**IA – Recherche de groupe de synonyme à l’aide de l’exploration de donnée**

**Vincent Vinh** Cegep du Vieux Montréal

**Hoanui Sartore – Devasse** Cegep du Vieux Montréal

**Xi Duo Zhang** Cegep du Vieux Montréal

**Abstract**

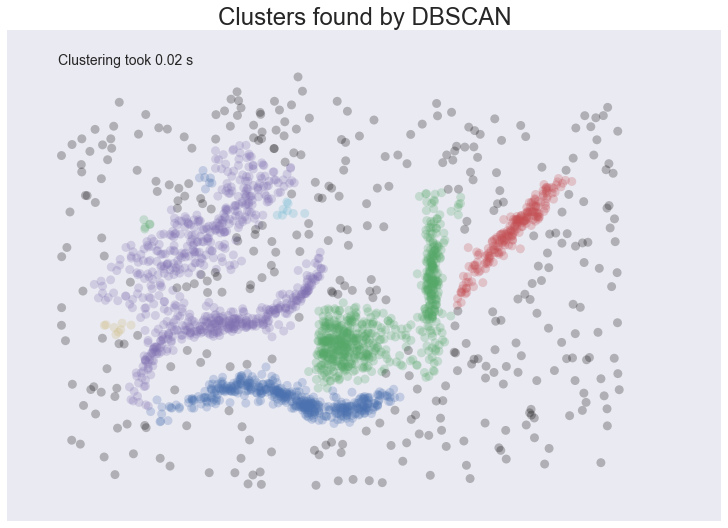
Suite à la réalisation d’un TP sur la recherche de synonyme dans des textes fournis, de façon autonome à l’aide de différent algorithme (produit scalaire, least square et city block), ce projet porte son attention sur la recherche de groupes de synonymes de façon intelligente (autonome) grâce à la mise en place de regroupement en cluster et de recherche de centroïdes dans celle-ci jusqu’à obtention de groupe de mots.

**Introduction**

La recherche automatisée à l’aide d’intelligence artificiel est un sujet d’actualité quand on voit toutes les « smart » technologies et objets produits ces dernières années. Pouvoir délégué des tâches à un ordinateur peut s’apparenter à en faire un membre d’équipe ou un employé à part entière, qui ne connait fatigue pas et réalise exactement ce qu’on lui demande de faire. Sa seule limite est l’étoffe de la machine sur laquelle il est exécuté. Cependant, les étapes à mettre en place avant l’aboutissement du projet n’est pas simple car cela implique certain algorithme tel que le « least square » et certaines notions de calcule à mettre en place. Aussi, l’analyse des résultats engendré demande une réflexion sur le sens qu’il faut leurs donner.

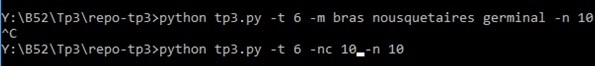
Pour cette raison, et en se basant sur nos travaux précédents, essentiellement basé sur la recherche de synonymes dans des textes donnés et sans alimenter les programme d’un dictionnaire au préalable, nous nous sommes intéressé au regroupement de synonymes en fonction de leur sens (ex : bras, pied, main dans un même cluster). Comment pouvions-nous procédé au regroupement de nos synonymes à l’aide de l’IA, qu’elle serait la notion de délais en fonction de la quantité de textes fournis et est-ce que les résultats seraient précis.

*Figure 1. Exemple de regroupement de point (mots pour notre projet) dans des clusters colorés.*



**Fonctionnement**

Pour le fonctionnement global de la méthode de recherche, nous avons eu besoin d’une matrice (matrice de cooccurrence) composé à l’aide de nos textes. Aussi, il nous a fallu mettre en place une base de donnée (Oracle) et lui insérer notre matrice ainsi que notre liste de vocabulaire. Notre programme récupère les éléments de la base de donné. Nous avons deux fonctions d’initialisation du cluster. La première laisse le choix à l’usager de *n* centroide à utiliser, puis le programme place les mots aléatoirement dans *n* clusters. Pour la seconde méthode, l’usager fournis les mots qu’il veut utiliser comme centroide. En fonction des mots saisie, le programme récupère le vecteur de chaque mot présent dans la matrice de cooccurrence.



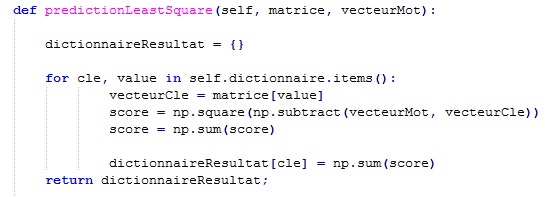
La création des clusters reste identique dépendamment de la méthode utilisée. Nous créons à présent deux listes :

1. *NouvelleListe* - Une liste qui contient le cluster correspondant à chaque mot.
2. *AncienneListe* - Une liste qui garde en mémoire les données de la précédente itération.

À l’aide de l’*AncienneListe* nous calculons les nouveaux vecteurs de centroides, et les centroides eux même appelé aussi barycentre.

Nous faisons ensuite un *leastsquare* (calcule de distance) entre chaque vecteur de mot dans la matrice et le vecteur du centroide.

Les résultats sont stockés dans une liste où chaque résultat de cluster est un dictionnaire contenant le mot et le score du *leastsquare.*



Pour la dernière étape, nous prenons le plus petit score de chaque mot et on définit le cluster auquel il appartient. Ensuite, nous comparons la *NouvelleListe* avec *AncienneListe* pour voir s’il y a eu des changements de clusters, si oui, le programme réitère la méthode pour affiner les résultats, le cas échéant le programme s’arrête. Pour chaque résultat de chaque cluster nous trions les scores du plus petit au plus grand. Nous pouvons consulter le résultat affichant les synonymes les plus proche, sauvegarder dans un fichier texte.

**Conclusion**

Les résultats fournis après divers tests de notre programme affichent effectivement des résultats prometteurs. Le fait que le programme puisse analyser en toute autonomie les changements de cluster et autres, sont de prime abord un des enjeux important de ce projet réaliser qui ouvre la voie vers ce qu’on peut appeler l’IA. Cependant, cette méthode nécessite une machine assez performante étant donné la complexité des calculs et les délais d’attente avant l’obtention de résultat.