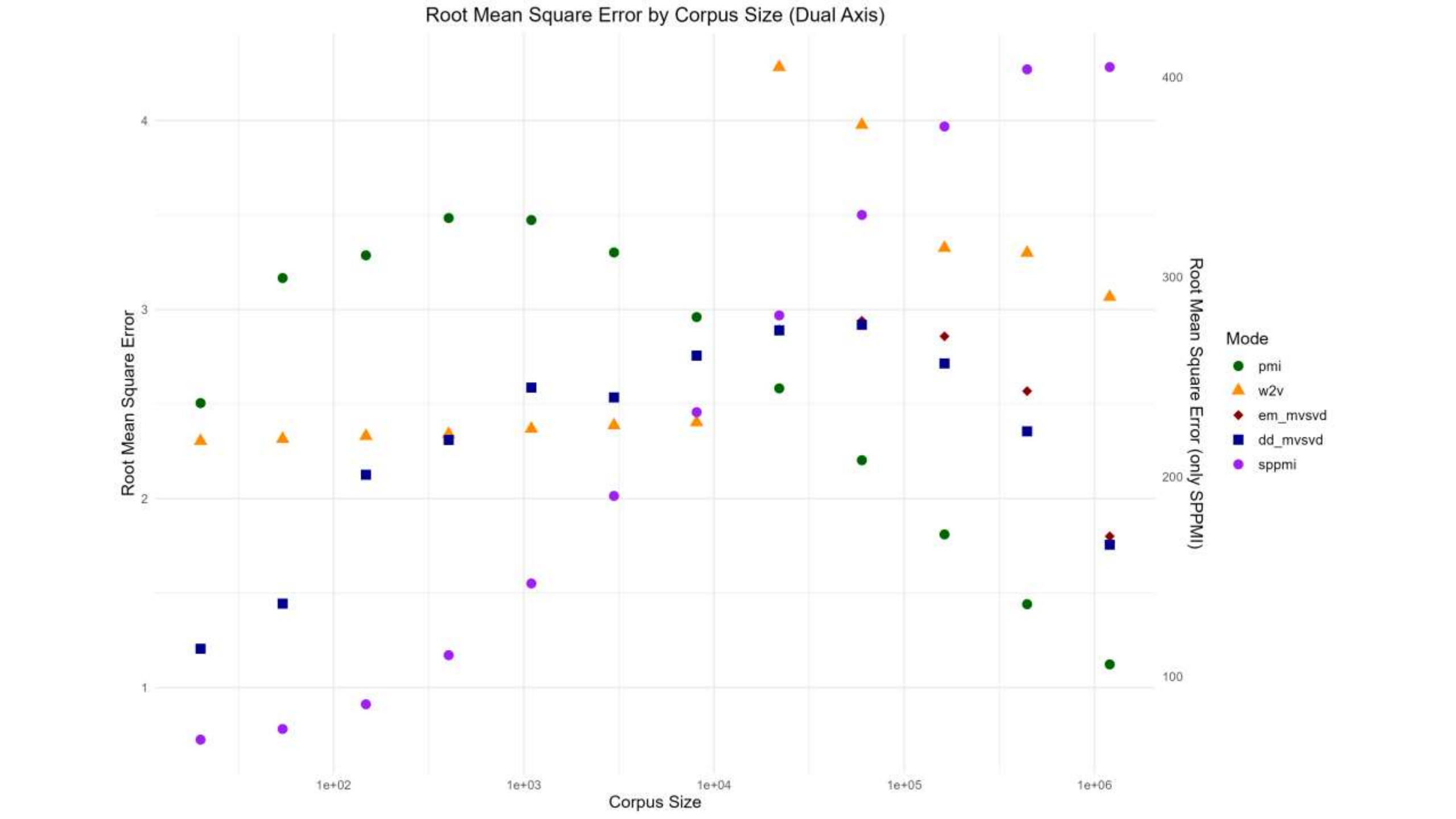
- **Plot:**
  - RMSE vs. corpus size (20 trials per point)
  - Methods: Word2Vec, SVD-SPPMI, EM-MVSVD, DD-MVSVD
  - Std. errors < 0.5 → no visible error bars
- **Interpretation:**
  - **SVD-SPPMI** worsens with more data
  - **MVSVD** stays close to Word2Vec → consistent & reliable
- **Takeaway:**
  - MVSVD = interpretable, statistically grounded, and empirically strong






EM-MVSVD
DD-MVSVD
W2V
SPPMI
Root Mean Square Error
Corpus Size
0
50
100
150
200
250
300
350
400
10^1
10^2
10^3
10^4
10^5
10^6


Root Mean Square Errors by Corpus Size (no SPPMI)
Root Mean Square Error
Corpus Size
Mode
dd_mvsvd
em_mvsvd
pmi
w2v
1
2
3
4
1e+02
1e+03
1e+04
1e+05
1e+06


Root Mean Square Error by Corpus Size (Dual Axis)
Root Mean Square Error
Root Mean Square Error (only SPPMI)
Corpus Size
1e+02
1e+03
1e+04
1e+05
1e+06
1
2
3
4
100
200
300
400
Mode
pmi
w2v
em_mvsvd
dd_mvsvd
sppmi

Thank you for your attention

- Rafael Ignacio Urbina Hincapie
- Alessandro Carella