Improving Supervised Bilingual Mapping of Word Embeddings

Written by: Armand Joulin, Piotr Bojanowski, Tomas Mikolov, and Edouard Grave

Presented by: Ehsan Amjadian @RBC

Summary

- Objective:
  - Improving the alignment of continuous word representations from different languages
- Method:
  - Leveraging a small bilingual lexicon to learn linear transformation
  - Uses retrieval criterion as the loss function (as opposed to square loss)
- Contributions:
  - Proposing a new convex objective function
  - Avoids the hubness problem during retrieval
  - Achieves state of the art results in word translation task
  - Shows the orthogonal mapping constraint does not improve the quality of translations

Intro: Idea

- Possible to learn word translation by linear mapping from one vector space to the other (same d) (Mikolov et al., 2013)
- A small bilingual lexicon is used as supervision
- A regression problem
- Learnt transformation generalizes well to unseen words

Intro: Applications

- Transferring predictive models
- $\text{Model}_{\text{lang\_A}} \rightarrow \text{Model}_{\text{lang\_B}}$
- Sentiment analysis
- Spam detection etc.

Intro: Some background

- Square loss is sub-optimal → hubness problem
- Instead classification or retrieval criteria can be used e.g., Cross-domain Similarity Local Scaling (CSLS)
- Other methods to improve the results:
  - Semi-supervised: *refinement procedure*
  - Weak supervision: string matches between vocabs as additional examples
- More to come on all of the above.

Intro: Main Contribution

- A new convex objective function
- Can be minimized by the projected subgradient method

Task

- Learning bilingual lexicon given:
  - Monolingual vectors
  - A set of pairs of words (seeds)
- Estimate mapping of the words in the different languages
- Infer word translations for non-seeds

Goal: Learn Linear Mapping Between Seeds

$\mathbf{W} \in \mathbb{R}^{d \times d}$	Linear mapping
$i \in \{1, \ldots, N\}$	Entire vocab
$\mathbf{x}_i \in \mathbb{R}^d$	Vector in source
$\mathbf{y}_i \in \mathbb{R}^d$	Vector in target
$(\mathbf{x}_i, \mathbf{y}_i)_{i \in \{1, \ldots, n\}}$	Seeds
$i \in \{n + 1, \ldots, N\}$	Unpaired

Goal: Learn Linear Mapping Between Seeds

$$\min_{\mathbf{W} \in \mathbb{R}^{d \times d}} \quad \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{W}\mathbf{x}_i, \mathbf{y}_i), \qquad (1)$$

$$\ell_2(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2^2$$

→ Linear Least Squares Problem → can be solved in Closed Form

Orthogonal Constraint Improvement

- L2 normalized vectors and orthogonal constraint are believed to improve the results
- $W^TW = I_d$
- Believed to preserve word vector distances hence word similarities
- Method to solve regression then: orthogonal Procrustes analysis
- These norms may discard critical information

Inference

$$t(i) \in \underset{j \in \{1,\ldots,N\}}{\arg\min} \ \ell(\mathbf{W}\mathbf{x}_i, \mathbf{y}_j). \tag{2}$$

Nearest Neighbor Search

$$t(i) \in \underset{j \in \{1,\ldots,N\}}{\arg\min} \ \|\mathbf{W}\mathbf{x}_i - \mathbf{y}_j\|_2^2. \tag{3}$$

The hubness problem with nearest neighbor search

- Hubs:
  - words that appear to frequently in the neighborhood of other words
- Antihubs:
  - words that are not nearest neighbors of any points
- Solutions:
  - Inverted Softmax (ISF, Smith et al., 2017)
  - Cross-domain Similarity Local Scaling (CSLS, Conneau et al., 2017)
- Problem:
  - Only for inference, that is, transformation’s loss stays the same for both ISF & CSLS

Resolving the discrepancy in loss

- Directly optimize CSLS in (1)
  - Coherent learning and inference criteria
  - W (translation model) directly learnt

CSLS

- $$\mathrm{CSLS}(\mathbf{x},\mathbf{y}) = -2\cos(\mathbf{x},\mathbf{y}) + \frac{1}{k}\sum_{\mathbf{y}'\in\mathcal{N}_Y(\mathbf{x})}\cos(\mathbf{x},\mathbf{y}') + \frac{1}{k}\sum_{\mathbf{x}'\in\mathcal{N}_X(\mathbf{y})}\cos(\mathbf{x}',\mathbf{y}),$$
  - Assumptions: W an orthogonal matrix
  - $||x_i||_2 = 1$, $||y_i||_2 = 1$
  - $\cos(\mathbf{W}\mathbf{x}_i, \mathbf{y}_i) = \mathbf{x}_i^\top \mathbf{W}^\top \mathbf{y}_i,$
- $$\min_{\mathbf{W}\in\mathcal{O}_d} \frac{1}{n}\sum_{i=1}^{n} -2\mathbf{x}_i^\top \mathbf{W}^\top \mathbf{y}_i + \frac{1}{k}\sum_{\mathbf{y}_j\in\mathcal{N}_Y(\mathbf{W}\mathbf{x}_i)} \mathbf{x}_i^\top \mathbf{W}^\top \mathbf{y}_j + \frac{1}{k}\sum_{\mathbf{W}\mathbf{x}_j\in\mathcal{N}_X(\mathbf{y}_i)} \mathbf{x}_j^\top \mathbf{W}^\top \mathbf{y}_i. \quad (4)$$

Optimization

- So far minimizing a non-smooth cost over the manifold of orthogonal matrices $O_{d.}$ → *one solution: manifold optimization (computationally demanding)*
- *Alternatives: convex relaxation*

2 relaxations of O_d

- 1. Replace the set O_d *by its convex hull* C_d*: matrices with singular values < 1 (unit ball of the spectral norm)*

- 2. The ball of radius $\sqrt{d}$ in Frobenius norm denoted by β_d

A brief proof

- Since we're dealing with a *convex* domain

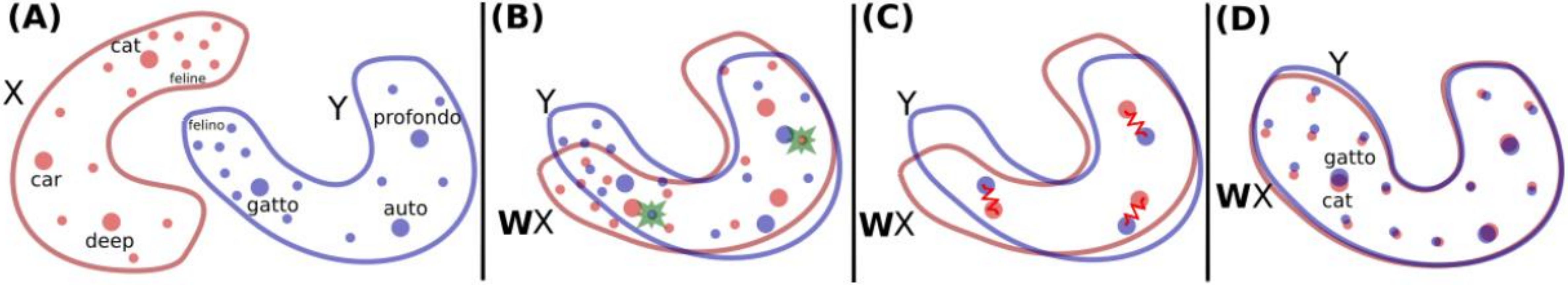
$$\sum_{\mathbf{y}_j \in \mathcal{N}_k(\mathbf{W}\mathbf{x}_i)} \mathbf{x}_i^\top \mathbf{W}^\top \mathbf{y}_j = \max_{S \in \mathcal{S}_k(n)} \sum_{j \in S} \mathbf{x}_i^\top \mathbf{W}^\top \mathbf{y}_j,$$

- *$S_k(n)$: set of all subsets of {1, … , n} of size k.*
- *Leads to the max of linear functions of W which is also convex.*
- → CSLS convex wrt W and is a piecewise linear function
- Projected subgradient method (descent :/) is used to minimize over C_d and β_d

Projections

- On the set C_d taking matrix SVD then and thresholding the singular values to 1.

- On β_d dividing the matrix by its Frobenius norm.



An Illustration from Conneau et al. @FB

Refinement Procedure

- After one iteration: augment the training lexicon by the best inferred translation (by W) then train W_{t+1}
- Worth emphasizing that previous methods used square loss to learn W and CSLS for inference. [divergence risk, no convergence guaranteed]
- The proposed directly optimizes CSLS loss and KNN leverages all the unlabeled as opposed to only labeled lexicon that is:
  - $\{y_1, \ldots, y_N\}$ instead of $\{y_1, \ldots, y_n\}$

Experiments

- Details for all language pairs:
  - Epochs = 10
  - Learning rate in {1, 10, 25, 50} divided by 2 when loss doesn’t decrease
  - Parameters selected using a validation set
  - All word vectors are L_2 unit normalized.
  - K = 10

Experiment 1

Method	en-es	es-en	en-fr	fr-en	en-de	de-en	en-ru	ru-en	en-zh	zh-en	avg.
Adversarial + refine	81.7	83.3	82.3	82.1	74.0	72.2	44.0	59.1	32.5	31.4	64.3
ICP + refine	82.2	83.8	82.5	82.5	74.8	73.1	46.3	61.6	-	-	-
Procrustes	81.4	82.9	81.1	82.4	73.5	72.4	51.7	63.7	42.7	36.7	66.8
Procrustes + refine	82.4	83.9	82.3	83.2	75.3	73.2	50.1	63.5	40.3	35.5	66.9
CSLS (spectral)	83.0	84.9	82.7	**84.1**	78.2	75.8	56.4	66.3	44.4	**45.6**	70.1
CSLS (Frobenius)	**84.5**	**86.4**	**83.1**	**84.1**	**79.1**	**75.9**	**57.0**	**67.1**	**44.6**	41.9	**70.4**

- Refinement minimally helps or damages orthogonal Procrustes.
- Preserving word vector distance seems not essential to word translation.

Experiment 2: Impact of extended normalization

- Uses only C_d Relaxation

	Full	Seeds
en-es	83.0	80.7
es-en	84.9	83.9
en-fr	82.7	81.7
fr-en	84.1	83.2
en-de	78.2	75.1
de-en	75.8	72.1
en-ru	56.4	51.1
ru-en	66.3	63.8
avg.	76.4	74.0

Comparison to the State of the Art

	en-it	it-en
Adversarial + refine + CSLS	45.1	38.3
Mikolov et al. (2013)	33.8	24.9
Dinu et al. (2014)	38.5	24.6
Artetxe et al. (2016)	39.7	33.8
Smith et al. (2017)	43.1	38.0
Procrustes + CSLS	44.9	**38.5**
CSLS (spectral)	45.3	37.9

- Word vectors learned on WaCky datasets (Baroni et al. 2009)
- Epochs: chosen from {1, 2, 5, 10} based on validation set performance
- → state of the art on en-it and comparable performance on it-en

Conclusion

- Retrieval Criterion Instead of Square Loss improves the supervised learning of the bilingual mapping.
- Proved CSLS is convex in W and can be used for learning.
- Resulting in same criterion for learning and inference.
- Expanded the KNN search to all the vocabs not only those in the labeled lexicon, which in turn improves performance.
- With the novel objective function the orthogonal mapping does not improve translation quality.

Adversarial Approach

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

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