Improving Vector Space Word Representations Using Multilingual Correlation

by M. Faruqui and C. Dyer

presented by Natalia Skachkova

Department of Computer Science
Saarland University

12.06.2019

Overview

Motivation

Distributional Hypothesis (Harris, 1954):
Words that are similar in meaning tend to occur in similar contexts.

Observation:

$$\text{vaayuyaan (Hindi)} \left\{ \begin{array}{l} \text{aeroplane} \\ \text{airplane} \\ \text{plane} \end{array} \right\} \Rightarrow \text{similar meaning}$$

Idea:
Knowing how words translate is a valuable source of lexico-semantic information.

Realization:
Incorporate translational context when constructing a vector space semantic model (VSM).

Incorporating translational context

Approach:

1. Construct independent monolingual VSMs for 2 languages.
2. Project them onto a common vector space.

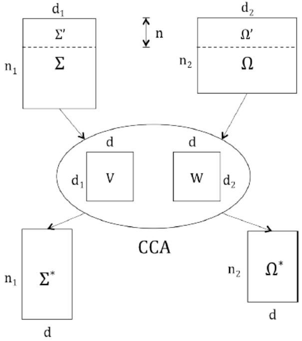
Step 1: Constructing monolingual VSMs with LSA

1. Construct a word co-occurrence frequency matrix:
   - a window of 10 words around the target word
   - words with frequencies < 10 are omitted
   - top 100 of the most frequent words are removed
2. Replace raw counts with PMI scores.
3. Factorize the matrix with SVD: $X = U\Psi V^T$
4. Obtain a reduced dimensional representation of words from size $|V|$ to k: $A = U_k \Psi_k$ (truncate columns)

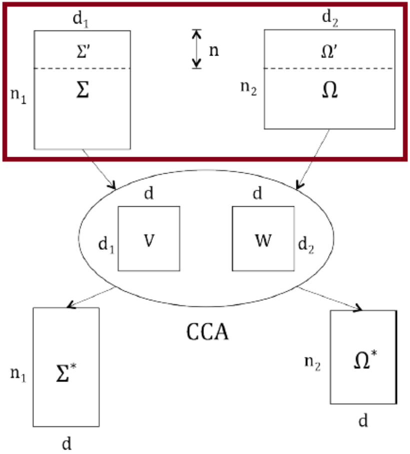
In the end A contains word vector representations in the reduced dimensional monolingual space.

Step 2: Projecting word vectors from 2 different VSMs onto a common vector space

Method: Canonical Correlation Analysis (CCA).
Objective: Measure the linear relationship between 2 multidimensional variables.

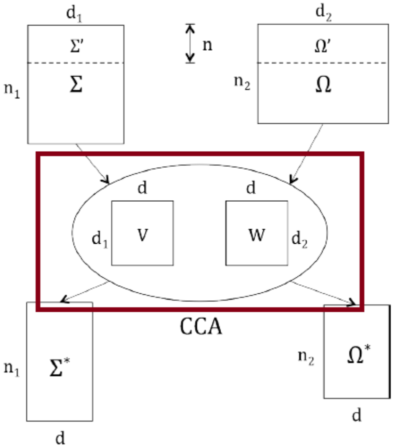


Step 2: CCA in detail



Take 2 different monolingual VSMs Σ and Ω (probably of different vocabulary sizes) and select *n* translation pairs resulting in $\Sigma' \subseteq \Sigma$ and $\Omega' \subseteq \Omega$.

Step 2: CCA in detail

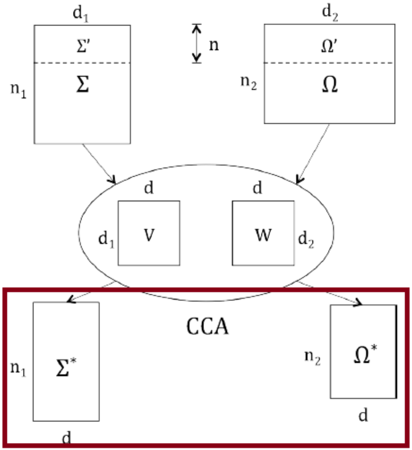


Find linear combinations of Σ' and Ω' which have maximal correlation with each other, namely $x' = \Sigma' v$ and $y' = \Omega' w$, s.t. the correlation $\rho(x', y')$ is maximized:

$$\rho(x', y') = \frac{E[x'y']}{\sqrt{E[x'^2]E[y'^2]}}$$

Vectors v and w are called **a canonical pair**.
The procedure is repeated d times, where $d = min(d_1, d_2)$.

Step 2: CCA in detail



Trancate the matrices V and W to reduce the number of dimensions.
Multiply the original co-occurrence matrices with the ones containing the projection vectors to get bilingual embeddings: $\Sigma^* = \Sigma V$, $\Omega^* = \Omega W$.

Types of tasks

- **Word Similarity**(4 benchmarks, human judgements):
  1. WS-353 dataset: WS-SIM (e.g. king queen 8.58) and WS-REL (e.g. baby mother 7.85)
     - besides generic words, the dataset includes phrases, proper names and technical terms
     - score range is 0-10
  2. RG-65 dataset (e.g. bird woodland 1.24):
     - includes only nouns, non-technical words
     - words range from synonymy pairs to unrelated words
     - score range is 0-4
  3. MC-30 dataset (e.g. midday noon 3.42):
     - includes only generic nouns, a subset of WS-353
     - score range is 0-4
  4. MTurk-287

  Similarity measure: **cosine similarity**.

  Spearman's rank correlation between the model's and humans' rankings.

Types of tasks

- **Semantic Relations** (4 relations)
  1. country-capital
  2. country-currency
  3. man-woman
  4. city-in-state

  E.g. *England:London::France:Paris*

  Pattern *a:b::c:d*

  $y = x_a - x_b + x_c$

  $x_w = \arg\max_{x_w} \frac{x_w \cdot y}{|x_w| \cdot |y|}$

- **Syntactic Relations** (9 relations)
  1. adjective-adverb
  2. opposites
  3. comparative
  4. superlative
  5. present-participle
  6. nation-nationality
  7. past tense
  8. plural nouns
  9. plural verbs

Data

- Monolingual news corpora WMT-2011 & WMT-2012 in 4 languages:
  - English
  - German
  - Spanish
  - French
- 300 M. tokens for each language
- Original monolingual vectors have dimension 640
- Multilingual embeddings truncated by 20%
- Language pairs: En-De, En-Es, En-Fr

Experiments' Results

Monolingual vs. Bilingual Embeddings

Lang	Dim	WS-353	WS-SIM	WS-REL	RG-65	MC-30	MTurk-287	SEM-REL	SYN-REL
En	640	46.7	56.2	36.5	50.7	42.3	51.2	14.5	36.8
En-De	512	68.0	**74.4**	64.6	**75.5**	**81.9**	53.6	43.9	**45.5**
En-Fr	512	**68.4**	73.3	**65.7**	73.5	81.3	**55.5**	43.9	44.3
En-Es	512	67.2	71.6	64.5	70.5	78.2	53.6	**44.2**	44.5

Table: Spearman's rank correlation on different tasks.

At least 20 points gain over the baseline!

Experiments' Results

Bilingual Embeddings vs. Embeddings obtained with Neural Networks

Vectors	Dim	Lang	WS-353	WS-SIM	WS-REL	RG-65	MC-30	MTurk-287	SEM-REL	SYN-REL
SVD	80	Mono	34.8	45.5	23.4	30.8	21.0	46.6	13.5	24.4
	48	Multi	58.1	65.3	52.7	**62.7**	**67.7**	62.1	23.4	33.2
RNN	80	Mono	23.6	35.6	17.5	26.2	47.7	32.9	4.7	18.2
	48	Multi	35.4	47.3	29.8	36.6	46.5	43.8	4.1	12.2
SG	80	Mono	**63.9**	69.9	**60.9**	54.6	62.8	**66.9**	**47.8**	**47.8**
	48	Multi	63.1	**70.4**	57.6	54.9	64.7	58.7	46.5	44.2

Table: Spearman's rank correlation on different tasks for different types of models.

Conclusion:

- Multilingual embeddings based on CCA perform better than monolingual ones based on LSA.
- Different language pairs demonstrate similar tendencies.
- Multilingual embeddings show a little bit worse results than Skipgram embeddings, but they are much easier and faster to obtain.
- They encode semantic information better than syntactic.

References:

- https://en.wikipedia.org/wiki/Canonical_correlation
- https://www.cs.cmu.edu/∼tom/10701_sp11/slides/CCA_tutorial.pdf
- https://www.mathematica-journal.com/2014/06/canonical-correlation-analysis/
- http://users.stat.umn.edu/∼ helwig/notes/cancor-Notes.pdf

Thank you for your attention!

A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings

Mikel Artetxe
Gorka Labaka
Eneko Agirre

Presented by Susann Boy
Seminar: Embeddings for NLP and IR
Lecturer: Cristina España i Bonet
Saarland University

Outline

- Introduction
- Proposed Method
  - Pre-processing
  - Initialization
  - self-learning procedure
  - improving dictionary induction
  - final refinement
- Results
  - Comparison with other methods
  - Ablation test

Recent work...

… manages to learn cross-lingual word embeddings without parallel data by mapping monolingual embeddings to a shared space through adversarial training

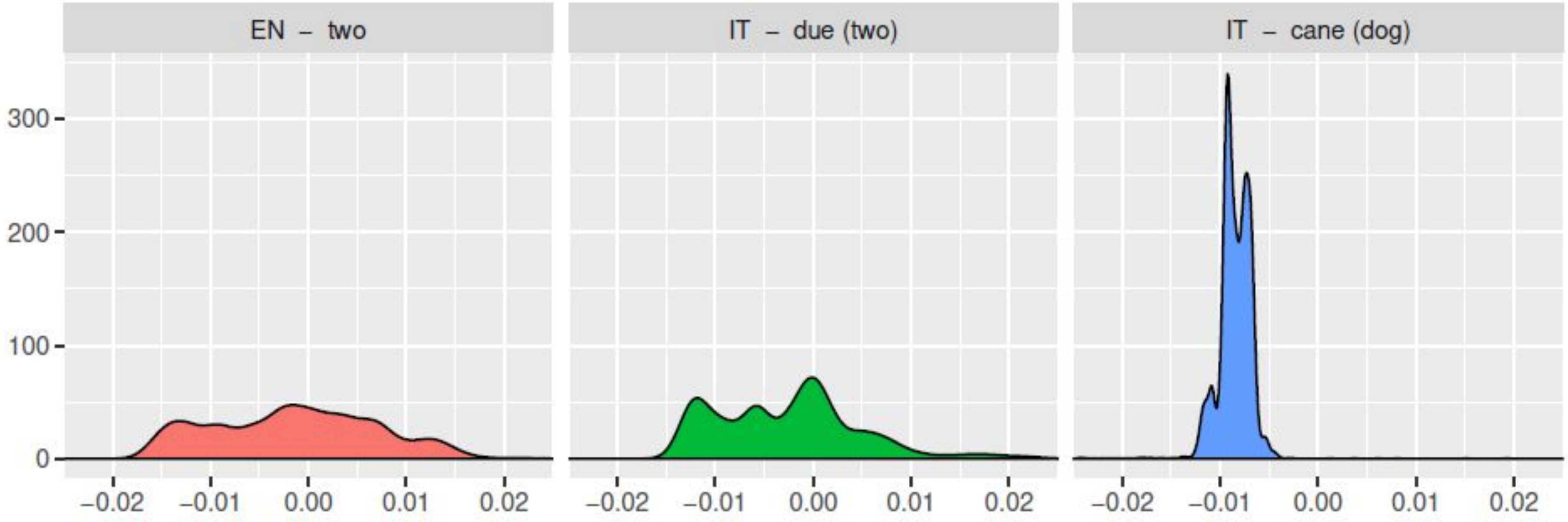
… uses mostly supervised methods and a bilingual dictionary to learn the mapping

and the evaluation has focused on favorable conditions

approach: fully unsupervised initialization that explicitly exploits the structural similarity of the embeddings + robust self-learning algorithm that iteratively improves this solution

Idea

two equivalent words in different languages should have a similar distribution



Idea

- independently train the embeddings in different languages using monolingual corpora, then map them to a shared space through a linear transformation
- unsupervised method to build an initial solution without the need of a seed dictionary
- combine initialization with a more robust self-learning method, which is able to start from the weak initial solution and iteratively improve the mapping

Method

X, Z = word embedding matrices in two languages, their ith row X_{i*} and Z_{i*} denotes the embeddings of the ith word in their respective vocabularies

goal: learn the linear transformation matrices W_X and W_Z so the mapped embeddings XW_X and ZW_Z are in the same cross-lingual space

build a dictionary between both languages, encoded as a sparse matrix D, $D_{ij} = 1$ if the jth word in the target language is a translation of the ith word in the source language

Four Key Steps

- pre-processing that normalizes the embeddings
- fully unsupervised initialization scheme that creates an initial solution
- robust self-learning procedure that iteratively improves this solution
- final refinement step that further improves the resulting mapping through symmetric re-weighting

Method

- **pre-processing** that normalizes the embeddings
- fully unsupervised initialization scheme that creates an initial solution
- robust self-learning procedure that iteratively improves this solution
- final refinement step that further improves the resulting mapping through symmetric re-weighting

Pre-processing

- length normalize embeddings
- mean center each dimension
- length normalize again

Why the second normalization?

- second length normalization guarantees the final embeddings to have a unit length
- dot product of any 2 embeddings is equivalent to their cosine similarity → can be taken as a measure of their similarity

Method

- pre-processing that normalizes the embeddings
- fully unsupervised **initialization** scheme that creates an initial solution
- robust self-learning procedure that iteratively improves this solution
- final refinement step that further improves the resulting mapping through symmetric re-weighting

Initialization

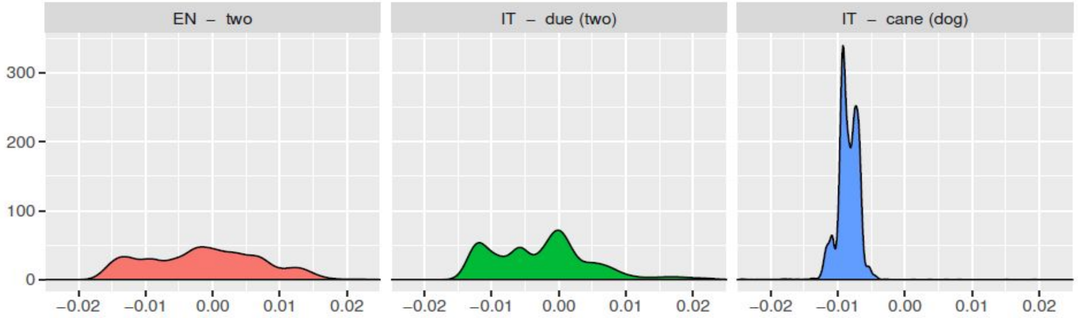
problem: X and Z are unaligned across both axes: no direct correspondence between both languages

construct two alternative representations *X′* and *Z′* that are aligned across their *j*th dimension X'_{*j} and Z'_{*j} which will be used to build the initial dictionary that aligns their respective vocabularies

- both axes of the corresponding similarity matrices of the original embeddings $M_X = XX^T$ and $M_Z = ZZ^T$ correspond to words
- assuming that embedding spaces are perfectly isometric, M_X and M_Z would be equivalent up to a permutation of their rows and columns, where the permutation defines the dictionary across both languages

Initialization

- sort values in each row of M_X and M_Z
- equivalent words would get exact same vector across languages: given a word and its row in sorted(M_X) apply nearest neighbor retrieval over the rows of sorted(M_Z) to find corresponding translation
- compute sorted($\sqrt{M_X}$) and sorted($\sqrt{M_Z}$) and normalize them: yields X' and Z' that are later used to build the initial solution for self-learning



Method

- pre-processing that normalizes the embeddings
- fully unsupervised initialization scheme that creates an initial solution
- **robust self-learning** procedure that iteratively improves this solution
- final refinement step that further improves the resulting mapping through symmetric re-weighting

self-learning procedure

training iterates through following 2 steps until convergence:

1. compute optimal orthogonal mapping maximizing the similarities for the current dictionary D:

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

   optimal solution is given by $W_X = U$ and $W_Z = V$, $USV^T = X^TDZ$ being the SVD of X^TDZ

2. compute optimal dictionary over similarity matrix of the mapped embeddings $XW_XW_Z^TZ^T$, uses typically nearest neighbor retrieval from the source language into target language, so

   $D_{ij} = 1$ if $j = \arg\max_k (X_{i*}W_X) \cdot (Z_{k*}W_Z)$ and $D_{ij} = 0$ otherwise

- underlying optimization objective is independent from initial dictionary and algorithm is guaranteed to converge to a local optimum of it
- method does not work if starting from a completely random solution

→ use unsupervised initialization procedure to build an initial solution

quality of initial method is not good enough to avoid poor local optima: **key improvements in dictionary induction step** to make self-learning more robust and learn better mappings:

- **stochastic dictionary induction**: by randomly keeping some elements in the similarity matrix with probability p and setting remaining ones to 0; the smaller the value of p, the more the induced dictionary will vary from iteration to iteration: enabling to escape poor local optima
- **frequency-based vocabulary cutoff**: size of similarity matrix grows quadratically with respect to that of vocabularies: restrict dictionary induction process to the k most frequent words in each language
- **CSLS retrieval**: nearest neighbor suffers from hubness problem (effect of curse of dimensionality, causes a few points (hubs) to be nearest neighbors of many other points)
- **bidirectional dictionary induction**: when dictionary is induced from source into target language, not all target language words will be present in it, some will occur multiple times: accentuates problem of local optima: inducing dictionary in both directions and taking their corresponding concatenation

Method

- pre-processing that normalizes the embeddings
- fully unsupervised initialization scheme that creates an initial solution
- robust self-learning procedure that iteratively improves this solution
- **final refinement step** that further improves the resulting mapping through symmetric re-weighting

Symmetric Re-Weighting

- given $USV^T = X^T DZ$, this is equivalent to taking $W_X = U$ and $W_Z = VS$, where X and Z are previously whitened and later de-whitened
- re-weighting accentuates also problem of local optima when incorporated into self-learning, it discourages to explore other regions of the search space: using it as final step once self-learning has converged to a good solution
- apply re-weighting symmetrically in both languages

Results

	ES-EN				IT-EN				TR-EN			
	best	avg	s	t	best	avg	s	t	best	avg	s	t
Zhang et al. (2017a), $\lambda = 1$	71.43	68.18	**10**	13.2	60.38	56.45	**10**	12.3	0.00	0.00	0	13.0
Zhang et al. (2017a), $\lambda = 10$	70.24	66.37	**10**	13.0	57.64	52.60	**10**	12.6	21.07	17.95	**10**	13.2
Conneau et al. (2018), code	76.18	75.82	**10**	25.1	**67.32**	**67.00**	**10**	25.9	32.64	14.34	5	25.3
Conneau et al. (2018), paper	76.15	75.81	**10**	25.1	67.21	60.22	9	25.5	29.79	16.48	7	25.5
Proposed method	**76.43**	**76.28**	**10**	**0.6**	66.96	66.92	**10**	**0.9**	**36.10**	**35.93**	**10**	**1.7**

Table 1: Results of unsupervised methods on the dataset of Zhang et al. (2017a). We perform 10 runs for each method and report the best and average accuracies (%), the number of successful runs (those with >5% accuracy) and the average runtime (minutes).

Results

	EN-IT				EN-DE				EN-FI				EN-ES			
	best	avg	s	t	best	avg	s	t	best	avg	s	t	best	avg	s	t
Zhang et al. (2017a), $\lambda = 1$	0.00	0.00	0	47.0	0.00	0.00	0	47.0	0.00	0.00	0	45.4	0.00	0.00	0	44.3
Zhang et al. (2017a), $\lambda = 10$	0.00	0.00	0	46.6	0.00	0.00	0	46.0	0.07	0.01	0	44.9	0.07	0.01	0	43.0
Conneau et al. (2018), code	45.40	13.55	3	46.1	47.27	42.15	9	45.4	1.62	0.38	0	44.4	36.20	21.23	6	45.3
Conneau et al. (2018), paper	45.27	9.10	2	45.4	0.07	0.01	0	45.0	0.07	0.01	0	44.7	35.47	7.09	2	44.9
Proposed method	**48.53**	**48.13**	**10**	**8.9**	**48.47**	**48.19**	**10**	**7.3**	**33.50**	**32.63**	**10**	**12.9**	**37.60**	**37.33**	**10**	**9.1**

Table 2: Results of unsupervised methods on the dataset of Dinu et al. (2015) and the extensions of Artetxe et al. (2017, 2018a). We perform 10 runs for each method and report the best and average accuracies (%), the number of successful runs (those with >5% accuracy) and the average runtime (minutes).

Comparison with state-of-the-art

Supervision	Method	EN-IT	EN-DE	EN-FI	EN-ES
5k dict.	Mikolov et al. (2013)	34.93†	35.00†	25.91†	27.73†
	Faruqui and Dyer (2014)	38.40*	37.13*	27.60*	26.80*
	Shigeto et al. (2015)	41.53†	43.07†	31.04†	33.73†
	Dinu et al. (2015)	37.7	38.93*	29.14*	30.40*
	Lazaridou et al. (2015)	40.2	-	-	-
	Xing et al. (2015)	36.87†	41.27†	28.23†	31.20†
	Zhang et al. (2016)	36.73†	40.80†	28.16†	31.07†
	Artetxe et al. (2016)	39.27	41.87*	30.62*	31.40*
	Artetxe et al. (2017)	39.67	40.87	28.72	-
	Smith et al. (2017)	43.1	43.33†	29.42†	35.13†
	Artetxe et al. (2018a)	45.27	44.13	**32.94**	36.60
25 dict.	Artetxe et al. (2017)	37.27	39.60	28.16	-
Init. heurist.	Smith et al. (2017), cognates	39.9	-	-	-
	Artetxe et al. (2017), num.	39.40	40.27	26.47	-
None	Zhang et al. (2017a), $\lambda = 1$	0.00*	0.00*	0.00*	0.00*
	Zhang et al. (2017a), $\lambda = 10$	0.00*	0.00*	0.01*	0.01*
	Conneau et al. (2018), code‡	45.15*	46.83*	0.38*	35.38*
	Conneau et al. (2018), paper‡	45.1	0.01*	0.01*	35.44*
	Proposed method	**48.13**	**48.19**	32.63	**37.33**

Table 3: Accuracy (%) of the proposed method in comparison with previous work. *Results obtained with the official implementation from the authors. †Results obtained with the framework from Artetxe et al. (2018a). The remaining results were reported in the original papers. For methods that do not require supervision, we report the average accuracy across 10 runs. ‡For meaningful comparison, runs with <5% accuracy are excluded when computing the average, but note that, unlike ours, their method often gives a degenerated solution (see Table 2).

Ablation test

self-learning does not work with random initialization

	EN-IT				EN-DE				EN-FI				EN-ES			
	best	avg	s	t	best	avg	s	t	best	avg	s	t	best	avg	s	t
Full system	48.53	48.13	10	8.9	48.47	48.19	10	7.3	33.50	32.63	10	12.9	37.60	37.33	10	9.1
- Unsup. init.	0.07	0.02	0	16.5	0.00	0.00	0	17.3	0.07	0.01	0	13.8	0.13	0.02	0	15.9
- Stochastic	48.20	48.20	10	2.7	48.13	48.13	10	2.5	0.28	0.28	0	4.3	37.80	37.80	10	2.6
- Cutoff (k=100k)	46.87	46.46	10	114.5	48.27	48.12	10	105.3	31.95	30.78	10	162.5	35.47	34.88	10	185.2
- CSLS	0.00	0.00	0	15.0	0.00	0.00	0	13.8	0.00	0.00	0	13.1	0.00	0.00	0	14.1
- Bidirectional	46.00	45.37	10	5.6	48.27	48.03	10	5.5	31.39	24.86	8	7.8	36.20	35.77	10	7.3
- Re-weighting	46.07	45.61	10	8.4	48.13	47.41	10	7.0	32.94	31.77	10	11.2	36.00	35.45	10	9.1

Table 4: Ablation test on the dataset of Dinu et al. (2015) and the extensions of Artetxe et al. (2017, 2018a). We perform 10 runs for each method and report the best and average accuracies (%), the number of successful runs (those with >5% accuracy) and the average runtime (minutes).

References

Mikel Artetxe, Gorka Labaka, Eneko Agirre. (2018). A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. Vol. 1. Pages 789-798

But what about language pairs that don’t share the same alphabet like English-Russian / English-Chinese?

WORD TRANSLATION WITHOUT PARALLEL DATA

A Paper By

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, Hervé Jégou

Presented by
Guadalupe Romero and Kathryn Chapman

Structure

- Introduction
- Model
  - Domain adversarial setting
  - Refinement
- Experiments/Evaluation
- Conclusion

Structure

- **Introduction**
- Model
  - Domain adversarial setting
  - Refinement
- Experiments/Evaluation
- Conclusion

Intro - Background

- Mikolov et al. (2013)
  - first noticed continuous word embedding spaces exhibit similar structures across languages
  - proposed using similarity by learning linear mapping from source to target
  - used parallel vocabulary as anchor points to learn mapping

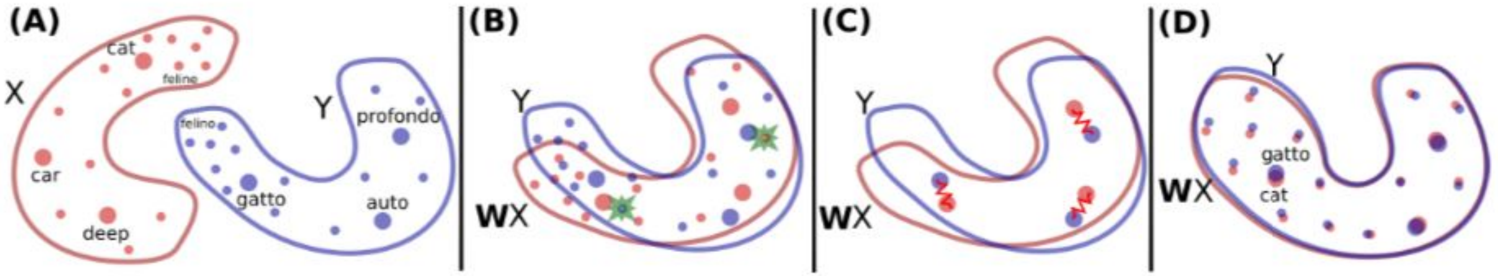
Intro - Background

- Mikolov et al. (2013)
  - supervised approach
- Current fully and semi-unsupervised methods have either:
  - not reached competitive performance
  - require parallel data (aligned corpora)
  - require seed lexicon

Intro - This Paper

- Introduces **unsupervised model** on par with, sometimes outperforming, current supervised models
  - therefore, no parallel data - only two large monolingual corpora (source and target)
- Uses **adversarial training** to map source to target space
- Extracts **parallel dictionary**
- Introduces unsupervised selection metric to select best performing model
- Important: goal here to do in unsupervised way what previous work has only done in a supervised way: creating a word-to-word mapping between natural languages
  - goal is NOT to create robust translator; rather, a dictionary

Intro - Pipeline



two word embedding distributions X and Y trained with fasText

use a GAN to learn a transformation matrix W that aligns X and Y

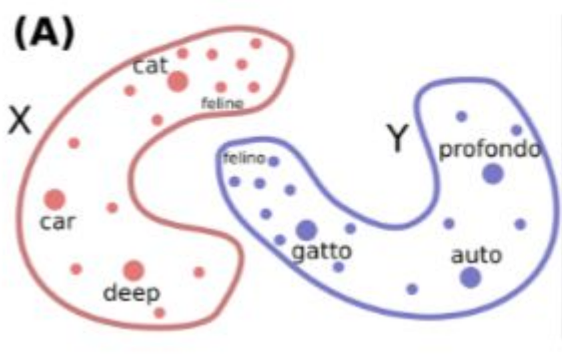
keep only translation pairs from WX and Y that are frequent and mutual K-NN

translate by using the mapping W and distance metric CSLS

Structure

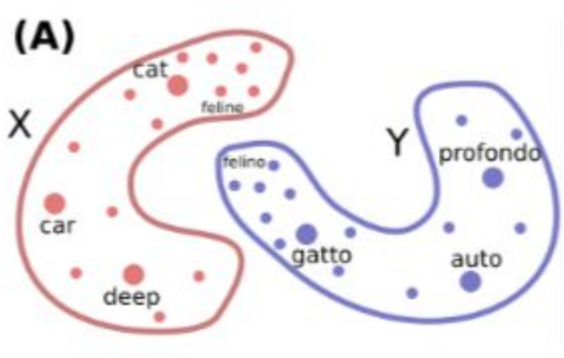
- Introduction
- **Model**
  - **Domain adversarial setting**
  - Refinement
- Experiments/Evaluation
- Conclusion

We start with A:



two word embedding distributions
X and Y trained with fastText

We start with A:

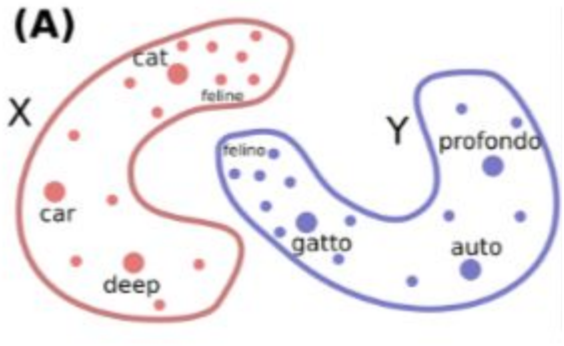


two word embedding distributions
X and Y trained with fastText

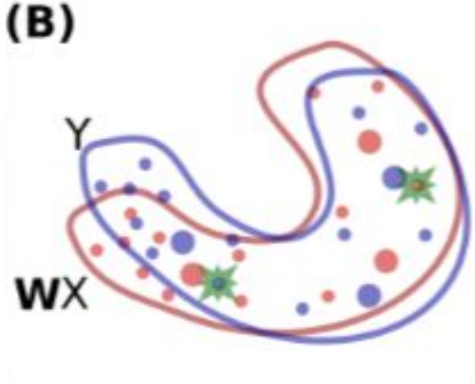
- Similar shapes
- Similar clusters

We start with A:

How do we get to B?



two word embedding distributions X and Y trained with fastText



use a GAN to learn a transformation matrix W that aligns X and Y

Generative Adversarial Network (GAN) - what does this mean?

- A GAN is actually two neural networks competing with each other
  - Generator vs Discriminator
- Generative algorithms:
  - given data, generates new data trying to mimic input
  - predict features given a label
- Discriminative algorithms:
  - given data, classifies it
  - predict label given features

Generative Adversarial Networks

- How do generative and discriminative algorithms work together?
- 1 generative neural network generates data instances
- 1 discriminative neural network evaluates authenticity of data instance
    - rather, decides whether each data instance belongs to actual training data, or synthetically generated data

Generative Adversarial Networks

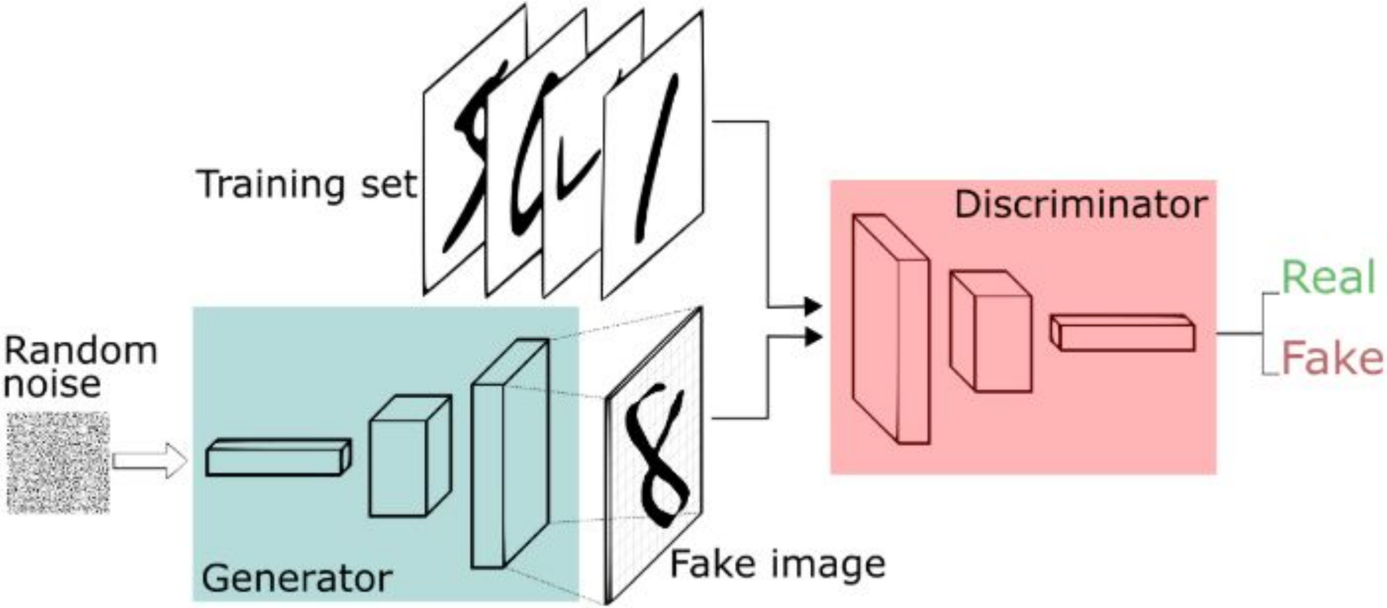
An example:

- We have a dataset of images of handwritten numerals
- Generator goal: create new, synthetic images to pass to discriminator and ‘trick’ discriminator into classifying them as authentic
- Discriminator goal: recognize that a numeral is either authentic or synthetic when given numeral as input

Generative Adversarial Networks

- Three steps:
  - Generator takes in random input & transforms it into what it “thinks” a number looks like
  - Generated image is passed to discriminator with images from authentic data
  - Discriminator returns authenticity probabilities between 0 and 1
    - 0 = fake, 1 = authentic

Generative Adversarial Networks



Source: https://www.freecodecamp.org/news/an-intuitive-introduction-to-generative-adversarial-networks-gans-7a2264a81394/

Generative Adversarial Networks

But, that’s not all

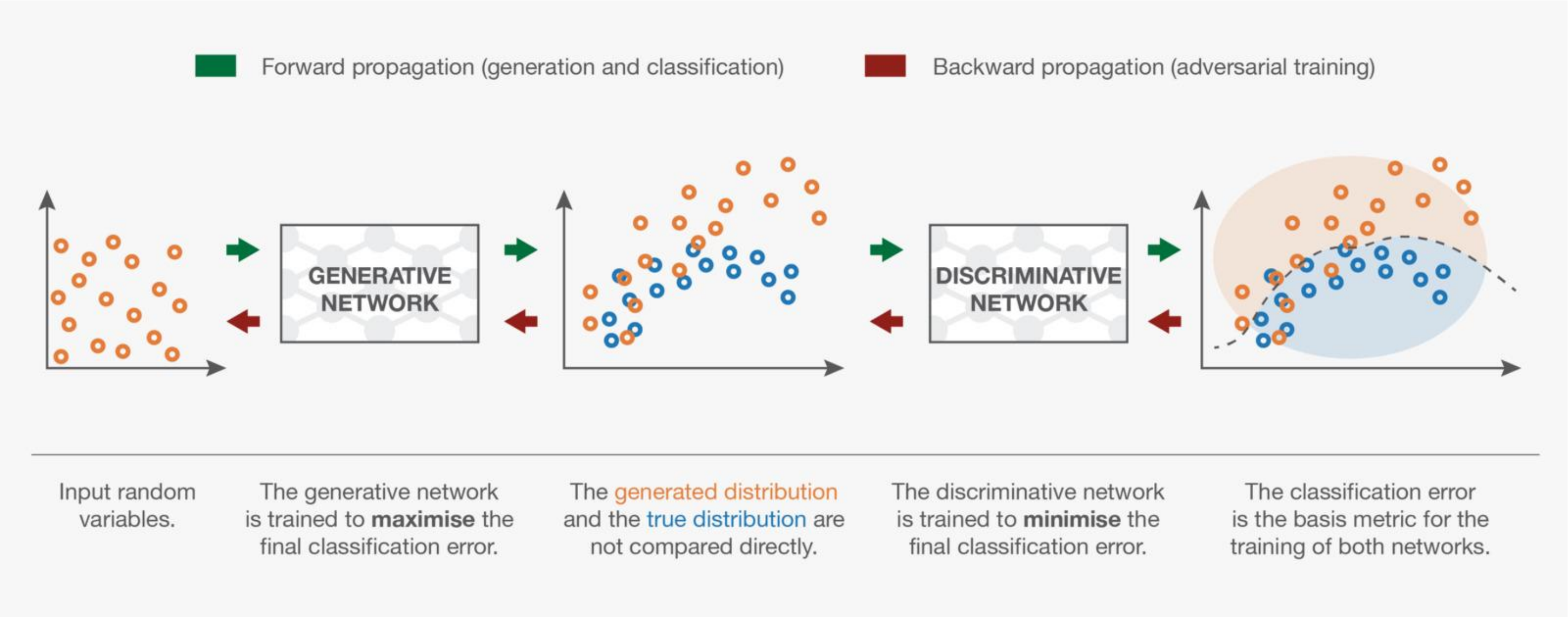
Also, double feedback loop:

- discriminator is in feedback loop with ground truth of images,

  generator in feedback loop with discriminator

- how the model improves

Generative Adversarial Networks



source: https://towardsdatascience.com/understanding-generative-adversarial-networks-gans-cd6e4651a29

Generative Adversarial Networks

In Conneau et al:

- Goal to generate matrix W thats maps source embeddings $X = \{x_1, \ldots, x_n\}$ to

  target embeddings $Y = \{y_1, \ldots, y_m\}$
- Model trained to discriminate between elements randomly sampled from $WX = \{Wx_1, \ldots, Wx_n\}$ & Y
- W trained to prevent discriminator from distinguishing origins of embeddings sampled from WX & Y

Generative Adversarial Networks

Discriminator objective We refer to the discriminator parameters as θ_D. We consider the probability $P_{\theta_D}(\text{source} = 1 | z)$ that a vector z is the mapping of a source embedding (as opposed to a target embedding) according to the discriminator. The discriminator loss can be written as:

$$\mathcal{L}_D(\theta_D|W) = -\frac{1}{n}\sum_{i=1}^{n} \log P_{\theta_D}(\text{source} = 1 | Wx_i) - \frac{1}{m}\sum_{i=1}^{m} \log P_{\theta_D}(\text{source} = 0 | y_i). \quad (3)$$

n = length of source embeddings

m = length of target embeddings

$P\theta_D$(source = 1| Wx_i) = probability that Wx_i is classified as a mapping of a source embedding

$P\theta_D$(source = 0| y_i) = probability that y_i is classified as a target embedding

GOAL: to maximize ability to determine that a mapped source embedding is a mapped source embedding, and that a target embedding is a target embedding; minimize L_D

Generative Adversarial Networks

Discriminator objective We refer to the discriminator parameters as θ_D. We consider the probability $P_{\theta_D}(\text{source} = 1 | z)$ that a vector z is the mapping of a source embedding (as opposed to a target embedding) according to the discriminator. The discriminator loss can be written as:

$$\mathcal{L}_D(\theta_D|W) = -\frac{1}{n}\sum_{i=1}^{n} \boxed{\log P_{\theta_D}(\text{source} = 1 | Wx_i)} - \frac{1}{m}\sum_{i=1}^{m} \boxed{\log P_{\theta_D}(\text{source} = 0 | y_i)}. \quad (3)$$

n = length of source embeddings

m = length of target embeddings

$P\theta_D(\text{source} = 1| Wx_i)$ = probability that Wx_i is classified as a mapping of a source embedding

$P\theta_D(\text{source} = 0| y_i)$ = probability that y_i is classified as a target embedding

GOAL: to maximize ability to determine that a mapped source embedding is a mapped source embedding, and that a target embedding is a target embedding; minimize L_D

Generative Adversarial Networks

Mapping objective In the unsupervised setting, W is now trained so that the discriminator is unable to accurately predict the embedding origins:

$$\mathcal{L}_W(W|\theta_D) = -\frac{1}{n}\sum_{i=1}^{n}\boxed{\log P_{\theta_D}\left(\text{source} = 0 \middle| Wx_i\right)} - \frac{1}{m}\sum_{i=1}^{m}\boxed{\log P_{\theta_D}\left(\text{source} = 1 \middle| y_i\right)} \quad (4)$$

n = length of source embeddings

m = length of target embeddings

$P\theta_D$(source = 0| Wx_i) = probability that Wx_i is classified as a target embedding

$P\theta_D$(source = 1| y_i) = probability that y_i is classified as a mapping of a source embedding

GOAL: to maximize ability to generate a mapping such that a mapped source embedding is classified as a target embedding, and that a target embedding is classified as a mapped source embedding; minimize L_W

Generative Adversarial Networks

Mapping objective In the unsupervised setting, W is now trained so that the discriminator is unable to accurately predict the embedding origins:

$$\mathcal{L}_W(W|\theta_D) = -\frac{1}{n}\sum_{i=1}^{n}\log P_{\theta_D}\left(\text{source} = 0\middle|Wx_i\right) - \frac{1}{m}\sum_{i=1}^{m}\log P_{\theta_D}\left(\text{source} = 1\middle|y_i\right). \quad (4)$$

n = length of source embeddings

m = length of target embeddings

$P\theta_D$(source = 0| Wx_i) = probability that Wx_i is classified as a target embedding

$P\theta_D$(source = 1| y_i) = probability that y_i is classified as a mapping of a source embedding

GOAL: to maximize ability to generate a mapping such that a mapped source embedding is classified as a target embedding, and that a target embedding is classified as a mapped source embedding; minimize L_W

Generative Adversarial Networks

In other words:

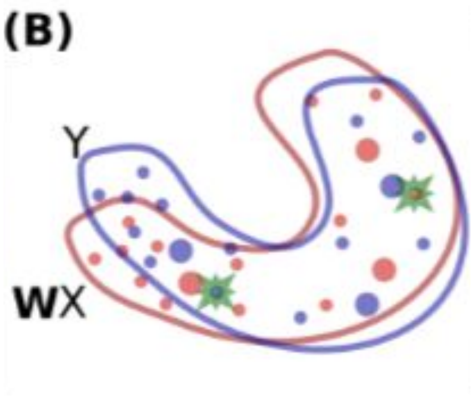
- An embedding is randomly sampled from WX or Y and the discriminator judges whether from WX or Y*
- The discriminator's judgement is fed back to generator, and generator alters its method of generating a matrix W so it can better fool the discriminator via stochastic gradient updates
- Both discriminator and generator competing to maximize their abilities
- Once discriminator cannot distinguish whether embedding is from WX or Y, we proceed to next step

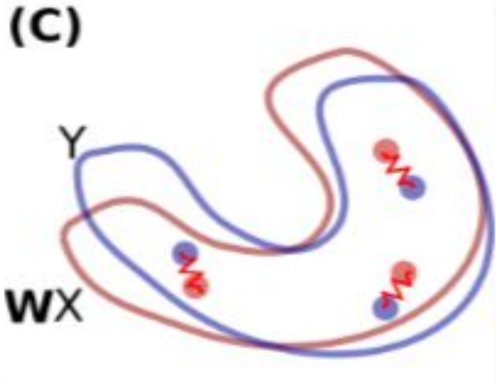
*Note: it is unclear whether two embeddings are fed to discriminator at a time and discriminator tries to determine if they are from same source, or if 1 embedding is fed to discriminator at a time and discriminator ties to determine source of embedding

Structure

- Introduction
- **Model**
    - Domain adversarial setting
    - **Refinement**
- Experiments/Evaluation
- Conclusion

Going from B to C:





(we just saw how to get B, now time to get C)

keep only translation pairs from WX and Y that are frequent and mutual K-NN

The Procrustes problem

matrix X

matrix Y

matrix X matrix Y

get the best linear map W

matrix X matrix Y

get the best linear map W

$$W^\star = \underset{W \in O_d(\mathbb{R})}{\operatorname{argmin}} \|WX - Y\|_F = UV^T$$

matrix X

matrix Y

get the best linear map W

i.e., the one that minimizes the
difference between WX and Y

$$W^\star = \underset{W \in O_d(\mathbb{R})}{\operatorname{argmin}} \|\overbrace{WX - Y}\|_{\mathrm{F}} = UV^T$$

matrix X

matrix Y

get the best linear map W

i.e., the one that minimizes the difference between WX and Y

$$W^{\star} = \underset{W \in O_d(\mathbb{R})}{\operatorname{argmin}} \|WX - Y\|_{\mathrm{F}} = UV^T$$

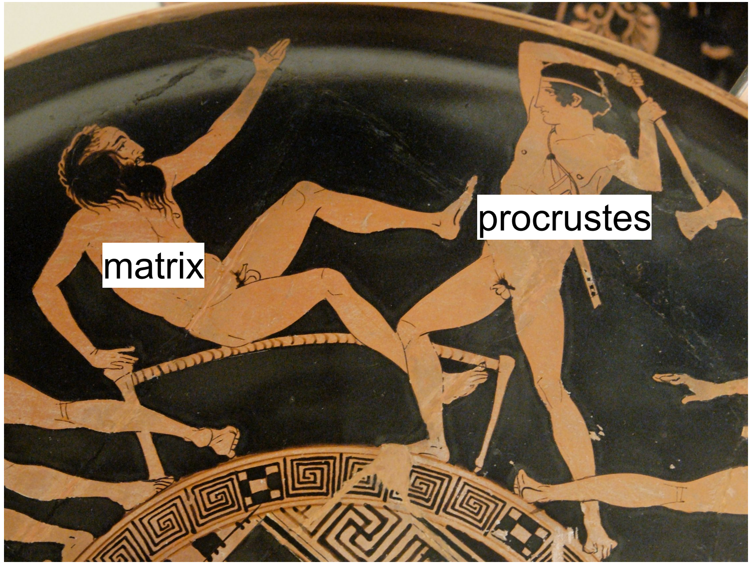
$$\mathrm{SVD}(YX^T)$$

$$\mathrm{SVD}(YX^T) = U\Sigma V^T$$

$$\mathrm{SVD}(YX^T) \quad = \quad U\cancel{\Sigma}V^T$$

$$\mathrm{SVD}(YX^T) = U\cancel{\Sigma}V^T = UV^T$$

$$\mathrm{SVD}(YX^T) \quad = \quad U\cancel{\Sigma}V^T \quad = \quad UV^T$$



Cross-domain Similarity Local Scaling (CSLS)

CSLS

to choose translation pairs from
WX and Y that are mutual K-NN

CSLS

CSLS

to choose translation pairs from
WX and Y that are mutual K-NN

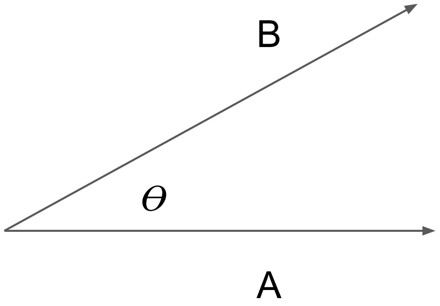
to evaluate the closeness
of the spaces WX and Y

CSLS

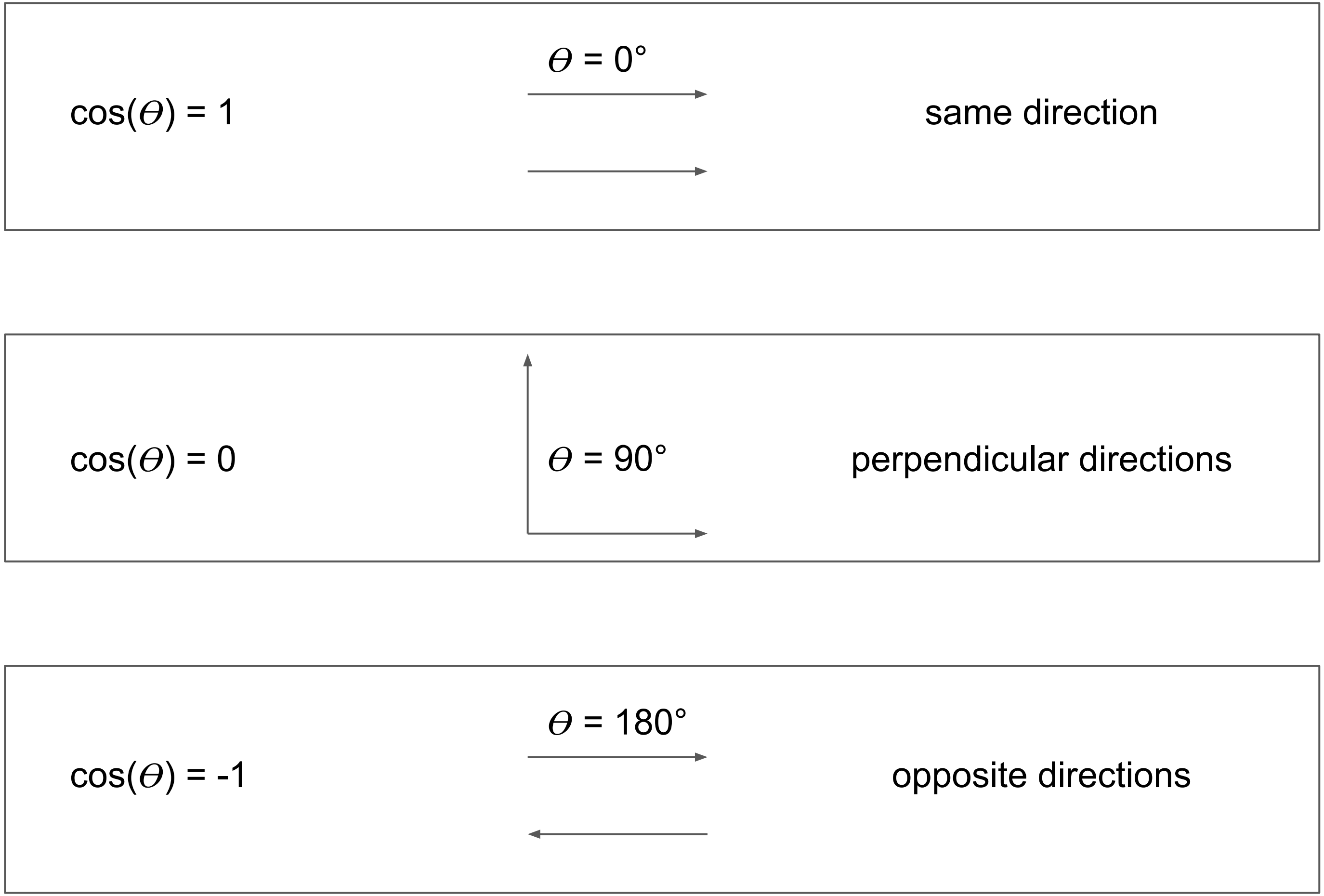
to choose translation pairs from
WX and Y that are mutual K-NN

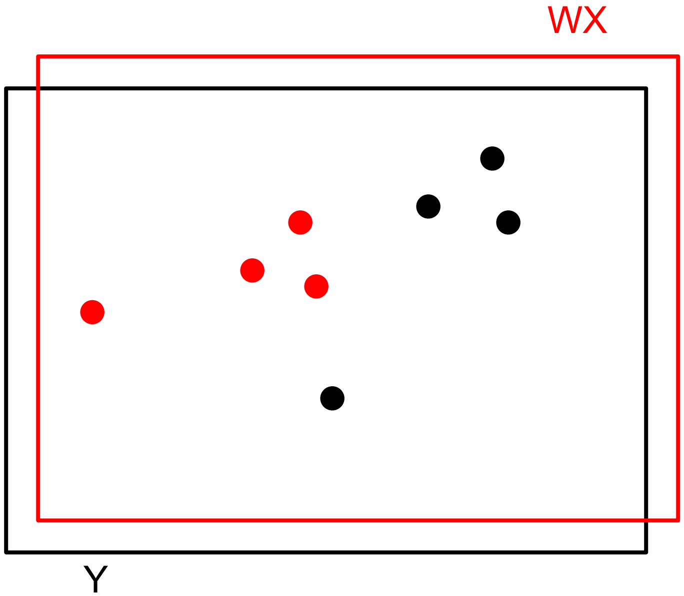
to evaluate the closeness
of the spaces WX and Y

since we don’t have a validation
set as in supervised approaches!

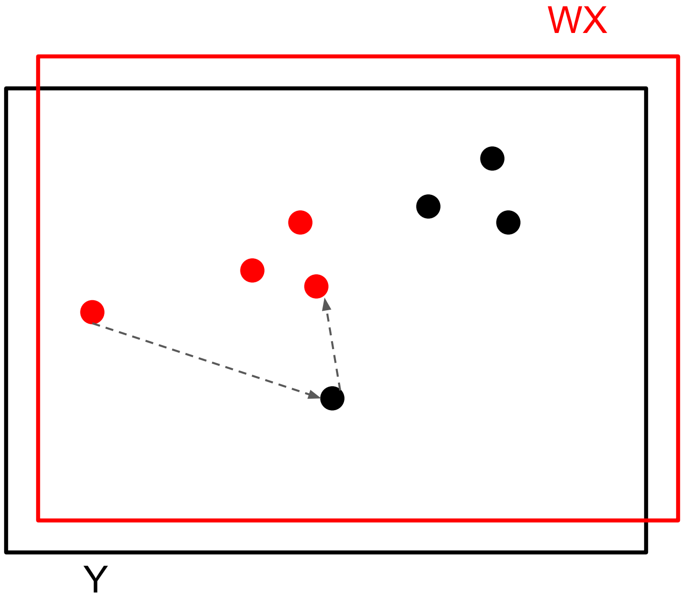


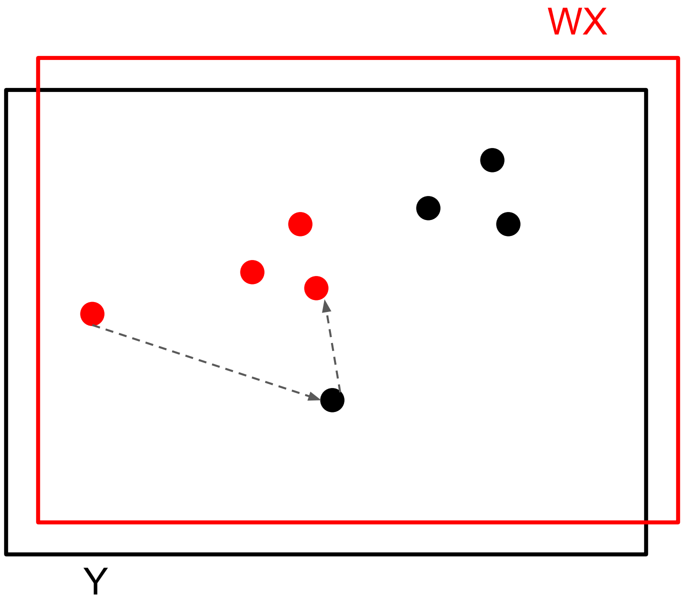
$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}$$


θ = 0°
cos(θ) = 1
same direction
cos(θ) = 0
θ = 90°
perpendicular directions
θ = 180°
cos(θ) = -1
opposite directions

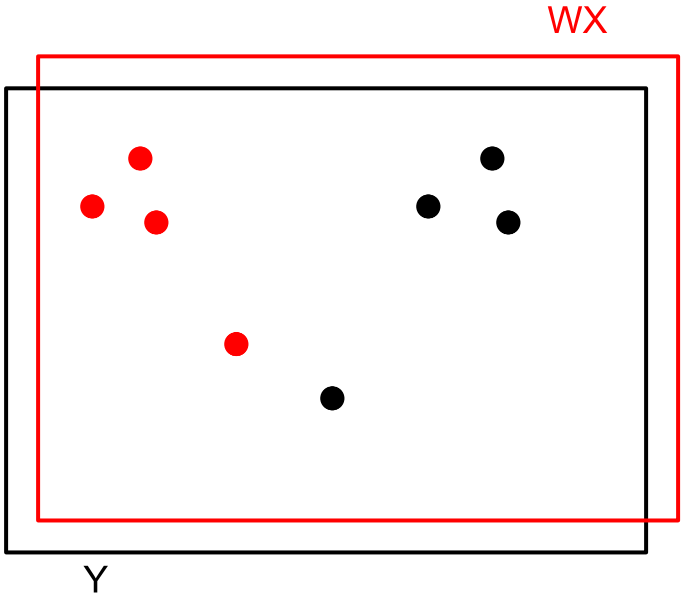


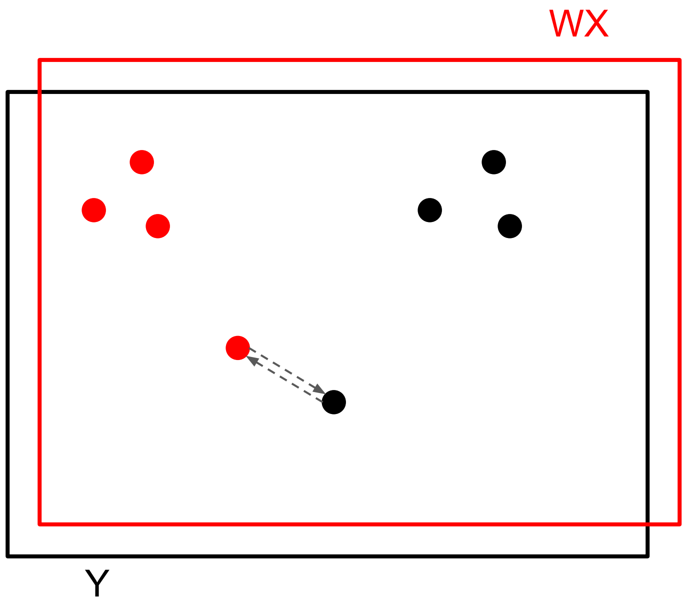
WX
Y

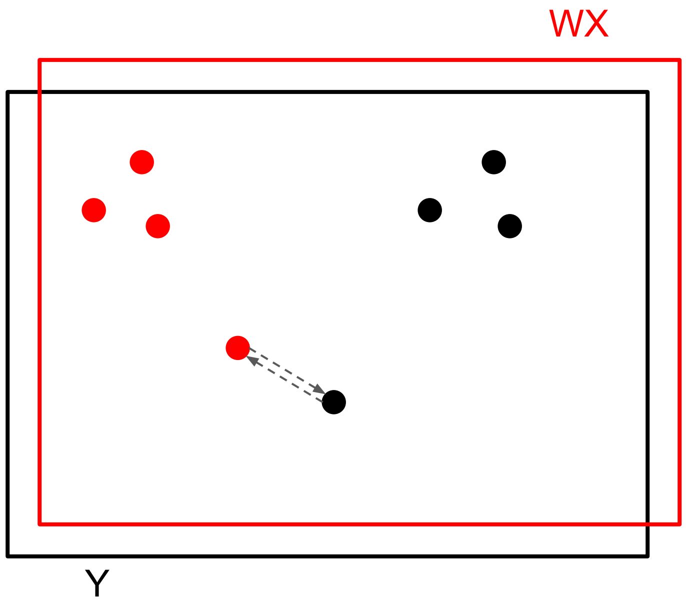

WX
Y



❌ not mutual nearest neighbours

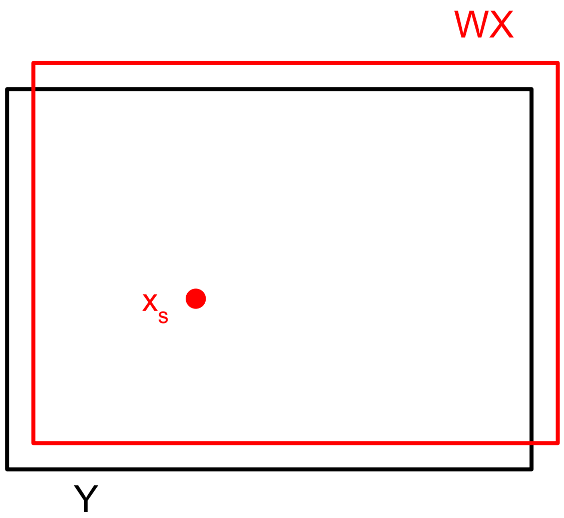

WX
Y


WX
Y



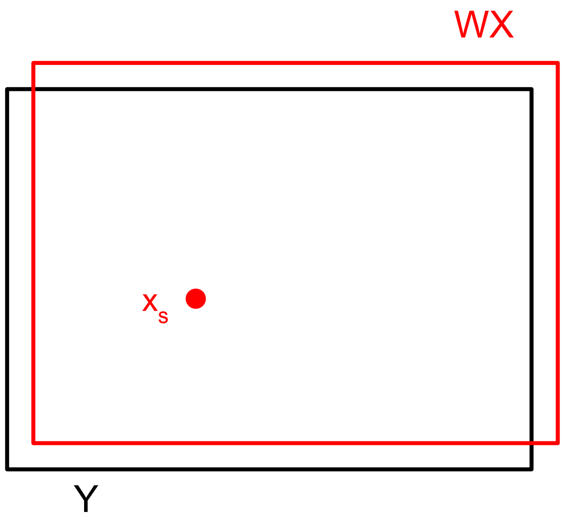
✓ mutual nearest neighbours

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



mean similarity

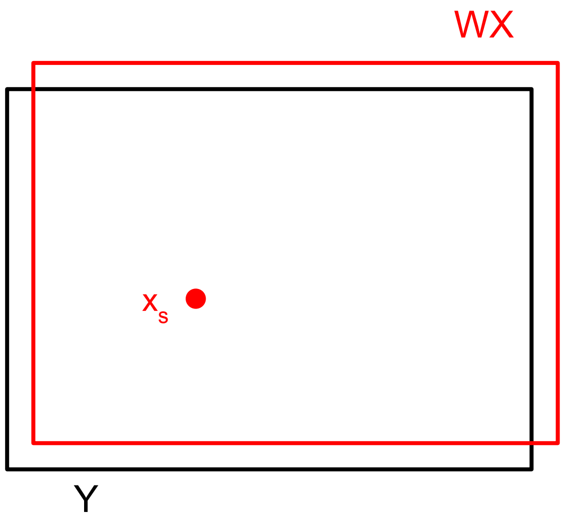
$$\boxed{r_{\mathrm{T}}(Wx_s)} = \frac{1}{K} \sum_{y_t \in \mathcal{N}_{\mathrm{T}}(Wx_s)} \cos(Wx_s, y_t),$$



mean similarity

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

number of target words
in target neighbourhood

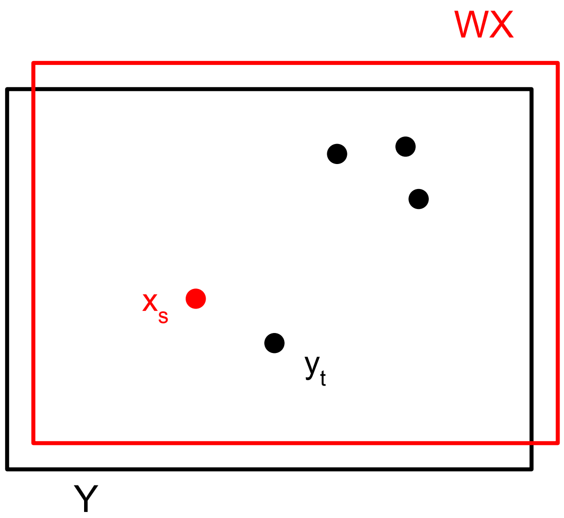


mean similarity

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

for target word in
target neighbourhood

number of target words
in target neighbourhood



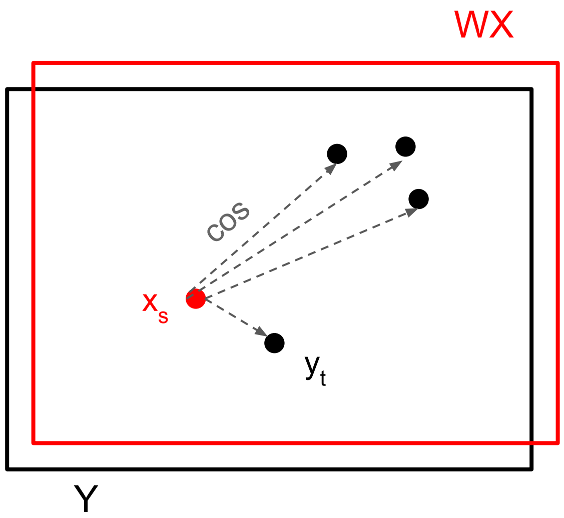
cosine similarity between source
word mapping and target word

mean similarity

$$\boxed{r_{\mathrm{T}}(Wx_s)} = \frac{1}{K} \underbrace{\sum_{y_t \in \mathcal{N}_{\mathrm{T}}(Wx_s)}}_{\text{for target word in target neighbourhood}} \overbrace{\cos(Wx_s, y_t)}^{\text{cosine similarity between source word mapping and target word}},$$

for target word in
target neighbourhood

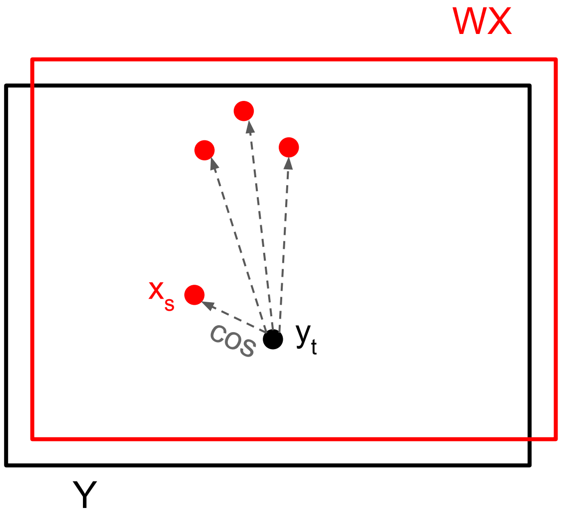
number of target words
in target neighbourhood



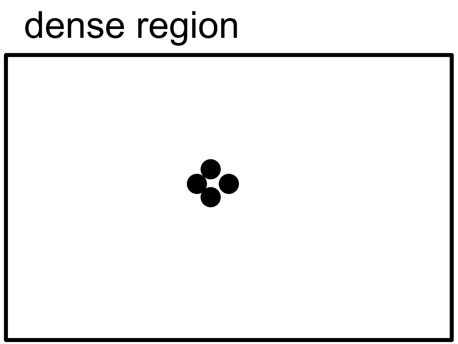
mean similarity

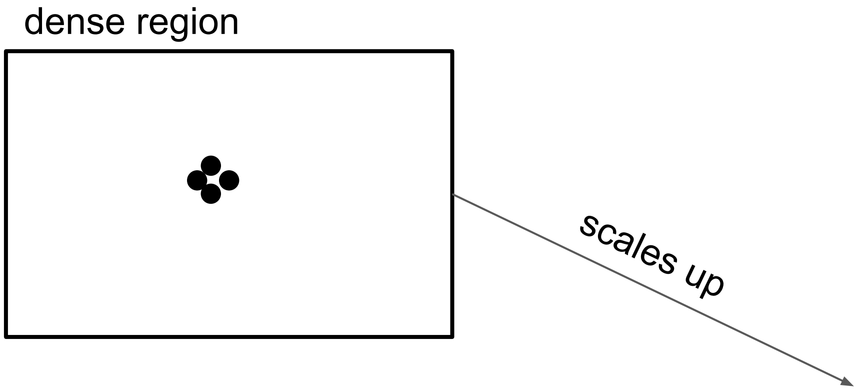
$$\boxed{r_S(y_t)} = \frac{1}{K} \underbrace{\sum_{Wx_s \in N_s(y_t)}}_{\text{for mapped word in mapped neighbourhood}} \overbrace{\cos(Y_t, Wx_s)}^{\text{cosine similarity between target and mapped word}},$$

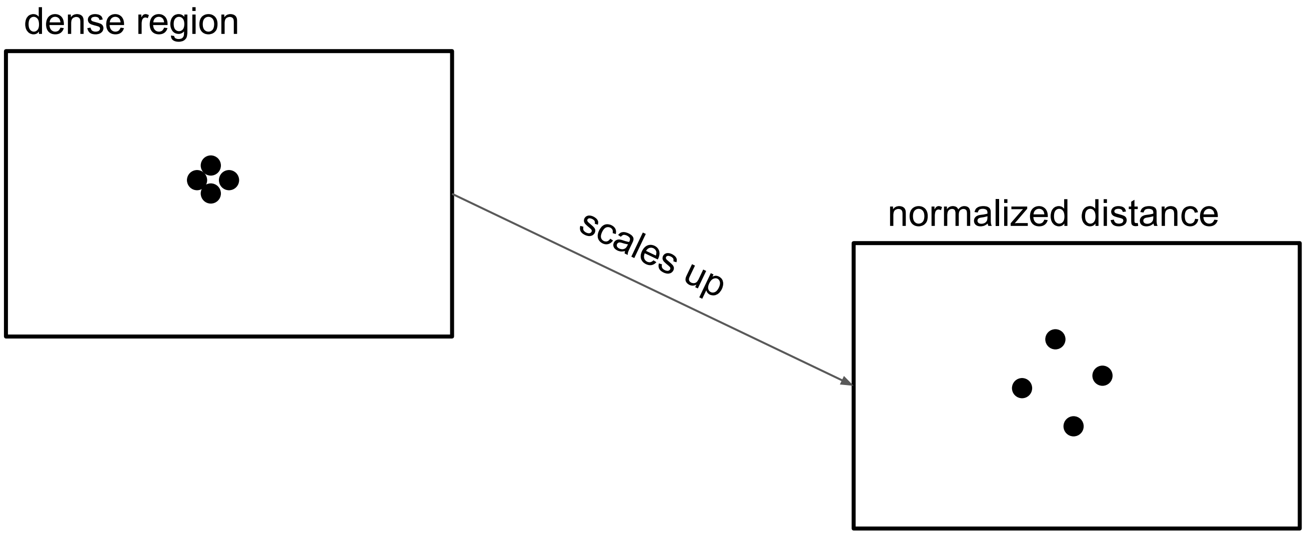
number of mapped words in
mapped neighbourhood

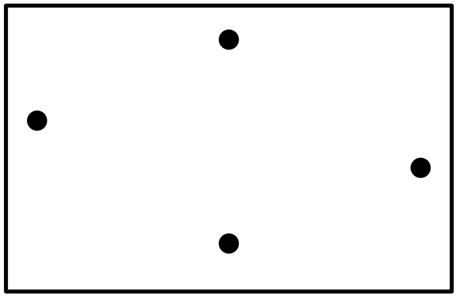


$$\mathrm{CSLS}(Wx_s, y_t) = 2\cos(Wx_s, y_t) - r_{\mathrm{T}}(Wx_s) - r_{\mathrm{S}}(y_t)$$

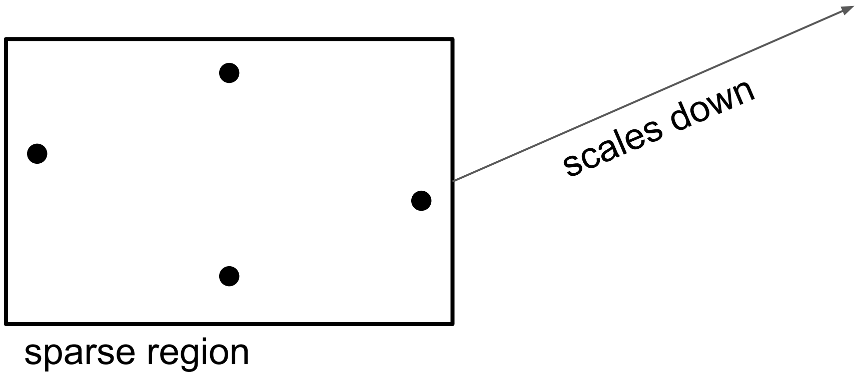

dense region

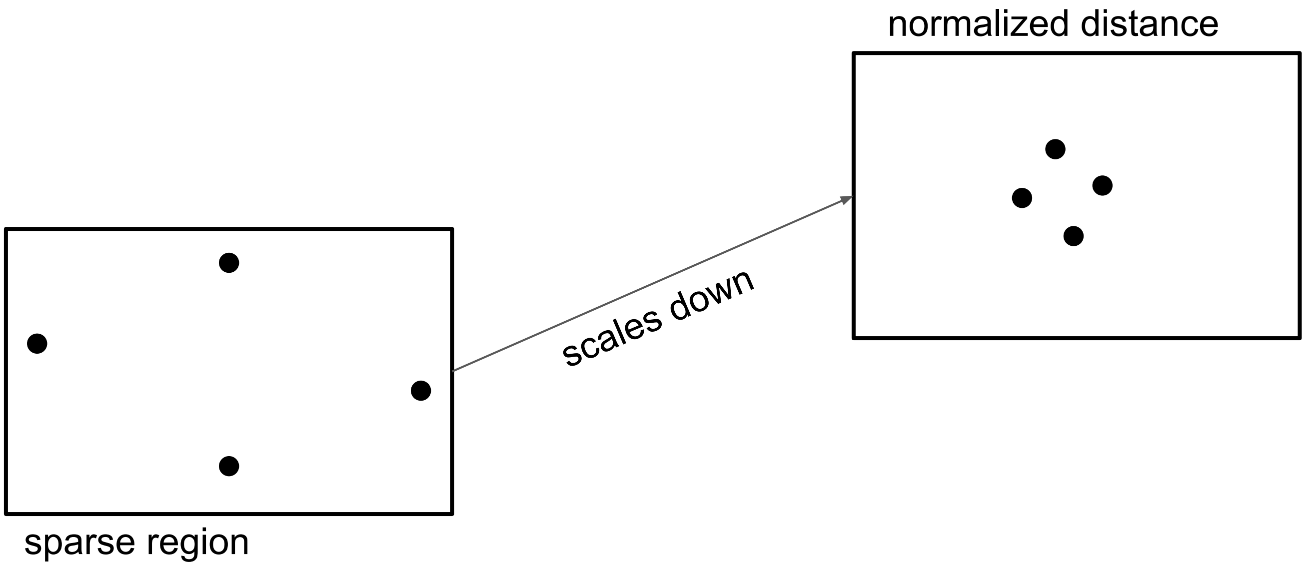

dense region
scales up


dense region
scales up
normalized distance

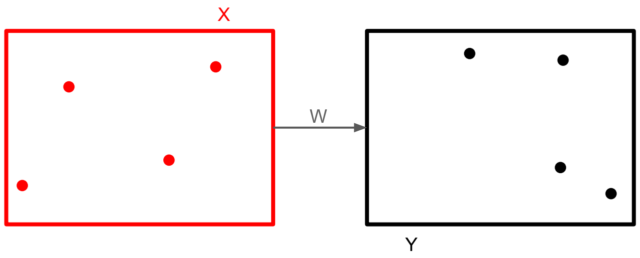


sparse region

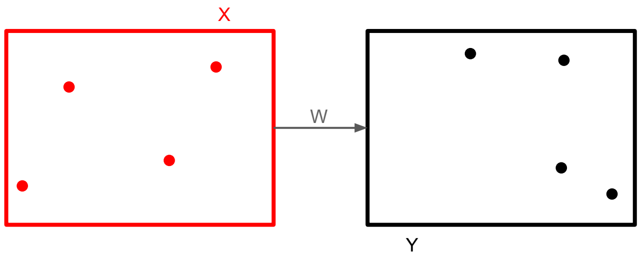

scales down
sparse region

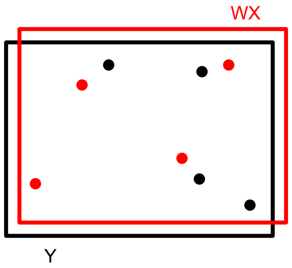

normalized distance
scales down
sparse region

Let’s take a look at the pipeline again

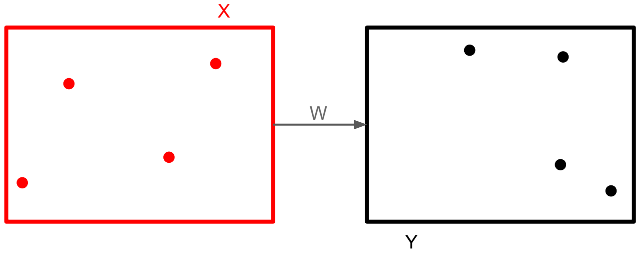


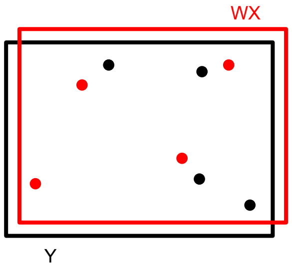
learn an initial W
with GANs

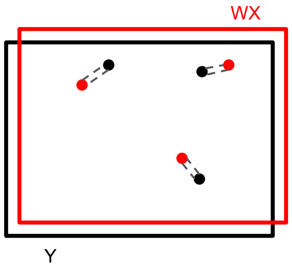




learn an initial W with GANs → get WX





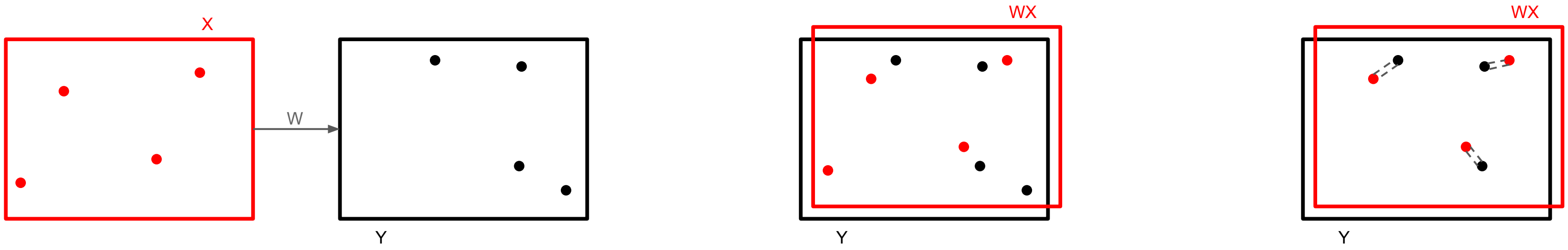


learn an initial W
with GANs

get WX

keep only translation pairs
from WX and Y that are
frequent and mutual K-NN*

(*) CSLS



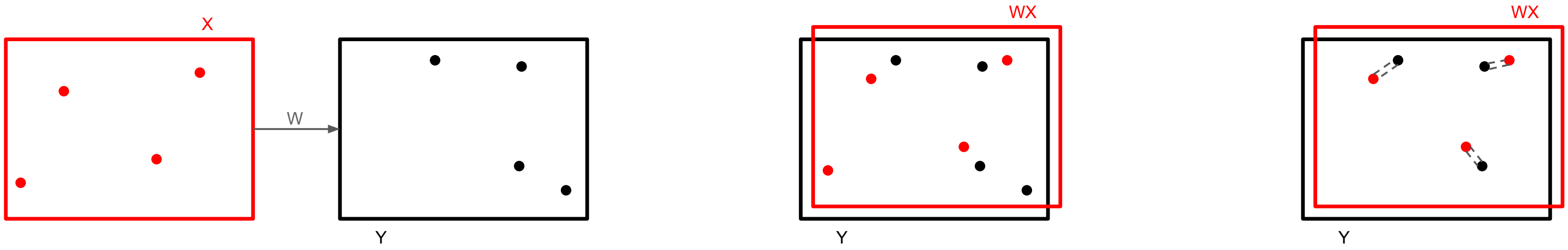
learn an initial W
with GANs

get WX

keep only translation pairs
from WX and Y that are
frequent and mutual K-NN*

use Procustes to
get a new W

(*) CSLS



learn an initial W with GANs

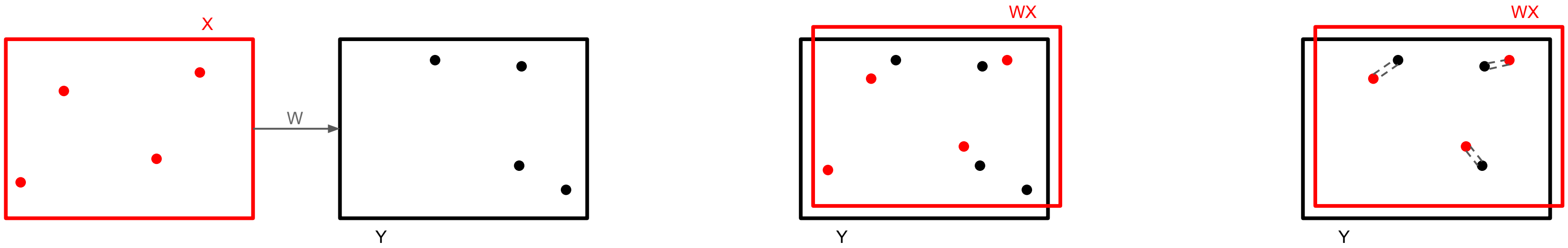
get WX

keep only translation pairs from WX and Y that are frequent and mutual K-NN*

use Procustes to get a new W

evaluate*

(*) CSLS



learn an initial W with GANs

get WX

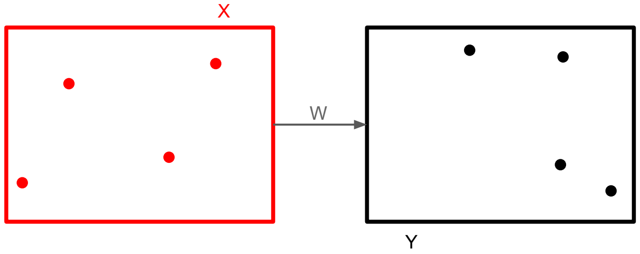
keep only translation pairs from WX and Y that are frequent and mutual K-NN*

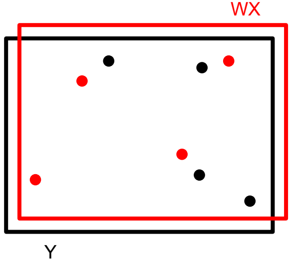
use Procustes to get a new W

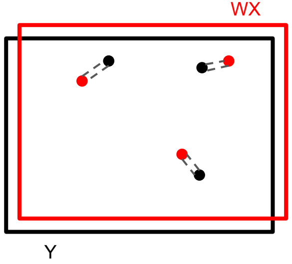
evaluate*

❌

(*) CSLS







learn an initial W with GANs

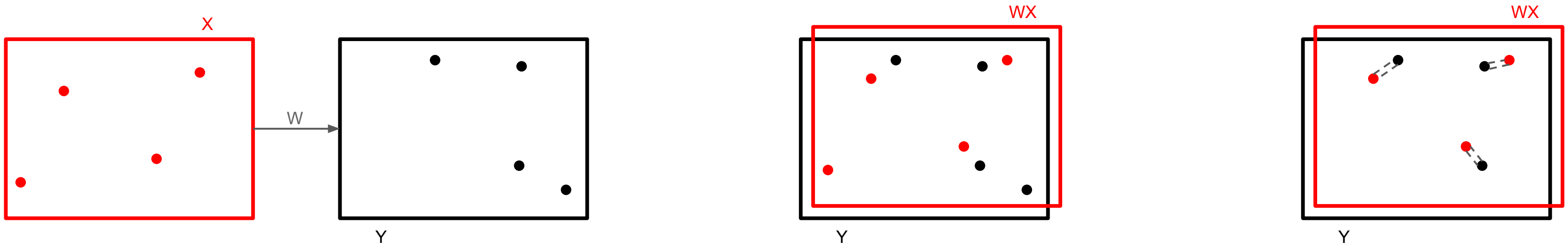
get WX

keep only translation pairs from WX and Y that are frequent and mutual K-NN*

use Procustes to get a new W

evaluate*

(*) CSLS



learn an initial W with GANs

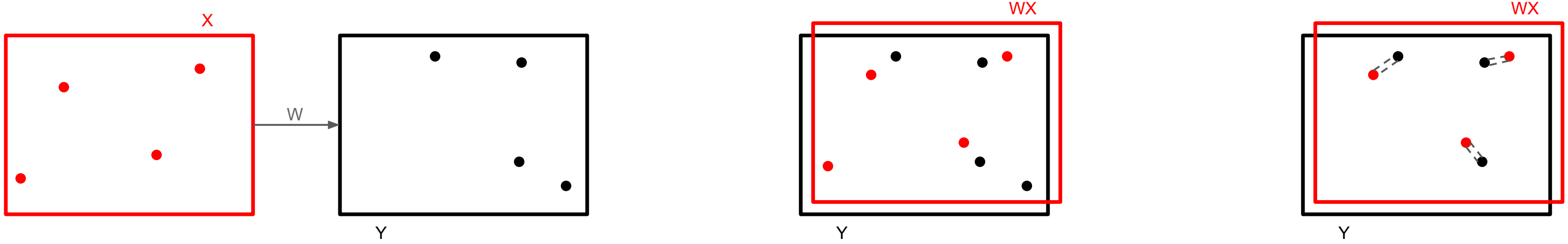
get WX

keep only translation pairs from WX and Y that are frequent and mutual K-NN*

use Procustes to get a new W

evaluate*

(*) CSLS



learn an initial W with GANs

get WX

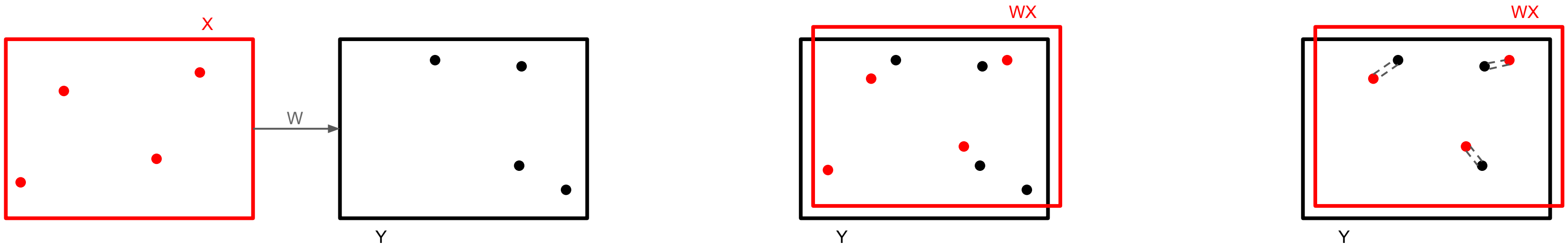
keep only translation pairs from WX and Y that are frequent and mutual K-NN*

use Procustes to get a new W

evaluate*

✓

(*) CSLS



learn an initial W
with GANs

get WX

keep only translation pairs
from WX and Y that are
frequent and mutual K-NN*

use Procustes to
get a new W

evaluate*

✔

(*) CSLS

Structure

- Introduction
- Model
  - Domain adversarial setting
  - Refinement
- **Experiments/Evaluation**
- Conclusion

(1) Word translation retrieval

	en-es	es-en	en-fr	fr-en	en-de	de-en	en-ru	ru-en	en-zh	zh-en	en-eo	eo-en
Methods with cross-lingual supervision and fastText embeddings												
Procrustes - NN	77.4	77.3	74.9	76.1	68.4	67.7	47.0	58.2	40.6	30.2	22.1	20.4
Procrustes - ISF	81.1	82.6	81.1	81.3	71.1	71.5	49.5	63.8	35.7	**37.5**	29.0	27.9
Procrustes - CSLS	81.4	82.9	81.1	**82.4**	73.5	**72.4**	**51.7**	**63.7**	**42.7**	36.7	**29.3**	25.3
Methods without cross-lingual supervision and fastText embeddings												
Adv - NN	69.8	71.3	70.4	61.9	63.1	59.6	29.1	41.5	18.5	22.3	13.5	12.1
Adv - CSLS	75.7	79.7	77.8	71.2	70.1	66.4	37.2	48.1	23.4	28.3	18.6	16.6
Adv - Refine - NN	79.1	78.1	78.1	78.2	71.3	69.6	37.3	54.3	30.9	21.9	20.7	20.6
Adv - Refine - CSLS	**81.7**	**83.3**	**82.3**	82.1	**74.0**	72.2	44.0	59.1	32.5	31.4	28.2	**25.6**

Table 1: Word translation retrieval P@1 for our released vocabularies in various language pairs. We consider 1,500 source test queries, and 200k target words for each language pair. We use fastText embeddings trained on Wikipedia. NN: nearest neighbors. ISF: inverted softmax. ('en' is English, 'fr' is French, 'de' is German, 'ru' is Russian, 'zh' is classical Chinese and 'eo' is Esperanto)

(1) Word translation retrieval

supervised

unsupervised

	en-es	es-en	en-fr	fr-en	en-de	de-en	en-ru	ru-en	en-zh	zh-en	en-eo	eo-en
Methods with cross-lingual supervision and fastText embeddings												
Procrustes - NN	77.4	77.3	74.9	76.1	68.4	67.7	47.0	58.2	40.6	30.2	22.1	20.4
Procrustes - ISF	81.1	82.6	81.1	81.3	71.1	71.5	49.5	63.8	35.7	**37.5**	29.0	27.9
Procrustes - CSLS	81.4	82.9	81.1	**82.4**	73.5	**72.4**	**51.7**	**63.7**	**42.7**	36.7	**29.3**	25.3
Methods without cross-lingual supervision and fastText embeddings												
Adv - NN	69.8	71.3	70.4	61.9	63.1	59.6	29.1	41.5	18.5	22.3	13.5	12.1
Adv - CSLS	75.7	79.7	77.8	71.2	70.1	66.4	37.2	48.1	23.4	28.3	18.6	16.6
Adv - Refine - NN	79.1	78.1	78.1	78.2	71.3	69.6	37.3	54.3	30.9	21.9	20.7	20.6
Adv - Refine - CSLS	**81.7**	**83.3**	**82.3**	82.1	**74.0**	72.2	44.0	59.1	32.5	31.4	28.2	**25.6**

Table 1: Word translation retrieval P@1 for our released vocabularies in various language pairs. We consider 1,500 source test queries, and 200k target words for each language pair. We use fastText embeddings trained on Wikipedia. NN: nearest neighbors. ISF: inverted softmax. ('en' is English, 'fr' is French, 'de' is German, 'ru' is Russian, 'zh' is classical Chinese and 'eo' is Esperanto)

(1) Word translation retrieval

combinations with different similarity measures

	en-es	es-en	en-fr	fr-en	en-de	de-en	en-ru	ru-en	en-zh	zh-en	en-eo	eo-en
Methods with cross-lingual supervision and fastText embeddings												
Procrustes - NN	77.4	77.3	74.9	76.1	68.4	67.7	47.0	58.2	40.6	30.2	22.1	20.4
Procrustes - ISF	81.1	82.6	81.1	81.3	71.1	71.5	49.5	63.8	35.7	**37.5**	29.0	27.9
Procrustes - CSLS	81.4	82.9	81.1	**82.4**	73.5	**72.4**	**51.7**	**63.7**	**42.7**	36.7	**29.3**	25.3
Methods without cross-lingual supervision and fastText embeddings												
Adv - NN	69.8	71.3	70.4	61.9	63.1	59.6	29.1	41.5	18.5	22.3	13.5	12.1
Adv - CSLS	75.7	79.7	77.8	71.2	70.1	66.4	37.2	48.1	23.4	28.3	18.6	16.6
Adv - Refine - NN	79.1	78.1	78.1	78.2	71.3	69.6	37.3	54.3	30.9	21.9	20.7	20.6
Adv - Refine - CSLS	**81.7**	**83.3**	**82.3**	82.1	**74.0**	72.2	44.0	59.1	32.5	31.4	28.2	**25.6**

supervised

unsupervised

Table 1: Word translation retrieval P@1 for our released vocabularies in various language pairs. We consider 1,500 source test queries, and 200k target words for each language pair. We use fastText embeddings trained on Wikipedia. NN: nearest neighbors. ISF: inverted softmax. ('en' is English, 'fr' is French, 'de' is German, 'ru' is Russian, 'zh' is classical Chinese and 'eo' is Esperanto)

(1) Word translation retrieval

combinations with different similarity measures

supervised

unsupervised

	en-es	es-en	en-fr	fr-en	en-de	de-en	en-ru	ru-en	en-zh	zh-en	en-eo	eo-en
Methods with cross-lingual supervision and fastText embeddings												
Procrustes - NN	77.4	77.3	74.9	76.1	68.4	67.7	47.0	58.2	40.6	30.2	22.1	20.4
Procrustes - ISF	81.1	82.6	81.1	81.3	71.1	71.5	49.5	63.8	35.7	**37.5**	29.0	27.9
Procrustes - CSLS	81.4	82.9	81.1	**82.4**	73.5	**72.4**	**51.7**	**63.7**	**42.7**	36.7	**29.3**	25.3
Methods without cross-lingual supervision and fastText embeddings												
Adv - NN	69.8	71.3	70.4	61.9	63.1	59.6	29.1	41.5	18.5	22.3	13.5	12.1
Adv - CSLS	75.7	79.7	77.8	71.2	70.1	66.4	37.2	48.1	23.4	28.3	18.6	16.6
Adv - Refine - NN	79.1	78.1	78.1	78.2	71.3	69.6	37.3	54.3	30.9	21.9	20.7	20.6
Adv - Refine - CSLS	**81.7**	**83.3**	**82.3**	82.1	**74.0**	72.2	44.0	59.1	32.5	31.4	28.2	**25.6**

Table 1: Word translation retrieval P@1 for our released vocabularies in various language pairs. We consider 1,500 source test queries, and 200k target words for each language pair. We use fastText embeddings trained on Wikipedia. NN: nearest neighbors. ISF: inverted softmax. ('en' is English, 'fr' is French, 'de' is German, 'ru' is Russian, 'zh' is classical Chinese and 'eo' is Esperanto)

(1) Word translation retrieval

combinations with different similarity measures

supervised

unsupervised

	en-es	es-en	en-fr	fr-en	en-de	de-en	en-ru	ru-en	en-zh	zh-en	en-eo	eo-en
Methods with cross-lingual supervision and fastText embeddings												
Procrustes - NN	77.4	77.3	74.9	76.1	68.4	67.7	47.0	58.2	40.6	30.2	22.1	20.4
Procrustes - ISF	81.1	82.6	81.1	81.3	71.1	71.5	49.5	63.8	35.7	**37.5**	29.0	27.9
Procrustes - CSLS	81.4	82.9	81.1	**82.4**	73.5	**72.4**	**51.7**	**63.7**	**42.7**	36.7	**29.3**	25.3
Methods without cross-lingual supervision and fastText embeddings												
Adv - NN	69.8	71.3	70.4	61.9	63.1	59.6	29.1	41.5	18.5	22.3	13.5	12.1
Adv - CSLS	75.7	79.7	77.8	71.2	70.1	66.4	37.2	48.1	23.4	28.3	18.6	16.6
Adv - Refine - NN	79.1	78.1	78.1	78.2	71.3	69.6	37.3	54.3	30.9	21.9	20.7	20.6
Adv - Refine - CSLS	**81.7**	**83.3**	**82.3**	82.1	**74.0**	72.2	44.0	59.1	32.5	31.4	28.2	**25.6**

Table 1: Word translation retrieval P@1 for our released vocabularies in various language pairs. We consider 1,500 source test queries, and 200k target words for each language pair. We use fastText embeddings trained on Wikipedia. NN: nearest neighbors. ISF: inverted softmax. ('en' is English, 'fr' is French, 'de' is German, 'ru' is Russian, 'zh' is classical Chinese and 'eo' is Esperanto)

✓ CSLS gives the best results

(1) Word translation retrieval

combinations with different similarity measures

supervised

unsupervised

	en-es	es-en	en-fr	fr-en	en-de	de-en	en-ru	ru-en	en-zh	zh-en	en-eo	eo-en
Methods with cross-lingual supervision and fastText embeddings												
Procrustes - NN	77.4	77.3	74.9	76.1	68.4	67.7	47.0	58.2	40.6	30.2	22.1	20.4
Procrustes - ISF	81.1	82.6	81.1	81.3	71.1	71.5	49.5	63.8	35.7	**37.5**	29.0	27.9
Procrustes - CSLS	81.4	82.9	81.1	**82.4**	73.5	**72.4**	**51.7**	**63.7**	**42.7**	36.7	**29.3**	25.3
Methods without cross-lingual supervision and fastText embeddings												
Adv - NN	69.8	71.3	70.4	61.9	63.1	59.6	29.1	41.5	18.5	22.3	13.5	12.1
Adv - CSLS	75.7	79.7	77.8	71.2	70.1	66.4	37.2	48.1	23.4	28.3	18.6	16.6
Adv - Refine - NN	79.1	78.1	78.1	78.2	71.3	69.6	37.3	54.3	30.9	21.9	20.7	20.6
Adv - Refine - CSLS	**81.7**	**83.3**	**82.3**	82.1	**74.0**	72.2	44.0	59.1	32.5	31.4	28.2	**25.6**

Table 1: Word translation retrieval P@1 for our released vocabularies in various language pairs. We consider 1,500 source test queries, and 200k target words for each language pair. We use fastText embeddings trained on Wikipedia. NN: nearest neighbors. ISF: inverted softmax. ('en' is English, 'fr' is French, 'de' is German, 'ru' is Russian, 'zh' is classical Chinese and 'eo' is Esperanto)

✓ CSLS gives the best results

✓ supervised and unsupervised approaches on par (thanks to boost with CSLS)

(1) Word translation retrieval

combinations with different similarity measures

outperforms for low-resourced languages

supervised

unsupervised

	en-es	es-en	en-fr	fr-en	en-de	de-en	en-ru	ru-en	en-zh	zh-en	en-eo	eo-en
Methods with cross-lingual supervision and fastText embeddings												
Procrustes - NN	77.4	77.3	74.9	76.1	68.4	67.7	47.0	58.2	40.6	30.2	22.1	20.4
Procrustes - ISF	81.1	82.6	81.1	81.3	71.1	71.5	49.5	63.8	35.7	**37.5**	29.0	27.9
Procrustes - CSLS	81.4	82.9	81.1	**82.4**	73.5	**72.4**	**51.7**	**63.7**	**42.7**	36.7	**29.3**	25.3
Methods without cross-lingual supervision and fastText embeddings												
Adv - NN	69.8	71.3	70.4	61.9	63.1	59.6	29.1	41.5	18.5	22.3	13.5	12.1
Adv - CSLS	75.7	79.7	77.8	71.2	70.1	66.4	37.2	48.1	23.4	28.3	18.6	16.6
Adv - Refine - NN	79.1	78.1	78.1	78.2	71.3	69.6	37.3	54.3	30.9	21.9	20.7	20.6
Adv - Refine - CSLS	**81.7**	**83.3**	**82.3**	82.1	**74.0**	72.2	44.0	59.1	32.5	31.4	28.2	**25.6**

Table 1: Word translation retrieval P@1 for our released vocabularies in various language pairs. We consider 1,500 source test queries, and 200k target words for each language pair. We use fastText embeddings trained on Wikipedia. NN: nearest neighbors. ISF: inverted softmax. ('en' is English, 'fr' is French, 'de' is German, 'ru' is Russian, 'zh' is classical Chinese and 'eo' is Esperanto)

✓ CSLS gives the best results

✓ supervised and unsupervised approaches on par (thanks to boost with CSLS)

(1) Word translation retrieval

	English to Italian			Italian to English		
	P@1	P@5	P@10	P@1	P@5	P@10
Methods with cross-lingual supervision (WaCky)						
Mikolov et al. (2013b) †	33.8	48.3	53.9	24.9	41.0	47.4
Dinu et al. (2015)†	38.5	56.4	63.9	24.6	45.4	54.1
CCA†	36.1	52.7	58.1	31.0	49.9	57.0
Artetxe et al. (2017)	39.7	54.7	60.5	33.8	52.4	59.1
Smith et al. (2017)†	43.1	60.7	66.4	38.0	58.5	63.6
Procrustes - CSLS	44.9	61.8	66.6	38.5	57.2	63.0
Methods without cross-lingual supervision (WaCky)						
Adv - Refine - CSLS	45.1	60.7	65.1	38.3	57.8	62.8
Methods with cross-lingual supervision (Wiki)						
Procrustes - CSLS	63.7	78.6	81.1	56.3	76.2	80.6
Methods without cross-lingual supervision (Wiki)						
Adv - Refine - CSLS	**66.2**	**80.4**	**83.4**	**58.7**	**76.5**	**80.9**

Table 2: English-Italian word translation average precisions (@1, @5, @10) from 1.5k source word queries using 200k target words. Results marked with the symbol † are from Smith et al. (2017). Wiki means the embeddings were trained on Wikipedia using fastText. Note that the method used by Artetxe et al. (2017) does not use the same supervision as other supervised methods, as they only use numbers in their initial parallel dictionary.

(1) Word translation retrieval

waCky

Wiki

	English to Italian			Italian to English		
	P@1	P@5	P@10	P@1	P@5	P@10
Methods with cross-lingual supervision (WaCky)						
Mikolov et al. (2013b) †	33.8	48.3	53.9	24.9	41.0	47.4
Dinu et al. (2015)†	38.5	56.4	63.9	24.6	45.4	54.1
CCA†	36.1	52.7	58.1	31.0	49.9	57.0
Artetxe et al. (2017)	39.7	54.7	60.5	33.8	52.4	59.1
Smith et al. (2017)†	43.1	60.7	66.4	38.0	58.5	63.6
Procrustes - CSLS	44.9	61.8	66.6	38.5	57.2	63.0
Methods without cross-lingual supervision (WaCky)						
Adv - Refine - CSLS	45.1	60.7	65.1	38.3	57.8	62.8
Methods with cross-lingual supervision (Wiki)						
Procrustes - CSLS	63.7	78.6	81.1	56.3	76.2	80.6
Methods without cross-lingual supervision (Wiki)						
Adv - Refine - CSLS	**66.2**	**80.4**	**83.4**	**58.7**	**76.5**	**80.9**

Table 2: English-Italian word translation average precisions (@1, @5, @10) from 1.5k source word queries using 200k target words. Results marked with the symbol † are from Smith et al. (2017). Wiki means the embeddings were trained on Wikipedia using fastText. Note that the method used by Artetxe et al. (2017) does not use the same supervision as other supervised methods, as they only use numbers in their initial parallel dictionary.

(1) Word translation retrieval

waCky

Wiki

	English to Italian			Italian to English		
	P@1	P@5	P@10	P@1	P@5	P@10
Methods with cross-lingual supervision (WaCky)						
Mikolov et al. (2013b) †	33.8	48.3	53.9	24.9	41.0	47.4
Dinu et al. (2015)†	38.5	56.4	63.9	24.6	45.4	54.1
CCA†	36.1	52.7	58.1	31.0	49.9	57.0
Artetxe et al. (2017)	39.7	54.7	60.5	33.8	52.4	59.1
Smith et al. (2017)†	43.1	60.7	66.4	38.0	58.5	63.6
Procrustes - CSLS	44.9	61.8	66.6	38.5	57.2	63.0
Methods without cross-lingual supervision (WaCky)						
Adv - Refine - CSLS	45.1	60.7	65.1	38.3	57.8	62.8
Methods with cross-lingual supervision (Wiki)						
Procrustes - CSLS	63.7	78.6	81.1	56.3	76.2	80.6
Methods without cross-lingual supervision (Wiki)						
Adv - Refine - CSLS	**66.2**	**80.4**	**83.4**	**58.7**	**76.5**	**80.9**

Table 2: English-Italian word translation average precisions (@1, @5, @10) from 1.5k source word queries using 200k target words. Results marked with the symbol † are from Smith et al. (2017). Wiki means the embeddings were trained on Wikipedia using fastText. Note that the method used by Artetxe et al. (2017) does not use the same supervision as other supervised methods, as they only use numbers in their initial parallel dictionary.

✓ outperforms all previous approaches

(2) Sentence translation retrieval

	English to Italian			Italian to English		
	P@1	P@5	P@10	P@1	P@5	P@10
Methods with cross-lingual supervision						
Mikolov et al. (2013b) †	10.5	18.7	22.8	12.0	22.1	26.7
Dinu et al. (2015) †	45.3	72.4	80.7	48.9	71.3	78.3
Smith et al. (2017) †	54.6	72.7	78.2	42.9	62.2	69.2
Procrustes - NN	42.6	54.7	59.0	53.5	65.5	69.5
Procrustes - CSLS	**66.1**	77.1	80.7	**69.5**	**79.6**	**83.5**
Methods without cross-lingual supervision						
Adv - CSLS	42.5	57.6	63.6	47.0	62.1	67.8
Adv - Refine - CSLS	65.9	**79.7**	**83.1**	69.0	**79.7**	83.1

Table 3: English-Italian sentence translation retrieval. We report the average P@k from 2,000 source queries using 200,000 target sentences. We use the same embeddings as in Smith et al. (2017). Their results are marked with the symbol †.

(2) Sentence translation retrieval

	English to Italian			Italian to English		
	P@1	P@5	P@10	P@1	P@5	P@10
Methods with cross-lingual supervision						
Mikolov et al. (2013b) †	10.5	18.7	22.8	12.0	22.1	26.7
Dinu et al. (2015) †	45.3	72.4	80.7	48.9	71.3	78.3
Smith et al. (2017) †	54.6	72.7	78.2	42.9	62.2	69.2
Procrustes - NN	42.6	54.7	59.0	53.5	65.5	69.5
Procrustes - CSLS	**66.1**	77.1	80.7	**69.5**	**79.6**	**83.5**
Methods without cross-lingual supervision						
Adv - CSLS	42.5	57.6	63.6	47.0	62.1	67.8
Adv - Refine - CSLS	65.9	**79.7**	**83.1**	69.0	**79.7**	83.1

Table 3: English-Italian sentence translation retrieval. We report the average P@k from 2,000 source queries using 200,000 target sentences. We use the same embeddings as in Smith et al. (2017). Their results are marked with the symbol †.

✔ CSLS outperforms all previous approaches

(2) Sentence translation retrieval

	English to Italian			Italian to English		
	P@1	P@5	P@10	P@1	P@5	P@10
Methods with cross-lingual supervision						
Mikolov et al. (2013b) †	10.5	18.7	22.8	12.0	22.1	26.7
Dinu et al. (2015) †	45.3	72.4	80.7	48.9	71.3	78.3
Smith et al. (2017) †	54.6	72.7	78.2	42.9	62.2	69.2
Procrustes - NN	42.6	54.7	59.0	53.5	65.5	69.5
Procrustes - CSLS	**66.1**	77.1	80.7	**69.5**	**79.6**	**83.5**
Methods without cross-lingual supervision						
Adv - CSLS	42.5	57.6	63.6	47.0	62.1	67.8
Adv - Refine - CSLS	65.9	**79.7**	**83.1**	69.0	**79.7**	83.1

Table 3: English-Italian sentence translation retrieval. We report the average P@k from 2,000 source queries using 200,000 target sentences. We use the same embeddings as in Smith et al. (2017). Their results are marked with the symbol †.

- ✓ CSLS outperforms all previous approaches
- ✓ unsupervised approach outperforms supervised 50% of the time

(3) Cross-lingual semantic word similarity

SemEval 2017	en-es	en-de	en-it
Methods with cross-lingual supervision			
NASARI	0.64	0.60	0.65
our baseline	0.72	0.72	0.71
Methods without cross-lingual supervision			
Adv	0.69	0.70	0.67
Adv - Refine	0.71	0.71	0.71

Table 4: Cross-lingual wordsim task. NASARI (Camacho-Collados et al. (2016)) refers to the official SemEval2017 baseline. We report Pearson correlation.

(3) Cross-lingual semantic word similarity

SemEval 2017	en-es	en-de	en-it
Methods with cross-lingual supervision			
NASARI	0.64	0.60	0.65
our baseline	0.72	0.72	0.71
Methods without cross-lingual supervision			
Adv	0.69	0.70	0.67
Adv - Refine	0.71	0.71	0.71

Table 4: Cross-lingual wordsim task. NASARI (Camacho-Collados et al. (2016)) refers to the official SemEval2017 baseline. We report Pearson correlation.

✔️ outperforms the SemEval baseline (human-label score)

Structure

- Introduction
- Model
    - Domain adversarial setting
    - Refinement
- Experiments/Evaluation
- **Conclusion**

Conclusion

Conneau et al shows:

- unsupervised approach of mapping source > target embeddings space
- for first time, unsupervised approach is on par w/ or outperforms supervised
- methodology:
  - initialize linear mapping using adversarial approach
  - mapping used to generate synthetic dictionary
  - then, same techniques applied as in supervised approaches like Procrustean optimization
  - introduce unsupervised validation metric and CSLS
- Finally, the high-quality dictionaries can be evaluated against those produced from supervised approaches

Sources

- Tomas Mikolov, Quoc V. Le and Ilya Sutskever. 2014. Exploiting Similarities among Languages for Machine Translation. CoRR, abs/1309.4168.
- Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, Hervé Jégou. 2018. Word Translation without Parallel Data. arXiv:1710.04087v3