March, 2020

ATHENS course : **TPT-37**

Algorithmic Information and Artificial Intelligence

Micro-study

teaching.dessalles.fr/

Name: Joachim CARVALLO

Résumé

Cette courte étude propose une technique de plongement lexical basée sur la *Normalized Information Distance*. Pour cela, nous introduisons une variante à la distance Google : la distance Wikipedia.

Introduction

Le plongement lexical permet de représenter chaque mot d'un dictionnaire par un vecteur de nombres réels. Cet ensemble de techniques a pour objectif d'assigner des vecteurs proches à des mots utilisés dans des contextes similaires. Cette idée se base sur l'hypothèse que des mots utilisés dans des contextes similaires doivent êtres proches sémantiquement.

La distance Google, ou *Normalized Google Distance* [1], mesure la similarité entre deux concepts x et y, en se basant sur des données extraites du web. Elle est définie par :

$$NGD(x,y) = \frac{max[log_2(g(x)), log_2(g(y))] - log_2(g(x,y))}{log_2(N) - min[log_2(g(x)), log_2(g(y))]}$$

où $g(x)$ correspond au nombre de pages web contenant x, $g(x,y)$ au nombre de pages contenant à la fois x et y, d'après Google, et N au nombre total de pages indexées par Google.

Cette distance est en fait une implémentation pratique de la *Normalized Information distance* [2], issue de la théorie algorithmique de l'information, où la complexité de Kologorov de x est estimée par : $log_2(N/g(x))$. Intuitivement, la distance Google quantifie la proximité de deux concepts en utilisant des contextes créés par l'homme dans lesquels ils coexistent.

Pour passer d'un dictionnaire de n mots à une représentation vectoriel de chacun d'entre eux, l'idée proposée est la suivante :

1. Construire la matrice des distances Google deux à deux entre les mots du dictionnaire ;
2. En déduire des coordonnées pour chacun des mots, dans un espace de dimension $\leq n$, cohérentes avec notre matrice des distances ;
3. Éliminer les dimensions de plus faible variance afin d'obtenir des vecteurs de dimension d.

La distance Wikipedia

La première étape, consistant à construire la matrice des distances Google, pose plusieurs problèmes. Pour remplir notre matrice, il nous faut récupérer le nombre de page contenants chacun des couples de mots de notre dictionnaire. Ceci implique d'envoyer n^2 requêtes à Google. Pour des dictionnaires de très petites tailles, cela est envisageable. Cependant, un dictionnaire permettant de travailler sur des tâches de traitement automatique du langage naturel, par exemple, nécessite plusieurs milliers, voir dizaines de milliers de mots. En prenant 5000 mots par exemple, et avec une machine envoyant une requête par seconde, cela prendrait 289 jours pour extraire nos données. De plus, Google bloque très rapidement les adresses IP envoyant un nombre anormal de requêtes. Ainsi, la construction de la matrice des distance Google n'est pas possible directement.

La solution que nous proposons pour contourner ce problème est de ce restreindre uniquement aux pages Wikipédia. Cela réduit le nombre total de pages possibles d'environ 2×10^{12} à 6 231 000 (pour les pages en anglais). Les pages Wikipédia possèdent plusieurs avantages majeurs. Elles sont en accès libre et il est possible de télécharger tout leur contenu texte, au prix de quelques giga-octets de mémoire (voir ici pour plus de détails). Ainsi, la création d'une matrice de distances Wikipédia ne dépend pas d'un moteur de recherche comme Google et peut être beaucoup plus rapide. De plus, malgré ce nombre de pages relativement faible, en comparaison à l'intégralité du web du moins, ces pages sont très structurés et touchent à des contenus extrêmement divers, assurant une diversité de représentation de tous les mots. Wikipédia étant une encyclopédie qui touche à presque tous les domaines de la connaissances humaines, les mots se trouvent représentés dans des domaines où leur sémantique est particulièrement importante, et donc les co-occurences de mots sont, à priori, particulièrement porteuses d'information.

Estimation des coordonnées

Pour passer de la matrice des distances à des coordonnés dans un espace à dimension $\leq n$, la méthode est la suivante :

Si l'on note $(D)_{ij}$ notre matrice de distances, on construit la matrice M telle que :

$$M_{ij} = \frac{D_{1j}^2 + D_{i1}^2 + D_{ij}^2}{2}$$

Puis, les coordonnées des points s'obtiennent par décomposition en valeurs propres de M: $M = USU^T$. Alors, la matrice $X = U\sqrt{S}$ donne les positions des points (ou chaque ligne correspondant à un point). Ainsi, la plus petite dimension (euclidienne) dans laquelle les points peuvent être incorporés est donnée par le rang de la matrice M. Cependant, ce procédé ne fonctionne pas si la matrice n'est pas semi-définie positive.

En pratique, la distance Wikipédia ne donne pas des matrices de distances semi-définie positive, car elle n'est pas formellement une distance : elle viole l'inégalité triangulaire. Ainsi, il est impossible d'obtenir des coordonnées respectant parfaitement la matrice des distances. Toutefois, en prenant la valeur absolue des valeurs propres de la matrice M, ou en les réduisant à zéro, des coordonnés raisonnables peuvent être estimées.

Réduction de la dimension

Pour obtenir des vecteurs de la dimension voulue d, il suffit de ne conserver que les d plus grandes valeurs propres de la matrice M, et mettre les autres à zéro dans le calcul de X.

Mise en pratique

Dans cette micro étude, nous évaluerons la méthode proposée de façon très succincte par manque de temps. Nous avons composé manuellement un jeu de données composé de 110 mots anglais, rentrants chacun dans l'une des dix catégories suivantes : couleurs, chiffres, animaux, métiers, vêtements, pièces de la maison, fruits, légumes, sentiments et prénoms. Nous allons procéder au plongement lexical de ce petit dictionnaire afin de voir si les relations entre nos mots respectent les catégories sémantiques dans lesquels ils rentrent intuitivement. Afin de visualiser nos mots dans leur espace de dimension très supérieur à 3, nous utiliserons l'algorithme t-SNE. Pour extraire notre matrice de distance, étant donné notre petit dictionnaire, nous nous contentons de récupérer nos données via web-scrapping (sur le moteur de recherche Exalead).

Résultats

Sur la figure ci-dessus, nous pouvons voir une représentation de deux dimensions de nos mots, réalisé par l'algorithme t-SNE, qui permet de mettre en valeur des structures dans nos données. Nous espérons que ces structures correspondent aux 10 catégories de mots utilisés pour créer notre dictionnaire. Ces catégories sont représentées par les couleurs. Attention, cette représentation met en valeur les relations locales entre les mots, en revanche, il n'est pas possible d'interpréter la position des structures entre elles.

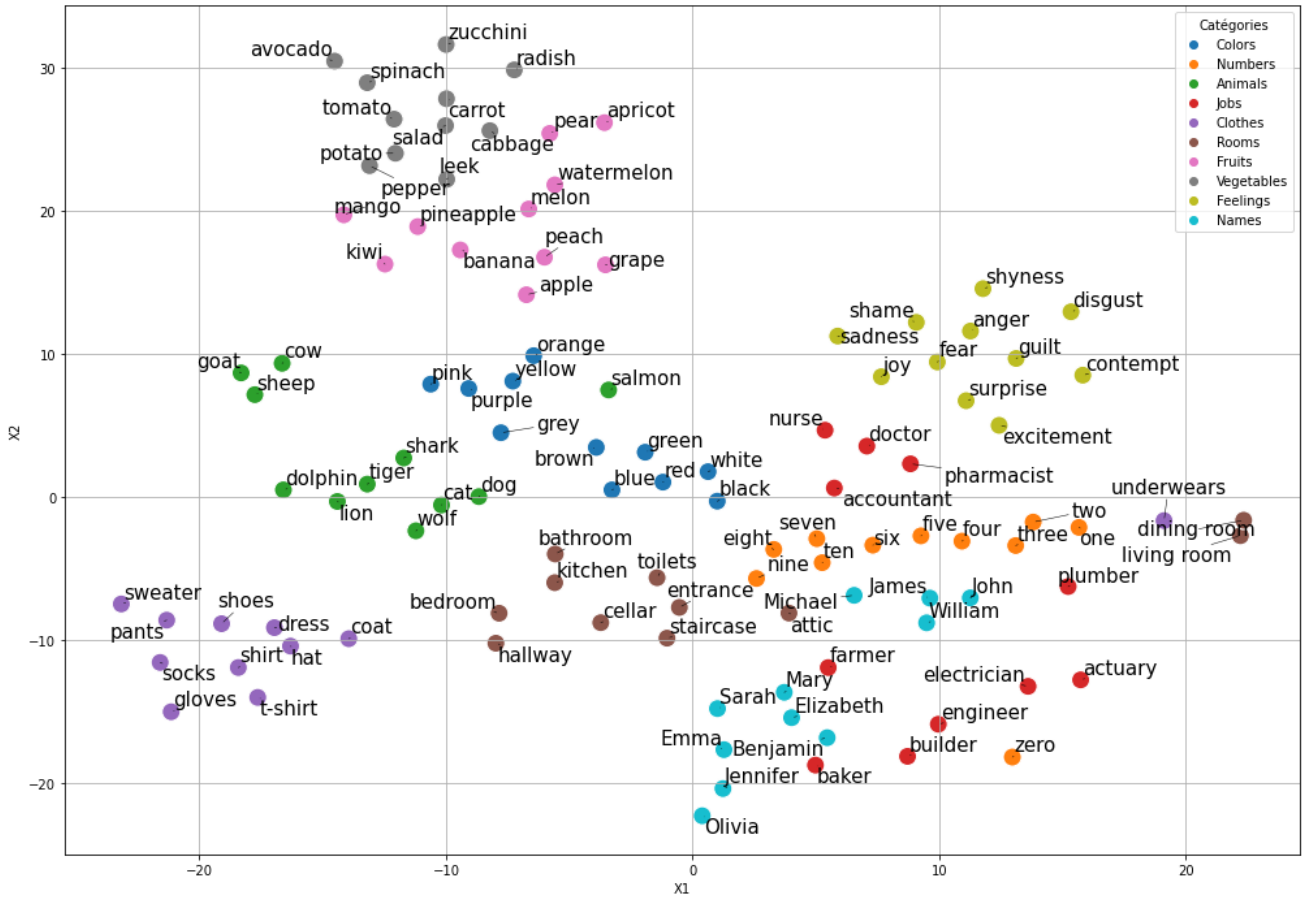

Figure 1: Visualisation du plongement lexical de nos mots

Notre plongement lexical semble visuellement très satisfaisant. Les mots de même catégorie sont presque toujours proches les uns des autres et les structures dans nos données correspondent bien à l'intuition.

La catégorie des sentiments est parfaitement identifiable et isolée des autres mots. Les fruits et légumes forment une grosse structure, bien séparé elle-même entre fruits et légumes. Les couleurs forment une structure bien identifiable, qui semble séparée en deux parties entre les couleurs "principales" (blanc, noir, rouge, vert et bleu) et les autres. Le mot saumon est inclus dans cette structure plutôt qu'avec les animaux : c'est un mot ambigu qui est également une couleur. Au sein du cluster des animaux, les regroupements suivent également l'intuition. Les pièces de la maison sont bien regroupés à l'exception de la salle à manger et du salon qui sont isolés du reste. Les vêtements sont également parfaitement identifiables, seul le mot sous-vêtements est isolé. Ce mot étant plus un classe de vêtement qu'un vêtement à proprement parler donc cet isolation peut s'interpréter.

Les autres catégories de mots sont légèrement moins bien regroupées entre elles mais tout de même très intéressantes. Les chiffres forment une ligne dans le bon ordre (à l'exception du zéro qui est isolé). Il semble donc que les chiffres ne sont pas tous autant liés les uns aux autres : plus ils sont proches en valeur numérique, plus ils sont liés dans notre représentation vectorielle. Les prénoms sont séparés entre prénoms de filles et de garçons (à l'exception de Benjamin qui est avec les prénoms de filles). De plus, les métiers "baker" et "farmer" sont inclus dans le cluster des prénoms et non des métiers. En effet, ils sont ambigus car étant également des prénoms. Pour finir, les métiers sont séparés en deux structures différentes.

Afin d'obtenir une quantification de la qualité de notre représentation vectorielle, nous allons appliquer l'algorithme des k-moyennes sur notre plongement lexical et comparer le partitionnement des données obtenu sans supervision aux vraies catégories. Sur le graphique ci-dessous, nous pouvons voir les catégories identifiées par les k-moyennes (avec k = 10). A noter que l'algorithme n'est pas lancé sur la représentation 2-D des mots mais bien sur les vecteurs complets.

Pour quantifier la qualité de ce partitionnement, nous employons l'indice de Rand et l'indice de Rand ajusté :

$$IR = 0.881\,; IRA = 0.431$$

L'IR correspond à la proportion des paires de points correctement identifiés comme étant dans le même cluster ou non, et l'IRA corrige l'IR du hasard, de façon à ce qu'une classification aléatoire obtienne 0 et une classification parfaite obtienne 1. Ces résultats semblent très satisfaisants.

Discussion

La méthode de plongement lexical présentée dans cette courte étude possède des avantages clairs : très intuitive, facile à implémenter, très économique en calculs en comparaison à des méthodes de plongement lexical basés sur des réseaux de neurones. Toutefois, la qualité de celle-ci reste à démontrer. Bien que les résultats préliminaires présentés dans cette étude semblent prometteurs, il conviendra de réaliser une comparaison à plus grande échelle, avec d'autres méthodes de plongement lexical.

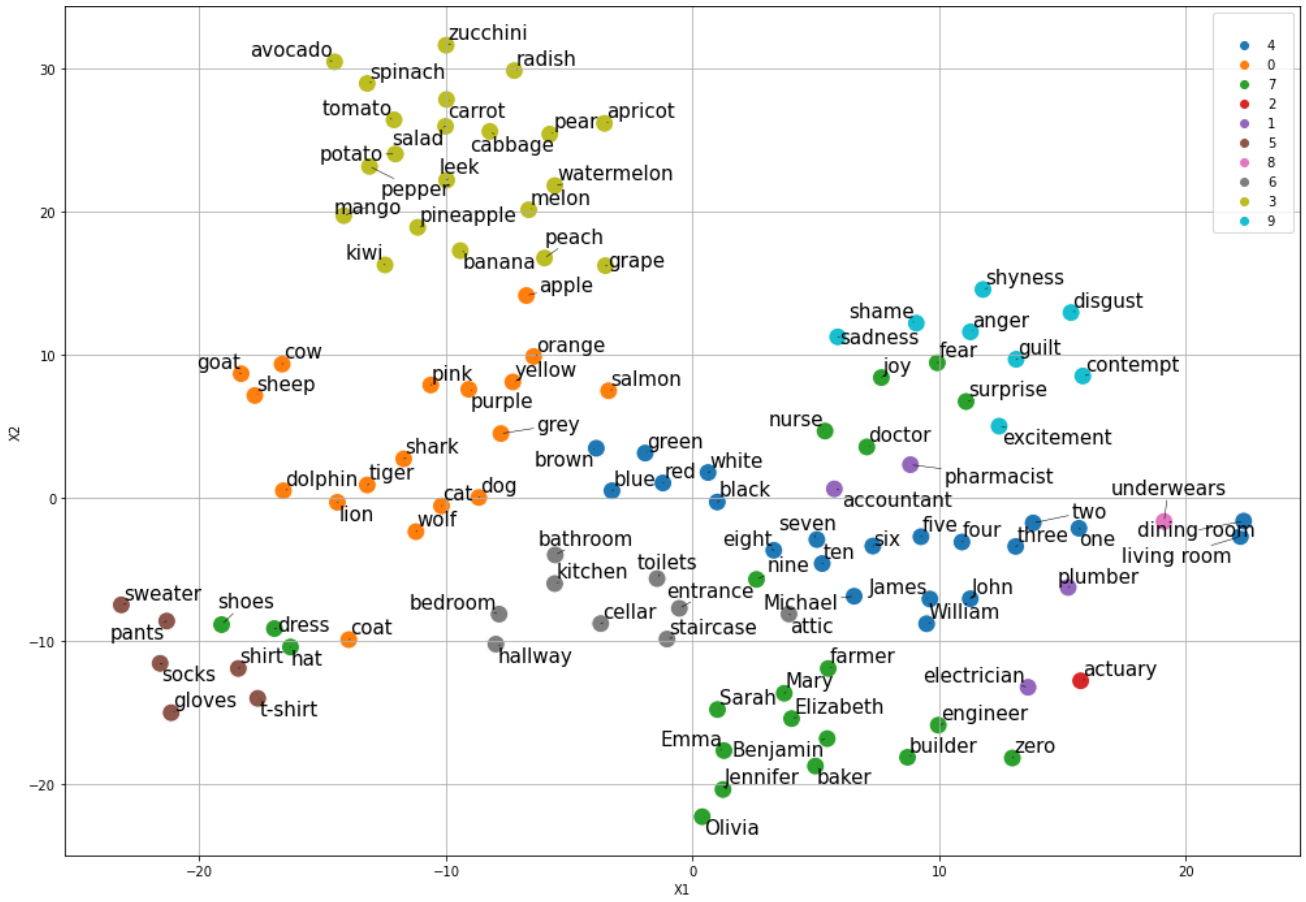

Figure 2: Visualisation des clusters obtenus par l'algorithme des k-moyennes

References

[1] Rudi Cilibrasi and Paul Vitányi. Automatic meaning discovery using google. 01 2006.

[2] Ming Li, Xin Chen, Xin Li, Bin Ma, and P. M. B. Vitanyi. The similarity metric. *IEEE Transactions on Information Theory*, 50(12):3250–3264, 2004.