# TRANSPORT OPTIMAL POUR ALIGNEMENT DE MOTS

ÉCOLE POLYTECHNIQUE
UNIVERSITÉ PARIS-SACLAY

EA MAP511

Paul Mortamet | Adrien Lagesse
X2019

# INTRODUCTION

- Déroulé de notre EA


- Sujet précisé : *Transport Optimal pour alignement de mots*

- Deux expérimentations afin d'optimiser les méthodes existantes
- Deux ajouts permettant d'affiner les méthodes existantes

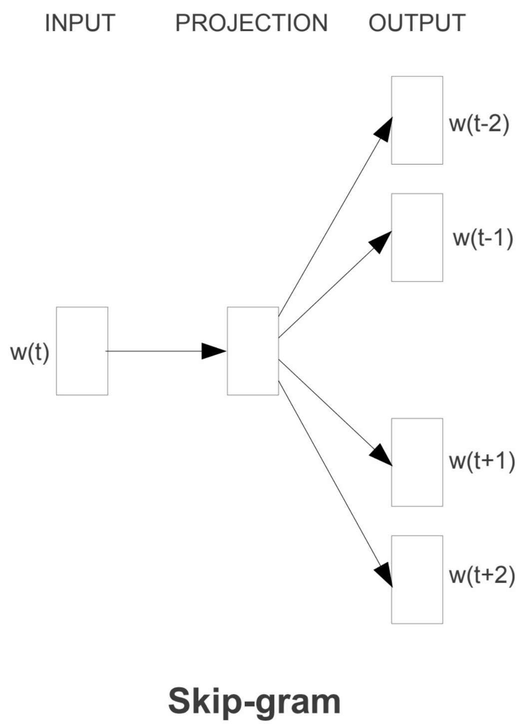
- Applications dans des différents domaines du NLP

# SOMMAIRE

# TRAVAUX PRÉLIMINAIRES

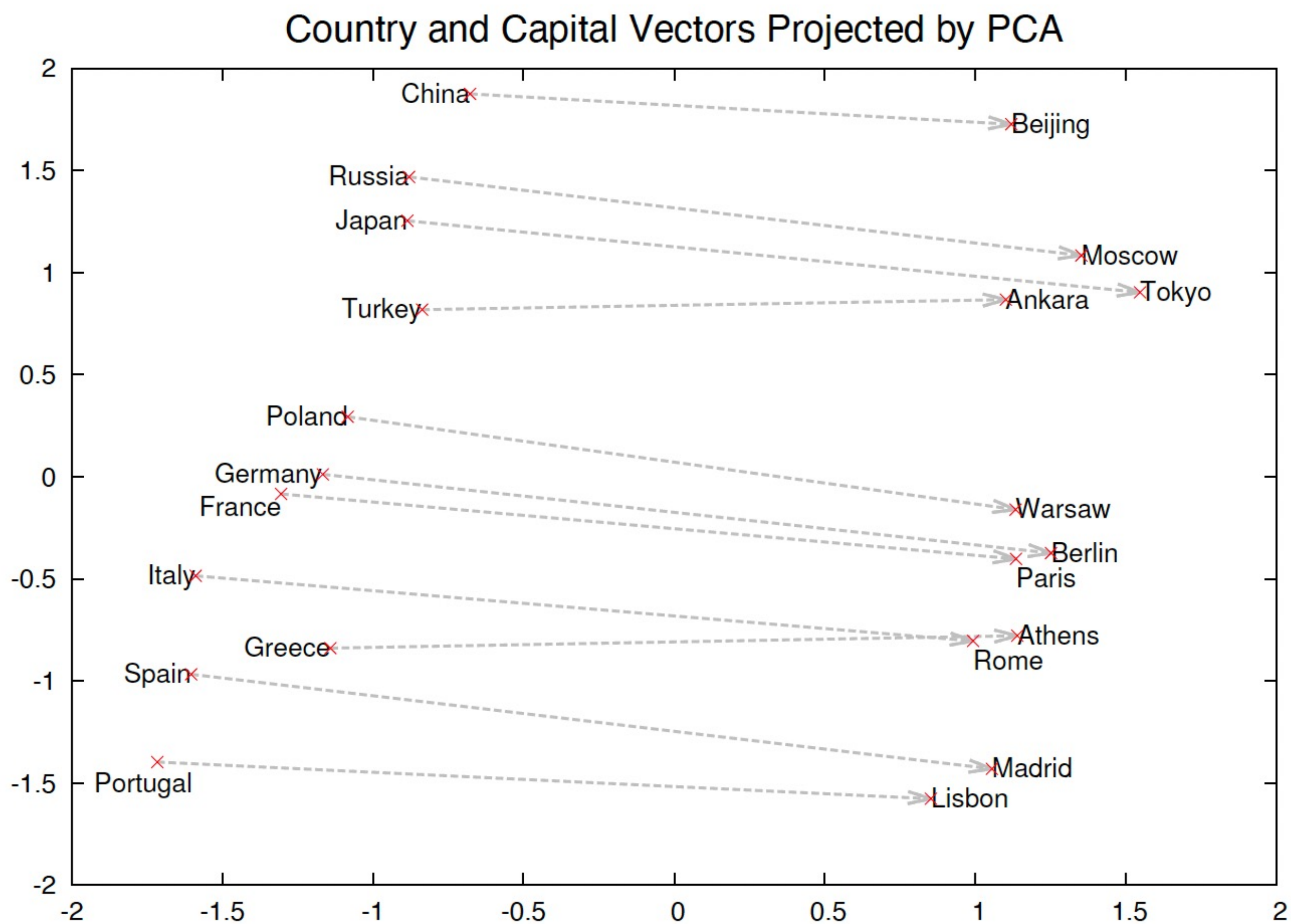
# Word Embeddings

## TRAVAUX DE T. MIKOLOV




- Notion de contexte
- Recherche de relations linéaires

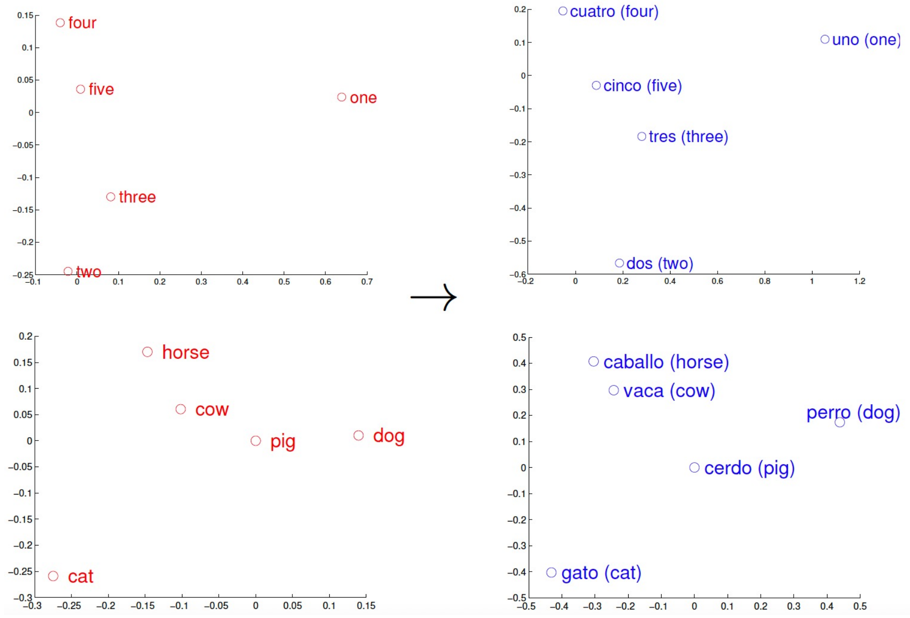
# RELATIONS LINÉAIRES DE MIKOLOV



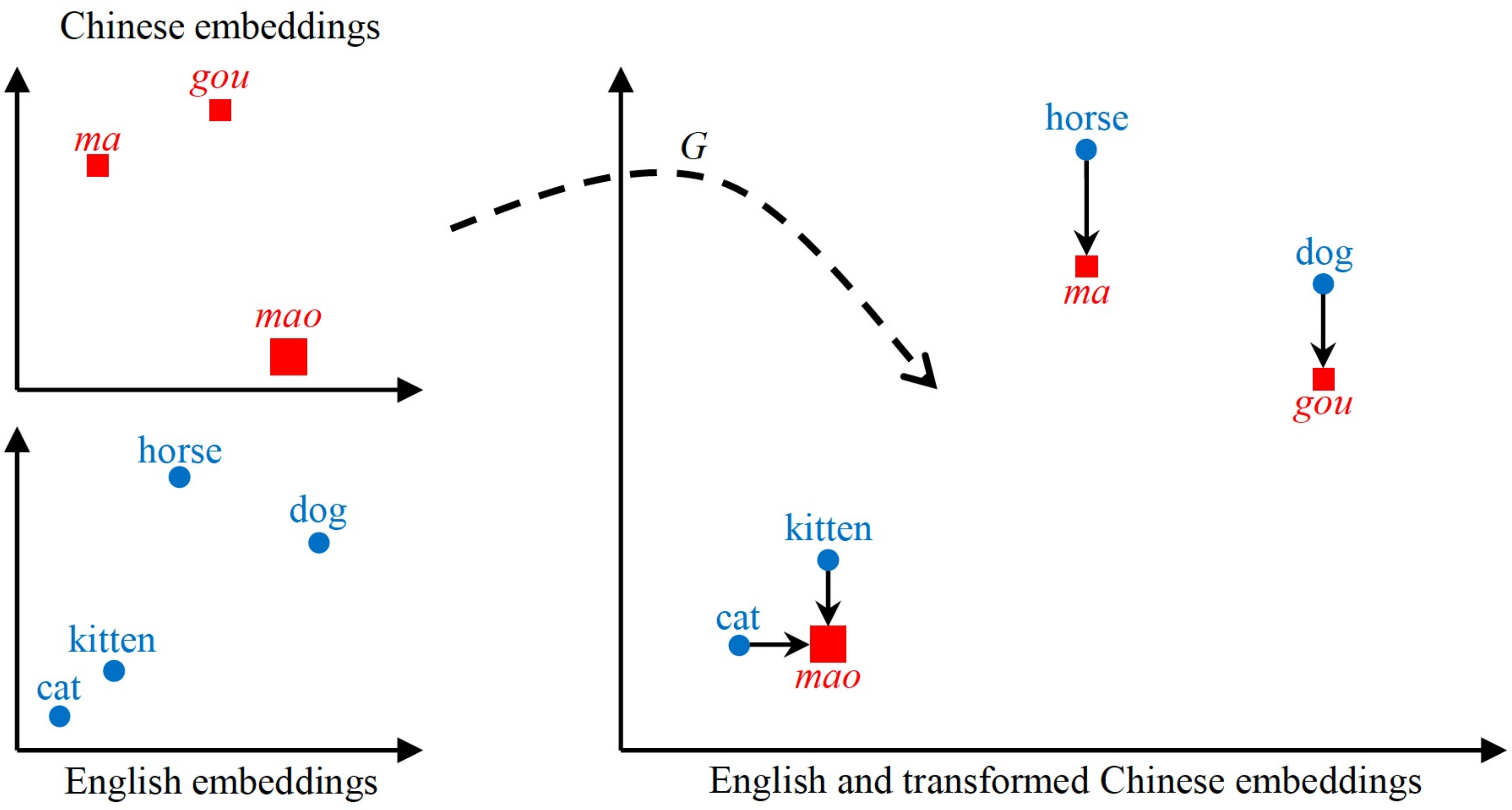
# Approche multilingue

## TRAVAUX DE MIKOLOV, XING, ARTEKA, ZHANG

- Conserver ces relations linéaires
- Translation d'un espace à un autre
- Regroupements de mots et de notions conservés *au maximum*



# TRANSPOSITION D'ESPACES D'EMBEDDINGS

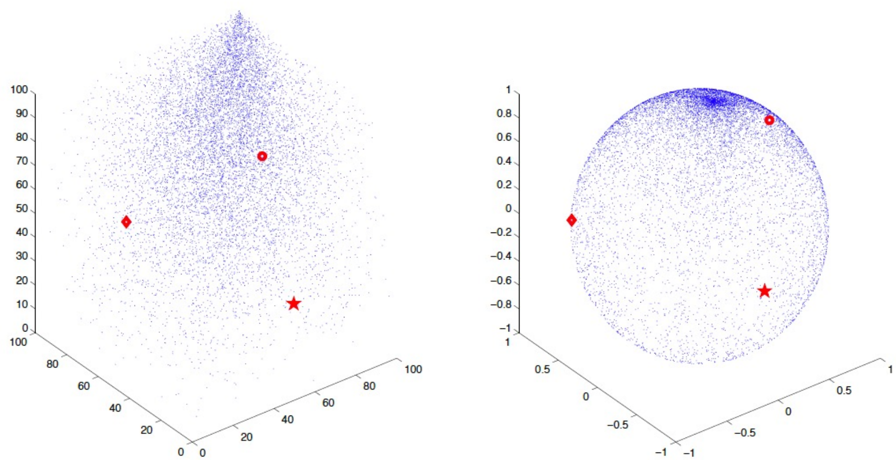


# Méthodes et améliorations

## TRAVAUX DE XING ET ARTETXE

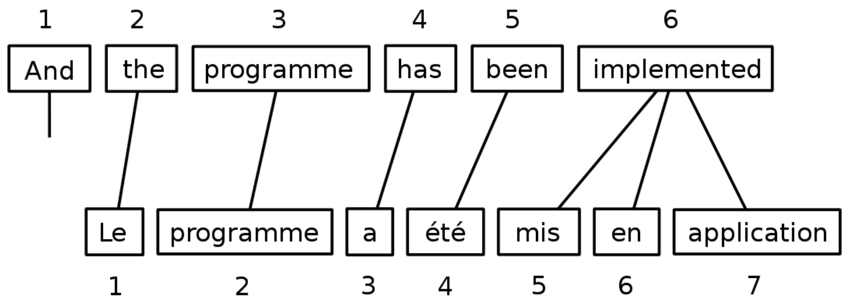
Méthode initiale

$$\arg\min_{W} \sum_{i} \|X_{i*}W - Z_{i*}\|^2$$



# Alignement de mots

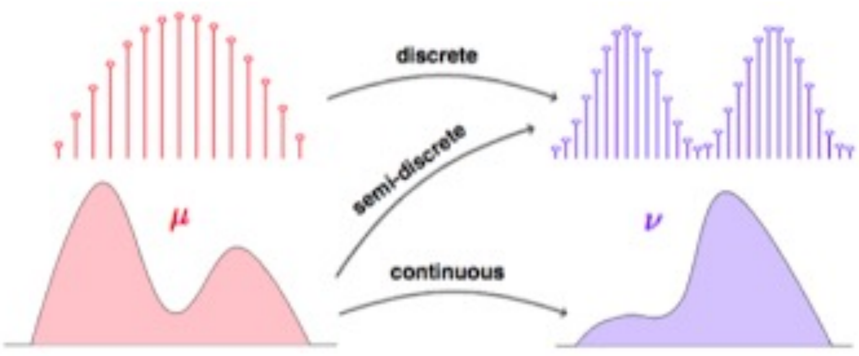
## DÉFINITION DU CADRE D'ÉTUDE




- Représentation avec des distributions
- Idée de mise en relation plutôt que de traduction un à un
- Visualisation plus simple

# Transport Optimal

## TRAVAUX DE CUTURI

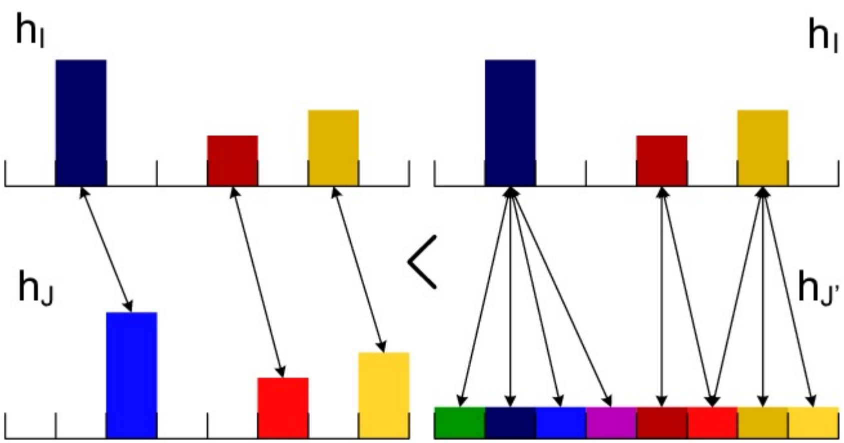



- Trouver le déplacement le plus économique entre deux distributions
- Nous travaillons avec des distributiosn discrètes

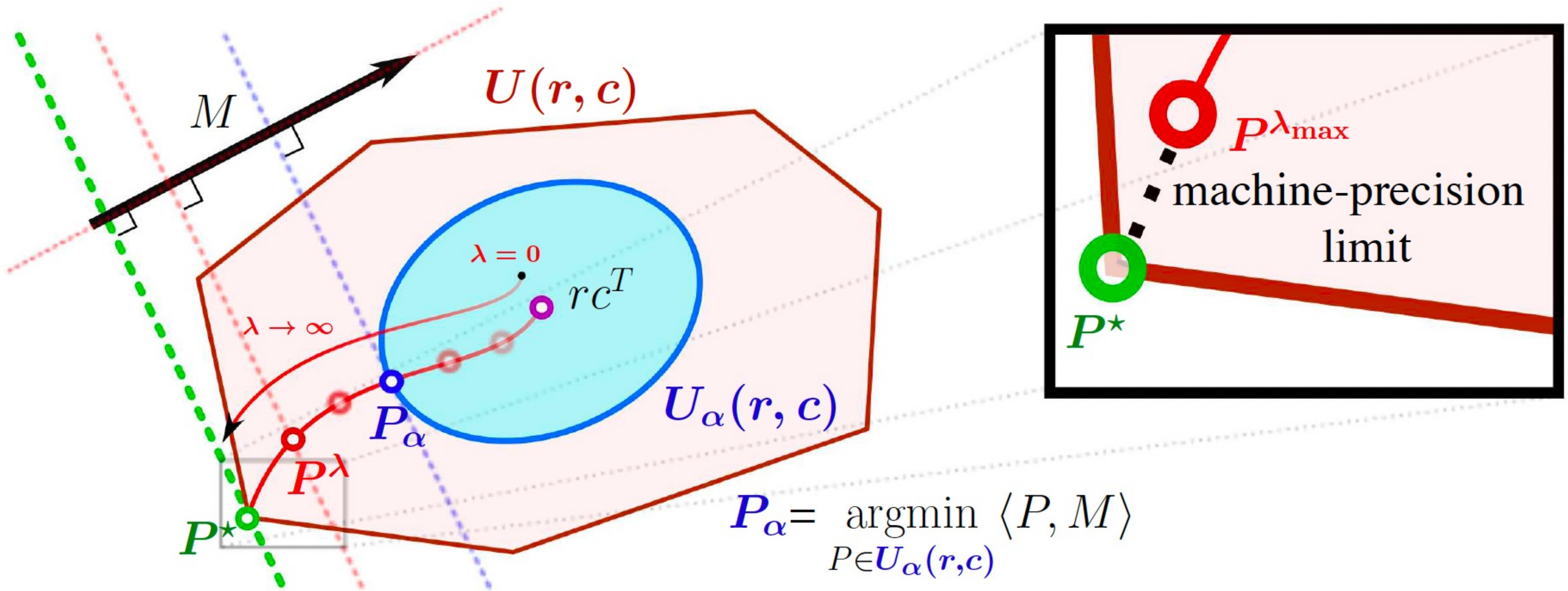
# Earth's Mover Distance

## TRAVAUX DE ZHANG

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




- Représentation des embeddings sous forme de distributions
- Idée de mise en relation plutôt que de traduction forte




$$P^\star = \underset{P \in U(r,c)}{\operatorname{argmin}} \langle P, M \rangle$$

$$\boxed{P^\lambda = \underset{P \in U(r,c)}{\operatorname{argmin}} \langle P, M \rangle - \frac{1}{\lambda} h(P)}$$

$\mathbb{R}^{d \times d}$

**Algorithm 1** Algorithme de Sinkhorn

**Require:** $M \in \mathbf{R}_+^{d\times d}$, $u, v \in \Sigma_d - \{\mathbf{0}_d\}$

$\boxed{K \leftarrow \exp(-\lambda M)}$

$x_1 \leftarrow \mathbf{ones}(d)/d$

$\tilde{K} \leftarrow \mathbf{diag}(1/v)K$

**while** u ne change plus **do**

$x_1 \leftarrow 1/(\tilde{K}(v/K^T x_1))$

**end while**

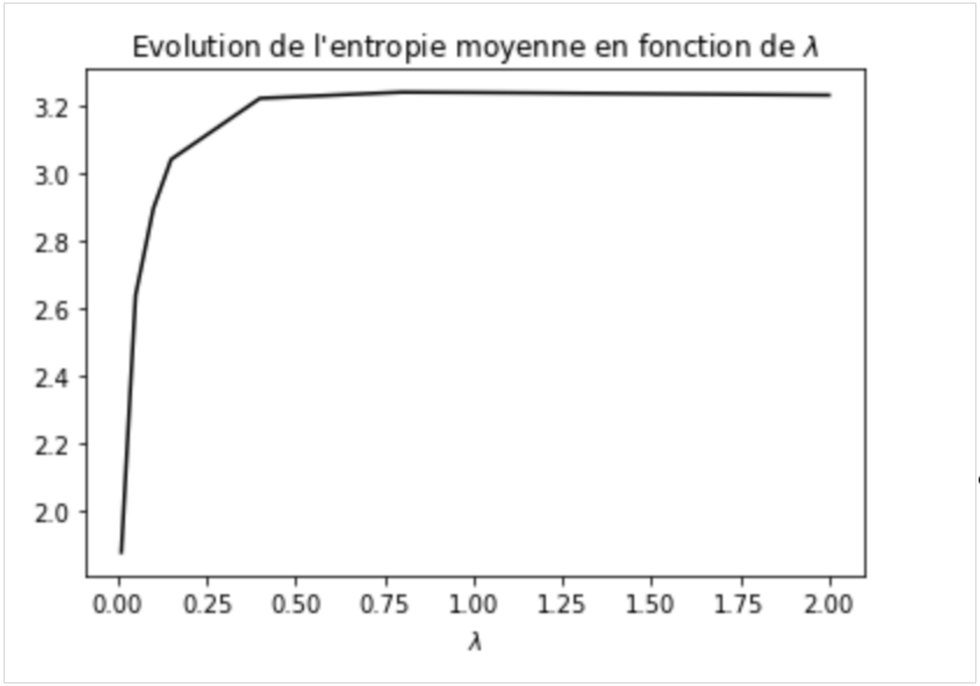
$x_2 \leftarrow v/(K^T x_1)$

**return** $\boxed{\mathbf{diag}(x_1)K\,\mathbf{diag}(x_2)}$

# OPTIMISATION DE L'ALGORITHME DE SINKHORN

# Régularisation de l’entropie

## OPTIMISATION DE L’ALGORITHME DE SINKHORN




- L’article ne nous dit pas comment choisir lambda
- En pratique le choix et très important. Résultats très différent en fonction de lambda.
- Peut pas être pris trop petit non plus.

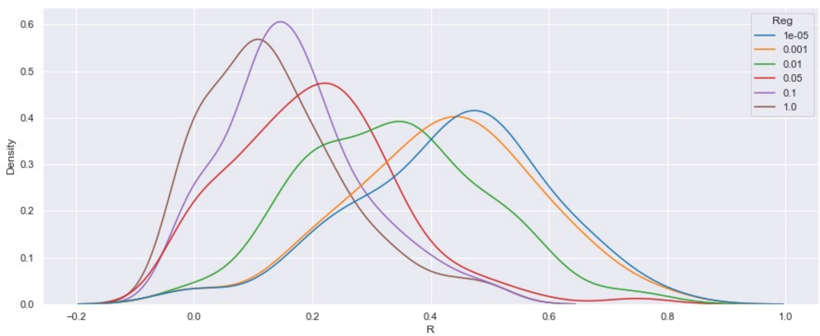
# Régularisation du coût

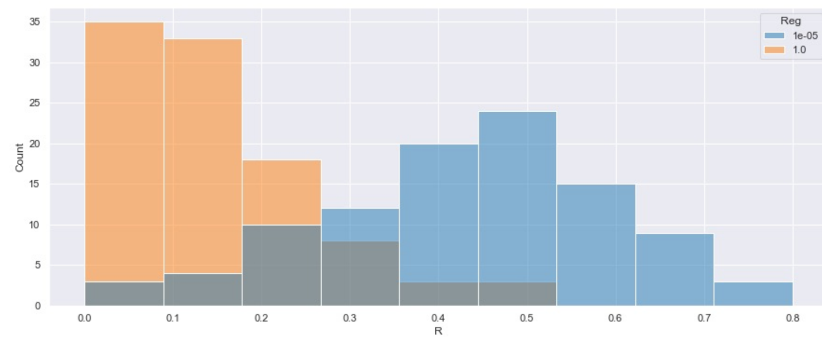
## OPTIMISATION DE L'ALGORITHME DE SINKHORN

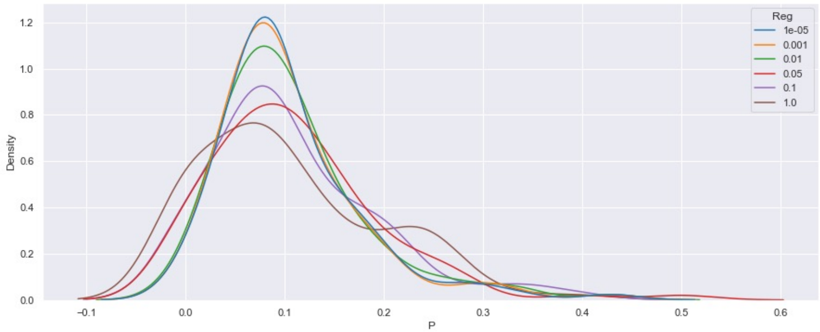
We suggest thus that you choose $\lambda$ by cross-validation and start with a small value. A small value for $\lambda$ means that the product of $\lambda$ with $\max_{ij} M_{ij}$ (where $M$ is the ground metric) should not be too large. You need to make sure that $\lambda \max_{ij} M_{ij} \leq 200$ which would be the threshold when numerical problems appear.

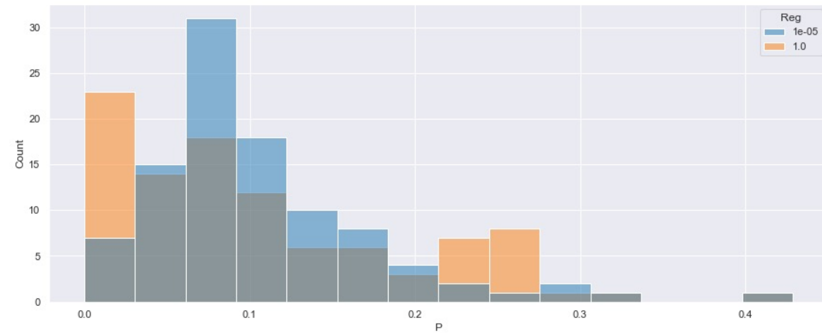
- Adapter la matrice M au paramètre de régularisation lagrangien
- Problèmes lorsque $\lambda$ est proche de 0
- Aucune littérature précise sur le sujet

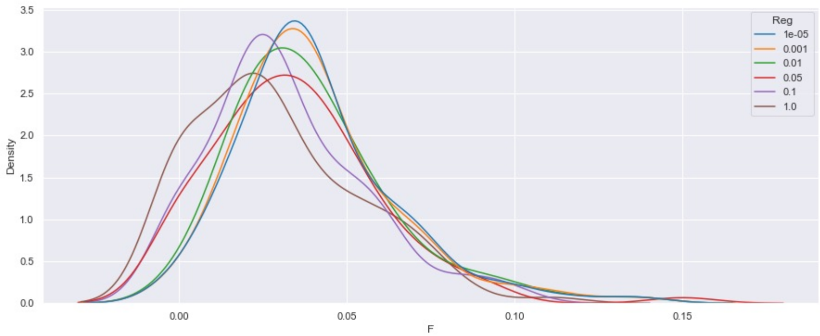
# RÉGULARISATION DU COÛT

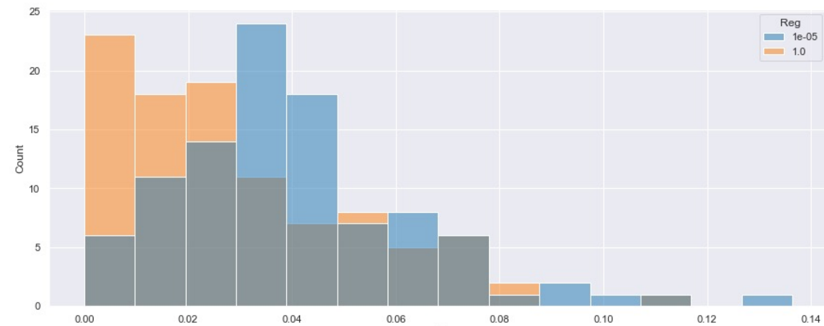


X

# OPTIMISATION DE LA MATRICE DE COÛT

# Distance et coût

## OPTIMISATION DE LA MATRICE DE COÛT

- Quelle notion utiliser? Distance ou coût?
- Quelles valeurs cherche-t'on à avoir?
- Modèle de base avec $1 - CosineSimilarity(u, v)$
- Transformer cette matrice de base pour tirer les résultats vers ce que l'ont cherche →

$$Cos\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \, \|\vec{b}\|} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2} \sqrt{\sum_1^n b_i^2}}$$

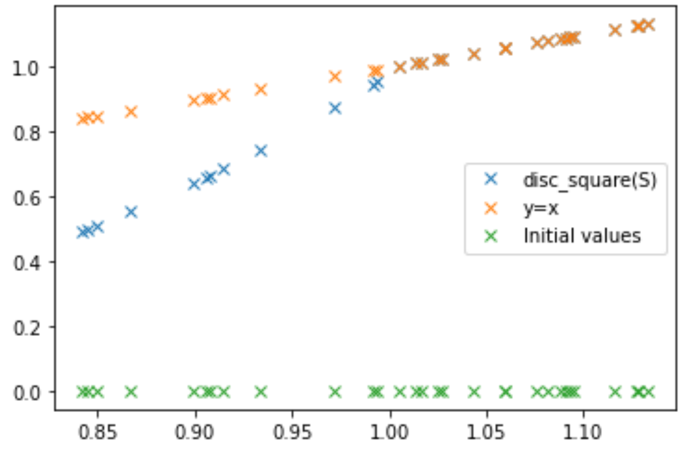
- Tirer vers 0 les vecteurs les plus similaires
- Surtout ceux qui se détachent
- Garder les blocs compacts

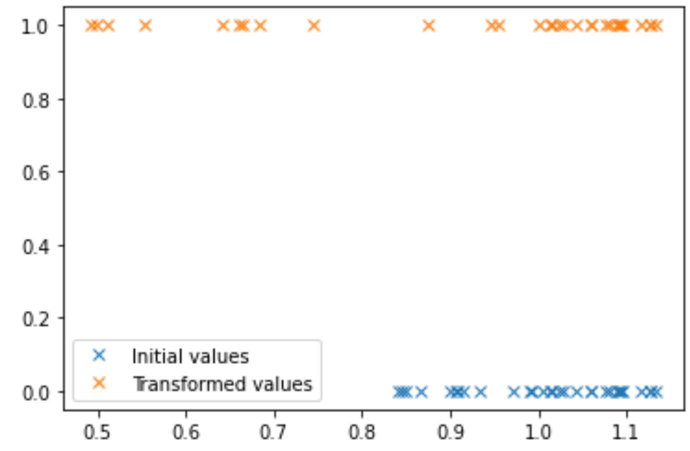
# OPTIMISATION DE LA MATRICE DE COÛT

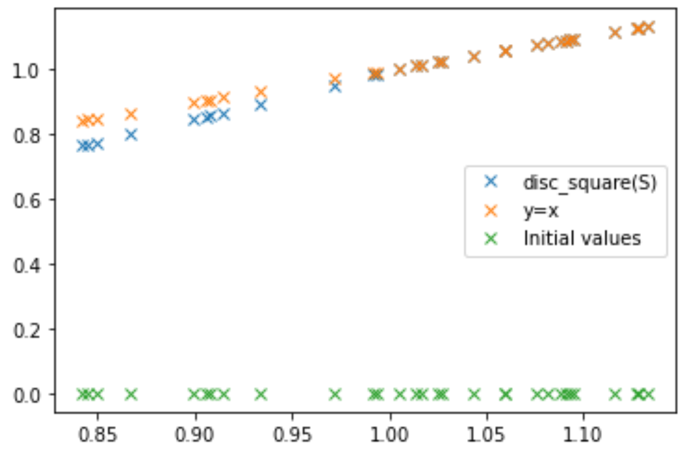
$$\mathbf{DiscSquare}(x) = \begin{cases} \frac{x^2}{\alpha} & \text{si } x \leq \alpha \\ x & \text{sinon.} \end{cases}$$

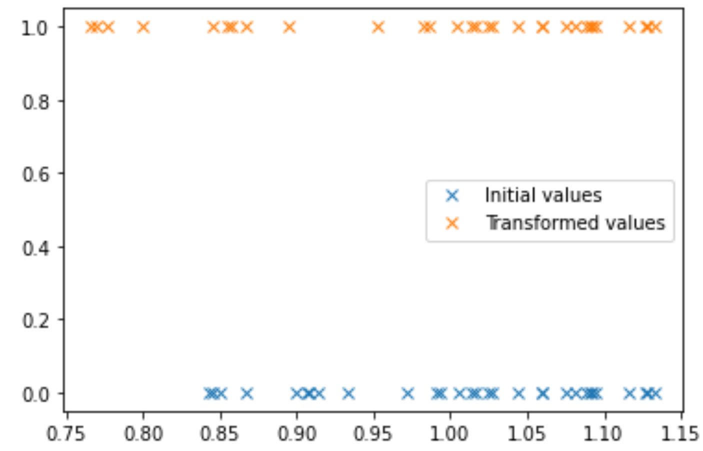
avec $\alpha = \mu(X) - \mathbf{IRQ}(X)$

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

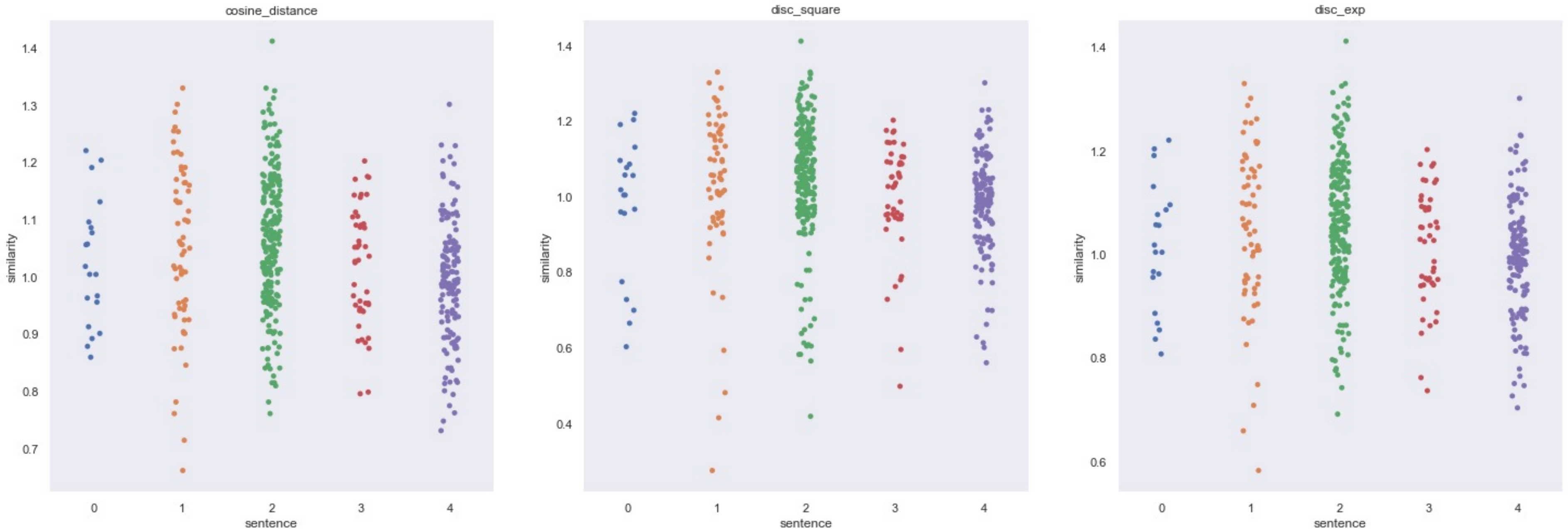


# OPTIMISATION DE LA MATRICE DE COÛT



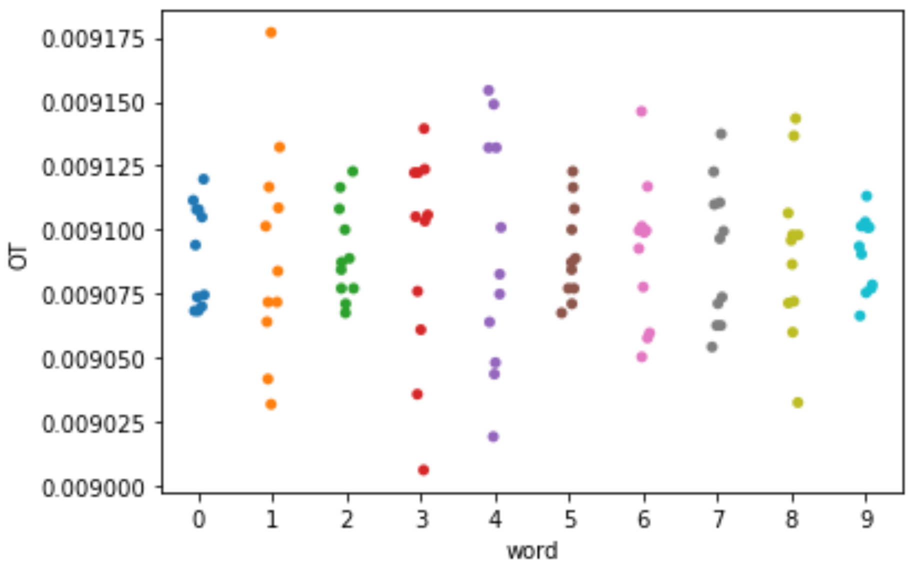
# Extraction des alignements

## OPTIMISATION DE LA MATRICE DE COÛT

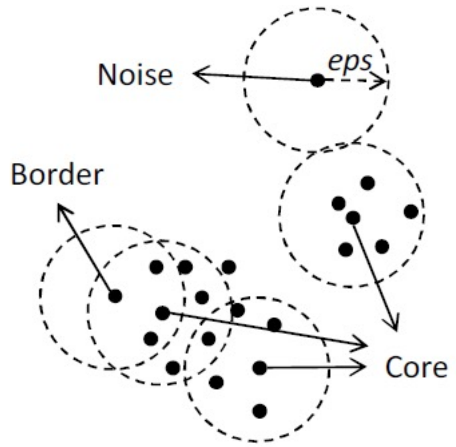
- Régularisation *MinMax*
- Extraction des éléments supérieurs à un seuil fixé

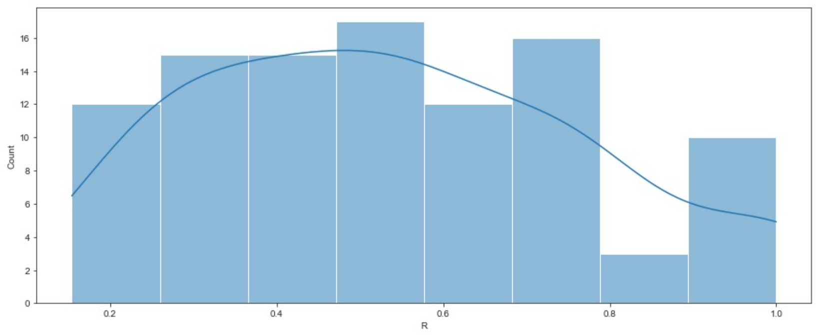
---

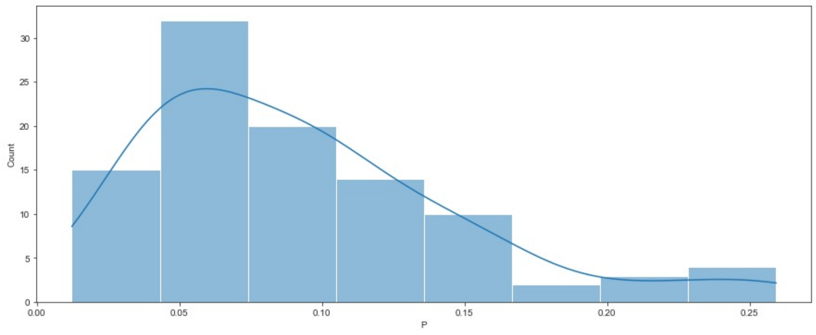
- *MinMax* impose la présence de 1 donc des alignements « forcés »
- Le seuil ne s'adapte pas aux différents distributions possibles

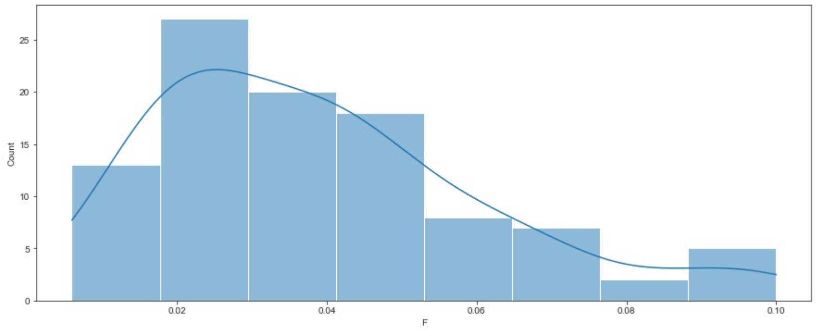


Méthode DBScan




Très sensible aux paramètres *eps* et *n_voisins*


X

# CONCLUSION

# CONCLUSION

- Un sujet très en vogue
- Le peu de littérature précise sur le sujet permet de nombreuses expérimentations
- Nous proposons deux modifications
  - Résultats meilleurs
  - Résultats à remettre en contexte
- La taille du jeu de données est aussi à repenser, ainsi qu'une évaluation des alignements de référence