Cross-lingual and Multi-lingual Word Embeddings

Md Shad Akhtar, Sukanta Sen and Zishan Ahmad
IIT Patna

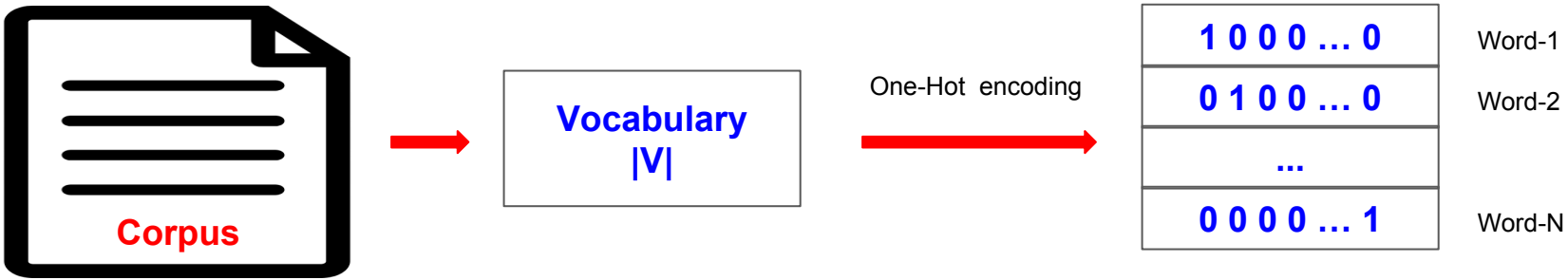
Outline

- Introduction and Objectives
- Supervised
  - Parallel Corpus - Luong et al. 2015
  - Comparable Corpus - Vulić and Moens, 2015
  - Bi-lingual dictionary Induction
    - Faruqui and Dyer 2014
    - Mikolov et al., 2013b
    - Artetxe et al., 2016
  - Almost no Bi-lingual dictionary
    - Artetxe et al., 2017
- Unsupervised
  - Artetxe et al., 2018
  - Conneau et al., 2018

Why do we need word representation?

- Many Machine Learning algorithms does not understand text data, they require input to be numeric. E.g. SVM, NN etc.
- Two types of representations
  - Local Representation
    - One hot
      - Cat = [0,0,0,0,1,0,0,0,0,0]
      - Sparse
      - No semantics
      - Curse of Dimensionality
  - Distributed Representation
    - Word embeddings
      - Cat = [2.4, 1.0,3.1,5.3]
      - Dense
      - Very good at capturing semantic relations.
      - Low Dimensionality

Word2Vec: Preprocessing



Example: The domestic **cat** is a small, furry, carnivorous mammal.

Context window size |C| = 6

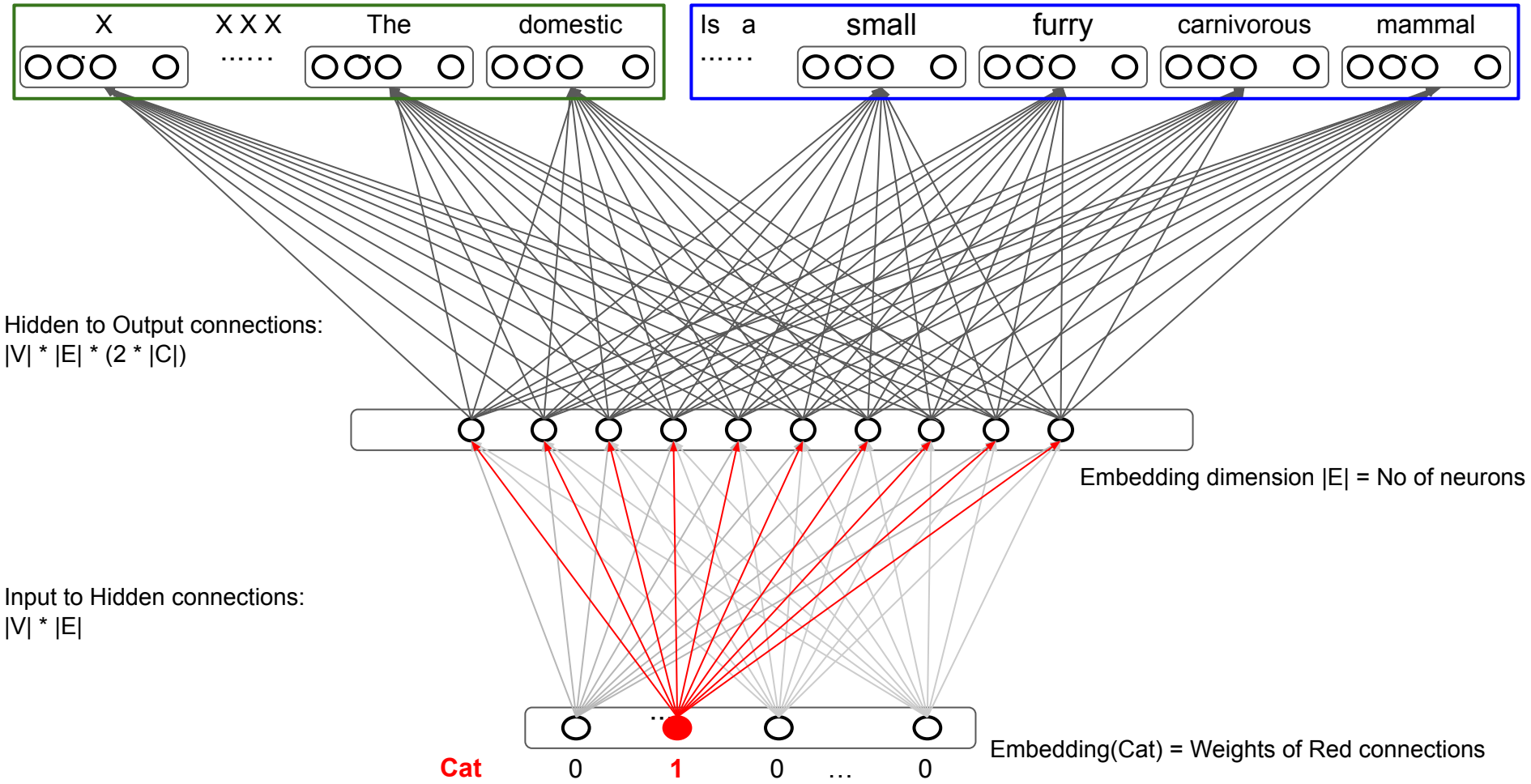
X X X X The domestic **cat** is a small furry carnivorous mammal .

Previous |C| tokens

Next |C| tokens

*Comma is omitted only for illustration convienence.

Skip-Gram (Mikolov et al., 2013a)

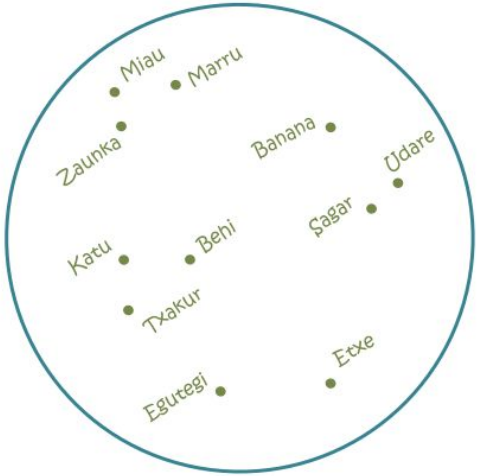


Why do we need Cross-lingual/Bi-lingual Embeddings?

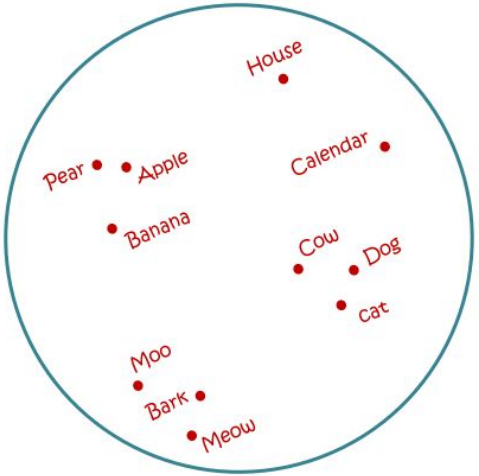
- Bridge the language divergence
- Applications
  - Leverage the resource-richness of one language (e.g., English) in solving a problem in resource-constrained languages (e.g., Hindi, Marathi etc.)
  - Code-mix text

What is Cross-lingual/Multi-lingual embeddings?

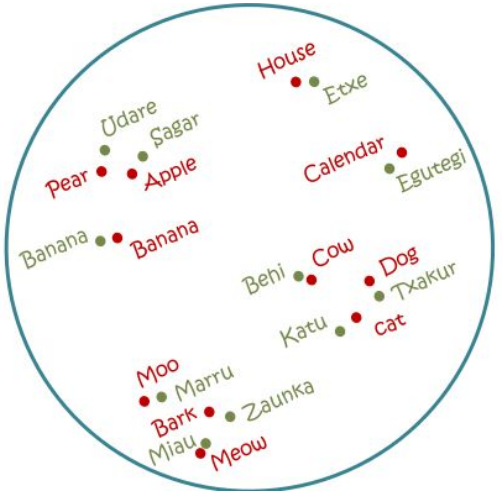
- Embedding of a word in one language (say, Spanish) and embedding of the same word (translated) in other language (say, English) *does not pose any association* between them.
- Therefore, they cannot represent each other in the vector space (i.e., they *cannot correlate*).



Embedding space for Spanish



Embedding space for English



Shared Embedding space for Spanish and English

Supervised Approaches

Bi-lingual Word Embeddings (1)

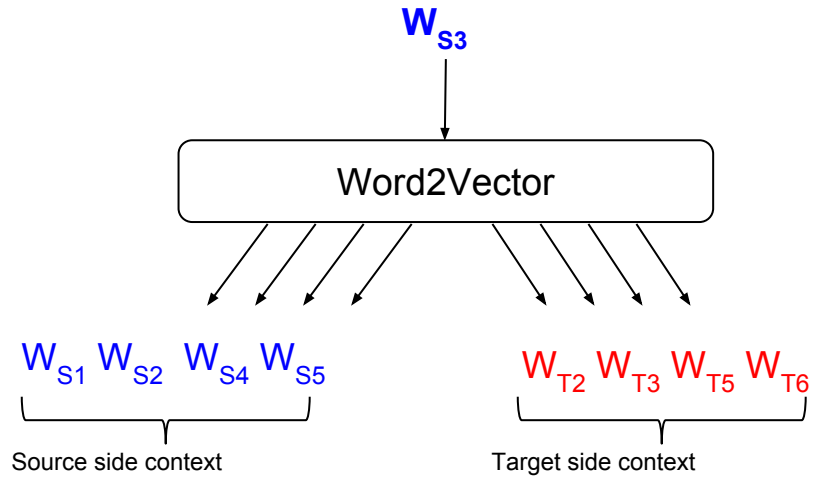
- Luong et al. 2015, Bilingual Word Representations with Monolingual Quality in Mind. In *NAACL Workshop on Vector Space Modeling for NLP*.
    - Bi-lingual word embeddings aims to *bridge the language divergence* in the vector space.
        - Idea is pretty simple
            - Utilize existing word2vec skip-gram model (Mikolov., 2013a)
            - For each word, define its context to include words from both the source and target languages.
        - Requires a *parallel corpus* and alignment information among parallel sentences.

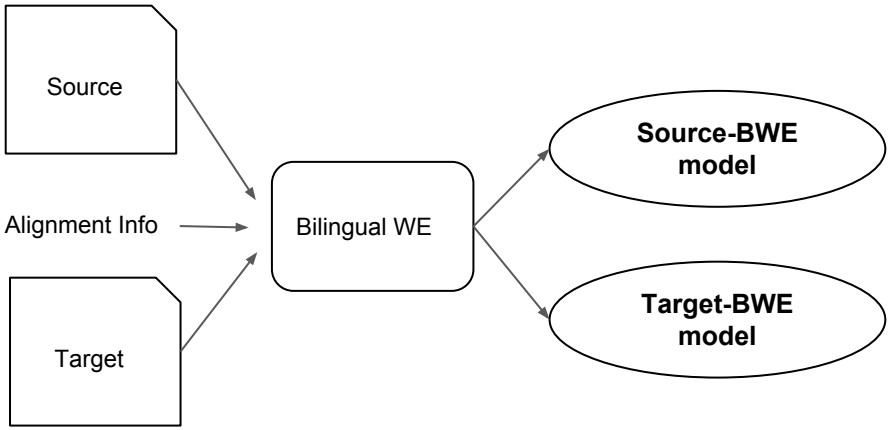
Bi-lingual Skip-gram model

Source: W_{S1} W_{S2} $\mathbf{W_{S3}}$ W_{S4} W_{S5} W_{S6}

Alignment

Target: W_{T1} W_{T2} W_{T3} $\mathbf{W_{T4}}$ W_{T5} W_{T6} W_{T7}





Bi-lingual word embeddings

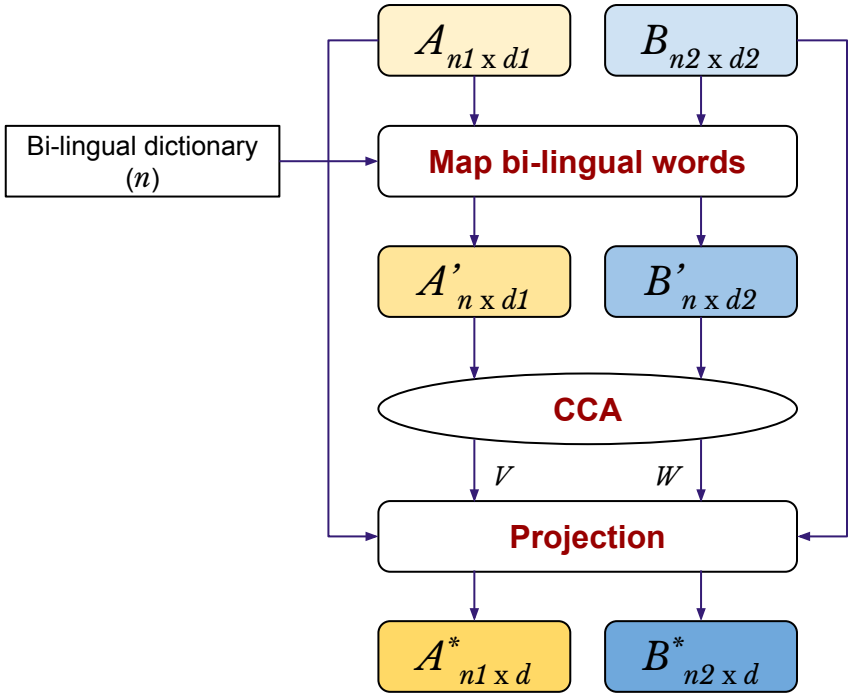
Bi-lingual Word Embeddings (2)

- Ivan Vulić and Marie-Francine Moens. 2015. Bilingual Word Embeddings from Non-Parallel Document-Aligned Data Applied to Bilingual Lexicon Induction. In ACL-IJCNLP.
  - Idea is similar to (Loung et al., 2015)
  - Requires a *comparable corpus* instead of parallel corpus.
  - *No alignment information* is required.
  - Approach
    - Take comparbale documents D_{Source} and D_{Target}
    - Merge D_{Source} and D_{Target} into D'
    - Shuffle words in D'
    - Exceute word2vec skip-gram model

Bi-lingual Word Embeddings (3)

- Manaal Faruqui and Chris Dyer. 2014. Improving Vector Space Word Representations Using Multilingual Correlation. In EACL.
    - Requires *two monologual embeddings* and *bi-lingual dictionary*.
    - Performs *canonical correlation analysis (CCA)* on two embeddings and project these into shared vector space where they are maximally correlated

Bi-lingual Word Embeddings (3)



Let x and y be two corresponding vectors from A' and B', and v and w be two projection directions.

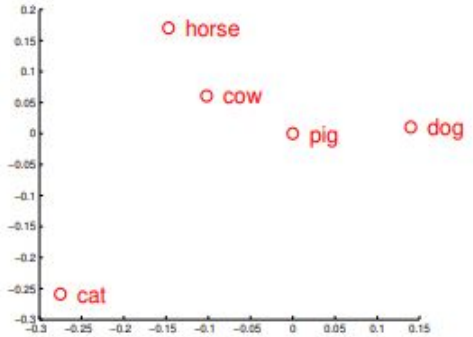
$$v, w = CCA(x, y)$$
$$x' = xv$$
$$y' = yw$$
$$\rho(x', y') = \frac{E[x'y']}{\sqrt{E[x'^2]E[y'^2]}}$$

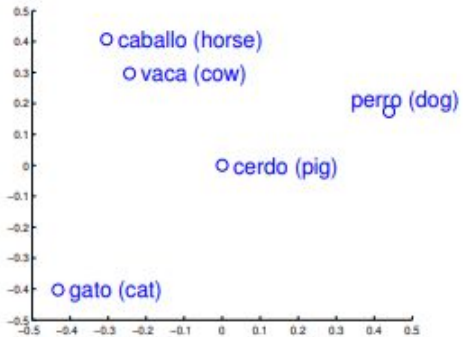
Maximize ρ,

argmax $\rho(xv, yw)$

Bi-lingual Word Embeddings (4)

- Tomas Mikolov, Quoc V. Le, and Ilya Sutskever, 2013. Exploiting Similarities among Languages for Machine Translation. In arXiv:1309.4168v1.
  - Requires
    - *Two monologual embeddings*
    - *Bi-lingual dictionary*.





- Approach
  - Suppose we are given a set of word pairs and their associated vector representations $\{x_i , z_i\}$.
  - Goal is to find a transformation matrix W

$$\min_W \sum_{i=1}^{n} \|Wx_i - z_i\|^2$$

  - For any given new word and its vector representation x, we can compute $z = Wx$.

Normalized word embedding and orthogonal transform for bilingual word translation (Xing et al. 2015)

- Remember, *Exploiting Similarities among Languages for Machine Translation (Mikolov et at. 2013)*
  - Given a set of n word pairs and their vector representations $\{x_i, y_i\}$, where x_i is a d_1 dimensional vector and y_i is a d_2 dimensional vector.
  - Goal is to find W (dimension: $d_2 \times d_1$) such that Wx_i approximates y_i $\min_W ||WX-Y||$

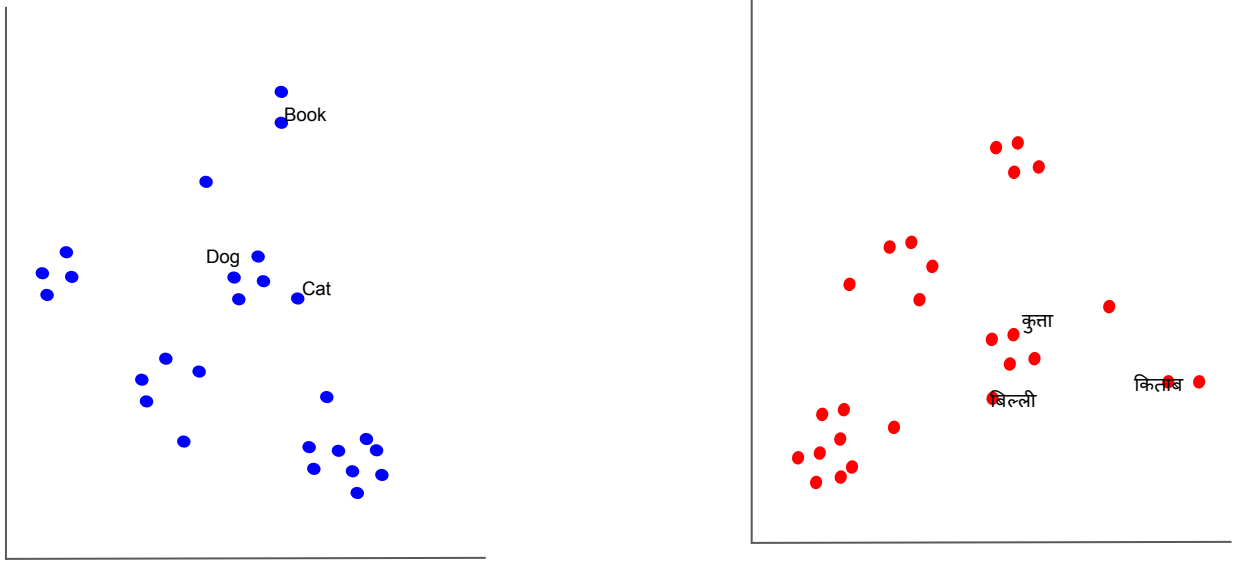
- These results can be improved by enforcing an orthogonality constraint on W

$$WW^T = I$$

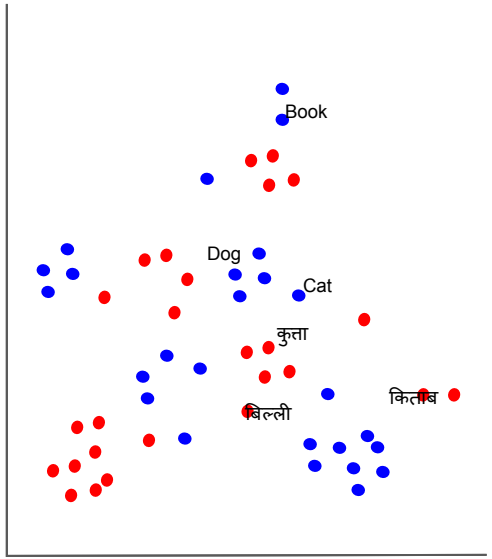
- Which reduced to ***Procrustes problem***.

Why orthogonality is important

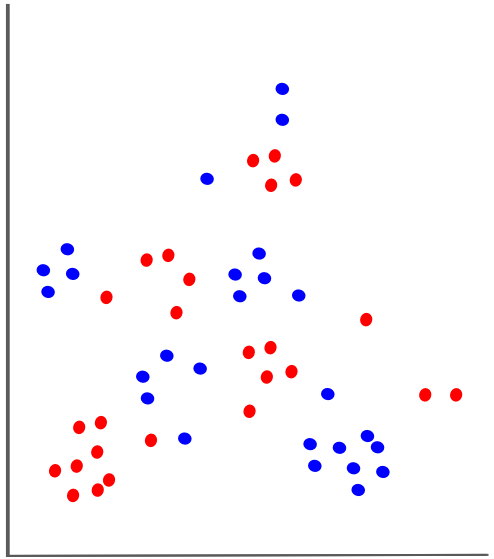
- Orthogonality is important to keep the monolingual property after transformation.
- Orthogonal transformation is length and angle preserving.
- Therefore it is an isometry of the Euclidean space (such as a rotation).



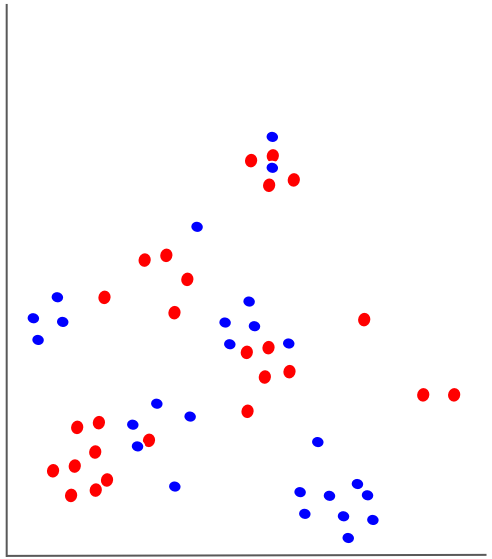
Orthogonal Rotation



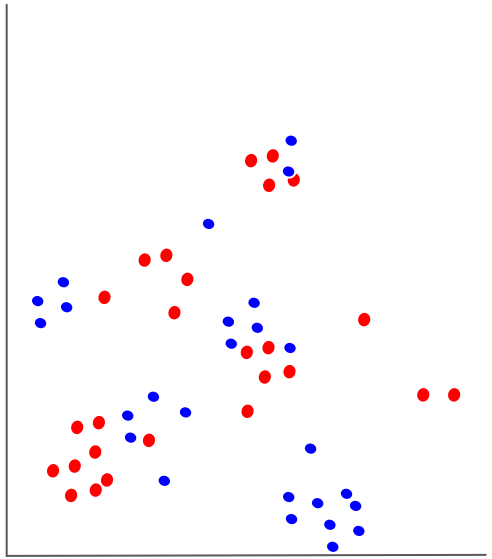
Orthogonal Rotation



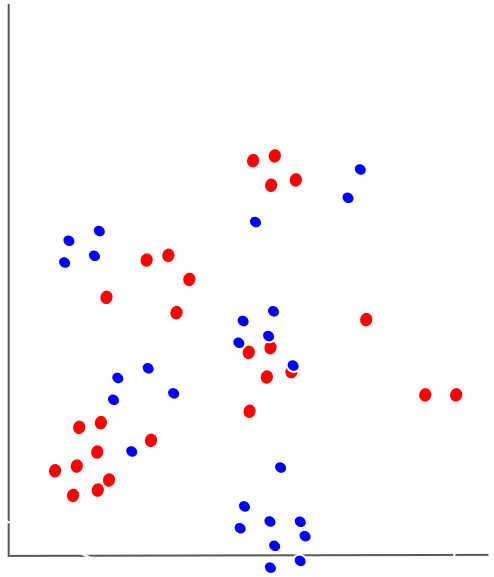
Orthogonal Rotation



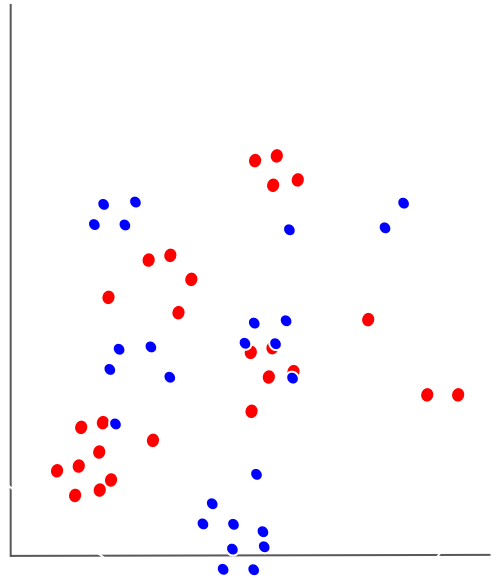
Orthogonal Rotation



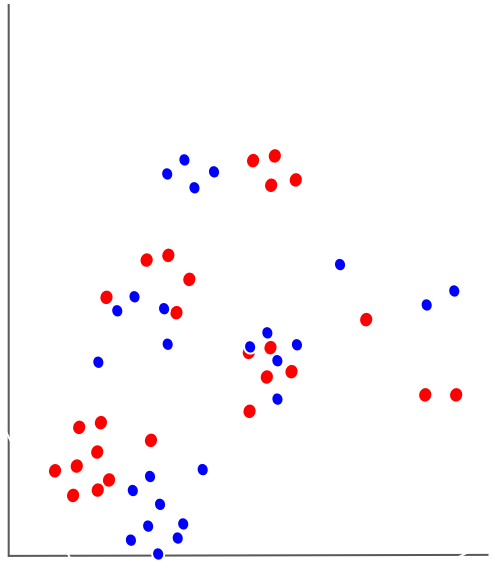
Orthogonal Rotation



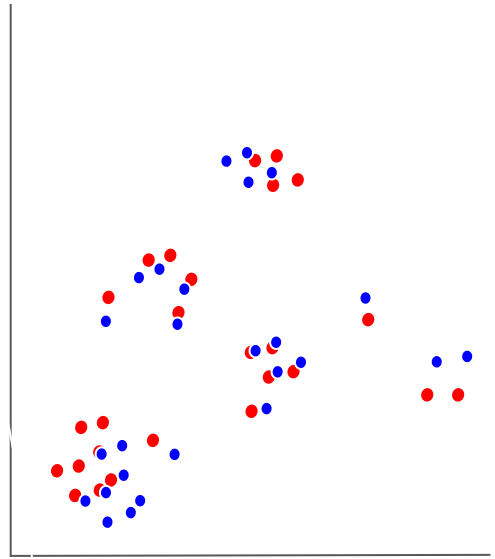
Orthogonal Rotation



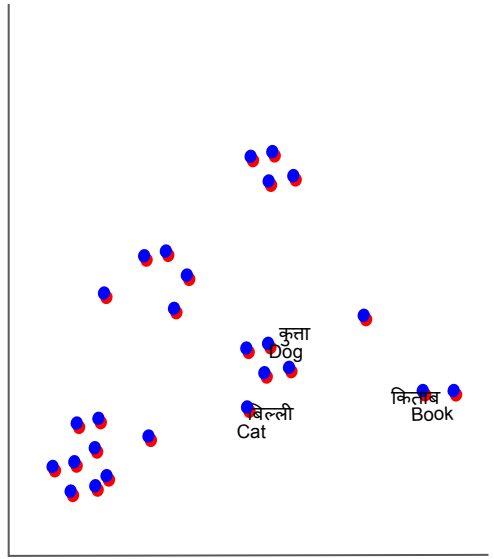
Orthogonal Rotation



Orthogonal Rotation



Orthogonal Rotation



Procrustes problem

- A matrix approximation problem in linear algebra.
- Given two matrices X and Y the problem is to find a orthogonal matrix W which closely maps X to Y

$$\text{argmin}_W \, ||WX\text{-}Y|| \, , \qquad \text{subject to } WW^T = I$$

- This problem was originally solved by *Peter Schönemann* in his thesis (1964).

- Solution:

$$W = UV^T \text{ where } U\Sigma V^T = SVD(YX^T)$$

Bi-lingual dictionary based Cross-lingual embeddings: Artetxe et al. 2016.

This paper improves the mapping between two languages by proposing the following steps:

- Orthogonality for transformation matrix for monolingual invariance
- Length normalization for maximum cosine

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. **Learning principled bilingual mappings of word embeddings while preserving monolingual invariance**. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289-2294

Orthogonality for monolingual invariance

- Monolingual invariance is needed to preserve the dot products after mapping
  - Mikelov et al. (2013) proposed bilingual mapping using dictionary between two languages using the following transformation:

$$\arg\min_{W} \sum_{i} \|X_{i*}W - Z_{i*}\|^2$$

X and Z denote the word embedding matrices in two languages for a given bilingual dictionary so that their ith row X_{i*} and Z_{i*} are the word embeddings of the ith entry in the dictionary

- Problems with the above transformation
  - The euclidean distance between the transformed source embeddings and the target embeddings is reduced, however when fetching the nearest words cosine similarity is generally used
  - The angles between vectors in the source embeddings and transformed source embeddings may not be preserved

Orthogonality for monolingual invariance

- The equation proposed by Mikolov et al. (2013) can be written as follows:

$$\arg\min_{W} \|XW - Z\|_F^2$$

F is the *Frobenius norm*

Orthogonality for monolingual invariance

This W can be obtained by using SVD.

Proof:

$$
\begin{aligned}
R &= \arg\min_{\Omega} ||\Omega A - B||_F^2 \\
&= \arg\min_{\Omega} \langle \Omega A - B, \Omega A - B \rangle \\
&= \arg\min_{\Omega} \|A\|_F^2 + \|B\|_F^2 - 2\langle \Omega A, B \rangle \\
&= \arg\max_{\Omega} \langle \Omega, BA^T \rangle \\
&= \arg\max_{\Omega} \langle \Omega, U\Sigma V^T \rangle \\
&= \arg\max_{\Omega} \langle U^T \Omega V, \Sigma \rangle \\
&= \arg\max_{\Omega} \langle S, \Sigma \rangle \quad \text{where } S = U^T \Omega V
\end{aligned}
$$

Here S is orthonormal matrix and is maximized when *S=I*

$$
\begin{aligned}
I &= U^T R V \\
R &= U V^T
\end{aligned}
$$

Orthogonality for monolingual invariance

Thus the solution for W is $W = VU^T$, where $Z^TX = U\Sigma V^T$ is the SVD factorization of Z^TX

- V and U^T are orthogonal matrices, thus making W orthogonal
- Thus transforming the original embedding with W will maintain the inner angles and distances of the source embedding
- SVD is done in linear time

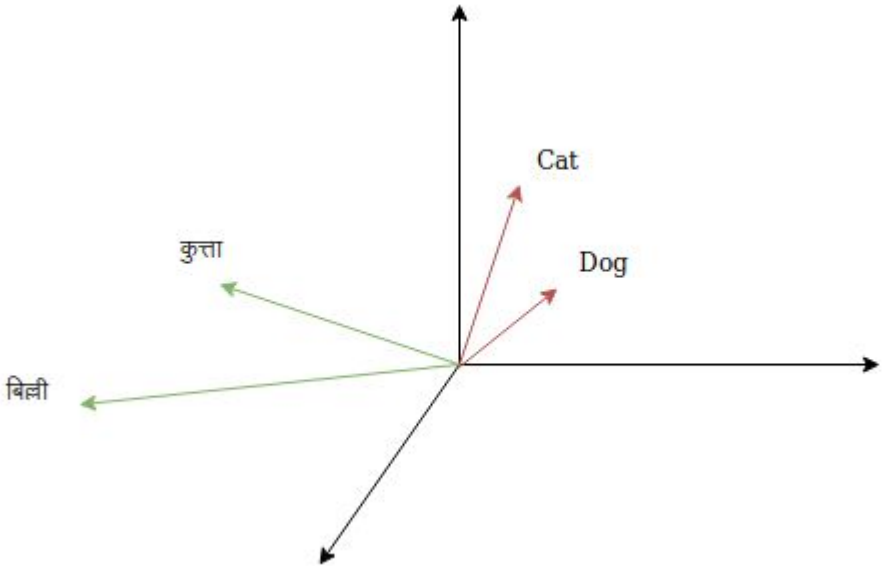
Length normalization for maximum cosine

- Using $W = VU^T$ reduces the transformation W only to a set of rotations V and U^T
- It converts the embedding space into a hyper-sphere
- Length normalization reduces our products to cosine similarity:

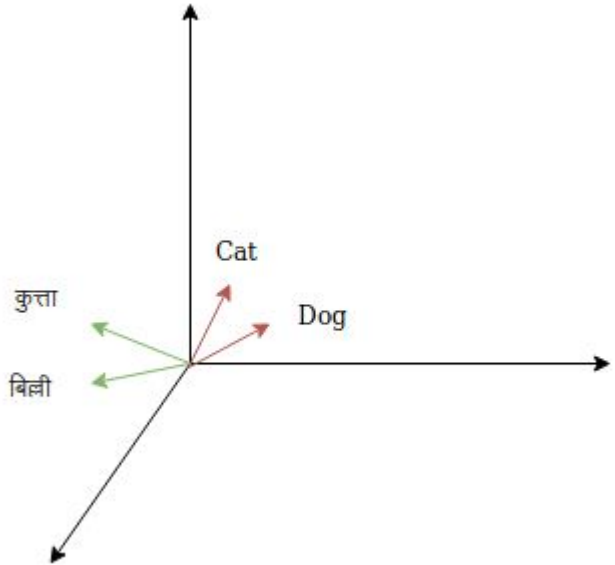
$$\arg\min_W \sum_i \left\| \frac{X_{i*}}{\|X_{i*}\|} W - \frac{Z_{i*}}{\|Z_{i*}\|} \right\|^2$$
$$= \arg\max_W \sum_i \cos\left(X_{i*}W, Z_{i*}\right)$$

Length normalization for maximum cosine

Visualization:



Unnormalized Embedding space



Normalized Embedding space

Almost no bilingual dictionary based Bi-lingual embeddings: Artetxe et al.[2]

- This paper shows that a good transformation of source embeddings can take place even with as little as 25 parallel words in the seed dictionary.
- An iterative self induction method for dictionary initialization is proposed in this paper

[2]Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. **Learning bilingual word embeddings with (almost) no bilingual data**. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451-462.

Traditional framework v/s Proposed self learning framework

Algorithm 1 Traditional framework

Input: X (source embeddings)
Input: Z (target embeddings)
Input: D (seed dictionary)
1: $W \leftarrow$ LEARN_MAPPING(X, Z, D)
2: $D \leftarrow$ LEARN_DICTIONARY(X, Z, W)
3: EVALUATE_DICTIONARY(D)

Algorithm 2 Proposed self-learning framework

Input: X (source embeddings)
Input: Z (target embeddings)
Input: D (seed dictionary)
1: **repeat**
2: $W \leftarrow$ LEARN_MAPPING(X, Z, D)
3: $D \leftarrow$ LEARN_DICTIONARY(X, Z, W)
4: **until** convergence criterion
5: EVALUATE_DICTIONARY(D)

Formulization

$$W^* = \arg\min_W \sum_i \sum_j D_{ij} ||X_{i*}W - Z_{j*}||^2$$

D is the dictionary matrix such that D_{ij}=1 if the i^{th} source language word is aligned with the j^{th} target language word
X_i is the word embedding of the ith word in the source embeddings matrix X
Z_j is the word embedding of the jth word in the target embedding matrix Z
W is the transformation matrix to be optimized

The above formula can also be written as:

$$W^* = \arg\max_W \mathrm{Tr}\left(XWZ^T D^T\right)$$

$W^* = UV^T$ Where $X^TDZ = U\Sigma V^T$ is the SVD of X^TDZ

Dictionary Inductoin

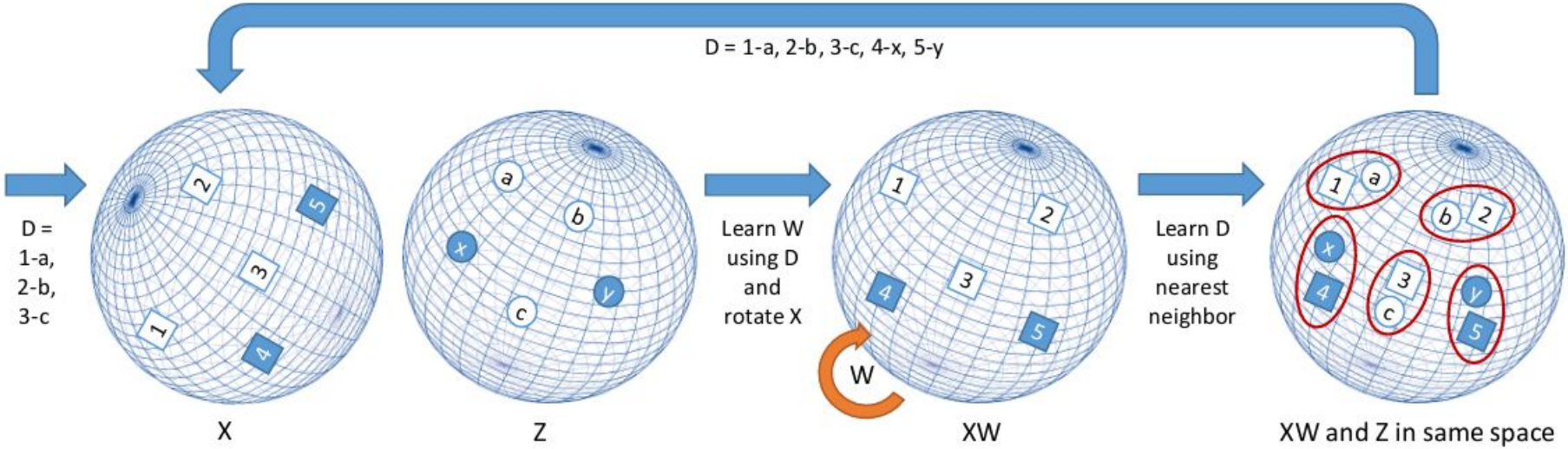
Assumption:

- The dictionary induced by training on the seed dictionary is better (at least in the sense that it is larger) than the initial seed dictionary

Steps:

- Length normalize and perform the 1st iteration of learning using the initial seed dictionary D
- Perform dictionary induction by the following steps:
  - $D_{ij} = 1$ if $j = \text{argmax}_k(X_{i*}W).Z_{k*}$
  - else $D_{ij} = 0$
  - Above steps are equivalent to taking cosine similarity and choosing the word pairs with maximum similarity

Visualization



Unsupervised Approaches

Unsupervised Cross-lingual embeddings: Artetxe et al.[3]

Assumption: The languages share an isometry between words, meaning that distribution of words in different languages is approximately the same

Intuitions behind the assumption:

- If we assume that embeddings of one language can be mapped with embeddings of other language, then we are indirectly assuming isometry

[3]Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. **A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings**. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*

Proposed Method

The proposed method consists of three sequential steps:

- A preprocessing step that normalizes the monolingual embeddings
- A fully unsupervised initialization scheme that creates an initial dictionary
- A robust self learning procedure that computes the weight matrix for mapping the monolingual embedding of a language to shared vector space

Unsupervised Dictionary Induction

Steps:

- Compute the similarity matrices for the word embedding matrix of both the languages:
  - $M_X = XX^T$ and $M_z = ZZ^T$ where X and Z are word embedding matrices of the two languages
  - Under the strict isometry condition, equivalent words would get the exact same vector across M_X and M_Z under some permutation
  - Sorting is used as permutation and the values in each row of M_X and M_z is sorted resulting in matrices sorted(M_x) and sorted(M_z)
  - Given a word and its row in sorted(M_x), apply nearest neighbour retrieval over the rows of sorted(M_z) to find its corresponding translation.
- Using these nearest neighbours, build a dictionary D between both languages encoded as a sparse matrix D where $D_{ij} = 1$ if the j_{th} word in the target language is the translation of the i_{th} word in the source language

Robust self-learning

The training iterates through the following two steps until convergence:

Steps:

- Compute the optimal orthogonal mapping maximizing the similarities for the current dictionary D:

$$\underset{W_X, W_Z}{\arg\max} \sum_i \sum_j D_{ij}((X_{i*}W_X) \cdot (Z_{j*}W_Z))$$

Where W_X and W_Z are linear transformation matrices that will map X and Z in the same vector space. An optimal solution is obtained by $W_X = U$ and $W_Z = V$, where $USV^T = X^TDZ$ is the singular value decomposition of X^TDZ

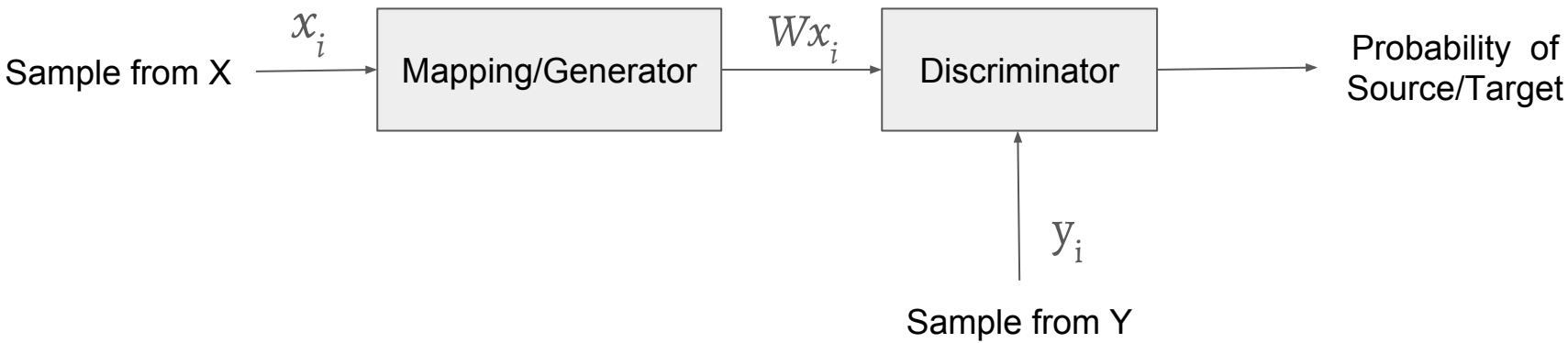
- Compute the optimal dictionary over the similarity matrix of the mapped embeddings $XW_XW_Z^TZ^T$. Using nearest neighbour retrieval update the dictionary D. So $D_{ij} = 1$ if $j = \text{argmaxk}(X_{i*}W_X).(Z_{j*}W_Z)$ and $D_{ij} = 0$ otherwise

Word translation without parallel data (Conneau et al. 2018)

- Proposed complete unsupervised approach to cross-lingual mapping:
- Basic steps:
  - Learn W from domain adversarial training
  - Use W to induce initial bilingual dictionary $X, Y = \{x_i, y_i\}^n_{i=1}$ using CSLS (Cross-domain Similarity Local Scaling) metric
  - Iteratively update, applying
    - $W = UV^T$ where $U\Sigma V^T = SVD(YX^T)$
    - And finding new $X, Y = \{x_i, z_i\}^n_{i=1}$ using CSLS metric

Adversarial training

- Let $X = \{x_1, x_2, x_3, ..., x_n\}$ and $Y = \{y_1, y_2, y_3, ..., y_m\}$ be two sets of n and m word embeddings coming from a source and a target language respectively.
- A model is trained to discriminate between elements randomly sampled from $WX = \{Wx_1, Wx_2, ..., Wx_n\}$ and Y



Adversarial training: Loss function

- Discriminator:

$$\mathcal{L}_D(\theta_D|W) = -\frac{1}{n}\sum_{i=1}^{n}\log P_{\theta_D}\left(\text{source} = 1\middle|Wx_i\right) - \frac{1}{m}\sum_{i=1}^{m}\log P_{\theta_D}\left(\text{source} = 0\middle|y_i\right).$$

- Mapping:

$$\mathcal{L}_W(W|\theta_D) = -\frac{1}{n}\sum_{i=1}^{n}\log P_{\theta_D}\left(\text{source} = 0\middle|Wx_i\right) - \frac{1}{m}\sum_{i=1}^{m}\log P_{\theta_D}\left(\text{source} = 1\middle|y_i\right)$$

Training

- Mapping:
  - A Feed-Forward network with d_1 is input dimension and d_2 as output dimension.
    - W is initialized to as a diagonal matrix with diagonal elements being 1. When $d_1 = d_2$, W is Identity matrix

- Discriminator:
  - A Feed-Forward network with two hidden layers of size 2048, and *Leaky-ReLU* activation functions
    - After each iteration W (weight matrix in mapping)is updated as:

$$W \leftarrow (1+\beta) W - \beta (WW^T)W$$

Cross-domain Similarity Local Scaling

- Initial bilingual dictionary $X, Y = \{x_i, y_i\}^n_{i=1}$
  - Nearest neighbour
- Nearest neighbour suffers from *hubness* problem*:*
- Some vectors, dubbed hubs, are with high probability nearest neighbors of many other points, while others (anti-hubs) are not nearest neighbors of any point.
- **Solution:** CSLC

$$\mathrm{CSLS}(Wx_s, y_t) = 2\cos(Wx_s, y_t) - r_{\mathrm{T}}(Wx_s) - r_{\mathrm{S}}(y_t)$$

$$r_{\mathrm{T}}(Wx_s) = \frac{1}{K} \sum_{y_t \in \mathcal{N}_{\mathrm{T}}(Wx_s)} \cos(Wx_s, y_t)$$

  - $r_T(Wx_s)$ is the mean similarity of a source embedding x_s to its K target neighborhood
  - $r_s(y_t)$ is the mean similarity of a target embedding y_t to its K source neighborhood

References

- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. **Efficient Estimation of Word Representations in Vector Space**. In *arXiv preprint arXiv:1301.3781*
- Tomas Mikolov, Quoc V. Le, and Ilya Sutskever, 2013b. **Exploiting Similarities among Languages for Machine Translation**. In *arXiv preprint arXiv:1309.4168v1*
- Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. **Bilingual Word Representations with Monolingual Quality in Mind**. In *NAACL Workshop on Vector Space Modeling for NLP*. Denver, United States, pages 151–159.
- Ivan Vulić and Marie-Francine Moens. 2015. **Bilingual Word Embeddings from Non-Parallel Document-Aligned Data Applied to Bilingual Lexicon Induction**. In *53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, Beijing, China, 719–725.
- Manaal Faruqui and Chris Dyer. 2014. **Improving Vector Space Word Representations Using Multilingual Correlation**. In *14th Conference of the European Chapter of the Association for Computational Linguistics.* Association for Computational Linguistics, Gothenburg, Sweden, 462–471.
- *Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016.* ***Learning principled bilingual mappings of word embeddings while preserving monolingual invariance****. In 2016 Conference on Empirical Methods in Natural Language Processing, pages 2289-2294*
- *Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017.* ***Learning bilingual word embeddings with (almost) no bilingual data****. In 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 451-462.*
- *Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018.* ***A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings****. In 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*
- Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. 2015. **Normalized word embedding and orthogonal transform for bilingual word translation**. In *Proceedings of NAACL*.
- Peter H Schonemann. 1966. **A generalized solution of the orthogonal procrustes problem**. *Psychometrika*, 31(1):1–10.
- Conneau, Alexis, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. **Word translation without parallel data."** In *arXiv preprint arXiv:1710.04087*.