Word Translation Without Parallel Data (2018)

A. Conneau, G. Lample, M. Ranzato, L. Denoyer, H. Jégou
Poster by: A. Puigdemont Monllor, J. Ciesko

PA164: Machine Learning and NLP
Faculty of Informatics, Masaryk University
November 11, 2024

Objectives

- Achieve word translation without using parallel data by aligning independently trained monolingual embeddings.
- Develop an adversarial training method to align embedding spaces.

Model Overview

Embedding Spaces and Translation Mapping

- Two sets of word embeddings X (source) and Y (target) are used, trained on monolingual data.
- Embedding spaces X and Y do **not need to be of the same dimension.**
- Objective: Find a linear mapping W that aligns translations in a shared space by minimizing:

$$W^* = \underset{W}{\operatorname{argmin}} ||WX - Y||_F \quad (1)$$

- Translation for a source word s is done by maximizing cosine similarity:

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

Adversarial Training

Domain-Adversarial Approach

- Domain-adversarial training aligns WX and Y without cross-lingual supervision.
- A **discriminator** is trained to classify embeddings as source (transformed WX) or target (Y):

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

- Mapping W is optimized to confuse the discriminator by minimizing the following:

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

Refinement Procedure with Procrustes

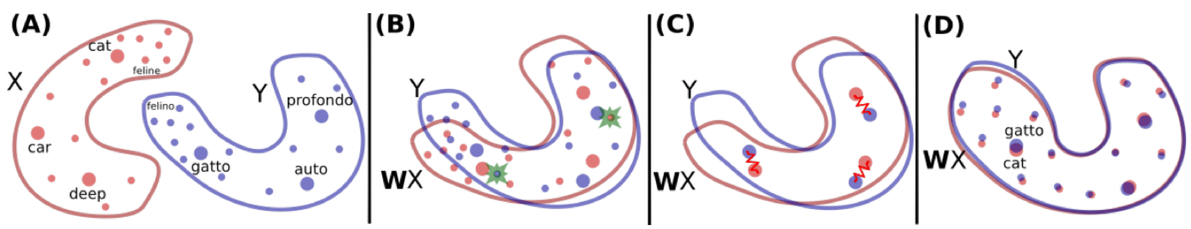
Refining the Mapping

- Initial mapping W aligns well but struggles with rare words.
- Procrustes (orthogonal refinement) improves accuracy by minimizing:

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

- This method iteratively aligns frequent words as anchors and is applied for a high-quality dictionary.

Method Illustration



(A) The embeddings of English (red, X) and Italian (blue, Y), showing word frequency. (B) Adversarial training aligns X and Y via W. (C) Procrustes refinement using frequent word anchors. (D) CSLS adjusts dense region distances, enhancing word alignment.

Cross-Domain Similarity Local Scaling (CSLS)

Improving Nearest Neighbor Matching

- CSLS reduces the effect of "hubs" in dense areas of the embedding space, where some vectors appear as nearest neighbors for many others.
- The CSLS similarity measure is:

$$CSLS(Wx_s, y_t) = 2\cos(Wx_s, y_t) - r_T(Wx_s) - r_S(y_t) \quad (6)$$

Where:

$$r_T(Wx_s) = \frac{1}{K}\sum_{y_t \in N_T(Wx_s)} \cos(Wx_s, y_t) \quad (7)$$

$$r_S(y_t) = \frac{1}{K}\sum_{x_s \in N_S(y_t)} \cos(x_s, y_t) \quad (8)$$

- CSLS adjusts similarity based on neighboring word density, enhancing translation accuracy.

References

Mikolov, T., et al. (2013). Distributed Representations of Words and Phrases and Their Compositionality. *NeurIPS*.
Goodfellow, I., et al. (2014). Generative Adversarial Nets. *NeurIPS*. ...