Rapport EA - MAP511

Optimal Transport for Cross-lingual Word Alignments

Paul Mortamet and Adrien Lagesse

Ecole Polytechnique

January 28, 2022

Abstract

Il existe peu d'études précises sur l'utilisation du Transport Optimal dans les problèmes d'alignement de mots entre des phrases de différentes langues (*Word Alignment*). Notre étude se place à la suite des travaux de Zi-Yi Dou et Graham Neubig [7] qui introduisent légèrement cette méthode dans les problématiques de *Word Alignment*. Nous avons étudié les implications de l'algorithme de Sinkhorn et de la construction de la matrice de coût dans l'approximation du plan de Transport Optimal. En observant les différentes étapes successives de la méthode de Dou et Neubig [7], nous proposons une optimisation de l'algorithme de Sinkhron à travers une étude sur le paramètre de régularisation entropique. Ensuite, nous proposons une modification de la construction de la matrice de coût, en définissant deux nouvelles distances. Nous terminons notre étude en proposant une nouvelle méthode d'extraction des alignements, plus souple et adaptée aux différentes distributions.

Keywords Optimal Transport, Word Alignment, Sinkhorn, Skip-gram, Word Embeddings.

1 Introduction

Les problématiques autour de la notion de *Word Embedding*, c'est-à-dire la représentation de mots sous forme de vecteurs, sont devenues centrales en Traitement du Langage. Ce passage du monde lettré à un espace mathématique bien défini, sans omettre le sens de chaque mot, seul et au sein de son contexte, est un chemin parsemé d'embuches que les méthodes modernes doivent éviter.

1.1 Espace d'embeddings

Les travaux de Mikolov et al. [10] ont fait émerger des méthodes innovantes (Skip-gram, CBOW) de représentations vectorielles de mots, conditionées par leur contexte au sein d'un corpus donné. L'innovation apportée par ces travaux est la capacité de traduire dans les représentations vectorielles des relations syntaxiques, sémantiques mais aussi idiomatiques entre les mots. En effet, le coeur de problème de *Word Embedding* est de construire cette capacité à associer des mots sémantiquement différents comme "Air" et "France" afin de comprendre leur association, ici "Air France", et de

traduire dans la somme des vecteurs le sens recherché à l'association, ici celui d'une compagnie aérienne française.

1.2 Transposition multilingue

Dans la continuité de ces méthodes, la construction d'un espace vectoriel d'*embeddings* saisissant les règles sémantiques, synthaxiques mais aussi linguistiques, a pris un tournant multilingue. En effet, une fois ces espaces construits les informations linguistiques, peuvent être mises à contribution dans les problèmes de tranduction où elles sont essentielles. Les travaux de Mikolov et al. [9] de 2018 ont proposé une première méthode simple et efficace visant à trouver une matrice de traduction optimale. Etant donné un jeu de mots $\{x_i, y_i\}_{i=1}^n$ sous forme vectorielle avec $x_i \in \mathbf{R}^{d_1}$ et $y_i \in \mathbf{R}^{d_2}$, le but est de trouver une matrice de traduction telle que :

$$\tilde{W} = \underset{W}{\operatorname{argmin}} \sum_{i=1}^{n} \|Wx_i - y_i\|^2$$

en utilisant une descente de gradient. Cette méthode très simple affiche des résultats remarquables mais met en lumière cette question de translation entre espaces de vecteurs et donc espaces d'*embeddings*. Les travaux de Xing et al. [13] ainsi que ceux de Arteka et al. [3] consolident les aspects mathématiques à la méthode proposée précédemment. Parmi ces précisions, ressortent la normalisation des vecteurs ainsi que le caractère orthogonal de la matrice de traduction. La normalisation permet de cofondre similarité cosinus (**CosineSimilarity**) et produit vectoriel, et apporte donc une certaine cohérence dans les distances utilisées dans les différentes étapes de l'aglorithme de Skip-gram. L'orthogonalité de la matrice de traduction, quant à elle, permet une traduction de qualité tout en simplifiant grandement l'optimisation de la matrice puisque dans ce cas, en notant $X = (x_1, ..., x_n)^T$ et $Y = (y_1, ..., y_n)^T$ nous avons :

$$\underset{W \in \mathcal{O}_d}{\operatorname{argmin}} \|WX - Y\|^2 = \underset{W \in \mathcal{O}_d}{\operatorname{argmax}} \operatorname{Tr}(XWY^T)$$

qui se calcule simplement en utilisant le théorème spectral et par diagonalisation.

1.3 Earth Mover's Distance

Cette problématique naissante de translation multilingue entre espaces de vecteurs a pris une nouvelle dimension avec les travaux de Zhang et al. [14]. En partant des espaces vectoriels donnés par les algorithmes de *Word Embedding*, ces travaux proposent une méthode d'alignement isomorphique des *embeddings* d'un espace de départ dans un espace d'arrivée. Cette notion d'alignement et non de traduction un-à-un est clé dans cette méthode puisqu'elle utilise une notion linguistique (comme Mikolov dans Skip-gram [10], [9]) celle des invariances sémantiques entre deux langues différentes. Il est démontré que les espaces d'*embedding* de deux langues différentes se ressemble dans leur construction et sont même invariants selons certains points. Au delà de cette nouvelle approche, ces travaux introduisent une nouvelle métrique dans cette problématique de traduction, la *Earth Mover's Distance* (*EMD*) définie, pour deux distributions de probabilités discrètes $\mathbf{P}_1 = \sum_i u_i \delta_i$ et $\mathbf{P}_2 = \sum_j v_j \delta_j$, comme suit :

$$\mathrm{EMD}(\mathbf{P}_1, \mathbf{P}_2) = \min_{T \in \mathcal{U}(u,v)} \sum_i \sum_j T_{ij} c(x_i, y_j)$$

avec $c(x_i, y_j)$ la distance entre les points x_i et y_j et $\mathcal{U}(u, v)$ est le polytope de transport défini comme :

$$\{T/T_{ij} \geq 0, \sum_j T_{ij} = u_i, \sum_i T_{ij} = v_j, \forall i, j\}$$

.

Intuitivement, cette mesure cherche à minimiser les coûts de déplacement de chaque "tas" de probabilité d'une distribution vers ceux de la seconde. Nous nous rapprochons donc du problème de Transport Optimal qui fait partie intégrante de notre travail et sera présenté plus loin.

L'introduction de cette nouvelle distance permet une définition plus fine du problème et permet d'obtenir une traduction plus robuste face aux règles linguistiques. En effet, le problème devient donc d'aligner les vocabulaires, vus comme des distributions de probabilités, tout en minimisant le cout de déplacement, défini comme la distance des vecteurs d'*embedding* de chaque vocabulaire. Cet alignement est représenté par la matrice de translation T qui joue un rôle similaire au rôle de la matrice de traduction W introduite par Mikolov, tout en permettant une traduction plus nuancée notamment en permettant de faire des associations *many-to-one*. Ce point est loin d'être négligeable puisqu'il est linguistiquement obligatoire de pouvoir réaliser ce type d'associations. La forme de la matrice de translation doit répondre à certaines caractéristiques, notamment concernant la sparsité dont nous discuterons plus loin.

Les travaux de Zhang et al. [14] proposent deux méthodes pour déterminer cette matrice T dont une s'appuyant sur les travaux de Arjovsky et al. [2] qui introduit le Wassersrtein GAN. La distance de Wassterstein se définit comme suit :

$$\mathrm{W}(\mathbf{P}_1, \mathbf{P}_2) = \inf_{\gamma \in \Gamma(\mathbf{P}_1, \mathbf{P}_2)} E_{(x,y)\ \gamma}[c(x, y)]$$

où $\Gamma(\mathbf{P}_1, \mathbf{P}_2)$ correspond à l'ensemble des distributions $\gamma(x, y)$ de distributions marginales $\mathbf{P}_1$ et $\mathbf{P}_2$. De part sa définition, la *EMD* se rapproche de cette distance de Wassterstein dans le cas discret. Les travaux de Arjovsky et al. [2] proposent une utilisation des propriétés mathématiques d'un *Generative Adversarial Network* ou GAN pour résoudre le problème de :

$$\operatorname*{argmin}_{G \in \mathbf{R}^{d_1 x d_2}} \mathrm{W}(\mathbf{P}^{G(S)}, \mathbf{P}^T)$$

où S désigne l'espace de départ, T l'espace d'arrivée et G la matrice de translation entre espaces d'*embedding*. En utilisant la dualité Kantorovich-Rubinstein établie par Villani, Arjovsky et al. [2] proposent une solution fondée sur l'alterance de ces deux programmes de minimisations :

$$T^{(k)} = \operatorname*{argmin}_{T \in \mathcal{U}(f^S, f^T)} \sum_s \sum_t T_{st} c(G^{(k)} w_s^S, w_t^T)$$

$$G^{(k+1)} = \operatorname*{argmin}_{G \in \mathcal{O}(d)} \sum_s \sum_t T_{st}^{(k)} c(G w_s^S, w_t^T)$$

avec une initialisation $G^{(0)}$ et en travaillant sur les fréquences de mots dans le vocabulaire f^S et f^T.

Ces deux derniers ouvrages constituent le fondement de nos expérimentations. Les premiers travaux sur Skip-gram et ses optimisations mathématiques nous ont introduits à l'optimisation mathématique des espaces vectoriels d'*embeddings* pour conserver les proporiétés linguistiques du langage. Les travaux suivants ont cherché à faire correspondre deux espaces vectoriels d'*embeddings* tout en conservant les propriétés linguistiques, invariantes d'un espace à l'autre. La problématique est donc devenue la détermination d'une matrice de translation entre deux espaces vectoriels. Les dernières méthodes proposées ont fait émerger une nouvelle vision du problème fondée sur les distributions de probabilités et en introduisant de nouvelles distances. Ces approches nous

ont fait nous intéresser aux conditions de sparsité de notre matrice de translation et au Transport Optimal comme moyen de mise en relation de différents *embeddings*.

Notre expérimentation s'est donc concentrée sur le Transport Optimal et son efficacité dans les problèmes de traduction. Afin de donner un cadre plus précis à nos recherches, nous nous somme concentrés sur un problème de traduction précis : celui du *Word Alignment*. Semblable aux problèmes précédents, il permet de se détacher des papiers étduiés et de visualiser plus facilement les observations que nous cherchions à faire sur la sparsité de la matrice de traduction.

2 Word Alignment

L'alignement des mots est une tâche de traitement du langage naturel qui consiste à identifier les relations de traduction entre les mots d'un texte et d'un autre texte traduit. Cette opération donne un graph biparti entre les textes, où un arc existe entre deux mots si et seulement si ces mots sont des traductions l'un de l'autre. L'alignement des mots est une tâche importante pour la plupart des méthodes de traduction automatique statistique. Nous pouvons notamment citer l'utilisation des alignements de mots pour générer des matrices d'attention (Liu et al [8], Chen et al [5]) ou encore leur utilisation pour le décodage par Alkhouli et al [1].

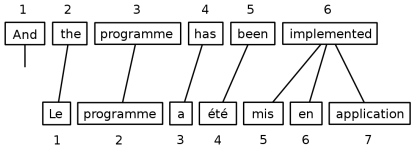

Figure 1: Alignements entre une phrase en anglais et en français

Comme nous voyons sur la figure 2, le problème d'alignement est très compliqué. En effet, tous les mots n'ont pas de traduction dans la seconde phrase, un mot peut être traduit en plusieurs mots et il arrive même que plusieurs mots soient traduit en un seul mot. Historiquement, déterminer les alignements de mots se faisaient avec des méthodes statistiques (chaînes de Markov cachées par Vogel et al [12]) et en les déterminant parallèlement de le traduction. Néanmoins ces méthodes demandent énormément de données et leurs performances diminuent rapidement avec peu de données. Sabet et al [11] propose d'utiliser les *embeddings* des mots afin de calculer les alignement de mots. En effet, comme décrit en introduction, il est possible de créer des espaces de mots multilingue [13] [3] [14] et ensuite de comparer les mots deux à deux en utilisant par example la distance cosinus. Ensuite, Sabet et al [11] créeent des matrices de similarités qui seront utilisées par un algorithme appelé *IterMax* afin de faire les calculs des alignements. Cet algorithme affine l'idée de construction d'alignments suivant :

- Soit $(A_{i,j})_{i,j}$ la matrice d'alignement où $A_{i,j} = 1$ si les mots i et j sont alignés et 0 sinon.
- $A_{i,j} = 1$ si $(i = \text{argmax}_l\, S_{l,j}) \wedge (j = \text{argmax}_l\, S_{l,i})$ avec S la matrice de similarité.

Cette première méthode nous permet déjà de repérer des qualités importantes d'un bon algorithme de prédiction d'alignement. En effet, on s'attend déjà à ce que l'alignement soit une matrice très creuse car la traduction associe la majorité du temps un seul mot à un autre mots. De plus, pour des traductions du type Anglais/Français on s'attend aussi que la matrice se rapproche d'une certaine manière de la matrice identité (la diagonale dominante), mais il faut prendre cette heuristique avec des pincettes

(exemple: inversion adjectif/nom entre l'anglais et le français).

Finalement, l'existence d'alignement optimaux pour de nombreuses phrases traduites nous permet de tester la précision et le rappel d'un certain algorithme d'alignement.

3 Optimal Transport

Le problème du Transport Optimal, introduit précédemment avec la *EMD*, est un problème très intuitif. Historiquement formulé par Monge au XVIIIe siècle, il consiste à chercher le moyen le plus économique pour transporter des objets entre un ensemble de points de départ et de points d'arrivée. Mathématiquement parlant et d'une manière plus adaptée à notre problème, il consiste à minimiser la distance *EMD* entre deux distributions de probabilités $\mathbf{P}_u$ et $\mathbf{P}_v$ représentées par des vecteurs de probabilités u et v du simplexe $\Sigma_d = \{x \in \mathbf{R}_+^d, x^T \cdot \mathbf{1}_d = 1\}$, espacées par une matrice de coût $M \in \mathbf{R}_+^{d \times d}$. Le problème se définit donc comme suit :

$$P = \underset{P \in \mathcal{U}(u,v)}{\operatorname{argmin}} \langle P, M \rangle$$

où $\langle \cdot, \cdot \rangle$ est le produit de Frobenius. Lorsque la dimension (nous nous sommes contentés au cas où les deux vecteurs sont de même dimension par soucis de simplicité) augmente, ce problème devient long et couteux à résoudre correctement.

Avec une approche entropique, Cuturi [6] propose une méthode fondée sur l'aglorithme de Sinkhorn pour résoude efficacement le problème de Transport Optimal. En définissant l'entropie d'un vecteur $x \in \Sigma_d$:

$$h(x) = -\sum_{i=1}^{d} x_i \log(x_i)$$

le problème de Transport Optimal devient :

$$P_\lambda = \underset{P \in \mathcal{U}(u,v)}{\operatorname{argmin}} \langle P, M \rangle - \frac{1}{\lambda} h(P)$$

Ajouter un terme de régularisation entropique simplifie la résolution du problème pusique la solution P_λ a finalement une forme très simple. En effet, d'après le théorème de Sinkhorn (1967), pour tout $\lambda \in \mathbf{R}_+^*$, la solution $P_\lambda \in \mathcal{U}(u,v)$ est unique et est de la forme :

$$P_\lambda = \mathbf{diag}(x_1) K \, \mathbf{diag}(x_2), K = e^{-\lambda M}$$

Les vecteurs positifs $x_1, x_2 \in \mathbf{R}_+^d$ sont uniques et calculés par doubles itérations successives $(x_1, x_2) \leftarrow (u/Kx_2, v/K^T x_1)$. Puisque nous travillons avec des distributions de probabilités, il est raisonnable de poser, sans contrainte, $u > \mathbf{0}_d$. Sous cette hypothèse, la double itération peut être réduite à la simple itération suivante $x_1 \leftarrow 1/(\tilde{K}(v/K^T x_1))$ avec la notation $\tilde{K} = \mathbf{diag}(1/u)K$.

Ainsi, l'algorithme de Sinkhorn permet de résoudre efficacement le problème du Transport Optimal, en se débarrassant des contraintes de dimension et en donnant une forme simple à la solution grace à la régularisation entropique. La méthode de calcul, simple à implémenter, est illsutrée par l'**Algorithme 1**.

Algorithm 1 Algorithme de Sinkhorn

Require: $M \in \mathbf{R}_+^{d \times d}$, $u, v \in \Sigma_d - \{\mathbf{0}_d\}$
  $K \leftarrow \exp(-\lambda M)$
  $x_1 \leftarrow \mathbf{ones}(d)/d$
  $\tilde{K} \leftarrow \mathbf{diag}(1/v)K$
  while u ne change plus **do**
    $x_1 \leftarrow 1/(\tilde{K}(v/K^T x_1))$
  end while
  $x_2 \leftarrow v/(K^T x_1)$
return $\mathbf{diag}(x_1) K \, \mathbf{diag}(x_2)$

Particulièrement utile, cet algorithme introduit néanmoins un paramètre de

régularisation qui nécessite une attention particulière. Influant sur la solution, il est nécessaire de mesure son impact sur la vitesse de convergence mais aussi sur la qualité de la solution. Puisqu'il résoud le problème par approximation, la solution peut s'éloigner de la solution optimale $P^\star$. Au delà de ces préoccupations classiques, il est important de vérifier la qualité de la matrice trouvée à travers sa sparsité. La régularisation ne doit pas proposer une solution P^λ peu sparse là où $P^\star$ l'est. Les travaux de Blondel et al. [4] démontrent que les régularisations entropiques peuvent être propices aux résultats sparses, ce qui est particulièrement positif dans notre sujet, mais sous certaines conditions à respecter.

Ainsi, nous débutons nos expérimentations dans l'optique d'appliquer le Transport Optimal au problème de *Word Alignment*. En utilisant l'aglorithme de Sinkhorn pour résoudre le problème de Transport Optimal, nous nous intéresserons à l'effet du paramètre de régularisation Lagrangien sur l'efficacité de l'algorithme mais aussi sur la qualité de la solution proposée. Une fois ce paramètre optimisé, nous nous intéresserons à une optimisation de la matrice de coût M qui joue un rôle crucial dans l'algorithme mais où très peu d'informations et de critères sont fixés. Nous explorerons donc les possibilités de construction de cette matrice en jouant notamment sur les distances utilisées et les procédés de régularisation des valeurs obtenues.

4 Recherche

4.1 Régularisation entropique

Dans l'article de Zi-Yi Dou et Graham Neubig [7] la formulation de la solution théorique est basée sur le calcul d'un plan de transport optimal, ce plan de transport peut être vu comme un alignement de mots. Cependant, comme il a été dit dans la partie 3, en pratique c'est le problème de transport optimal régularisé suivant (attention on a remplacé λ par λ^{-1}) qui est résolu:

$$P_\lambda = \underset{P \in (u,v)}{\operatorname{argmin}} \langle P, M \rangle - \lambda h(P)$$

Intuitivement, le terme de régularisation vise à rendre la matrice de transport moins creuse. Or, comme évoqué dans la partie *Word Alignement*, notre problématique d'alignement de mots requiert une matrice de transport le plus creuse possible.

Nous avons alors voulu quantifier à quel point le terme de régularisation impacte donc le caractère creux de la matrice. Pour ce faire, nous avons choisi d'utiliser la fonction entropie h définie précédemment. En effet, une matrice d'alignement avec uniquement des 0 et des 1 a une entropie nulle, ce qui nous permet donc de quantifier correctement la sparsité de la matrice. Nous mesurons ci-contre l'entropie de la matrice de transport en fonction du paramètre de régularisation lagrangien λ:

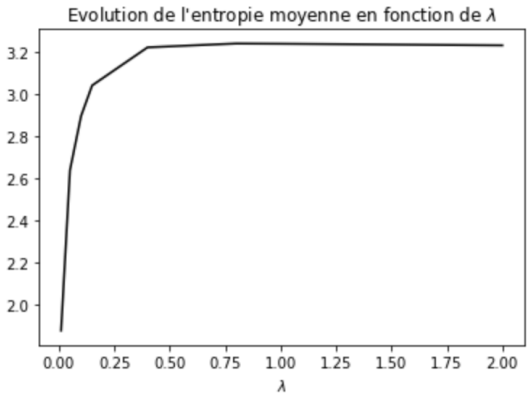

Figure 2: Perte du caractère creux en fonction de λ

De manière nettement visible, nous concluons qu'il est primordial de bien choisir le λ, c'est à dire proche de 0, pour

ne pas perdre le caractère creux recherché de la matrice de transport.

Néanmoins, s'il semble suffir de prendre $\lambda = 10^-9$ par exemple, nous remarquons que dans l'algorithme Sinkhorn 1, il est nécessaire de travailler avec des grandeurs de l'ordre de $e^{\lambda^{-1}}$.

4.2 Matrice de coût

Le second axe de recherche sur lequel nous nous sommes focalisé est le rôle de la matrice de coût intervenant dans le problème de Transport Optimal. Cette matrice est la pierre angulaire du problème puisqu'elle aiguille la répartition de la distribution de départ sur celle d'arrivée.

Dans notre cas du *Word Alignment*, la matrice de coût indique quels mots peuvent être facilement mis en relation de part leur similarité sémantique et linguistique. La difficulté réside donc dans la construction d'une matrice transmettant une information suffisante pour obtenir une matrice de traduction optimale. Malgré le rôle essentiel de cette matrice, peu voire aucun papier de recherche n'expose précisément les règles auxquelles elle doit répondre. Par exemple, certains parlent de matrice de *distance* où les coefficients sont à valeurs dans $[0, 1]$, d'autres de matrice de *coût* donc borne fixée au préalable. Quelle forme donner à M pour optimiser le résultat final? Quelles sont les règles auxquelles la matrice M doit obéir pour résoudre le problème de *Word Alignment* avec le Transport Optimal?

Nous sommes partis des travaux de Dou et al. [7] qui introduisent récemment le Transport Optimal sur la thématique du *Word Alignment* mais sans l'évaluer concrètement. Cette implémentation constitue donc notre base depuis laquelle nous menons nos expérimentations et améliorerons les performances.

Algorithm 2 Transport Optimal pour Word Alignment

Require: $(w_i^{\text{FR}})_{i \in n_{\text{FR}}}, (w_i^{\text{ES}})_{i \in n_{\text{ES}}} \in \mathbf{R}^d$
  $M \leftarrow 1 - \textbf{CosineSimilarity}(w^{\text{FR}}, w^{\text{ES}})$
  $\mathbf{P}_{\text{FR}} \leftarrow \textbf{ones}(n_{\text{FR}})/n_{\text{FR}}d$
  $\mathbf{P}_{\text{ES}} \leftarrow \textbf{ones}(n_{\text{ES}})/n_{\text{ES}}d$
  $P_{\text{ES}}^{\text{FR}} \leftarrow \textbf{Sinkhorn}(P_{\text{FR}}, P_{\text{ES}}, M, \lambda)$
  $A \leftarrow \textbf{MinMax}(P_{\text{ES}}^{\text{FR}})$
return $\textbf{Where}(A < 0, 1, 0)$

Les performances de cette méthode ont donc été mesurées sur les 100 phrases Français - Espagnol du jeu de données *golden collection*. Les grandeurs relevées pour chaque phrase sont le *rappel* R, la *précision* P et le *f-score* F, définies comme suit :

$$R = \frac{|L_{pred} \cap L_{true}|}{|L_{true}|}$$
$$P = \frac{|L_{pred} \cap L_{true}|}{|L_{pred}|}$$
$$F = \frac{P \cdot R}{2(P + R)}$$

où L_{pred} et L_{true} sont des listes de paires $(v_{w_{i_1}^{\text{FR}}}, v_{w_{j_1}^{\text{ES}}})$ représentants des alignements de mots deux-à-deux. Ces trois grandeurs sont mesurées pour $\lambda = 0.002$ à la vue des conclusions précédentes. Cette valeur permet de maximiser les écarts de valeurs de la solution, notamment en donnant une matrice très sparse.

Une fois ce modèle de référence déterminé et mesuré, certaines expérimentations peuvent être entreprises à partir de ce modèle. La première fait le lien entre nos études sur le paramètre de régularisation de l'algorithme de Sinkhorn. Les travaux de Cuturi [6] (plus précisément son site internet) exposent des contraintes supplémentaires que devrait vérifier la matrice M. Ces critères flous, "il faut que $\lambda \max_{i,j} M_{ij} \leq 200$" nous ont aiguillé vers un premier travail sur

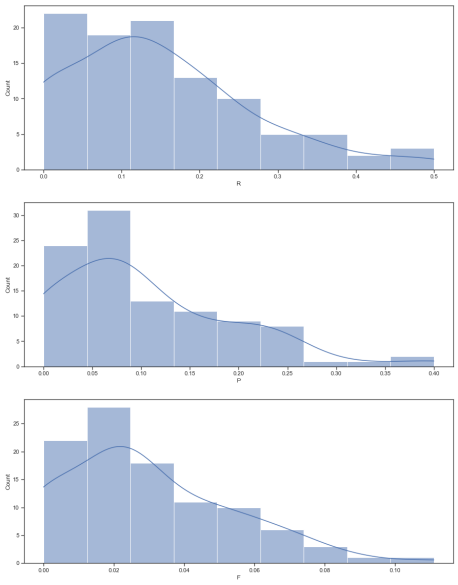

Figure 3: Histogrammes et densités empiriques des grandeurs R,P et F de référence.

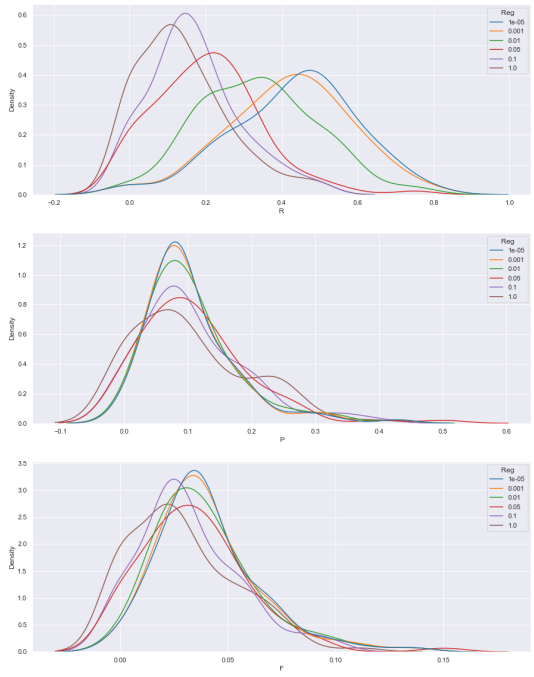

Figure 4: Densités estimées des valeurs de R,P,$F1$ en fonction du paramètre $\epsilon_r eg$ pour $\lambda = 0.002$

la régularisation de la matrice de coût. Nous avons donc exploré quelles valeurs de $\epsilon_r eg$ optimisait les résultats (donc les grandeurs R,P et $F1$) en utilisant $M \times \epsilon_r eg$ dans notre algorithme de Sinkhorn (Figure 3).

Le choix d'un paramètre *reg* très proche de 0 permet à l'algorithme de Sinkhorn de produire, pour un λ donné, des résultats alignements de meilleure qualité. Néanmoins, il reste possible que son rôle ne soit simplement que de favoriser la sparsité de la matrice finale et d'augmenter le nombre d'alignements, ce qui reste encore à être étudié. L'amélioration du *f-score* est quand même un indice positif de l'effet de ce paramètre puisque pour de mêmes conditions initiales, le nombre de $f-score$ égaux ou très proches de 0 diminue drastiquement.

Ces premières expérimentations ouvrent la porte à un travail sur les mesures qui déterminent la matrice de coût. Si le paramètre $\epsilon_r eg$ tend à augmenter le nombre d'alignements, une matrice de coût plus optimale pourrait finalement concilier résultats cohérents et régularisation de la matrice pour l'algorithme de Sinkhorn.

Dans la méthode de référence, la matrice de coût est donc définie comme la *cosine distance*, définie par $1 - \textbf{CosineSimilarity}(\cdot)$. Plusieurs travaux connexes écrivent utiliser simplement **CosineSimilarity**, c'est-à-dire le *cosinus*.

Néanmoins, cette fonction renvoie des valeurs dans $[-1, 1]$, où 1 signifie que les vecteurs sont similaires et -1 qu'ils sont opposés, en sens comme en forme, ceci dépendant du modèle d'*embedding* utilisé pour construire les vecteurs. Logiquement, pour que l'algorithme du Transport Optimal mette en relation deux vecteurs $v_{w_i^{\text{FR}}}$,$v_{w_j^{\text{ES}}}$ et similaires, il est nécessaire que M_{ij} soit proche de 0. Sans preuve mathématique, cela se comprend intuitivement en revenant à la formulation initiale du problème de Trans-

port Optimal qui est de déplacer le plus économiquement possible des "tas" de probabilités. Si deux "tas" sont "proches" ils seront donc facilement mis en relation pour limiter le coût total. Il faut donc que notre matrice de coût attribue des poids faibles pour les vecteurs à aligner.

La qualité d'une métrique sera donc sa capacité à évaluer proche de 0 des vecteurs à aligner et d'attribuer une "grande valeur" à deux vecteurs éloignés sémantiquement et linguistiquement. Cette notion de "grande valeur" fait l'objet d'un premier questionnement. Est-il préférable d'utiliser une distance à valeurs dans $[0, 1]$ ou une fonction de coût dans $[0, +\infty[$ dans la plupart des cas bornée. Nous prennons la décision de nous rapprocher de la formulation historique du problème du Transport Optimal et de ne pas nous contenter à des distances dans la boule unité. En effet, aucune contrainte ne semble nous en empêcher et nous trouvons intéressant de travailler, en plus des fonctions de coût, sur la répartition statistique (variance, quartiles,...) optimale des coûts pour orienter le système vers la bonne décision.

Nous cherchons donc une méthode non-linéaire de transformation de la distance initiale $(u, v) \rightarrow 1 - \textbf{CosineSimilarity}(u, v)$ afin de détacher vers 0 les vecteurs similaires, s'il en existe, tout en gardant un "ventre mou" compact et loin de l'origine. Ainsi, dans le cas où un vecteur est proche de tous les autres comme c'est le cas de mots clés, commme "le", "du" ou "un", cette méthode permet de détacher les plus pertinents tout en gardant la compacité de l'ensemble des autres coûts calculés. Nous proposons ici deux méthodes nommées **DiscExp** et **DiscSquare**, définies pour $x \in X$ ensemble d'éléments de $\mathbf{R}$, telles que :

$$\text{avec } \alpha = \mu(X) - \textbf{IRQ}(X)$$

$$\textbf{DiscExp}(x) = \begin{cases} \alpha \cdot \frac{e^x - 1}{e^\alpha - 1} & \text{si } x \leq \alpha \\ x & \text{sinon.} \end{cases}$$

$$\textbf{DiscSquare}(x) = \begin{cases} \frac{x^2}{\alpha} & \text{si } x \leq \alpha \\ x & \text{sinon.} \end{cases}$$

Ces fonctions étant dépendantes de la variance des données, elle s'adaptent sans rigidité à chaque échantillon. Elles jouent un rôle d'étirement des valeurs aux extrémités et de compression à l'intérieur, leur effet étant illustré Figure 4 en comparaison avec la méthode de référence.

L'intégration de ces métriques dans l'algorithme de Sninkhorn ne permet pas de tirer de réelles conclusions quant à l'efficacité de ce travail sur les distances. Comme illustré Figure 6, les résultats des grandeurs d'évaluations diffèrent peu de la méthode de référence.

Au delà du travail et de la réflexion sur la construction de la matrice de coût, une réflexion sur la méthode choisie pour extraire les alignements finaux est également intéressante. La méthode de référence propose comme de nombreuses autres papiers, une régulrisation $MnMx(\cdot)$ suivie d'un *thresholding* pour extraire les connections dépassant un certain seuil défini et fixé en amont. La régularisation $MnMx(\cdot)$ est discutable puisqu'elle oblige à faire au moins une association par mot. Pourtant, aucune règle linguistique ne l'indique et il doit pouvoir être possible qu'un mot soit aligné avec aucun autre. Toutes les langues ont leur particularités qui ne sont pas transposables dans les autres et cela doit pouvoir être pris en compte. De plus, la méthode de *thresholding* est peu flexible et pourrait être repensée plus finement.

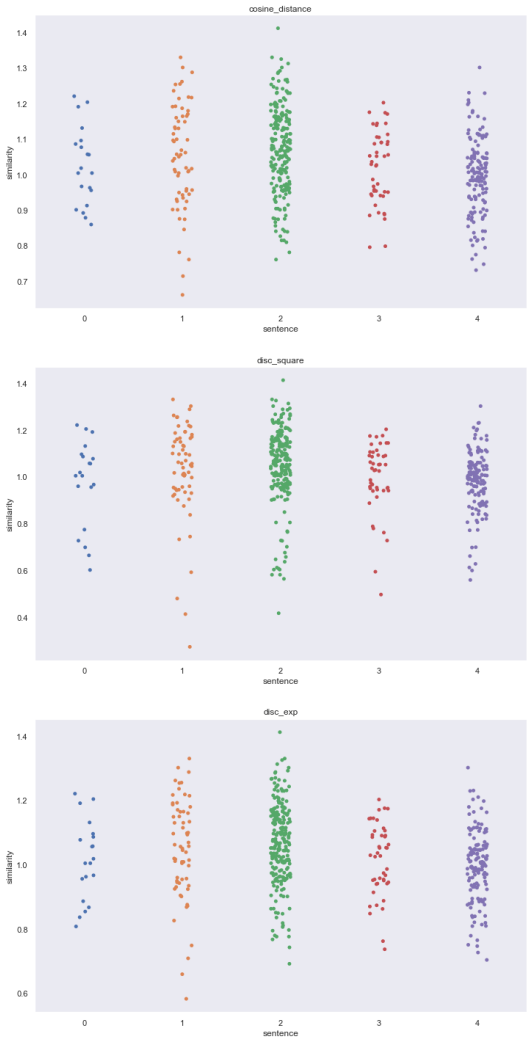

Figure 5: Répartition des poids de la matrice M pour les 5 premières phrases du jeu de données pour les 3 métriques.

Nous proposons une solution fondée sur la détection d'anomalies dans un jeu de données en utilisant l'algorithme de clustering DBScan de la librairie Scikit-learn. Cet algorithme va déterminer des clusters de voisins, faisant émerger des points dits d'*intérieur* et des points d'*extérieur*. Ces derniers sont donc les fameux *outliers*, ceux qui se détachent du reste des points vis-à-vis de la distribution de l'ensemble. Ainsi, nous obtenons une méthode moins rigide et prêtant plus d'attention à la répartition des données en permettant, entre autres, qu'un mot ne soit aligné avec aucun autre.

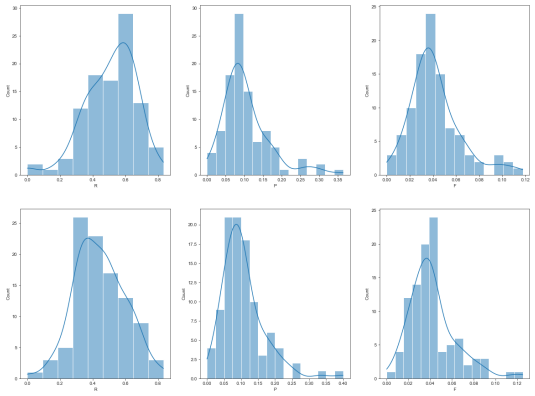

Figure 6: Evaluation des performances **DiscExp** (en haut) et **DiscSquare** (en bas) pour $\lambda = 0.002$ et $reg = 0.0001$.

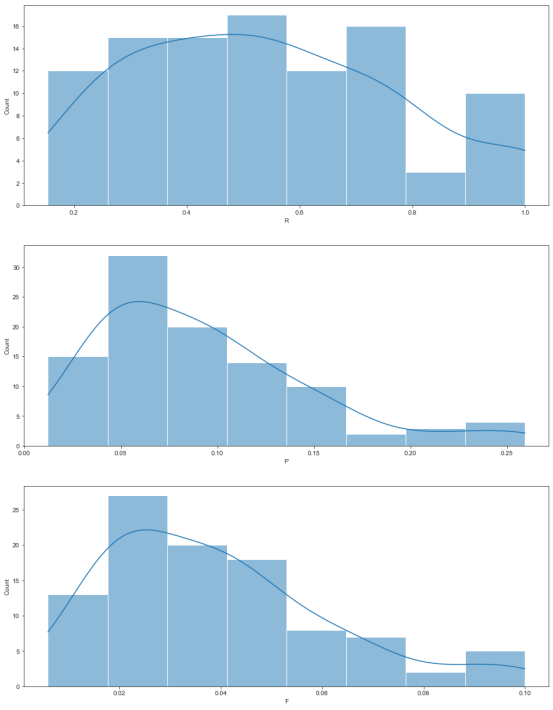

Figure 7: Evaluation des performances la méthode de clustering pour l'extraction des alignements, $\lambda = 0.002$ et $reg = 0.0001$.

Les performances de cette méthode est finalement négligeable puisque comme le montre la Figure 7, nous ne constatons aucune modification majeure des valeurs prises par R, P, et F.

La méthode DBScan est une méthode

paramétrique très sensible, ce qui rend son évaluation difficile sans réelle étude approfondie. Ses paramètres sont *eps*, la distance maximale pour définir deux voisins, et *minSamples*, le nombre de voisins minimal pour constituer un cluster. Le choix de ces paramètres est crucial pour une solution de bonne qualité puisque la méthode y est très sensible. Nous avons décidé, après de nombreux essais, de prendre comme valeur :

$$eps = \sigma(X)/4$$
$$minSamples = \max(2, \lfloor \frac{|X|}{2} \rfloor)$$

puisqu'il est intuitif de prendre des valeurs basées sur la distribution des données. Le cardinal et l'écart-type sont deux grandeurs empiriques capables de fournir suffisamment d'information pour traduire la distribution des valeurs et donc optimiser le rôle de DBScan.

Conclusions

En partant de nombreux articles récents sur la thématique du *Word embedding*, nous nous sommes orienté vers l'utilisation du Transport Optimal pour une application au problème du *Word Alignment*. Nous nous sommes donc intéressés à des outils d'optimisation de l'algorithme de Sinkhorn dans la résolution du Transport Optimal.

Ensuite, nous avons cherché à optimiser la matrice de coût à différentes étapes de notre méthode initiale. La première étape a été la régularisation de la matrice de coût afin d'augmenter l'efficacité de l'algorithme de Sinkhorn. La seconde étape, quant à elle, a été de transformer les coûts de notre matrice afin d'aiguiller le système dans la bonne direction. Finalement, nous avons cherché à repenser la façon dont l'extraction des alignements finaux était faite en proposant une solution plus souple fondée sur une méthode de clustering.

D'un point de vue global, nos expérimentations ont permis une amélioration légère mais visible des performances de l'algorithme de référence. De nombreuses pistes sont encore à explorer, notamment l'utilisation d'*embeddings* positionnels, une recherche d'optimisation des différents paramètre plus profonde et une étude sur un jeu de données plus grand.

References

[1] T. Alkhouli, G. Bretschner, J.-T. Peter, M. Hethnawi, A. Guta, and H. Ney. Alignment-based neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 54–65, Berlin, Germany, Aug. 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-2206. URL https://aclanthology.org/W16-2206.

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan, 2017.

[3] M. Artetxe, G. Labaka, and E. Agirre. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294, Austin, Texas, Nov. 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1250. URL https://aclanthology.org/D16-1250.

[4] M. Blondel, A. F. T. Martins, and V. Niculae. Learning classifiers with fenchel-young losses: Generalized entropies, margins, and algorithms, 2018.

[5] W. Chen, E. Matusov, S. Khadivi, and J.-T. Peter. Guided alignment training for topic-aware neural machine translation, 2016.

[6] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transportation distances, 2013.

[7] Z.-Y. Dou and G. Neubig. Word alignment by fine-tuning embeddings on parallel corpora, 2021.

[8] L. Liu, M. Utiyama, A. M. Finch, and E. Sumita. Neural machine translation with supervised attention. *CoRR*, abs/1609.04186, 2016. URL http://arxiv.org/abs/1609.04186.

[9] T. Mikolov, Q. V. Le, and I. Sutskever. Exploiting similarities among languages for machine translation, 2013.

[10] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality, 2013.

[11] M. J. Sabet, P. Dufter, F. Yvon, and H. Schütze. Simalign: High quality word alignments without parallel training data using static and contextualized embeddings, 2021.

[12] S. Vogel, H. Ney, and C. Tillmann. Hmm-based word alignment in statistical translation. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 2*, COLING '96, page 836–841, USA, 1996. Association for Computational Linguistics. doi: 10.3115/993268.993313. URL https://doi.org/10.3115/993268.993313.

[13] C. Xing, D. Wang, C. Liu, and Y. Lin. Normalized word embedding and orthogonal transform for bilingual word translation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1006–1011, Denver, Colorado, May–June 2015. Association for Computational Linguistics. doi: 10.3115/v1/N15-1104. URL https://aclanthology.org/N15-1104.

[14] M. Zhang, Y. Liu, H. Luan, and M. Sun. Earth mover's distance minimization for unsupervised bilingual lexicon induction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1934–1945, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1207. URL https://aclanthology.org/D17-1207.