Chapitre 1

Implémentation

1.1 Objectifs

Notre but étant de pouvoir proposer aux étudiants de première année les éventuelles voix qu'ils sont susceptibles de poursuivre tout au long de leurs études. Notre solution doit pouvoir répondre aux questions suivantes :

— Donner une filière de départ, quel est le parcours suivi ?

— Pouvoir fournir une représentation de ce parcours

Dans cette optique, le premier objectif de cette étude est dans un premier temps de pouvoir regrouper les différents CV en fonction des différentes filières. Cette première phase de classification nous permettra par la suite de conditionner l'algorithme utilisé. Une fois ce premier traitement effectué, il sera ensuite nécessaire de récupérer toutes les filières parcourues par l'individu. Enfin, une fois cette étape de catégorisation du CV et la récupération du chemin parcouru réalisée. L'objectif final est de pouvoir représenter le résultat sous forme graphique interprétable par les étudiants.

1.2 Préparation et caractérisation des données

1.2.1 Le jeu de données

Pour cette étude, le jeu de données utilisé contiendra environ 2000 CV d'un projet pouvant être trouvé à l'adresse suivante :

— `https://github.com/JAIJANYANI/Automated-Resume-Screening-System`

Celui-ci contient différents CV au format PDF,DOC et docx. La majorité de ces documents sont construits de la façon suivante :

— Expérience professionnelle

— Éducation

— Loisirs et autre

À noter, bien que cette structure soit la plus récurrente, l'ordre des différentes sections peut tout de même changer en fonction des documents.

Un premier tri a été effectué afin de filtrer les documents en fonction du format utiliser afin de ne retenir que ceux au format **.doc** ou **.docx** pour faciliter les traitements visant ensuite à les classifier. En outre, ce premier tri permettra par la suite de minimiser la charge de calcul afin que les étapes de traitement et d'analyse ne

soient pas impactées par un temps de traitement trop important. À noter que bien que cette première étape de tri puisse être réalisée de façon manuelle. L'objectif final étant de proposer cette solution sous la forme d'un programme, il est plus intéressant que cette phase de tri puisse s'effectuer de façon automatique une fois le jeu de données fourni.

1.2.2 Choix de l'algorithme

Afin de pouvoir choisir l'algorithme le plus adapté à notre cas, celui-ci doit pouvoir répondre aux critères suivant :

— Capacité à traiter des documents
— Capacité à gérer une montée en charge
— Consommation de mémoire

Comme vu précédemment, l'algorithme K-means, bien que celui-ci puisse traiter des documents texte, il impose toutefois de connaitre le nombre de clusters souhaités afin de pouvoir effectuer son traitement.
Dans notre cas, cette contrainte ne représente pas de difficultés les CV étant regroupés en fonction des secteurs d'activités (figure à voir). En outre, l'algorithme choisi étant utilisé à travers la librairie python « scikit-learn », celle-ci propose un tableau comparatif (figure 2.1) des différents algorithmes qu'il est possible d'implémenter en plus de DBSCAN et Agglomerative clustering. Au vu des différents cas

Method name	Parameters	Scalability	Usecase	Geometry (metric used)
K-Means	number of clusters	Very large `n_samples`, medium `n_clusters` with MiniBatch code	General-purpose, even cluster size, flat geometry, not too many clusters	Distances between points
Affinity propagation	damping, sample preference	Not scalable with n_samples	Many clusters, uneven cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Mean-shift	bandwidth	Not scalable with `n_samples`	Many clusters, uneven cluster size, non-flat geometry	Distances between points
Spectral clustering	number of clusters	Medium `n_samples`, small `n_clusters`	Few clusters, even cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Ward hierarchical clustering	number of clusters or distance threshold	Large `n samples` and `n_clusters`	Many clusters, possibly connectivity constraints	Distances between points
Agglomerative clustering	number of clusters or distance threshold, linkage type, distance	Large `n samples` and `n_clusters`	Many clusters, possibly connectivity constraints, non Euclidean distances	Any pairwise distance
DBSCAN	neighborhood size	Very large `n_samples`, medium `n_clusters`	Non-flat geometry, uneven cluster sizes	Distances between nearest points
OPTICS	minimum cluster membership	Very large `n_samples`, large `n_clusters`	Non-flat geometry, uneven cluster sizes, variable cluster density	Distances between points
Gaussian mixtures	many	Not scalable	Flat geometry, good for density estimation	Mahalanobis distances to centers
Birch	branching factor, threshold, optional global clusterer.	Large `n_clusters` and `n_samples`	Large dataset, outlier removal, data reduction.	Euclidean distance between points

FIGURE 1.1 – Comparatif des algorithmes de clustering

d'utilisation proposés par ce tableau, les trois algorithmes étudiés dans ce document peuvent être implémentés. K-means étant le plus courant et possédant une facilité d'implémentation sera l'algorithme utilisé étant donné qu'il est également possible de l'utiliser dans le traitement de documents. La représentation des données choisie n'est pas adaptée à l'utilisation de DBSCAN. De plus, nous sommes dans un cas d'apprentissage non supervisé, cet algorithme nécessite une configuration établie en amont par un utilisateur. En revanche,K-means bien qu'ayant besoin du nombre de clusters souhaités peut fonctionner sans que des labels lui soit fourni. La méthode utilisée sera donc un clustering par partitionnement en ayant recours à l'algorithme K-means.

1.3 Caractérisation et classification des parcours

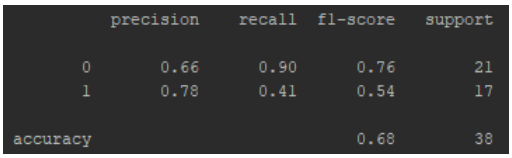
1.3.1 Démarche

Afin de pouvoir utiliser l'algorithme K-Means, il est nécessaire dans un premier temps de transformer notre base de données en un format interprétable par nos algorithmes. Afin de pouvoir fonctionner, K-means utilise des données numériques sous la forme illustrée la figure 2.2. Par conséquent, grâce à la structure des dossiers ou sont stockés, il est possible de créer une énumération afin de pouvoir attribuer une valeur aux différentes catégories. Cette structure nous permet de déterminer le K maximum qu'il est possible d'adopter pour les différentes opérations de l'algorithme. Une fois cette énumération créée, celle-ci nous permet donc de catégoriser les différents CV en fonction des différents secteurs intégrés par un individu. Cette première étape apporte un premier élément de réponse quant à la modélisation des trajectoires. En effet, les différentes clés étant stockées dans l'ordre ou celles-ci sont rencontrées ceci nous permet dans un premier temps d'obtenir le chemin suivi.

	File Name	Secteur Key_	Secteur Key_	Secteur Key_	Secteur Key_	Secteur Key_	Secteur Key_	Secteur Key_	Secteur Key_	Secteur Key_8
2	ASK_Carlyn M	2	4	12	14	0	0	0	0	0
3	Cyrus Global	2	3	4	8	10	11	12	14	15
4	Partners_Bet	2	4	8	12	0	0	0	0	0
5	WorldQuant_	2	11	12	14	0	0	0	0	0

FIGURE 1.2 – Extrait du jeu de données

Comme illustrée dans la figure ci-dessus, chaque mot clé correspondant à un secteur d'activité est renseigné dans l'ordre ou ceux-ci sont rencontré. Ceci permet d'obtenir un premier aperçu des filières parcourues par un individu. Afin de pouvoir prendre en compte les éventuelles répétitions des mots clés recherchés durant les traitements, lorsque l'un de ceux-ci est rencontré, celui-ci est stocké dans un dictionnaire avec la clé associée. Ceci permet d'éviter la répétition de clé dans le fichier csv produit à la fin du traitement. À noter que la valeur 0 sert à représenter un vide afin que le jeu de données puisse être interprété par K-means.
Une fois cette première étape exécutée, il est alors possible d'utiliser K-means pour la création de nos clusters. Dans notre cas, nous avons tenté de lancer l'algorithme sans lui fournir de labels au préalable afin de pouvoir mesurer son taux de précision.(figure 2.3)



	precision	recall	f1-score	support
0	0.66	0.90	0.76	21
1	0.78	0.41	0.54	17
accuracy			0.68	38

FIGURE 1.3 – Résultat de création de clusters par K-means

Dans la figure ci-dessus :

- Précision fait référence au score obtenu par la précision de classification utilisée.
- Recall correspond à la proportion de documents pertinents récupérés.
- F1-score mesure la précision du test.

Dans cet exemple, nous avons créé un cluster contenant tous les CV dont le propriétaire a commencé par une filière administrative. On peut observer un taux de 68% de réussite dans la détermination de l'appartenance à la filière recherchée sans aucun apport de la part de l'utilisateur. Cependant, ce premier essai a été effectué sur une fraction du jeu de données et globale. Cette première tentative permet dans un premier temps d'évaluer le comportement et la précision de K-means pour nos documents. À noter que la mesure de distance utilisée est la distance euclidienne utilisée par défaut par l'algorithme K-mean.

Environnement

Ci-dessous les caractéristiques de la machine utilisée durant cette étude ainsi que les différents logiciels et librairies.

- Mémoire Ram : 16 Go
- Espace disque : 500 Go
- Logiciel utilisé : Pycharm
- Langage : Python
- Librairies utilisées : sklearn,numpy,pandas et seaborn

1.3.2 Problématiques restantes

Nous avons pu voir qu'il existe une multitude de méthodes applicables afin d'ordonner nos données, cependant la première problématique rencontrée pour cette implémentation a été le parsing des CV. Il est en effet possible de récupérer aisément les mots clés recherchés tels que « Sales » ou encore « Law » comme filière universitaire ou secteur d'activité professionnelle. La difficulté vient toutefois du fait qu'il est nécessaire de différencier dans quelle rubrique du CV, dans notre cas « Education » ou « Work experience » ces mots apparaissent. Ce type de découpage de document est en général utilisé par les services de recrutement d'entreprise lorsqu'un recruteur cherche un profil en fonction de mots clés ce qui fait que ces solutions sont propriétaires. Un projet open source disponible à l'adresse suivante :

— `https://github.com/tramyardg/CVparser`

Ce projet aurait pu représenter une solution possible. Celui-ci renvoie une chaine de caractère au format JSON contenant les différentes rubriques découpées. Malencontreusement, ce projet n'est plus fonctionnel ni maintenu par son créateur. La seconde problématique restante vient du fait que pour notre cas d'étude, une fois les différents parcours universitaires ainsi que le secteur occupé ont été classifiés, afin de pouvoir modéliser ces trajectoires sous forme graphique. Il serait nécessaire de pouvoir récupérer le contenu d'un objet lui-même contenu dans un cluster donné et pouvoir considérer chaque ligne comme une unique occurrence afin de pouvoir modéliser celle-ci sous forme de point. Cette seconde problématique est par conséquent fortement liée à la première. En effet, le fait qu'il soit nécessaire dans un premier temps de pouvoir traiter les différentes rubriques d'un document une à une et d'être en mesure de ne récupérer que des données pertinentes.

1.4 Conclusion

Dans ce mémoire, nous avons présenté les différentes méthodes de clustering, dont le clustering par partitionnement qui semble correspondre à notre cas. Toutefois, le clustering hiérarchique reste prometteur. Cependant le traitement des objets regroupés et le découpage de document ont entrainé la rencontre de nouvelle problématique durant cette étude.
L'analyse de données étant un élément essentiel de domaine qui de nos jours possède une forte popularité tel que le Big data. Les méthodes de clustering utilisées dans ce domaine représentent une opportunité prometteuse pour notre problématique. En effet, celles-ci nous permettent dans un premier temps d'ordonner nos données afin de pouvoir en dégager les populations principales. Cette première étape de classification et représentation des données permet ensuite de faciliter les différents traitements.
Dans la dernière partie, nous avons présenté les premières étapes d'une méthode visant à tirer profit de la classification obtenue à travers des processus de clustering. Plus particulièrement, nous avons eu recours à un clustering par partitionnement afin de pouvoir identifier les différentes populations. Cependant, cette méthode pose de nouvelles problématiques dans le cas de parcours contenant plusieurs filières et n'est par conséquent pas la plus idéale. De plus, il reste encore une partie conséquente de l'implémentation regardant l'extraction de texte à réaliser.