Plongements Lexicaux Multilingues

Guillaume Potel, Mohamed Malhou
supervisé par: François Yvon

March 2021

1 Introduction

Dans le cadre de l'étude des plongements lexicaux, différentes méthodes ont vu le jour dans la littérature, notamment les modèles de Skip-Gram - prédire le contexte à partir du mot actuel - et de "sac de mots" - prédire le mot à partir de son contexte - (Mikolov)[5]. Leurs travaux ont permis d'obtenir des plongements lexicaux donnant une structure aux mots, qui encode différentes régularités et motifs. On peut citer notamment une additivité des sens; c'est ainsi que la somme des vecteurs "France" et "capitale" sera proche du vecteur "Paris", tandis que la différence des vecteurs "Rome" et "Italie" sera lui proche du vecteur "capitale".

Il a été constaté que de nombreuses invariances se produisent dans les plongements lexicaux entre différents langages entrainés indépendément. C'est à partir de cette idée que différentes méthodes de cartographie de plongements lexicaux bilingues ont vu le jour, avec pour objectif que deux mots ayant le même sens au travers des deux langages se retrouvent avec des vecteurs les représentant proche. Dans cette étude, nous présenterons et analyserons une méthode de plongements multilingues basé sur la minimisation de la distance de Wasserstein entre les distributions des deux plongements, et cela sans supervision inter-langues.

2 Plongements bilingues, Earth's Mover Distance et distance de Wasserstein

Cartographier deux plongements lexicaux dans le même espace a de nombreuses application, notemment pour obtenir la traduction d'un mot dans l'autre langue. L'une des approches supervisées les plus communes[3] est d'entraîner indépendemment les deux plongements puis appliquer une transformation linéaire à l'un d'entre eux qui minimise les distance des vecteurs des mots équivalent dans un dictionnaire bilingue. Cette transformation apprit peut ensuite être utilisé pour trouver les traductions de mots manquant dans un dictionnaire - malgré la simplicité

de la méthode, Mikolov arrive à une précision de 90 pourcent en considérant les 5 vecteurs les plus proches d'un vecteur de mot donné.

Nous sommes alors en droit de nous demander si il est possible d'obtenir cette transformation sans supervision multilingue. Bien que cela semble formidable, l'existence d'isomorphisme structurelle que l'on peut constater au travers de différents espaces de plongements semble indiquer la faisabilité de cette tache. Dans cette étude, nous nous intéresserons bien à une méthode de plongements multilingues sans supervision inter-langue. L'idée principale de celle-ci est de voir les espaces de plongements comme des distributions, et que la transformation désirée doit rendre les ditributions "proches". Afin de quantifier cette proximité, nous utiliserons la distance EMD introduite plus loin. Notre problème se ramène ainsi à minimiser cette distance EMD entre le plongement source transformé et le plongement cible - puisque cette minimisation se fait sur les distributions, aucune supervision au niveau des mots n'est nécessaire. D'autres articles semblent indiquer que les transformations les plus efficaces étaient les transformations orthogonales: on ajoute donc à notre problème cette contrainte sur la transformation.

Introduisons donc la "Earth's Mover Distance" (EMD) [9] entre deux distributions de probabilités discrètes: si deux probabilités discrètes s'écrivent $\mathbb{P}_1 = \sum_i u_i \delta_{x_i}$ et $\mathbb{P}_2 = \sum_j u_j \delta_{x_j}$ avec $\delta_\cdot$ la fonction dirac, on a:

$$\mathsf{EMD}(\mathbb{P}_1, \mathbb{P}_2) = \min_{T \in \mathcal{U}(u,v)} \quad \sum_i \sum_j T_{i,j} c(x_i, y_j) \tag{1}$$

où c(x,y) est une distance entre x et y, et $\mathcal{U}(u, v)$ est le polytope de transport, défini par:

$$\mathcal{U}(u, v) \quad = \left\{ T | T_{i,j} \geq 0, \sum_j T_{i,j} = u_i, \sum_i T_{i,j} = v_j, \forall i, j \right\} \tag{2}$$

On introduit également la distance de Wasserstein qui généralise l'EMD en permettant des distributions continues:

$$\mathbf{W}(\mathbb{P}_1, \mathbb{P}_2) = \quad \inf_{\gamma \in \Gamma(\mathbb{P}_1, \mathbb{P}_2)} \mathbb{E}_{(x,y)\sim\gamma}[c(x, y)] \tag{3}$$

où $\Gamma(\mathbb{P}_1, \mathbb{P}_2)$ correspond à l'ensemble des distributions jointes $\gamma(x, y)$ de distributions marginales $\mathbb{P}_1$ et $\mathbb{P}_2$ en premier et second facteurs respectivement. Nous assimilerons les deux distances dans la suite de cette étude.

3 Minimisation de la distance EMD par transformation orthogonale (EMDOT) - Théorie

Le problème primal[9] sous contrainte orthogonale auquel on va s'intéresser est le suivant :

$$\min_{G \in \mathcal{O}(d)} \quad \mathsf{EMD}(\mathbb{P}^{G(S)}, \mathbb{P}^T) \tag{4}$$

où $\mathcal{O}(d)$ est le groupe orthogonal de dimension d.

Trouver la solution exacte de ce problème est cependant NP-difficile, mais on dispose d'une procédure de minimisation qui est garantit de converger vers un minimum local. En partant d'une matrice orthogonale $G^{(0)}$, on calcule successivement:

$$T^{(k)} = \quad \arg \min_{T \in \mathcal{U}(f^S, f^T)} \sum_{s=1}^{V^S} \sum_{t=1}^{V^T} T_{st} c(G^{(k)} \omega_s^S, \omega_t^T) \tag{5}$$

$$G^{(k+1)} = \arg \min_{G \in \mathcal{O}(d)} \sum_{s=1}^{V^S} \sum_{t=1}^{V^T} T_{st}^{(k)} c(G \omega_s^S, \omega_t^T) \tag{6}$$

Où f^S et f^T sont les vecteurs de fréquences des langages S et T, définis par $\mathbb{P}(w_s^S) = f_s^S$, $\mathbb{P}(w_t^T) = f_t^T$ pour tous mots w_s^S de S et w_t^T de T,

D'une part, si la distance c est la distance euclidienne au carré L_2^2, alors le problème (6) est une extension du problème de Procrustes[6] orthogonal, dont on sait calculer la solution:

$$G^{(k+1)} = UV^\top \tag{7}$$

avec la décomposition en valeurs singulières:

$$\sum_{s=1}^{V^S} \sum_{t=1}^{V^T} T_{st}^{(k)} \omega_t^T \omega_s^{S\top} = USV^\top \tag{8}$$

D'autre part, il existe des solveurs pour le problème (5), mais le coût de ces algorithmes est au moins de $\mathrm{O}(d^3 \log d)$. On utilisera alors un solveur approximatif pour une plus grande adaptabilité, qui se base sur la distance de Sinkhorn[1].

Le problème de minimisation associé au problème (5) se réécrit:

$$d_{M^{(k)}}(u, v) = \min_{T \in \mathcal{U}(u,v)} \quad \langle T, M^{(k)} \rangle \tag{9}$$

avec $u = f^S$, $v = f^T$, $M^{(k)} = (c(G^{(k)}w_s^S, w_t^T))_{s,t}$, et $\langle ., . \rangle$ le produit scalaire canonique de $\mathcal{M}_d(\mathbb{R})$.

En introduisant l'entropie h par:

$$h(u) = \sum_{i=1}^{d} u_i \log u_i \text{ et } h(T) = \sum_{i,j=1}^{d} T_{ij} \log T_{ij}$$

La distance de Sinkhorn entre u et v s'écrit alors:

$$d_{M^{(k)},\alpha}(u, v) = \min_{T \in \mathcal{U}_\alpha(u,v)} \langle T, M^{(k)} \rangle \tag{10}$$

avec α un paramètre réel et $\mathcal{U}_\alpha(u, v)$ l'ensemble convexe défini par:

$$\mathcal{U}_\alpha(u, v) = \{T \in \mathcal{U}(u, v) \mid h(T) \geq h(u) + h(v) - \alpha\} \tag{11}$$

On note que pour α assez grand, $\mathcal{U}_\alpha(u, v)$ et $\mathcal{U}(u, v)$ coincident.

Pour λ strictement positif, on introduit alors la divergence dual de Sinkhorn par:

$$d^\lambda_{M^{(k)}}(u, v) = \langle T^\lambda, M^{(k)} \rangle, \; T^\lambda = \arg\min_{T \in \mathcal{U}(u,v)} \langle T, M^{(k)} \rangle - \frac{1}{\lambda} h(T) \tag{12}$$

La théorie de la dualité nous indique alors que à chaque α réel correspond à un λ strictement positif tel que $d_{M^{(k)},\alpha}(u, v) = d^\lambda_{M^{(k)}}(u, v)$: calculer T^λ pour λ bien choisi nous permet bien de minimiser la distance de sinkhorn. Or, il est facile de calculer exactement T^λ à un coût bien moins important que le problème original (5): c'est donc ce T^λ pour $u = f^S$, $v = f^T$, et $M^{(k)} = (c(G^{(k)}w_s^S, w_t^T))_{s,t}$ que nous allons calculer comme valeur approximative du $T^{(k)}$ dans ce problème.
Ainsi, afin de répondre à (4), on applique l'algorithme suivant:

Algorithm 1 Minimizing Wasserstein distance

Input S and T matrices, f_S and f_T frequency vectors
Output G an approximation of the optimal OT matrix
while $\delta d \geq \epsilon$ **do**
  $C = \mathsf{dist}(SG^\top, T)$
  $W = \mathsf{sinkhorn}(f_S, f_T, C, \lambda)$
  Let $R = S^\top WT$,
  $U, \Delta, V^\top = \mathsf{Svd}(R)$
  the new G is $G = UV^\top$
  $d = \langle C, W \rangle$
end while

4 Le modèle Wasserstein GAN (WGAN)

Une autre manière d'aligner les distributions des plongements lexicaux consiste à faire appel au problème dual du problème (4). Cette approche est proposée

dans l'article de Zhang[9] et est appelée WGAN pour sa ressemblance avec les modèles antagonistes génératives. En effet, dès lors que la distance c correspond à la distance euclidienne L_2, par dualité de Kantorovich-Rubinstein [7] on peut montrer que:

$$\mathbf{W}(\mathbb{P}_1, \mathbb{P}_2) = \quad \frac{1}{K} \sup_{\|f\|_L \leq K} \mathbb{E}_{y \sim \mathbb{P}_2}[f(y)] - \mathbb{E}_{y \sim \mathbb{P}_1}[f(y)] \tag{13}$$

Où l'on maximise sur les fonctions K-lipschitziennes. En pratique, le discriminateur ,qui est représenté par un réseau de neurones, joue le rôle de la fonction f.

Les travaux de Mikolov[4] inspirent les chercheurs à chercher l'isomorphisme entre les plongements lexicaux parmi les transformations linéaires orthogonales. Le générateur est donc simplement une matrice ou une couche cachée sans fonction d'activation ni biais. Le générateur vise donc à minimiser la distance (10), ce qui revient à minimiser :

$$\min_{G \in \mathbb{R}^{d \times d}} -\mathbb{E}_{x \sim \mathbb{P}_S}[f_D(Gx)] \tag{14}$$

On verra plus tard que cet algorithme n'est pas stable et la contrainte d'orthogonalité de G n'est pas implémentée. Pour l'imposer, on peut utiliser Procrustes afin de chercher à chaque pas de la descente de gradient la matrice orthogonale la plus proche de G en résolvant le problème suivant:

$$\min_{\overline{G}} \|G - \overline{G}\| \quad s.t. \quad \overline{G}^\top \overline{G} = I \tag{15}$$

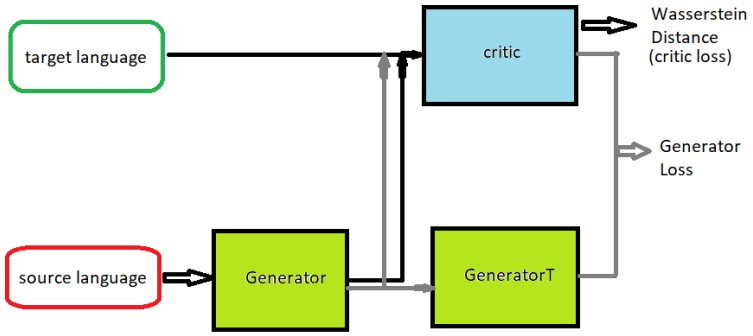
Ce problème admet une solution[8] par application d'une décomposition en valeurs singulières de G.

La deuxième méthode pour respecter l'orthogonalité de G est de rajouter à la fonction de perte (11) un terme de pénalisation comme suit :

$$\min_{G \in \mathbb{R}^{d \times d}} -\mathbb{E}_{x \sim \mathbb{P}_S}[f_D(Gx)] + \eta \mathbb{E}_{x \sim \mathbb{P}_S}[\|G^\top Gx - x\|^2] \tag{16}$$

avec η un hyper-paramètre à ajuster.

Figure 1: WGAN with OT constraint

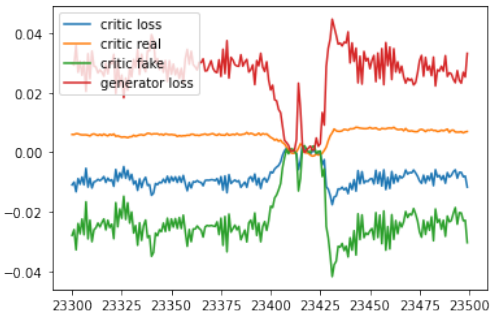


5 Expériences

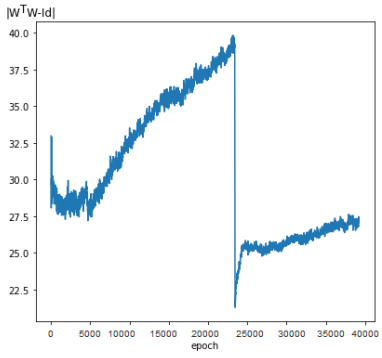
5.1 L'implémention de WGAN

Nous abordons d'abord l'approche WGAN pour analyser son comportement d'apprentissage. Nous utilisons le cadre tensorflow afin d'implémenter ce modèle, on emploie l'optimiseur RSMProp basé sur la descente de gradient stochastique car il explore mieux l'espace des paramètres. Pour reproduire les mêmes performances de l'article de Zhang [9], nous le testons sur un jeu de données chinois-anglais qui sont des plongements en dimension 50 obtenus par l'algorithme CBOW de Mikolov[5].

Figure 2: L'entrainement du WGAN (sans contrainte d'orthogonalité) sur un jeu de donnée anglais-chinois en dimension 50 et de taille 5000



(a) les fonctions de perte de WGAN



(b) Mesure de l'orthogonalité du générateur

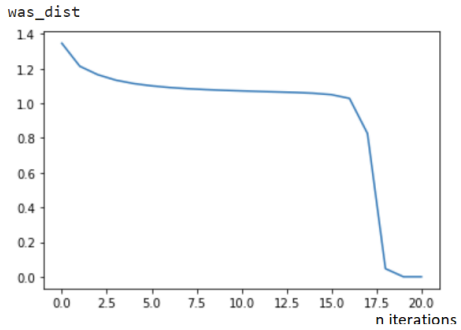
L'entraînement n'est pas stable, mais ce qui est intéressant de remarquer est la baisse soudaine du critère d'orthogonalité quand on s'approche de l'optimum qui correspond à des pertes nulles. Ceci suggère encore une fois que la meilleure transformation linéaire entre les plongements de CBOW est orthogonale. Plusieurs recherches sont faites pour stabiliser l'algorithme, Mescheder montre dans son article[2] que le WGAN dans la configuration qu'on a décrit ne converge pas alors que d'autres versions avec certains types de pénalité du gradient convergent.

Imposer l'orthogonalité par le biais des deux méthodes introduites en (15) et (16) certes accomplit son but, cependant, cela n'accélère pas la convergence malgré l'intuition que le fait de réduire l'espace des paramètres à explorer entraîne une certaine amélioration.

5.2 Implémention et résultats de la méthode EMDOT

Afin de tester cet algorithme, nous appliquons au plongement anglais une transformation orthogonale aléatoire et on essaye de la retrouver en appliquant EMDOT. On trace la distance de Wasserstein en fonction des itérations. Ci-dessous est le résultat de cette expérience après plusieurs initialisations, la convergence est assez lente.

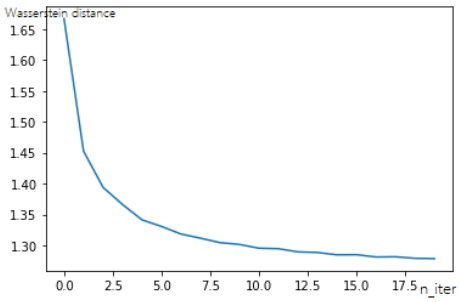
Figure 3: la EMD au cours des itérations pour une transformation linéaire orthogonale bien connue. Cette expérience est faite plusieurs fois en générant les matrices orthogonale avec la fonction ortho_group du paquet scipy.stats. L'algorithme converge rarement vers la bonne matrice au bout de 20 itérations



On applique ensuite cette approche aux plongements anglais-italien. On ne prend que 5000 mots de chaque langue en dimension 300 (une proportion du jeu de données fasttext) de sorte que chaque mot dans une langue a une traduction dans l'autre.

Ne rajoutant pas d'informations dans le modèle, la distance calculée semble converger lentement vers une valeur de 1.28 qui n'est pas, et on le verra plus tard, la valeur optimale. En sélectionnant le plus proche voisin, la précision mesurée n'est que de .06% après une vingtaine d'itérations.

Figure 4: EMDOT sur le jeu de données parallèle anglais-italien

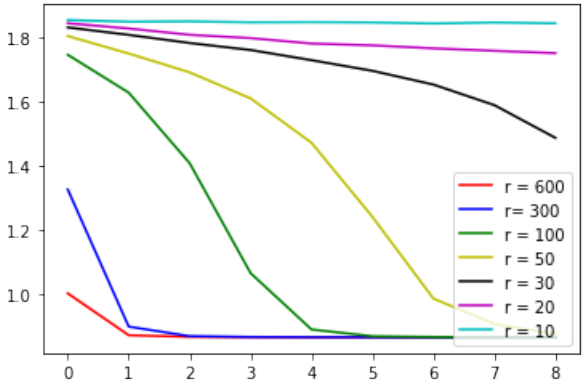


Nous essayons ensuite d'étudier la convergence de l'algorithme si l'on introduit de l'information qui consiste à initialiser la matrice de transport en une matrice : $\mathrm{W} = \begin{pmatrix} I_r & 0 \\ 0 & R \end{pmatrix}$

où R est une matrice quelconque. Ceci correspond à associer les r premiers mots.

Pour $r = N$, la distance converge vers une valeur autour de 0.86. La précision est alors de 86.20%. Notons qu'on ne peut pas atteindre de tels scores si l'on travaille sur les jeux de données massifs qui ne sont pas tout simplement des paires de (mot, mot traduit). Ci-dessous sont présentés les scores correspondants aux différentes valeurs de r.

Figure 5: EMDOT sur le jeu de données anglais-italien



r	5000	3000	1000	600	100	50	30	20	10	0
Accuracy	86.20%	86.17%	85.74%	85.85%	85.85%	85.01%	40.73%	5.90%	0.96%	0.06%

Figure 6: Précisions réalisées

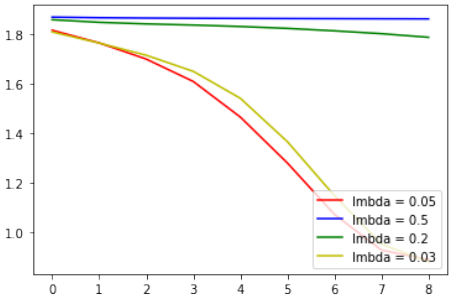


Figure 7: Les résultats du EMDOT en fonction du paramètre λ

λ	0.05	0.03	0.1	0.2	0.5
le TE en s	539	541	143	67	62

Figure 8: les temps d'exécution réalisés

6 Conclusion

Cette étude s'intéresse à des méthodes de plongements lexicaux multilingues sans supervision inter-langues. Nous y introduisons un algorithme approché de minimisation de la distance EMD, distance qui produit est utile pour quantifier les différences entre langages. Malgré l'apparente contrainte de non supervision, ce modèle arrive à rivaliser avec d'autres méthodes supervisées. Nos expérimentations mettent en avant l'efficacité du modèle ainsi que la convergence des distances des méthodes décrites et de l'orthogonalité du générateur. Il serait intéressant dans le futur d'évaluer plus en détail la qualité de la distance de Wasserstein comme bon indicateur de distance entre deux distributions de plongements, ainsi que de proposer d'autres distances qui pourraient peut-être mieux convenir à notre problème.

References

[1] Marco Cuturi. "Sinkhorn Distances: Lightspeed Computation of Optimal Transport". In: (2013). DOI: https://arxiv.org/pdf/1306.0895.pdf.

[2] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. "Which Training Methods for GANs do actually Converge?" In: (2018). DOI: https://arxiv.org/pdf/1801.04406.pdf.

[3] Thomas Mikolov, Quoc V Le, and Ilya Sutskever. "Exploiting Similarities among Languages for Machine Translation". In: (2013). DOI: https://arxiv.org/pdf/1309.4168.pdf.

[4] Tomas Mikolov, Ilya Sutskever, and Kai Chen. "Distributed Representations of Words and Phrases and their Compositionality". In: (2013). DOI: https://arxiv.org/pdf/1310.4546.pdf.

[5] Tomas Mikolov et al. "Efficient Estimation of Word Representations in Vector Space". In: (2013). DOI: https://arxiv.org/pdf/1301.3781.pdf.

[6] Peter H Schonemann. "A generalized solution of the orthogonal procrustes problem". In: (1966). DOI: https://link.springer.com/content/pdf/10.1007/BF02289451.pdf.

[7] Cédric Villani. *Optimal Transport: Old and New*. 2009.

[8] Chao Xing et al. "Normalized Word Embedding and Orthogonal Transform for Bilingual Word Translation". In: (2015). DOI: https://www.aclweb.org/anthology/N15-1104.

[9] Meng Zhang et al. "Earth Mover's Distance Minimization for Unsupervised Bilingual Lexicon Induction". In: (2017). DOI: https://zmlarry.github.io/publications/emnlp2017.pdf.