MUNI
FI


# Word Translation Without Parallel Data (2018)

A. Conneau, G. Lample, M. Ranzato, L. Denoyer, H. Jégou

**Poster by: A. Puigdemont Monllor, J. Čieško**

Faculty of Informatics, Masaryk University

November 11, 2024

# Model Overview

## Embedding Spaces and Translation Mapping

- We assume two sets of embeddings $X$ (source) and $Y$ (target), trained independently on monolingual data.
- Objective: Learn a linear mapping matrix $W$ such that translations are close in a shared embedding space. This is achieved by minimizing:

$$W^* = \underset{W \in M_d(\mathbb{R})}{\operatorname{argmin}} ||WX - Y||_F$$

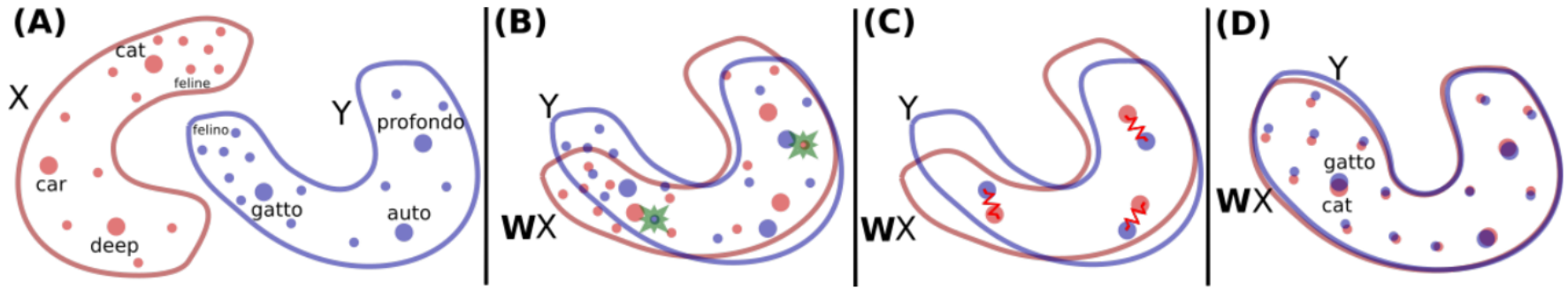
$$||A||_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2} \tag{1}$$

- Translation of a source word $s$ is computed by finding the target $t$ that maximizes the cosine similarity:

$$t = \underset{t}{\operatorname{argmax}} \cos(Wx_s, y_t) \tag{2}$$

- The approach draws on prior work (Mikolov et al., 2013) that demonstrated successful alignment with a linear mapping.

# Method Illustration




*(A) The embeddings of English (red, X) and Italian (blue, Y) words are shown, where dot size indicates word frequency. (B) Adversarial learning is used to align X and Y with a rotation matrix W, tested using randomly selected words. (C) The mapping W is refined through Procrustes, using frequent words as anchor points to minimize an energy function. (D) Translation uses W and CSLS, which adjusts distances in dense regions to reduce "hub" effects, ensuring better word vector alignment.*

# Adversarial Training

## Domain-Adversarial Approach

- We use a domain-adversarial approach to learn the mapping $W$ without relying on cross-lingual supervision.
- Let $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_m\}$ be two sets of word embeddings from a source and a target language, respectively.
- A **discriminator** is trained to distinguish between the transformed embeddings $WX$ and the target embeddings $Y$. This forms a two-player game:
  - The discriminator aims to maximize its ability to identify the origin of an embedding.
  - The mapping $W$ is optimized to minimize the discriminator's predictive accuracy by making $WX$ and $Y$ as similar as possible.
- Discriminator's objective:

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

# Mapping Objective

## Training the Mapping

- The mapping matrix $W$ is optimized to make it difficult for the discriminator to accurately predict the origins of the embeddings:

$$L_W(W|\theta_D) = -\frac{1}{n}\sum_{i=1}^{n} \log P_{\theta_D}(\text{source} = 0|Wx_i) - \frac{1}{m}\sum_{i=1}^{m} \log P_{\theta_D}(\text{source} = 1|y_i) \quad (4)$$

- This training process involves **stochastic gradient updates**, alternating between training the discriminator and updating $W$.
- The goal is to ensure that $WX$ and $Y$ become indistinguishable, thus learning a more robust mapping.
- This process follows the standard adversarial training protocols established by Goodfellow et al. (2014), where models are trained in opposition to each other to improve performance.

# Refinement Procedure

## Refining the Mapping with Procrustes

- The initial mapping $W$ from adversarial training performs well but struggles with rare words, which often have less reliable embeddings.
- Frequent words serve as reliable anchors for refinement to enhance alignment quality.
- We aim to minimize the difference between aligned embeddings:

$$W^* = \underset{W \in O_d(\mathbb{R})}{\operatorname{argmin}} ||WX - Y||_F \tag{5}$$

- A synthetic vocabulary is formed using mutual nearest neighbors among frequent words for a high-quality dictionary.
- The Procrustes method is applied iteratively for further refinement, but improvements beyond the first iteration are typically small, often below 1%.

# Cross-Domain Similarity Local Scaling (CSLS)

## Improving Nearest Neighbor Matching

- CSLS enhances the reliability of matching pairs across languages by adjusting the similarity metric.
- The similarity measure is defined as:

$$CSLS(Wx_s, y_t) = 2\cos(Wx_s, y_t) - r_T(Wx_s) - r_S(y_t) \tag{6}$$

- Where:

$$r_T(Wx_s) = \frac{1}{K} \sum_{y_t \in N_T(Wx_s)} \cos(Wx_s, y_t) \quad \text{(mean similarity to target neighbors)} \tag{7}$$

$$r_S(y_t) = \frac{1}{K} \sum_{x_s \in N_S(y_t)} \cos(x_s, y_t) \quad \text{(mean similarity to source neighbors)} \tag{8}$$

- CSLS effectively addresses the hubness problem, where some words (hubs) serve as nearest neighbors for many others, leading to inaccuracies in translation.
- This method improves accuracy by scaling similarity based on the density of neighboring words, enhancing translation quality.

Thank You for Your Attention!