Article Scientifique

Présenté par :

Nazer Lee-Sténio

Sarah Mei Léonard

Iprahim Tutuncu

## Résumé:

*Dans le cadre du cours B52 : Développement Base de données, nous avons été chargé de trouver des synonymes à un mot donné à l’aide de plusieurs algorithmes différents. Parmi les résultats obtenus l’algorithme du Least Square semblait être le meilleur, mais les synonymes obtenus laissaient à désirer.*

## Intro :

Nous nous sommes demandé si les résultats de la recherche de synonymes pouvaient être plus proche de la réalité s’il y avait un moyen de regrouper les mots avant la recherche. De là est venu l’idée de Clustering. Si on peut partitionner tous les mots de notre base de donnée, les synonymes de notre mot recherché serait beaucoup plus précis dans une des partitions.

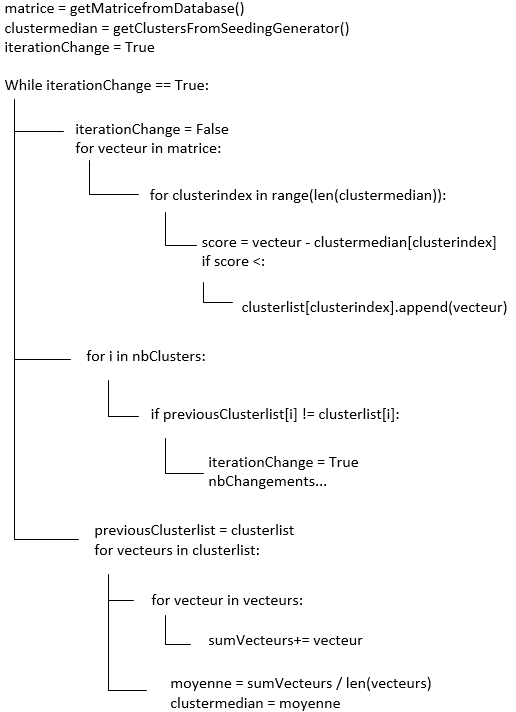
Les logiciels de synonymes utilisent surement une manière de catégoriser leurs mots. Par exemple, par type de mot ou par racine grammaticale. Nous espérons savoir si la catégorisation non-explicite par partition incrémentale, générera des groupes de mots avec un thème, une forme ou un concept similaire. Et de ce fait, améliorera la précision de synonymes que l’on peut trouver.

## Matériel et méthodes :

Les données que nous utilisons seront générés à partir du projet antérieur, d’où notre idée est venue. A partir de textes fournis par notre enseignant, nous recueillons tous les mots uniques qu’ils contiennent ainsi que toutes les instances que deux mots se retrouvent à l’intérieur d’une fenêtre donnée. Ces résultats sont enregistrés dans une base de donnée SQL et par la suite recueillie pour déterminer les synonymes. Ces mêmes données seront utilisées pour ce projet.

Pour notre expérience de partition, nous utilisons l’algorithme du K-means. Pour un nombre de partitions voulus, une matrice sera générée. La valeur initiale de chaque élément est un nombre plus grand ou égale à 0 et elles seront générés de deux façons, aléatoirement et données à partir d’un mot existant. Chaque mot de notre dictionnaire est associé à un vecteur dont la taille est le nombre de mots de notre dictionnaire et les vecteurs initiaux de nos partitions seront de la même taille.

Pour chaque vecteur de la matrice du dictionnaire on détermine à quelle partition celui-ci correspond. Pour ce faire, on effectue une \*\*\*Soustraction vectorielle?\*\*\* entre ce vecteur et chaque vecteur associés aux partitions, et le vecteur résultant le \*\*\*plus près de 0 \*\*\* est celui auquel le mot appartient. Lorsque tous les vecteurs ont étés calculés, nous effectuons la moyenne des vecteurs associés à chaque partition. Le résultat sera un nouveau vecteur qui deviendra le vecteur représentant la partition. Nous effectuons ces étapes jusqu’à ce les mots de chaque partition se stabilisent.



## Résultats :

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Date | Nb Clusters (-nc) | Taille fenêtre (-t) | nb Mots Dictionnaire | nb Concurrences | type d’algorithme d’initiation | Nombre d’itérations | Temps total | Étudiant responsable |
| 24 Novembre 2017 | 50 | 7 | 24639 | 549522 | Seeding Aléatoire | 48 | 0 hre  30 min 33 sec | Lee |
| 29 Novembre | 50 | 5 | 24639 | 379327 | positionnementOrthogonalSeed | 83 | 0 hre  57 min  48 sec | Lee |
| 29 Novembre | 25 | 7 | 24639 | 549522 | positionnementOrthogonalSeed | 48 | 0 hre  27 min 21 sec | Lee |
| 6 Décembre | 75 | 7 | 24639 | 549522 | positionnementOrthogonalSeed | 61 | 1 hre 31 min 40sec | Lee |
| 6 Décembre | 100 | 5 | 24639 | 379328 | RANDOM CLUSTERS | 90 | 2 hre  53 min  57 min | Sarah |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

## Conclusion

Le processus de partitionnement prends un temps linéairement proportionnel au nombre de partitions, avec un ratio près de 1:1. Plus le nombre de partitions est élevé, plus les résultats concordent à un certain thème ou concept. On peut remarquer des clusters de verbes conjugués temps similaires, des articles ou conjonctions, de déterminants et même de mots anglais. Donc, d’après nos résultats, on peut conclure que notre hypothèse est confirmée. Une autre application que l’on peut utiliser ce partitionnement est dans la recherche de synonymes que nous avons entamé. On peut rechercher un synonyme dans la partition la plus près du mot que nous recherchons, ainsi, les synonymes trouvés devraient être plus fiables.