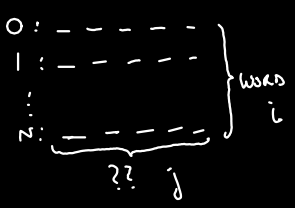
Word Embedding Matrices

X, Z

X: row X_i / Z_i = embedding of ith word in language.

X:



Steps

- normalize the embeddings.
- unsupervised initialization scheme to create initial solution
- self learning procedure to iteratively improve solution
- Final refinement step.

Goal:
- learn linear transformation matrices W_X, W_Z s.t. XW_X, ZW_Z in same space.
- Build a dictionary between both
  - encoded as D, $D_{ij} = 1$ if trans(i, source) = j target

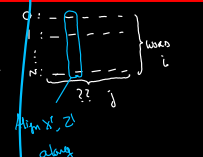
Embedding Normalization

- length normalise embeddings
- mean-center each dimension
- length normalise embeddings.

⇒ dot product ≃ cosine similarity & euclidian distance

Fully unsupervised initialization

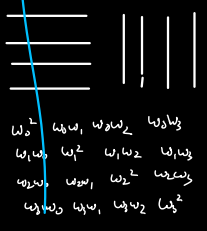
Construct X', Z' s.t ∃ alignment along X'_{*j}, Z'_{*j}



Idea:

$M_X = XX^T$

$M_Z = ZZ^T$ ⇒



Assuming isometry, M_X and M_Z have a matching in some combination of rows, columns.

→ Cannot try all permutations, computational limits.

So:
- Sort each row of $\sqrt{M_X}$, $\sqrt{M_Z}$
- normalize $\underbrace{\text{sorted}(\sqrt{M_X})}_{X'}$, $\underbrace{\text{sorted}(\sqrt{M_Z})}_{Z'}$

Robust self-learning

Algorithm:

- Computes optimal ortho mapping to maximize similarities for dictionary D.

$$\text{argmax} \sum_i \sum_j D_{ij}\left((X_{i*}W_X).(Z_{j*}W_Z)\right)$$

- Computes optimal dictionary over sim. mat. of mapped embeddings. ($XW_XW_Z^TZ^T$)

$$D_{ij} = \begin{cases} 1 \text{ if } j = \text{argmax}_k (X_{i*}W_X).(Z_{k*}W_Z) \\ 0 \text{ otherwise.} \end{cases}$$

Dictionary was created in previous step and so it does not stuck in poor local optima.

Init Dict
- Compute X', Z'

IMPROVEMENTS

- Stochastic dictionary induction
- Freq-vocab cutoff.
- CSLS retrieval
- Bidirectional dictionary induction.

Symmetric Re-weighting

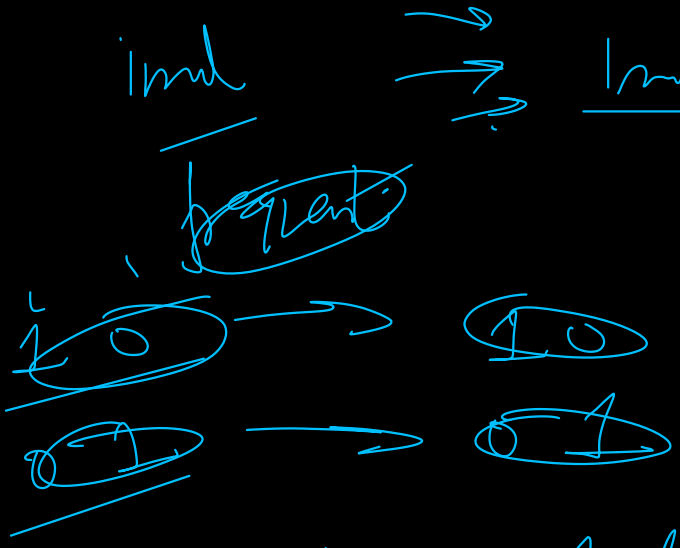
Possible Improvements

1. Most frequent Content Words instead of most freq. words. Will not get embeddings Accurately enough maybe?
2. Solutions of Wahba's problem.

I will get some water from the fridge.

मैं Fridge से पानि ले कर आऊंगा।

> 1ml ⇒ 1ml



XTX → all pairs of dot product.

YTY → all pairs

yy → 0.1
yx → 0.7
xy → 0.3

x.y = 0.3 0.1 = yy

y.y = 0.1 0.3 = xy

0.7 = x.x

y → (y.xi)

yα