Université Paris Cité

Rapport du projet TER

Large Language Models Enable Few-Shot Clustering

Préparé par

MAKHLOUF Mouloud

Salah Eddine Mohamed OUIKENE

MONCEF Naik

BAKIR Yagoub

Superviseur

Lazhar LABIOD

Code source du projet accessible sur
https://github.com/Salaheddine-ouikene/LLM-Clustering

Tableau des matières

Tableau des figures

Liste des abréviations

LLM : Large Language Model
CMVC : Clustering Multi-Vue Contextualisation
PCK-Means : Pairwise Constrained K-means
NMI : Normalized Mutual Information

Abstract

Le clustering est une technique d'apprentissage non supervisé qui consiste à regrouper des données similaires en ensembles homogènes appelés clusters. L'objectif principal du clustering est de diviser un ensemble de données en groupes ou clusters de sorte que les éléments au sein d'un même cluster partagent des caractéristiques similaires, or définir ces caractéristiques est une tâche très ardue et souvent impossible à réaliser efficacement comme l'indique "Caruana (2013)" selon ces dires ce manque de spécification rend le clustering "**probably approximately useless**." .

Pour palier ce problème cette étude se tourne vers le clustering semi-supervisé qui permet de diriger l'algorithme de clustering grâce à la labellisation de certaines données via "des avis d'experts" qui sont généralement générés par des intéractions homme machines très coûteux, une autre méthode abordée est l'utilisation d'un LLM dans l'extraction de mots clefs dans le but d'améliorer le clustering, l'objectif de ce travail de recherche est d'évaluer la capacité des LLMs (Large Langage Model) de substituer le facteur humain pour la génération de ces avis pour un coût beaucoup moindre.

1. Introduction

Dans le contexte actuel de la recherche en machine learning et en science des données, la capacité de mettre en application des concepts théoriques à des situations pratiques est essentielle. Ce rapport présente les résultats d'un projet tuteuré où notre groupe d'étudiants a exploré les implications d'un article scientifique intitulé "Large Language Models Enable Few-Shot Clustering" de Vijay Viswanathan, Kiril Gashteovski, Carolin Lawrence, Tongshuang Wu, et Graham Neubig. Cet article propose une nouvelle approche pour le clustering de texte en introduisant des LLMs (grands modèles de langage).
Cette utilisation d'un LLM est faite en 3 endroits :

- **Avant clustering :** Un expert sait quelles sont les aspects **importants** à capturer durant le clustering, au lieu de forcer l'algorithme de clustering à chercher par soi-même ces aspects, nous allons utiliser un LLM pour les extraires, l'extraction se base sur la fourniture de plusieurs exemple au LLM avant de le laisser faire l'extraction de façon complètement autonome.

- **Durant clustering :** Cette méthode s'appuie sur l'utilisation d'algorithmes de clustering sous contraintes en fournissant des contraintes "**must link**" et "**must not link**" générées à l'aide d'un LLM à qui on a donné plusieurs exemples en guise d'entraînement.

- **Après clustering :** Dans cette partie nous allons nous intéresser à la correction de l'assignation des points au clusters en identifiant les "**low confidence points"** en cherchant les **k** points avec la marge de distance la plus faible entre le cluster le plus proche et le deuxième cluster le plus proche et en demandant au LLM de vérifier la correction de l'assignation des points au clusters.

2. État de l'art

Dans le but de l'appropriation de l'état de l'art du projet nous allons nous intéresser à la méthode **d'expansion de la représentation de document** qui a été utilisée, nous aborderons ensuite les **algorithmes de clustering par contraintes** puis la **méthode de correction des clusters**.

1.1 Définition du clustering non-supervisé

Le clustering non supervisé, aussi appelé apprentissage non supervisé par partitionnement, est une technique d'apprentissage automatique qui vise à **regrouper des données non étiquetées** en fonction de leurs similarités.

Contrairement à l'apprentissage supervisé, où les données sont déjà classées et étiquetées, le clustering non supervisé ne dispose **d'aucune information préalable sur la structure des données**. L'algorithme doit donc **détecter par lui-même les patterns et les groupes** présents dans les données.

Pour ce faire, il utilise différentes techniques, comme la distance entre les points de données, la densité des points ou la cohésion des groupes. L'objectif est de **créer des groupes homogènes**.

Algorithmes de clustering non supervisé

Il existe de nombreux algorithmes de clustering non supervisé, chacun avec ses propres avantages et inconvénients. Parmi les plus populaires, on peut citer :

- **K-means (étudié dans ce projet)**: Un algorithme simple et efficace qui partitionne les données en un nombre prédéfini de groupes (K).
- **Clustering hiérarchique**: Cet algorithme crée une arborescence de clusters, en regroupant itérativement les données les plus proches les unes des autres.
- **Clustering par mélange de modèles**: Ce type d'algorithme suppose que les données proviennent de plusieurs distributions probabilistes et cherche à identifier ces distributions.

Limites du clustering non supervisé

- Le choix du nombre de groupes (K) peut être difficile.
- Les résultats du clustering peuvent être subjectifs et dépendre de l'algorithme utilisé.

- Il peut être difficile d'interpréter les groupes obtenus.

1.2 Définition du clustering semi-supervisé

Le clustering semi-supervisé est une technique d'apprentissage automatique qui se situe **entre le clustering non supervisé et le clustering supervisé** et vise à combler les lacunes rencontrées avec le clustering non-supervisé.

Contrairement au **clustering non supervisé**, qui n'a aucune information sur les données, le clustering semi-supervisé dispose de **deux types de données**:

- **Des données étiquetées**: une petite quantité de données pour lesquelles la classe est connue.
- **Des données non étiquetées**: une grande quantité de données pour lesquelles la classe est inconnue.

L'algorithme de clustering semi-supervisé utilise les données étiquetées pour **apprendre les caractéristiques des différentes classes**, puis utilise ces connaissances pour **regrouper les données non étiquetées**.

Avantages du clustering semi-supervisé

Le clustering semi-supervisé présente plusieurs avantages :

- **Il permet d'exploiter des données non étiquetées**, qui sont souvent beaucoup plus nombreuses que les données étiquetées.
- **Il peut améliorer les performances du clustering**, car les données étiquetées permettent à l'algorithme d'apprendre des patterns plus précis.
- **Il peut être utile dans des situations où il est difficile ou coûteux d'étiqueter toutes les données**.

Algorithmes de clustering semi-supervisé

Il existe plusieurs algorithmes de clustering semi-supervisé, dont les plus populaires sont :

- **Propagation d'étiquettes**: Cet algorithme propage les étiquettes des données étiquetées aux données non étiquetées en fonction de leur similarité.
- **Co-apprentissage**: Cet algorithme utilise plusieurs algorithmes de clustering pour apprendre les données de manière itérative.
- **Clustering par apprentissage par faible supervision**: Cet algorithme suppose que les données non étiquetées ont des contraintes structurelles (par exemple, des
- points de données appartenant à la même classe sont plus proches les uns des autres) et utilise ces contraintes pour les regrouper.
- **apprentissage semi-supervisé par contraintes**:

  On approfondira notamment les algorithmes :

  - COP K-means
  - PCK-means (Pairwise Constrained K-means)

Les deux algorithmes repose sur des contraintes :

- **(a,b)** ∈ must-link ⊆ **D**x**D** ↔ l'instance de **a** et **b** doivent être dans le même cluster
- **(a,b)** ∈ cannot-link ⊆ **D**x**D** ↔ l'instance de **a** et **b ne** doivent **pas** être dans le même cluster
- La contrainte must-link est une relation symétrique, réflexive et transitive (relation reflexive)
- La contrainte cannot-link est une relation symétrique
- Si **(a,b)** ∈ must-link ∧ **(b,c)** ∈ cannot-link → **(a,c)** ∈ cannot-link

1.2.1 COP K-means :

Respecte toutes les contraintes, si une seule contrainte ne peut pas être respectée il échoue.

Algorithme simplifié :

Algorithm 1: COP-KMeans

(0) initialization of the initial centroids $C_1,..., C_k$.
(1) for each $x_j \in D$, *assign it to the cluster* y_j *such that the corresponding centroid is the nearest and such that the must-link and cannot-link constraints are respected. If no such cluster exists, the algorithm* ***fail****.*
(2) update centroids $C_1,..., C_k$ *as in the standard* $K-Means$ *algorithm.*
(3) iterate (1) and (2) until a stop criterion is not meet.
(4) return $C_1,..., C_k$.

Figure 1 : Algorithme COP-KMeans

1.2.2 PCK-means :

L'algorithme peut être vu comme une version moins stricte de l'algorithme COP-KMeans. Cette version de l'algorithme tolère que des contraintes ne soient pas respectées ,l'algorithme n'échoue pas dans ce cas là.

Algorithme simplifié :

Algorithm 2: PC-KMeans

(0) initialization of the initial centroids $C_1,\ldots, C_k$.
(1) for each $x_j \in D$:

$$y_j = \underset{i=1,\ldots,k}{\operatorname{argmin}} \left\{ ||x_j - C_i||^2 + \sum_{(x_j,x_s)\in must\_link} w_{\{j,s\}} \cdot \mathbb{1}(y(s) \neq i) + \right.$$
$$\left. + \sum_{(x_j,x_s)\in cannot\_link} \bar{w}_{\{j,s\}} \cdot \mathbb{1}(y(s) = i) \right\}$$

(2) update centroids $C_1,\ldots, C_k$ *as in the standard* $K-Means$ *algorithm.*
(3) iterate (1) and (2) until a stop criterion is not meet.
(4) return $C_1,\ldots, C_k$.

Figure 2 : Algorithme PC-KMeans

1.3 Grands Modèles de Langage (LLMs)

Les LLMs, tels que GPT-3 (gpt-3.5-turbo-0301 utilisé dans cette étude), ont révolutionné le domaine de la science des données et du traitement du langage naturel (NLP). Ils sont capables de comprendre et de générer du texte de manière sophistiquée, ce qui les rend utiles pour de nombreux contextes, y compris le clustering de texte.
L'utilisation de LLMs dans cette étude vise à améliorer les résultats obtenus via des algorithmes et ce via différentes méthodes :

- **Amélioration des représentations textuelles :** Les LLMs peuvent générer des représentations plus riches des données textuelles, ce qui facilite le clustering.
- **Pseudo-oracles pour les contraintes :** Les LLMs peuvent être utilisés pour simuler des experts humains et fournir des contraintes pour alimenter des algorithmes de clustering sous contraintes.
- **Correction après clustering :** Les LLMs peuvent aider à corriger des clusters mal assignés ou à ajuster des regroupements après le processus initial de clustering.

1.4 Approche proposée par l'article étudié

L'article "Large Language Models Enable Few-Shot Clustering" propose une approche innovante pour le clustering semi-supervisé en utilisant des LLMs. L'article examine trois étapes où les LLMs peuvent être intégrés au processus de clustering via 3 méthodes :

- Expansion de la représentation du document
- Pseudo-Oracle Pairwise Constraint Clustering
- Utiliser un LLM pour corriger un Clustering

1.4.1 **Expansion de la représentation du document**

Avant de produire des clusters, les experts savent généralement quels aspects de chaque document ils souhaitent capturer lors du regroupement.
Plutôt que de forcer les algorithmes de clustering à extraire ces facteurs clés à partir de zéro, il pourrait être plus efficace de mettre en évidence ces aspects de manière globale (et ainsi de spécifier les points d'intérêt de la tâche) au préalable. Pour ce faire, nous utilisons un LLM pour rendre la représentation textuelle de chaque document dépendante de la tâche, en l'enrichissant et en l'étendant avec des éléments pertinents au besoin de clustering.
Plus précisément, chaque document est passé par un LLM qui génère des phrases clés, ces phrases clés sont encodées par un modèle d'embedding, et l'embedding des phrases clés est ensuite concaténé à l'embedding du document original.

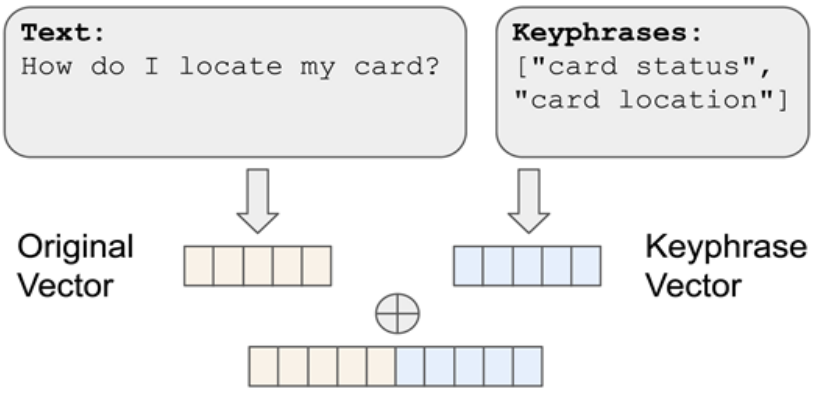

Figure 3 : Représentation de la méthode d'augmentation de la représentation d'un document en utilisant les keyphrases

Nous générons des phrases clés en utilisant un LLM, Nous fournissons une courte démonstration au LLM pour l'aider à la compréhension de la tâche qu'il lui est demandée:

1.4.2 Pseudo-Oracle Pairwise Constraint Clustering

Ce texte propose d'utiliser des grands modèles de langage (LLM) pour guider le processus de clustering semi-supervisé en se basant sur les connaissances d'experts.
L'approche consiste à fournir à un LLM des instructions spécifiques au domaine et des exemples de contraintes par paires pour l'aider à identifier des points similaires à regrouper. Les contraintes identifiées par le LLM sont ensuite utilisées avec comme **pseudo-Oracle** pour l'algorithme de clustering PCK-means pour créer des groupes cohérents avec les directives de l'expert.

1.4.3 Utiliser un LLM pour corriger un Clustering

Cette méthode explore comment améliorer la qualité d'un ensemble de clusters existant en apportant des modifications après la fin du clustering.
Pour ce faire, on utilise le même pseudo-oracle à base de contraintes générées par un LLM (même traitement que dans la

On identifie d'abord les points dit "low confidence points", les points ayant la plus faible marge entre le cluster le plus proche et le deuxième plus proche.
Pour représenter textuellement chaque cluster, on utilise les entités les plus proches du centroïde de ce cluster dans l'espace d'embedding.
Pour chaque point peu fiable, on demande d'abord au LLM si ce point est correctement associé à l'un des points représentatifs de son cluster actuel. Si le LLM prédit que le point ne devrait pas être associé au cluster actuel, on considère les 4 clusters suivants les plus proches dans l'espace d'embedding comme des candidats pour un reclassement, triés par proximité. Pour reclasser le point, on demande au LLM s'il doit être associé aux points représentatifs de chaque cluster candidat. Si le LLM répond positivement, on réajuste le point à ce nouveau cluster. Si le LLM répond négativement pour tous les choix alternatifs, on maintient l'affectation au cluster existant.

1.5 CMVC

L’architecture CMVC (Clustering Multi-Vue Contextualisé), est un cadre novateur conçu pour améliorer l'organisation et l'analyse des données sous différents angles. Il exploite deux points de vue distincts sur la connaissance : le point de vue factuel, issu des triplets extraits, et le point de vue contextuel, issu du contexte source de ces triplets, pour améliorer la canonisation des bases de connaissances ouvertes (OKB). En considérant conjointement ces points de vue et en utilisant des algorithmes de regroupement innovants, CMVC identifie et regroupe efficacement les expressions nominales et les relations synonymes au sein des OKB, améliorant ainsi leur cohérence et leur utilité sans nécessiter d'annotations manuelles.

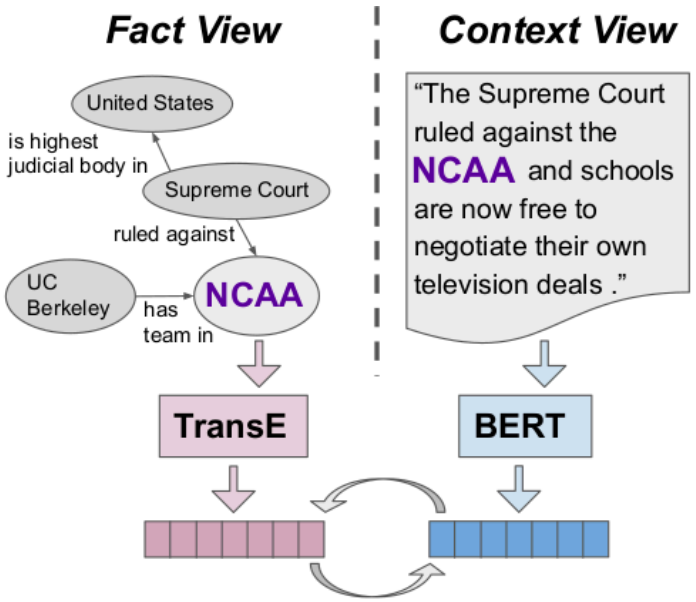

Figure 4 : Représentation de l’architecture CMVC

L'approche décrite dans l'article montre que l'utilisation de LLMs peut améliorer l'efficacité du clustering semi-supervisé tout en réduisant le besoin d'un retour d'information intensif de la part des experts.

3. Reproduction des résultats:

3.1 Datasets utilisés:

Dans cette étude 5 datasets ont été traités en utilisant 2 approches pour le chargement et le prétraitement, la première concerne est la **canonicalisation d'entités en utilisant l'architecture CMVC** (2 datasets), la seconde se basera la vectorization de texts courts en utilisant l'encodeur INSTRUCTOR pour générer les embeddings (3 datasets).

Par soucis de **coût** nous allons tester un dataset de chaque type, nous avons préféré garder des ressources pour explorer d'autres approches (utilisation d'autres versions de LLMs)

3.1.1 OPIEC59K:

Présentation du dataset : OPIEC59K est un ensemble de données comprenant 22000 expressions nominales regroupées en 490 clusters basés sur 2138 formes de surface d'entités uniques. Ces clusters sont ancrés dans des textes d'ancrage de Wikipedia qui renvoient aux mêmes articles.

Nous utilisons OPIEC59K pour l'entity canonicalization , En raison de la pertinence de l'ensemble de données pour l'entity canonicalization , nous avons opté pour CMVC pour notre approche.
Dans leur mise en œuvre de CMVC pour l'entity canonicalization , ils ont adopté une méthodologie inspirée de travaux antérieurs. Tout d'abord, les mentions d'entités individuelles ont été regroupées globalement en fonction de formes de surface uniques, en ignorant les contextes de mention spécifiques. Ce regroupement a servi de base pour les clusters au niveau des mentions. En s'appuyant sur l'approche de regroupement multi-vues décrite par Shen et al., ils ont représenté chaque expression nominale en utilisant des mentions textuelles provenant d'Internet et un graphe de connaissances ouvert extrait d'un système OIE. Ils ont utilisé un encodeur BERT pour capturer le contexte textuel entourant chaque entité (la "vue contextuelle") et un encodeur de graphe de connaissances TransE pour représenter les nœuds dans le graphe de connaissances ouvert (la "vue factuelle"). Leurs améliorations comprenaient le réglage fin de l'encodeur BERT en utilisant une supervision faible des entités corrélées et l'augmentation des représentations du graphe de connaissances. Ces deux vues de chaque entité ont ensuite été combinées pour créer une représentation unifiée. Pour simplifier le processus de regroupement, ils ont concaténé les représentations de chaque vue et ont utilisé le regroupement K-Means avec initialisation K-Means++ dans un espace vectoriel partagé, obtenant des performances comparables à l'approche originale. De plus, ils ont déterminé le nombre de centres de clusters en utilisant la méthode Log-Jump, en sélectionnant 490 clusters pour l'ensemble de données OPIEC59k.
Dans notre cas, nous avons utilisé plusieurs algorithmes et obtenu les résultats suivants :

Algorithmes/metrics	Macro_f1 (reproduction/originale)	micro_f1 (reproduction/originale)	Pairwise_f1 (reproduction/originale)
KMeans	0.236/0.535	0.814/0.91	0.726/0.856
GPTPairwiseClustering	0.295/0.587	0.823/0.915	0.719/0.861
GPTExpansionClusterig	0.477/0.603	0.892/0.925	0.831/0.873

3.1.2 Bank77 :

Présentation du dataset : Bank77 (Casanueva et al., 2020) contient 3 080 requêtes utilisateur pour un assistant bancaire en ligne classées en 77 catégories de requêtes.

Les résultats sont plus proches de de ceux de l'autre car **exactement** toutes les sources utilisées par l'auteur ont été trouvées de façon certaine.

Algorithmes/metrics	NMI (reproduction/original)	Acc (reproduction/original)
KMeans	0.806/**0.817**	0.593/**0.64**
GPTPairwiseClustering	**0.805**/0.796	0.594/**0.596**
GPTExpansionClusterig	0.822/**0.824**	**0.674**/0.653

Algorithmes/metrics	NMI	Acc
Clustering optimal (Etat de l'art sur le dataset)	0.841	0.712

4. Analyse Critique

Cette section examine les points forts et les limites de l'approche et propose des idées pour des évaluations complémentaires.

4.1 Points Forts de l'Approche

L'approche proposée par l'article présente plusieurs avantages significatifs :

- **Réduction du coût de la supervision humaine :** L'utilisation de grands modèles de langage (LLMs) comme pseudo-oracles permet de réduire le besoin de retour d'information direct des experts. Cela rend le clustering semi-supervisé plus économique et plus accessible à grande échelle.
- **Amélioration de la qualité des clusters :** L'utilisation des LLMs pour enrichir les représentations textuelles et pour ajouter des contraintes au processus de clustering contribue à améliorer la qualité des clusters obtenus, comme le montrent les résultats empiriques.
- **Facilité d'intégration :** L'approche proposée peut être intégrée dans des processus de clustering existants sans nécessiter de changements majeurs dans les algorithmes sous-jacents.

4.2 Limites de l'Approche

Malgré ses avantages, l'approche proposée présente certaines limites et défis :

- **Coût des LLMs :** Bien que les LLMs réduisent le besoin de supervision humaine, ils peuvent être coûteux à utiliser en raison des ressources de calcul requises. Les coûts d'utilisation de services LLM basés sur le cloud peuvent s'accumuler rapidement, surtout pour des ensembles de données volumineux.
- **Précision des Pseudo-Oracles :** Les LLMs, bien qu'utiles, peuvent introduire des erreurs en raison de leur nature probabiliste. Les erreurs dans les contraintes ou les corrections peuvent entraîner des clusters incorrects.

- **Impact sur la Reproductibilité :** L'utilisation de LLMs peut rendre la reproduction des résultats plus difficile, car les réponses des LLMs peuvent varier en fonction de différents facteurs, y compris les mises à jour des modèles ou les changements dans les paramètres de génération.

5. Évaluations Complémentaires :

5.1 Expérimentation d'un nouveau dataset "BBC news" (GPT 3.5, LAMA, MISTRAL) :

5.1.1 Avec GPT-3.5 comme LLM :

Dans cette section on analyse la performance des techniques de clustering appliquées à divers sous-ensembles du dataset de BBC News. L'analyse compare le clustering traditionnel k-means à une méthode améliorée utilisant la méthode "Key Phrase Expansion" avec **GPT-3.5** qu'on a implémenté nous même à partir de zéro. Nous avons utilisé différents sous-ensembles de données, allant de 250 à 2250 articles, et évalué la performance sous diverses graines aléatoires (42, 5, 25, 100). Les métriques utilisées pour évaluer la performance du clustering étaient l'Information Mutuelle Normalisée (**NMI**) et la précision (**acc**).

Méthodologie :

Les sous-ensembles de données ont été traités en utilisant deux approches de clustering :

- Clustering k-means simple qui a servi de référence.
- Clustering k-means amélioré par GPT-3.5 où les mot-clés générés par GPT-3.5 ont été utilisés pour augmenter les données textuelles, capturant vraisemblablement des significations sémantiques plus profondes et améliorant potentiellement les résultats de clustering.

L'efficacité du clustering a été évaluée en comparant les clusters prédits avec les catégories réelles des articles en utilisant l'NMI et la précision. Ces métriques fournissent un aperçu de la pureté et de la justesse des clusters générés par les modèles

Aperçu des Résultats :

Seed		42		5		25		100	
Metric		NMI	acc	NMI	acc	NMI	acc	NMI	acc
250 Articles	Simple	0.369	0.496	0.349	0.65	0.393	0.544	0.481	0.712
	GPT-3.5	0.881	0.956	0.656	0.704	0.668	0.776	0.886	0.96
500 Articles	Simple	0.659	0.806	0.639	0.8	0.35	0.558	0.35	0.558
	GPT-3.5	0.697	0.684	0.554	0.67	0.737	0.866	0.737	0.866
1000 Articles	Simple	0.71	0.57	0.61	0.605	0.63	0.658	0.59	0.70
	GPT-3.5	0.789	0.919	0.804	0.927	0.715	0.844	0.70	0.67
1500 Articles	Simple	0.669	0.798	0.732	0.852	0.765	0.901	0.581	0.73
	GPT-3.5	0.558	0.689	0.655	0.788	0.743	0.874	0.81	0.931
1850 Articles	Simple	0.69	0.82	0.57	0.551	0.695	0.838	0.696	0.707
	GPT-3.5	0.811	0.935	0.815	0.937	0.667	0.669	0.62	0.72
2250 Articles	Simple	0.815	0.926	0.637	0.733	0.668	0.704	0.697	0.805
	GPT-3.5	0.769	0.901	0.824	0.938	0.625	0.715	0.772	0.902

Les résultats indiquent une performance variable à travers différentes tailles de dataset et de graines, comme détaillé dans le tableau résumé :

250 Articles : GPT-3.5 surpasse significativement l'approche simple sur toutes les graines, avec des améliorations notables tant en IMN qu'en précision. Par exemple, sous la graine 42, l'approche GPT-3.5 atteint un IMN de 0.881 et une précision de 0.956, comparé respectivement à 0.369 et 0.496 pour l'approche simple.

500 Articles : Des tendances similaires sont observées, bien que la marge d'amélioration varie. Sous la graine 25, GPT-3.5 améliore la précision de 0.558 à 0.866.

1000 à 2250 Articles : À mesure que le nombre d'articles augmente, l'amélioration de la performance en utilisant GPT-3.5 reste évidente bien qu'elle devienne légèrement moins consistante dans certains cas. Par exemple, à 1500 articles, la graine 100 voit une montée en précision de 0.73 à 0.931 avec GPT-3.5.

Conclusions

La méthode de clustering améliorée utilisant l'expansion de mot-clé par GPT-3.5 offre généralement des améliorations substantielles par rapport au clustering k-means simple en

termes NMI et de précision. Ces améliorations sont les plus prononcées avec des datasets plus petits mais restent significatives avec des plus grands. Les résultats suggèrent que l'incorporation de mot-clés sémantiques peut aider à capturer des thèmes d'article plus nuancés, menant à des clusters plus cohérents et précis.

L'efficacité de cette méthode, particulièrement dans les datasets avec des sujets divers et nuancés comme les articles de nouvelles, souligne le potentiel de l'intégration de modèles de langage avancés dans les tâches de clustering de données. Cependant, les améliorations variables à travers différentes graines mettent également en lumière l'impact potentiel de l'initialisation et des processus stochastiques dans les algorithmes de clustering.

Travaux Futurs

Des recherches supplémentaires pourraient explorer la scalabilité de cette approche pour des datasets encore plus grands et son application à d'autres domaines.

5.1.2 En utilisant LIAMA et MISTRAL comme LLM :

Dans cette section, on examine l'efficacité de l'utilisation des modèles de langage open source, **Llama2** et **Mistra**l, pour améliorer la performance du clustering par rapport à une approche k-means simple. L'analyse se concentre sur un sous-ensemble du dataset de BBC News, testant à la fois des sous-ensembles plus petits (50 articles) et légèrement plus grands (100 articles) sous la graine 42. Les métriques d'évaluation utilisées étaient l'Information Mutuelle Normalisée (IMN) et la précision (acc).

Méthodologie

L'expérience impliquait trois approches de clustering :

Clustering k-means simple, servant de méthode de base.

Clustering amélioré par Llama2, utilisant le modèle de langue Llama2 pour la génération de phrases clés afin d'augmenter les données textuelles.

Clustering amélioré par Mistral, employant le modèle de langue Mistral visant de manière similaire à améliorer la compréhension sémantique des textes.
La performance était mesurée en termes d'**NMI** et de **précision**, quantifiant l'alignement et la justesse des clusters générés par rapport aux catégories réelles des articles.

Aperçu des Résultats :

Seed		42	
Metric		NMI	acc
50 Articles	Simple	0.436	0.6
	Llama2	0.583	0.7
100 Articles	Simple	0.609	0.77
	Llama2	0.818	0.86

Seed		42	
Metric		NMI	acc
50 Articles	Simple	0.436	0.6
	Mistral	0.605	0.64

Les métriques de performance à travers différentes méthodes et tailles de datasets sont les suivantes :

50 Articles :

- Approche Simple : A atteint une NMI de 0,436 et une précision de 0,6.
- Approche Llama2 : A montré une amélioration avec une NMI de 0,583 et une précision de 0,7.
- Approche Mistral : A également montré une amélioration par rapport à la méthode simple avec une NMI de 0,605 et une précision de 0,64.

100 Articles :

- Approche Simple : IMN de 0,609 avec une précision de 0,77.
- Approche Llama2 : Performance nettement supérieure avec une NMI de 0,818 et une précision de 0,86.

Analyse :

Les résultats indiquent que Llama2 et Mistral améliorent la performance du clustering par rapport au clustering k-means simple. Spécifiquement, Llama2 montre une amélioration significative, en particulier avec 100 articles, suggérant une meilleure scalabilité avec

l'augmentation de la taille du dataset. Mistral améliore également les métriques de clustering, mais dans une moindre mesure que Llama2.

Cependant, il existe un compromis considérable en termes d'efficacité computationnelle. Llama2 nécessite un temps de traitement nettement plus long (par exemple, 2 heures pour 50 articles) comparé à GPT-3, qui traite la même quantité en environ 2 minutes. Cette disparité met en lumière les défis de l'utilisation de modèles open source dans des applications gourmandes en ressources.

Conclusions :

Le clustering amélioré en utilisant des LLMs open source comme Llama2 et Mistral donne généralement lieu à une meilleure pureté et justesse des clusters à travers les sous-ensembles testés du dataset de BBC News. Bien que l'utilisation de Llama2 soit particulièrement efficace, l'augmentation de la taille du dataset conduit à des gains de performance qui compensent potentiellement ses longs temps de traitement.

Cela suggère que pour des applications où la précision est plus critique que la rapidité, Llama2 présente une alternative viable à des modèles plus coûteux comme GPT-3. Cependant, pour des applications sensibles au temps ou à grande échelle, le coût computationnel doit être pris en compte. Des recherches futures pourraient explorer des techniques d'optimisation pour réduire les temps de traitement ou comparer les performances à travers une gamme plus large de LLM open source.

5.2 Utilisation de la version 4.0 de GPT :

Dans cette partie nous allons étudier si l'utilisation d'une version plus performante de GPT (version 4.0 contre 3.5) améliore les résultats obtenus durant l'étude.

Voici les résultats obtenus

Algorithmes/metrics	NMI (GPT 4.0/GPT 3.5/original)	Acc (GPT 4.0/GPT 3.5/original)
KMeans	0.806/0.806/**0.817**	0.593/0.593/**0.64**
GPTPairwiseClustering	0.806/0.822/**0.824**	0.590/0.594/**0.596**
GPTExpansionClusterig	**0.832**/0.822/0.796	**0.677**/0.674/0.596

Algorithme/metrics	NMI	Acc
Clustering optimal (Etat de l'art sur le dataset)	0.841	0.712

Nous remarquons que l'utilisation de la version **4.0** de GPT améliore les résultats de la méthode **GPTExpansionClusterig** et s'approche grandement de l'état de l'art pour ce dataset.

5.2 Etude de l'impacte du nombre de contraintes dans l'efficacité de la méthode Pseudo-Oracle Pairwise Constraint Clustering :

Dans cette partie nous allons introduire 2 LLMs (GPT version 4.0 et 3.5) pendant le clustering pour générer les contraintes utiliser par l'algorithme **PCK-means.**

GPT 4.0

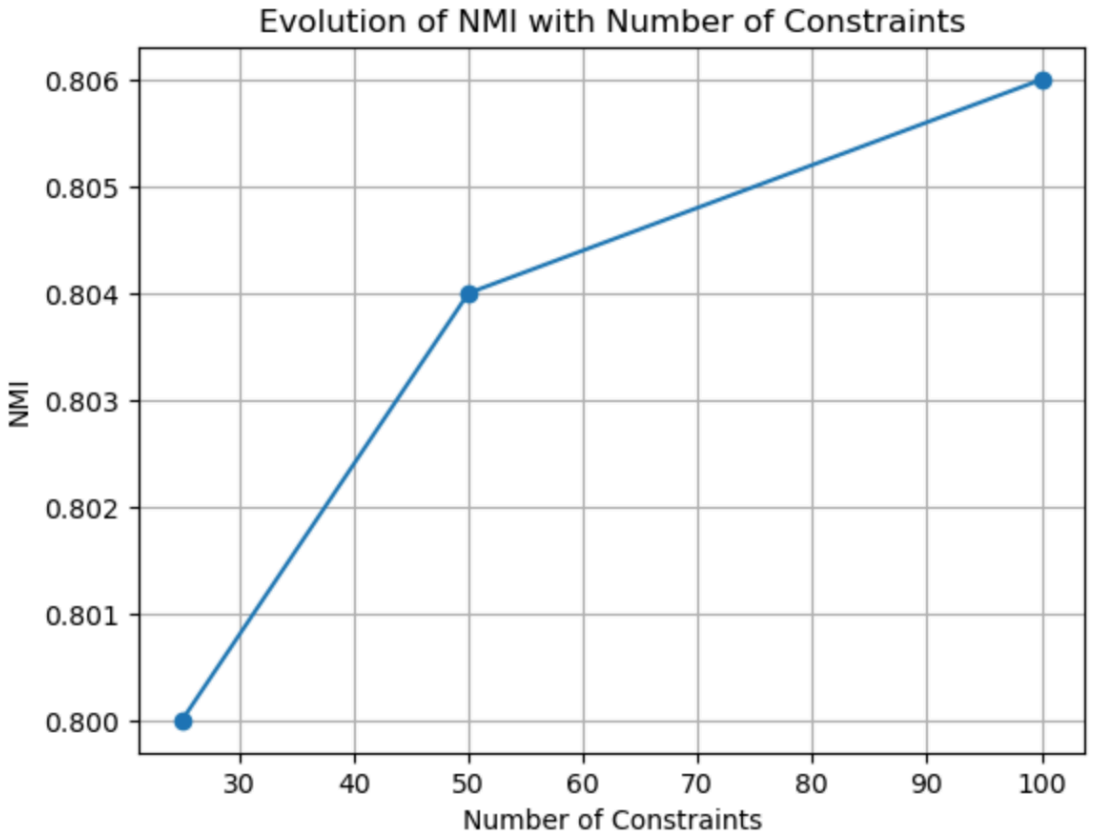

Figure 5 : Evolution de la métrique NMI selon le nombre de contraintes

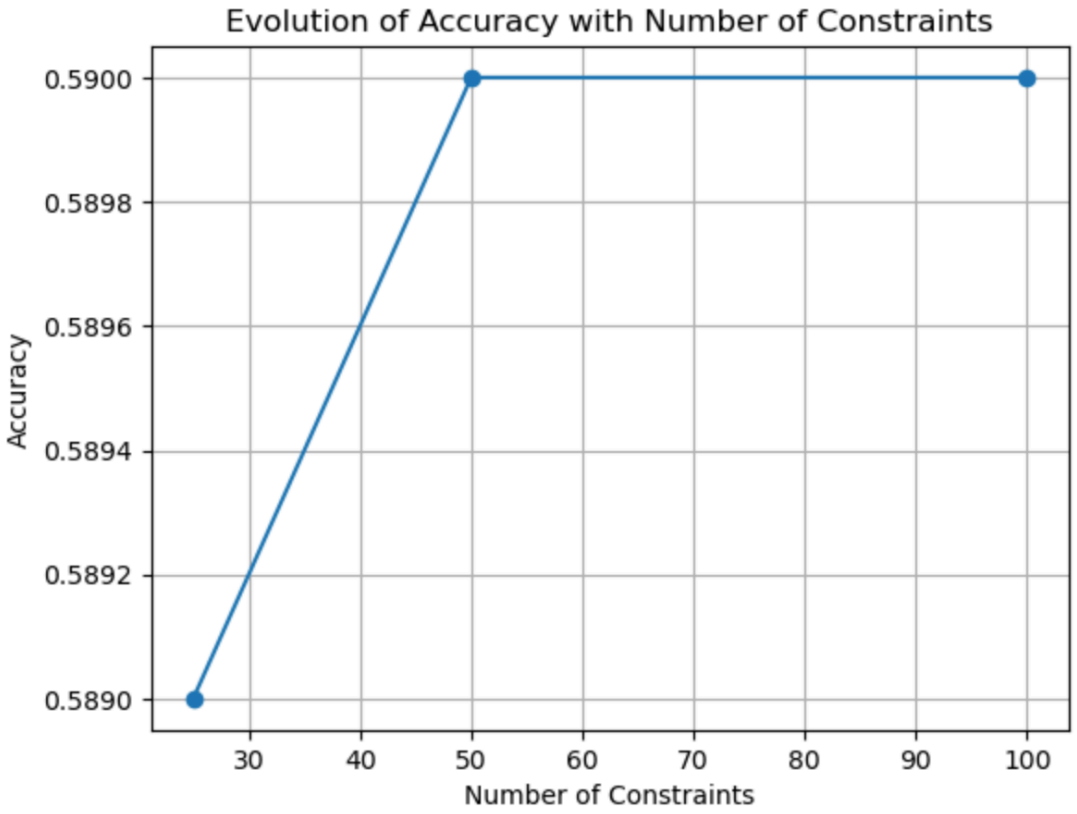

Figure 6 : Evolution de l'accuracy selon le nombre de contraintes

GPT 3.5

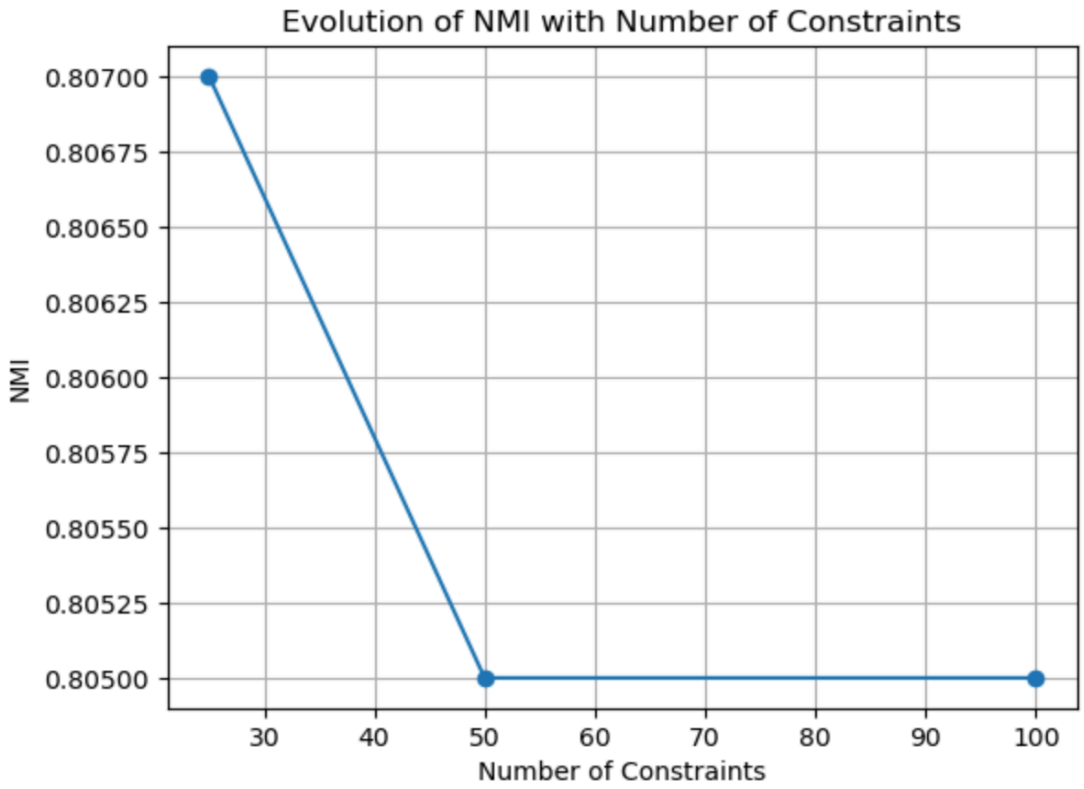

Figure 7 : Evolution de la métrique NMI selon le nombre de contraintes GPT 3.5

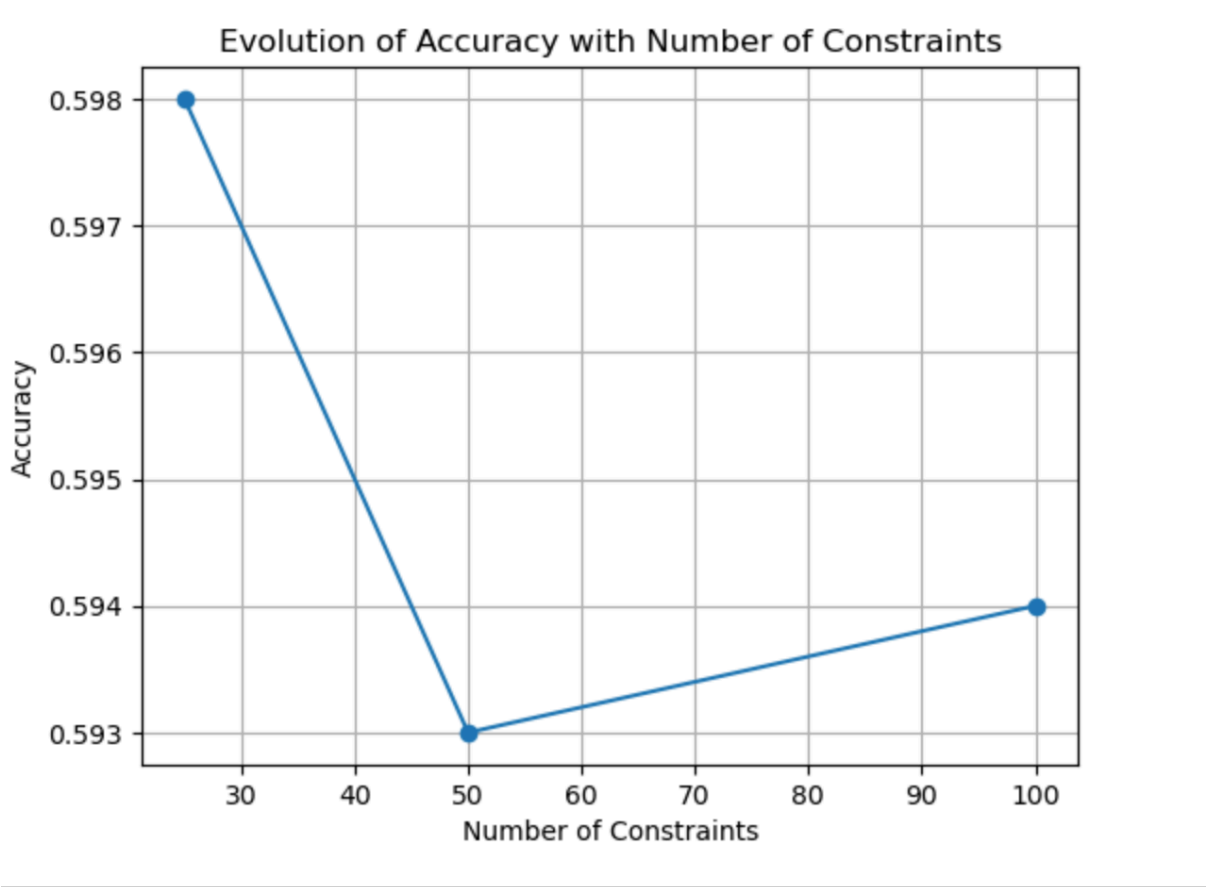

Figure 8 : Evolution de l'accuracy selon le nombre de contraintes GPT 3.5

Remarque : Les metrics (Accuracy et NMI) restent relativement stables malgré une différence considérable du nombre de contraintes.

6. Conclusion

Durant ce projet nous avons pu observer à quel point la révolution des Larges Langages Models peut intervenir dans le domaine de l'apprentissage non-supervisé, le fait de permettre le passage d'un contexte non-supervisé à un contexte semi-supervisé permet d'améliorer les résultats en prenant en compte les spécificités des données mais permet aussi la labellisation des données chose qui selon T Fredriksson et al "Les recherches actuelles estiment que plus de 80% des tâches d'ingénierie dans un projet d'apprentissage automatique concernent la préparation et l'étiquetage des données."

De plus les futurs versions des LLMs qui seront plus performantes promettent de donner de meilleurs résultats ce qui encourage à continuer à chercher de nouveaux domaines d'application dans le monde de l'apprentissage non-supervisé

7. Références

Wagstaff, K., Cardie, C., Rogers, S., & Schroedl, S. (2001, June). Constrained K-Means Clustering with Background Knowledge. In *ICML* (Vol. 1, pp. 577–584).

Basu, S., Banerjee, A., & Mooney, R. J. (2004, April). Active Semi-Supervision for Pairwise Constrained Clustering. In *Proceedings of the 2004 SIAM international conference on data mining* (pp. 333–344). Society for Industrial and Applied Mathematics.

Fredriksson, Teodor & Issa Mattos, David & Bosch, Jan & Olsson, Helena. (2020). Data Labeling: An Empirical Investigation into Industrial Challenges and Mitigation Strategies. 10.1007/978-3-030-64148-1_13.

CIKM '13: Proceedings of the 22nd ACM international conference on Information & Knowledge ManagementOctober 2013Pages 1259–1260https://doi.org/10.1145/2505515.2514692