Gromov-Wasserstein Alignment of Word Embedding Spaces

David Alvarez-Melis, Tommi S. Jaakkola | Massachusetts Institute of Technology

Summary

- A **direct optimization** approach to cross-lingual word embedding alignment
- The Gromov-Wasserstein distance is well-suited for this task because it:
  - Relies on **relational** rather than **positional** similarities across spaces
  - Applies to embeddings of different algorithms and dimensionality too!
- Unsupervised objective **strongly predictive** of final accuracy

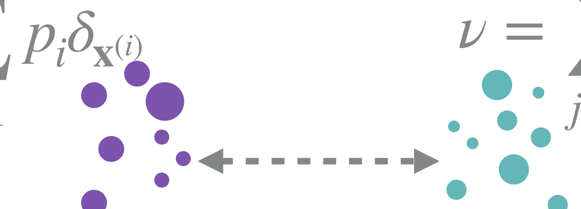
Motivation

- Many tasks in NLP rely on learning cross-domain correspondences
- Parallel data not always available $\Longrightarrow$ **unsupervised** methods
- Word-word translation (*bilingual lexical induction*)- a simple, but important litmus test
- Recent fully unsupervised methods perform on par with supervised counterparts [1, 2]
- ... but adversarial training is slow and often unstable

Background

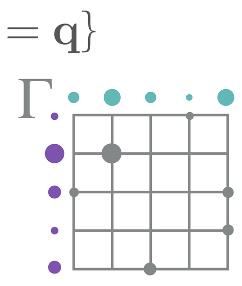
Discrete Optimal Transport

$$\mu = \sum_{i=1}^{n} p_i \delta_{\mathbf{x}^{(i)}} \qquad \nu = \sum_{j=1}^{m} q_j \delta_{\mathbf{y}^{(j)}}$$



$$C_{ij} = C(\mathbf{x}^{(i)}, \mathbf{y}^{(j)})$$

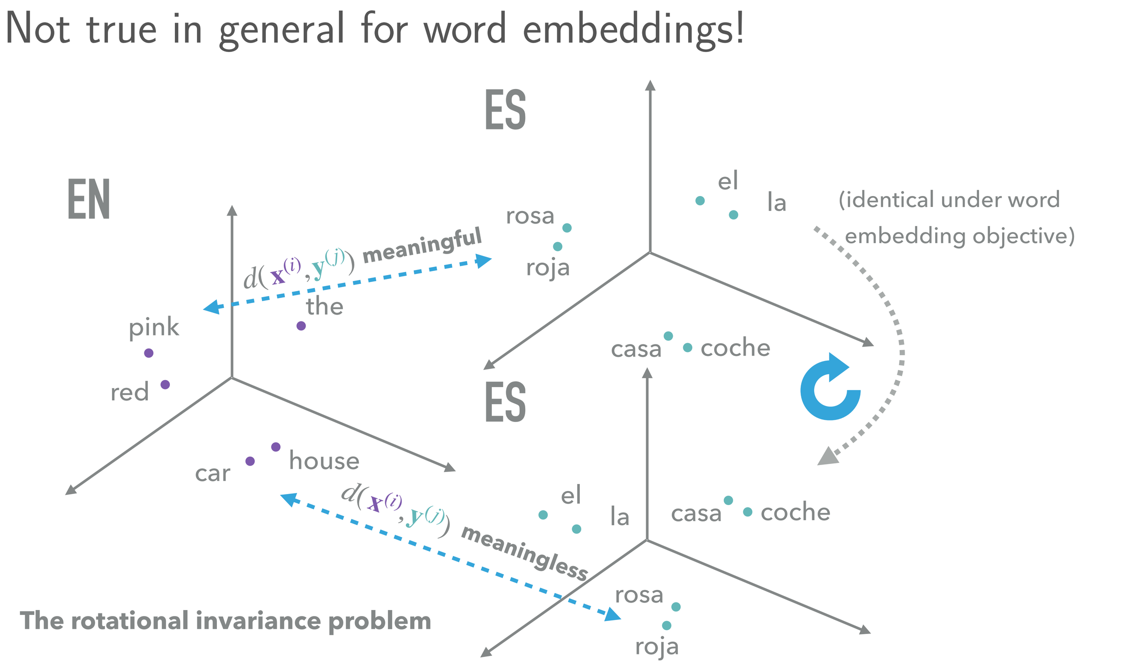
- Discrete distributions: $\mu = \sum_{i=1}^{n} p_i \delta_{\mathbf{x}^{(i)}}$, $\nu = \sum_{j=1}^{m} q_j \delta_{\mathbf{y}^{(j)}}$
- Pairwise costs: $C_{ij} = C(\mathbf{x}^{(i)}, \mathbf{y}^{(j)})$.
- Feasible couplings, $\Gamma \in \mathbb{R}^{n \times m}$ in:
  $$\Pi(\mathbf{p}, \mathbf{q}) = \{\Gamma \mid \Gamma \mathbb{1} = \mathbf{p},\ \Gamma^\top \mathbb{1} = \mathbf{q}\}$$



- The problem:
  $\min_{\Gamma \in \Pi(\mathbf{p},\mathbf{q})} \sum_{i,j} \Gamma_{ij} C_{ij}$

Optimal Transport between Word Embeddings

- Previous applications:
  - Word Mover's Distance [Kusner et al., 2015]: sentence similarity
  - In Word Embedding Alignment [3]
- Treat embeddings as support points of discrete distribution
  $$C_{ij} = c(w_i^{EN}, w_j^{ES}) = d(v^{EN}(w_i), v^{ES}(w_j))$$
- But this assumes the two spaces are **registered**
- Not true in general for word embeddings!



Key References

[1] A. Conneau, G. Lample, M. Ranzato, L. Denoyer, and H. Jégou. "Word Translation Without Parallel Data". In: *ICLR*. 2018.

[2] M. Artetxe, G. Labaka, E. Agirre, and K. Cho. "Unsupervised Neural Machine Translation". In: *International Conference on Learning Representations*. 2018.

[3] M. Zhang, Y. Liu, H. Luan, and M. Sun. "Adversarial training for unsupervised bilingual lexicon induction". In: *ACL*. Vol. 1. 2017, pp. 1959–1970.

[4] F. Mémoli. "Gromov–Wasserstein distances and the metric approach to object matching". In: *Foundations of computational mathematics* 11.4 (2011), pp. 417–487.

[5] G. Peyré, M. Cuturi, and J. Solomon. "Gromov-Wasserstein averaging of kernel and distance matrices". In: *ICML*. 2016, pp. 2664–2672.

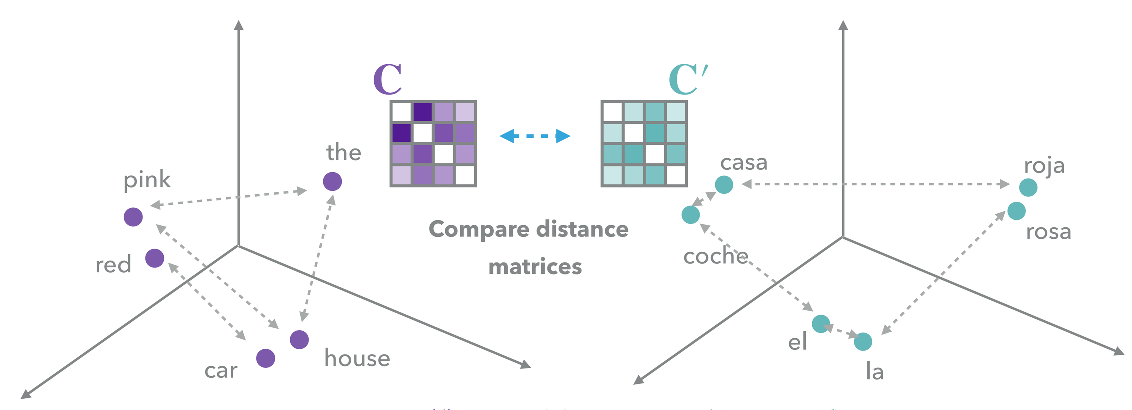
[6] G. Dinu, A. Lazaridou, and M. Baroni. "Improving zero-shot learning by mitigating the hubness problem". In: *arXiv preprint arXiv:1412.6568* (2014).

[7] M. Artetxe, G. Labaka, and E. Agirre. "A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings". In: *ACL*. 2018, pp. 789–798.

Approach

The Gromov-Wasserstein Distance

- Generalizes OT to the non-registered case
- Main idea: compare **distances** instead of absolute **positions**



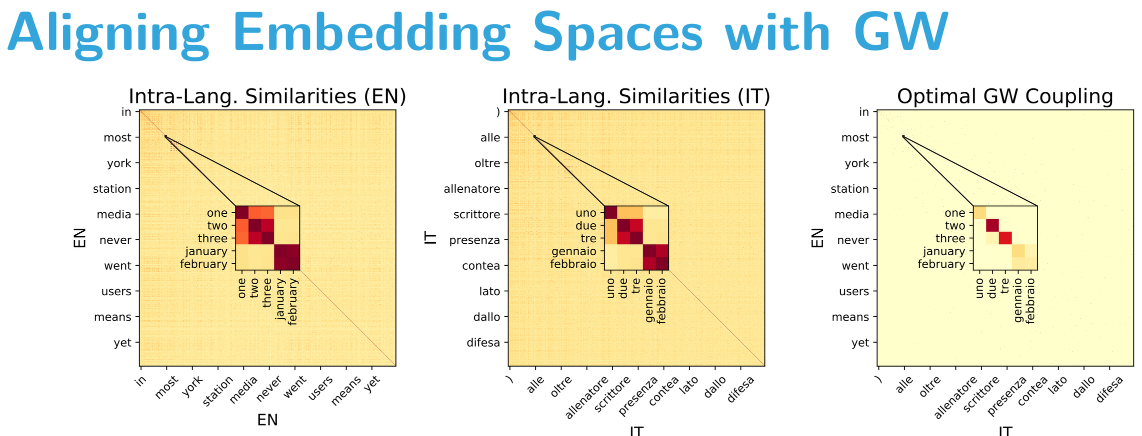
cost of matching $\mathbf{x}^{(i)}$ to $\mathbf{y}^{(j)}$ *and* $\mathbf{x}^{(k)}$ to $\mathbf{y}^{(l)}$ =

$$\mathcal{L}(d(\mathbf{x}^{(i)}, \mathbf{x}^{(i)})), d(\mathbf{x}^{(i)}, \mathbf{x}^{(i)}))$$

- The objective:

$$\mathrm{GW}(\mathbf{C}, \mathbf{C}', \mathbf{p}, \mathbf{q}) = \min_{\Gamma \in \Pi(\mathbf{p},\mathbf{q})} \sum_{i,j,k,l} \mathcal{L}(\mathbf{C}_{ik}, \mathbf{C}'_{jl}) \Gamma_{ij} \Gamma_{kl}$$

Aligning Embedding Spaces with GW



Desirable properties

- Simple, compact, stable objective, few hyperparameters
- For $\mathcal{L}(a, b) = |a - b|$, $\mathrm{GW}^{\frac{1}{2}}$ is a **(proper) distance** [4]

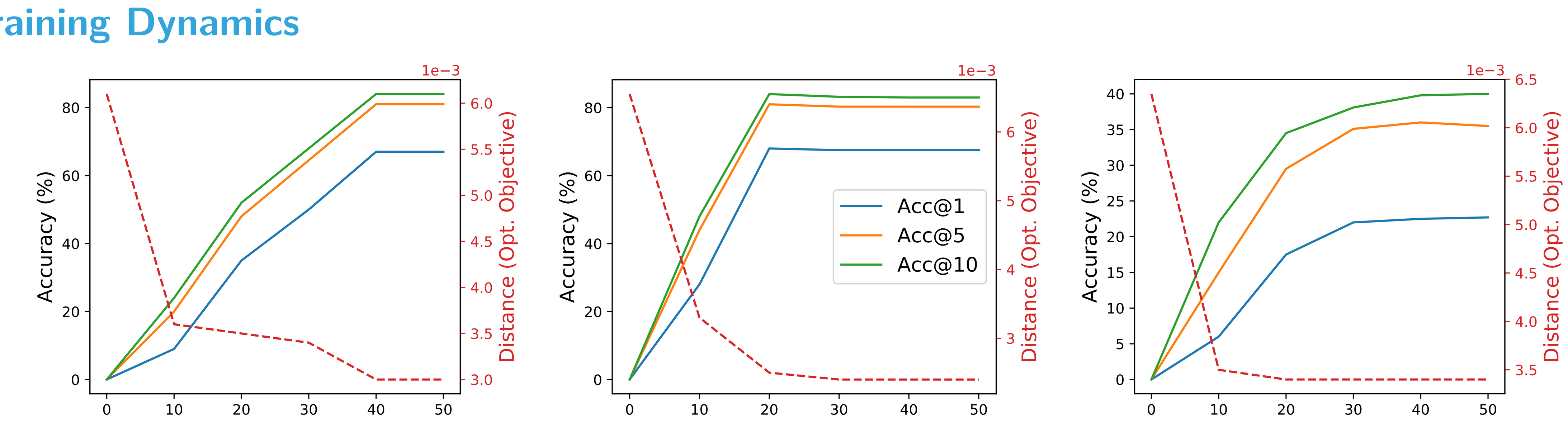
Optimization

- Non-convex problem (even with entropic regularization!)
- Naive solution requires storing 4th order tensor $\mathbf{L}_{i,j,k,l}$
- Yet, **solved efficiently** by projected gradient descent [5]
- Projections given by Sinkhorn-Knopp algorithm
- Algo make provable improvement ($\neq$ adversarial methods)
- For very large problems, we propose a two step approach:
  1. Learn coupling Γ on a subset of points
  2. Use pseudo-matches from Γ to learn orthogonal projection

Inputs: Embeddings $\mathbf{X}$, $\mathbf{Y}$ and probability vectors $\mathbf{p}, \mathbf{q}$, regularization parameter λ.
$\mathbb{C}_s \leftarrow \cos(\mathbf{X}, \mathbf{X})$, $\mathbb{C}_t \leftarrow \cos(\mathbf{Y}, \mathbf{Y})$ ▷ Compute intra-language similarities
$\mathbb{C}_{st} \leftarrow \mathbb{C}_s^2 \mathbf{p} \mathbb{1}_m^\top + \mathbb{1}_n \mathbf{q} (\mathbb{C}_t^2)^\top$
while not converged **do**
  $\hat{\mathbb{C}}_\Gamma \leftarrow \mathbb{C}_{st} - 2\mathbb{C}_s \Gamma \mathbb{C}_t^\top$ ▷ Compute pseudo-cost matrix
  $\mathbf{a} \leftarrow \mathbb{1}$, $\mathbf{K} \leftarrow \exp\{-\hat{\mathbb{C}}_\Gamma / \lambda\}$
  while not converged **do** ▷ Sinkhorn iterations
    $\mathbf{a} \leftarrow \mathbf{p} \oslash \mathbf{K}\mathbf{b}$, $\mathbf{b} \leftarrow \mathbf{q} \oslash \mathbf{K}^\top \mathbf{a}$
  end while
  $\Gamma \leftarrow \mathrm{diag}(\mathbf{a}) \mathbf{K} \mathrm{diag}(\mathbf{b})$
end while
$\mathbf{U}, \Sigma, \mathbf{V}^\top \leftarrow \mathrm{SVD}(\mathbf{X} \Gamma \mathbf{Y}^\top)$ ▷ Optionally (for large problems): Learn explicit projection
$\mathbf{P} = \mathbf{U}\mathbf{V}^\top$

Experiments

Training Dynamics



(a) EN →FR, 15K words, $\lambda = 5 \cdot 10^{-4}$ (b) EN →FR, 15K words, $\lambda = 10^{-4}$ (c) EN →RU, 15K words, $\lambda = 10^{-4}$

- Objective closely follows the metric of interest (accuracy, not available during training)
- Related languages lead to faster optimization
- Regularization λ trades-off speed vs accuracy

Translation Accuracy Results

- TL;DR: Comparable with SOTA
- significantly (order of magnitude) faster than adversarial approaches

Dataset of Conneau et al. [1]:

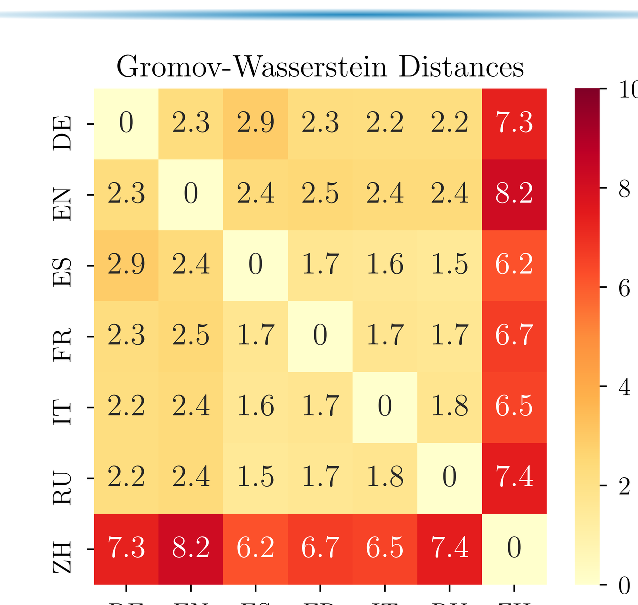
			EN-ES		EN-FR		EN-DE		EN-IT		EN-RU	
	Seeds	Time	→	←	→	←	→	←	→	←	→	←
PROCRUSTES	5K	3	77.6	77.2	74.9	75.9	68.4	67.7	73.9	73.8	47.2	58.2
+ CSLS	5K	3	81.2	82.3	81.2	82.2	73.6	71.9	76.3	**75.5**	51.7	63.7
ADV. [1]	–	957	**81.7**	**83.3**	**82.3**	82.1	74.0	72.2	77.4	76.1	**52.4**	**61.4**
GW ($\lambda = 10^{-4}$)	–	70	78.3	79.5	79.3	78.3	69.6	66.9	75.3	74.1	26.1	35.4
GW ($\lambda = 10^{-5}$)	–	37	**81.7**	80.4	81.3	78.9	71.9	**72.8**	**78.9**	75.2	45.1	43.7

Dataset of Dinu et al. [6]:

	EN-IT		EN-DE		EN-FI		EN-ES	
	P@1	Time	P@1	Time	P@1	Time	P@1	Time
ADVERSARIAL OT [3]	0	46.6	0	46.0	0.07	44.9	0.07	43.0
ADV [1]	45.4	46.1	47.3	45.4	1.62	44.4	36.2	45.3
SELF-LEARN[7]	48.5	8.9	**48.5**	7.3	**33.5**	12.9	**37.6**	9.1
GW	44.4	35.2	37.8	36.7	6.8	15.6	12.5	18.4
GW + NORM	**49.2**	36	46.5	33.2	18.3	42.1	**37.6**	38.2

NOTE: Times reported for first tree methods is in GPU, ours in CPU

The GW Linguistic Distance



- Recall: GW problem induces a (true) metric
- Notion of semantic-syntactic ling. distance

Discussion + Future Work

- Speed-ups using GPU + stochastic opt
- Experiments on different embedding algorithms and dimensionality
- Extension to sentence level translation

 Contact: {davidam|tommi}@csail.mit.edu