JK-Means : LLM-as-a-Judge au Service du Clustering Intentionnel

Arnaud Deleruyelle[1], Steve Bellart[1], Jérôme Mollier-Pierret[1]

[1] Talan – Centre de Recherche et d'Innovation, France

Résumé

Nous proposons JK-Means, un algorithme de clustering guidé par l'intention utilisateur en langage naturel. Notre contribution principale repose sur un mécanisme d'évaluation LLM-as-a-Judge en temps constant qui valide la cohérence des clusters et déclenche des opérations de split/-merge adaptatives. Évaluée sur trois jeux de données de référence, notre approche atteint une haute précision et un fort alignement sémantique avec les classes d'origine, démontrant sa capacité à découvrir des clusters pertinents et conformes aux attentes de l'utilisateur.

Mots-clés

Clustering, LLM as a Judge, Intention utilisateur, Embeddings sémantiques.

Abstract

We propose JK-Means, a clustering algorithm guided by user intent expressed in natural language. Our main contribution relies on a constant-time LLM-as-a-Judge evaluation mechanism that validates cluster coherence and triggers adaptive split/merge operations. Evaluated on three benchmark datasets, our approach achieves high accuracy and strong semantic alignment with the original classes, demonstrating its ability to discover relevant clusters that conform to user expectations.

Keywords

Clustering, LLM as a Judge, User intent, Semantic embeddings.

1 Introduction

Si le clustering permet d'organiser efficacement des données non annotées, son application se heurte souvent à une limite majeure : la déconnexion entre la formation des clusters et la volonté de l'utilisateur. Les algorithmes de type K-Means, largement plébiscités pour leur rapidité computationnelle, illustrent bien ce problème : leur logique purement géométrique et leur sensibilité à l'initialisation les empêchent d'intégrer l'expertise métier. Or, dans de nombreux contextes applicatifs (analyse de tickets de support, classification de documents, veille documentaire), l'utilisateur possède une intention spécifique sur la manière de catégoriser les données, une connaissance que les méthodes classiques ne permettent pas d'exploiter.

Face à ces limites, l'émergence des Grands Modèles de Langage (LLM) offre de nouvelles opportunités pour le clustering de textes. Ces modèles possèdent des capacités de compréhension sémantique riches et peuvent générer des descriptions textuelles interprétables. De plus, les embeddings modernes capturent des relations sémantiques complexes dans des espaces vectoriels de haute dimension. Cependant, les approches existantes utilisant les LLM pour le clustering se limitent souvent à la génération de labels *a posteriori* ou à la classification directe, sans véritablement guider la formation des clusters par l'intention de l'utilisateur.

Nous proposons **JK-Means**, un algorithme de clustering non supervisé guidé par l'intention utilisateur exprimée en langage naturel. Nos contributions s'articulent autour de trois mécanismes clés qui redéfinissent les étapes standards de K-LLMeans :

Un cadre de clustering intentionnel continu. L'intention de l'utilisateur guide désormais toutes les étapes de l'algorithme. Par exemple, pour l'intention « identifier les problèmes techniques SAP », l'algorithme génère automatiquement des centroïdes textuels spécifiques (« Erreurs de connexion à la base de données », « Problèmes d'autorisation », etc.), les raffine à chaque itération selon les documents assignés, et évalue leur pertinence par rapport au besoin métier. Cette approche transforme le clustering d'une tâche purement géométrique en une tâche sémantique ciblée, tout en restant non supervisée.

Une initialisation gloutonne maximisant la diversité. Pour contrer la sensibilité aux conditions initiales, nous proposons une heuristique de sélection des centroïdes textuels de départ. Afin d'éviter l'apparition de clusters "trop génériques" (ex : « Problèmes système ») qui absorbent trop de documents et créent des minima locaux, notre méthode sélectionne itérativement les titres les plus éloignés sémantiquement. Cela garantit un point de départ spécifique et bien segmenté.

Une évaluation LLM-as-a-Judge frugale et ciblée. Pour valider la cohérence des clusters sans faire exploser les coûts liés aux appels API des LLM, nous introduisons un mécanisme d'évaluation suivant : Plutôt que de faire analyser l'intégralité du jeu de données par le modèle, l'algorithme se concentre uniquement sur les cas les plus incertains : les documents les plus éloignés du centroïde. L'idée sous-jacente de cette approche étant que si ces documents périphériques sont jugés cohérents avec le titre du cluster, alors le cœur du cluster l'est également. Dans le cas contraire, un mécanisme de séparation (split) génère deux nouveaux titres plus spécifiques. Cette stratégie permet d'échapper aux minima locaux et d'affiner la catégorisation, tout en réduisant drastiquement les coûts computa-

tionnels et financiers.

Le reste de l'article est organisé comme suit : la section 2 présente l'état de l'art sur le clustering et les approches basées sur les LLM. La section 3 détaille notre méthode JK-Means. La section 4 présente le protocole expérimental et la section 5 se concentre sur les résultats obtenus. Enfin, la Section 6 conclut et discute des perspectives.

2 Travaux Antérieurs

2.1 Clustering Non Supervisé Classique

L'algorithme K-Means [10] reste une référence pour le clustering non supervisé grâce à sa simplicité et son efficacité. Il alterne entre assignation des points au centroïde le plus proche et mise à jour des centroïdes comme moyenne des points assignés. Cependant, K-Means souffre de limitations bien documentées : sensibilité à l'initialisation aléatoire, nécessité de fixer le nombre de clusters K a priori, et tendance à converger vers des minima locaux.

K-Means++ [2] améliore l'initialisation en sélectionnant les centroïdes initiaux de manière probabiliste. L'algorithme fonctionne comme suit : (1) le premier centroïde est choisi uniformément au hasard parmi les points de données, (2) pour chaque point x, on calcule la distance $D(x)$ au centroïde le plus proche déjà sélectionné, (3) le prochain centroïde est choisi avec une probabilité proportionnelle à $D(x)^2$, favorisant ainsi les points éloignés des centroïdes existants. Cette approche garantit que les centroïdes initiaux sont bien espacés, réduisant le risque de convergence vers des minima locaux et améliorant la qualité finale du clustering.

Les approches basées sur la densité comme DBSCAN [5] et HDBSCAN détectent automatiquement le nombre de clusters et gèrent les outliers, mais leur complexité computationnelle limite leur scalabilité. Le clustering hiérarchique offre une flexibilité via les dendrogrammes, mais reste difficile à interpréter pour de grandes quantités de données.

Dans le contexte du clustering de textes, l'utilisation d'embeddings sémantiques (Word2Vec [11], GloVe [13], BERT [3]) a significativement amélioré la qualité des clusters en capturant les relations sémantiques. Sentence-BERT [15] génère des représentations vectorielles de phrases entières, permettant un clustering plus cohérent. Les modèles d'embedding modernes comme text-embedding-3-large offrent des représentations de haute dimension (3072 dimensions) pouvant capturer des nuances sémantiques. Des travaux récents [12] ont démontré que les embeddings issus de LLM pré-entraînés (comme all-MiniLM-L6-v2 [19]) produisent des clusters significativement plus interprétables par des humains que les méthodes traditionnelles (doc2vec, LDA) sur des textes courts, avec une amélioration de 40% de la cohérence perçue. Cependant, ces approches restent non supervisées au sens strict : elles ne permettent pas d'intégrer l'expertise ou l'intention de l'utilisateur de manière structurée.

2.2 LLM pour le Clustering : K-LLMeans et NILC

Les LLM ont ouvert de nouvelles perspectives pour le clustering de textes. Plusieurs travaux récents explorent leur utilisation pour générer des labels de clusters *a posteriori*, améliorant l'interprétabilité sans modifier la formation des clusters [21]. TopicGPT [14] utilise des prompts structurés pour générer des descriptions hiérarchiques de topics, facilitant l'interprétation des résultats. Viswanathan et al. [17] proposent d'utiliser les LLM pour augmenter les représentations de documents et générer des contraintes par paires, permettant un clustering semi-supervisé efficace avec un nombre réduit d'annotations manuelles.

K-LLMeans. K-LLMeans [4] introduit l'utilisation de résumés textuels comme centroïdes, offrant une approche interprétable pour le clustering de textes. L'algorithme présente de nombreux avantages : (1) **Coûts faibles** car il repose essentiellement sur des embeddings pour l'assignation des documents, avec seulement K appels LLM par étape de résumé pour la mise à jour des centroïdes. (2) **Scalabilité** préservée grâce à l'utilisation d'embeddings modernes efficaces. (3) **Interprétabilité** améliorée car les centroïdes textuels sont directement lisibles et actionnables. (4) **Robustesse** grâce aux connaissances sémantiques apportées par les LLM.

ClusterLLM. Dans une approche complémentaire, ClusterLLM [21] exploite les LLM pour guider le fine-tuning d'embedders pré-entraînés. Le LLM est interrogé sur des triplets ⟨anchor, choice_1, choice_2⟩ pour déterminer quelle option est la plus proche de la référence selon une perspective donnée (ex : "Select based on topic"). Un échantillonnage entropique sélectionne les triplets les plus informatifs, permettant de réduire les coûts d'API tout en améliorant les performances sur 14 datasets testés. Les prédictions du LLM sont utilisées pour fine-tuner l'embedder via une loss contrastive, produisant un espace de représentation mieux aligné sur la perspective souhaitée.

NILC. Dans le cadre de NILC [18], l'approche est plus coûteuse mais offre une précision accrue. NILC utilise les LLM comme juges ("LLM as a Judge" [22]) pour évaluer la cohérence des clusters et guider leur formation. Cette approche pilote la procédure d'assignation via des Modèles de Langage, permettant une évaluation sémantique fine de la qualité des clusters. Le paradigme "LLM as a Judge" démontre l'efficacité de l'utilisation des LLM pour des tâches d'évaluation et de validation, au prix d'un nombre d'appels LLM plus élevé ($O(n^2)$ dans certaines configurations).

Notre approche se distingue par son caractère frugal : elle combine l'intégration de l'intention utilisateur en langage naturel avec une évaluation stratégiquement ciblée via le paradigme "LLM as a Judge", maintenant un coût constant de K appels LLM par itération. Là où NILC déploie le paradigme "LLM as a Judge" de manière exhaustive avec une complexité en $O(n^2)$, et où ClusterLLM nécessite un fine-tuning coûteux de l'embedder sur de nombreux triplets, notre méthode réalise une évaluation intentionnelle sélective en se concentrant uniquement sur les éléments ex-

trêmes de chaque cluster. Cette stratégie frugale maintient un coût constant indépendant de n, permettant de détecter efficacement les incohérences sémantiques et de déclencher, si nécessaire, un mécanisme de split de cluster. Elle corrige ainsi la trajectoire des centroïdes en temps réel, évitant la convergence vers des structures sémantiquement non pertinentes, tout en préservant le caractère non supervisé du clustering. Notre approche opère directement dans l'espace sémantique des résumés générés, offrant flexibilité et agnosticisme par rapport au choix de l'embedder, sans nécessiter de phase d'entraînement préalable.

Initialisation. Notre heuristique gloutonne s'inspire de K-Means++ en maximisant l'espacement des centroïdes initiaux, mais opère dans l'espace sémantique des titres générés par le LLM plutôt que dans l'espace des données. Nous remplaçons la sélection probabiliste par une sélection basée sur les relations sémantiques : à chaque étape, nous choisissons le titre ayant la plus faible similarité cosinus avec les titres déjà sélectionnés, garantissant une diversité sémantique maximale.

3 Notre contribution : JK-Means

3.1 Préliminaires : K-Means pour le Clustering de Textes

Soit un corpus de n documents textuels $\mathcal{D} = \{d_1, \ldots, d_n\}$. Chaque document d_i est représenté par un vecteur d'embedding $\mathbf{x}_i \in \mathbb{R}^d$ tel que :

$$\mathbf{x}_i = f_{emb}(d_i) \tag{1}$$

Nous supposons que tous les embeddings sont normalisés ($\|\mathbf{x}_i\| = 1$), rendant équivalentes la distance euclidienne et la similarité cosinus. L'objectif de K-Means est de partitionner ces n embeddings en k clusters, minimisant la variance intra-cluster :

$$\arg \min_{C_1,\ldots,C_k} \sum_{j=1}^{k} \sum_{i \in [C_j]} \|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2 \tag{2}$$

où C_j dénote l'ensemble des embeddings assignés au cluster j, $[C_j] = \{i | \mathbf{x}_i \in C_j\}$ dénote l'ensemble des indices assignés au cluster j, et $\boldsymbol{\mu}_j$ est le centroïde du cluster, calculé comme la moyenne des embeddings assignés :

$$\boldsymbol{\mu}_j = \frac{1}{|C_j|} \sum_{i \in [C_j]} \mathbf{x}_i \tag{3}$$

L'algorithme K-Means assigne itérativement chaque embedding $\mathbf{x}_i$ au centroïde le plus proche et met à jour les centroïdes jusqu'à convergence après T itérations. Cependant, K-Means souffre de limitations bien documentées : sensibilité à l'initialisation et tendance à converger vers des minima locaux en raison de la nature non-convexe de sa fonction objectif [10, 9].

3.2 Préliminaires : K-LLMeans

Pour améliorer K-Means pour le clustering de textes, K-LLMmeans [4] introduit le concept de **summary-as-centroid** : périodiquement, les centroïdes numériques sont remplacés par des résumés textuels générés par un LLM, puis ré-encodés dans l'espace d'embedding. Formellement, la distinction principale réside dans le mécanisme de mise à jour des centroïdes : toutes les l itérations, la mise à jour standard de l'équation (3) est remplacée par :

$$\boldsymbol{\mu}_j = f_{\text{emb}}\left(f_{\text{LLM}}(\mathcal{P}_j)\right) \tag{4}$$

où :

- f_{emb} est la fonction d'embedding (ex : text-embedding-3-small, e5-large) ;
- f_{LLM} est la fonction de génération du LLM (ex : GPT-4o, Claude-3.7) ;
- $\mathcal{P}_j = \text{Prompt}\left(\mathcal{I}, \{d_{z_i} \mid z_i \sim [C_j]\}_{i=1}^{m_j}\right)$ est le prompt construit à partir de :
  - $\mathcal{I}$: l'instruction (ex : *"The following is a cluster of online banking questions. Write a single question that represents the cluster concisely."*) ;
  - $\{d_{z_i}\}_{i=1}^{m_j}$: un échantillon de m_j documents du cluster C_j, où $z_i \sim [C_j]$ désigne un indice échantillonné sans remplacement et m est le nombre maximal de documents par prompt (typiquement $m = 10$ en mode *few-shot*, ou $m = |C_j|$ pour utiliser tous les documents du cluster).

Entre deux étapes de résumé, K-LLMmeans effectue des itérations standards de K-Means (assignation + mise à jour numérique des centroïdes), préservant ainsi l'objectif classique de minimisation de la variance intra-cluster. Le prompt utilisé pour la génération de résumés est donné en Annexe B.2.

Plutôt que de fournir tous les documents du cluster en entrée, le LLM traite un échantillon représentatif afin de fortement limiter le coût et préserver la taille du contexte en entrée au LLM.

Dans K-LLMmeans [4], les auteurs proposent d'utiliser un **échantillonnage K-means++** des embeddings du cluster pour sélectionner les m documents représentatifs à fournir au LLM. Cette approche maximise la distance entre les documents échantillonnés, pour diversifier l'aperçu du contenu sémantique du cluster. Les auteurs comparent empiriquement cette option avec plusieurs alternatives testées sur les embeddings text-embedding-3-small, notamment la sélection des m documents les plus proches du centroïde.

Notre approche diffère de K-LLMmeans en sélectionnant cette seconde stratégie, choix motivé par plusieurs considérations. Tout d'abord les **écarts de performance marginaux** entre les stratégies d'échantillonnage restent faibles (0.3–0.7% ACC sur CLINC/GoEmo). Ensuite, contrairement à K-LLMmeans qui utilise text-embedding-3-small, nous utilisons text-embedding-3-large et l'augmentation de la dimensionnalité renforce le phénomène de **concentration des distances** [1] ce qui va encore réduire les écart entre ces stratégies. Enfin, la sélection des m plus proches nécessite uniquement un tri par distance (complexité $O(n \log m)$), alors que K-means++ requiert un échantillonnage itératif (complexité $O(m \cdot n)$).

3.3 Architecture Générale de JK-Means

JK-Means étend K-LLMeans en intégrant l'intention utilisateur comme une variable au coeur de nos templates de prompts, que cela soit durant les itérations, B.2 mais aussi dès l'initialisation B.1 et également durant notre étape additionnelle d'évaluation ciblée des clusters B.3.
La Figure 1 présente l'architecture complète de notre approche.
Nous modifions l'initialisation aléatoire de K-Means ou probabiliste de K-Means++ pour un algorithme glouton afin de commencer avec des centroides cohérents avec l'intention de l'utilisateur. Ensuite, nous reprenons les mêmes première étapes de K-LLMeans dans notre boucle principale (mais avec notre propre prompt de regénération de titre avec une focalisation sur l'intention utilisateur B.2) suivi d'un mécanisme d'évaluation qui, en fonction de son résultat, va enclencher soit une séparation du cluster (split) si celui-ci est jugé incohérent, soit une sauvegarde dynamique si le cluster est validé. Enfin, un mécanisme de fusion permet de pallier les clusters de petites tailles car ils sont responsables d'optimums locaux.
Les sous-sections suivantes détaillent l'intégration de l'intention utilisateur dans notre approche lors de l'initialisation gloutonne (Section 3.4), dans la boucle principal (Section 3.5) ainsi que dans l'évaluation des clusters et les mécanismes de split/merge des clusters (Section 3.6) avant de finir sur les optimisations de coûts en appel LLM (Section 3.7).

3.4 Heuristique Greedy pour l'initialisation des titres

L'objectif de cette phase est d'éviter la création de **clusters attracteurs dominants** : une mauvaise initialisation qui génère un titre trop générique (ex : "Problème système") risque d'attirer la majeure partie des documents, piégeant l'algorithme dans un minimum local. Nous voulons obtenir des titres spécifiques qui couvrent le spectre de l'intention utilisateur (voir Annexe B, Prompt B.1) sans être génériques.
Le LLM génère d'abord N_{cand} titres candidats (typiquement $N_{cand} = 50$) basés sur l'intention utilisateur. Nous devons ensuite sélectionner K titres parmi ces candidats pour initialiser les clusters. Nous nous basons sur la similarité cosinus pour identifier les titres spécifiques. L'idée est qu'un titre ayant une faible similarité cosinus avec de nombreux autres doit avoir une faible généricité. Notre heuristique gloutonne fonctionne comme suit :
Étape 1 : Sélection du premier titre. Nous sélectionnons le titre t_1^* ayant la plus faible similarité cosinus moyenne avec tous les autres candidats :

$$t_1^* = \arg\min_{t \in \mathcal{T}_{cand}} \frac{1}{N_{cand} - 1} \sum_{t' \in \mathcal{T}_{cand} \setminus \{t\}} \cos(f_{emb}(t), f_{emb}(t')) \tag{5}$$

Étape 2 : Sélection gloutonne itérative. Une fois un titre sélectionné et ajouté au pool $\mathcal{T}_{sel}$, nous ajoutons itérativement les nouveaux titres un par un en choisissant celui qui minimise la similarité cosinus moyenne avec les titres déjà sélectionnés :

$$t_{i+1}^* = \arg\min_{t \in \mathcal{T}_{cand} \setminus \mathcal{T}_{sel}} \frac{1}{|\mathcal{T}_{sel}|} \sum_{t' \in \mathcal{T}_{sel}} \cos(f_{emb}(t), f_{emb}(t')) \tag{6}$$

Cette procédure garantit que chaque nouveau titre apporte une diversité sémantique maximale, forçant les clusters initiaux à être spécifiques et bien séparés. Les titres sélectionnés sont ensuite vectorisés pour former les centroïdes initiaux : $\mu_k = f_{emb}(t_k^*)$.

3.5 La régénération de titre avec intention

L'intention utilisateur guide l'ensemble du processus de clustering à travers les prompts LLM à plusieurs niveaux dont celui de la boucle principale visant à regénérer de nouveau titre de cluster. À chaque itération, lors du raffinement des titres, l'intention est intégrée dans le prompt pour générer des titres cohérents avec l'objectif métier (voir Annexe B, Prompt B.2). Cette intégration garantit que les titres générés restent alignés avec l'objectif métier tout au long des itérations.
De plus, dans la phase d'évaluation LLM as a Judge, l'intention se retrouve également dans le prompt d'évaluation (voir Annexe B, Prompt B.3) puisque le LLM évalue si un document appartient à un cluster en tenant compte de l'intention globale. Cette évaluation contextualisée garantit que les assignations respectent l'objectif métier et ne se basent pas uniquement sur la similarité sémantique brute de l'embedder.

3.6 Evaluation LLM as a Judge des clusters

À chaque itération, après l'assignation des documents aux clusters, nous évaluons la cohérence de chaque cluster pour identifier ceux nécessitant une subdivision (split) ou une fusion (merge). Cette évaluation repose sur le paradigme "LLM as a Judge" qui permet de détecter les clusters hétérogènes tout en maintenant un coût computationnel constant.

3.6.1 Évaluation LLM as a Judge

Pour chaque cluster C_k, nous évaluons la cohérence des N_{worst} documents les plus éloignés du centroïde (ceux avec les scores d'affinité les plus faibles). Soit D_{worst}^k l'ensemble de ces documents :

$$D_{worst}^k = \{d_i \in C_k \mid \text{affinity}(d_i, c_k) \in \text{Bottom}_{N_{worst}}(C_k)\} \tag{7}$$

Pour chaque document $d_i \in D_{worst}^k$, le LLM évalue si le document appartient bien au cluster en posant la question "Est-ce que d_i appartient à t_k ?" (voir Annexe B, Prompt B.3). Le score de cohérence du cluster est calculé comme le pourcentage de documents validés par le LLM parmi les N_{worst} documents évalués.
Si les documents les plus éloignés du centroïde (les plus difficiles à classer) sont validés par le LLM, alors nous considérons le cluster comme pertinent. Cette approche garantit un temps constant d'évaluation ($O(N_{worst})$ au lieu de

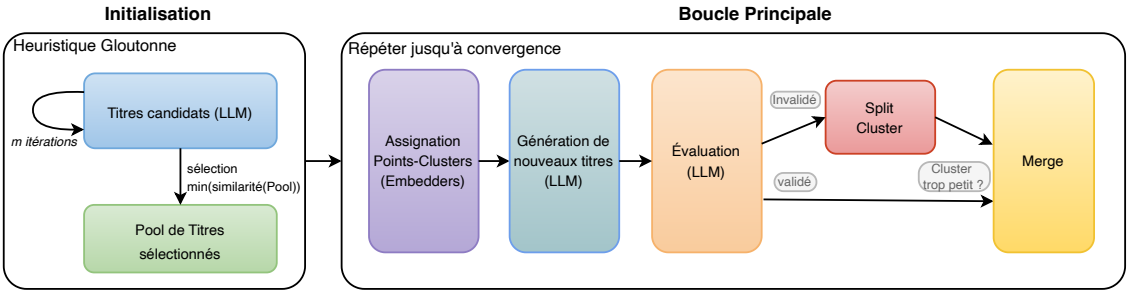

FIGURE 1 – Architecture de JK-Means. La phase d'initialisation repose sur une heuristique greedy de sélection des titres. La boucle principale a une structure séquentielle reprenant les étapes de K-LLMeans en ajoutant un mécanisme d'évaluation par LLM et des opérations de split et merge.

$O(|C_k|)$) tout en identifiant efficacement les clusters hétérogènes. Si un document est invalidé, cela signale que le cluster contient des sous-catégories distinctes nécessitant une subdivision. Cette évaluation est également réutilisée pour définir le critère d'arrêt de l'algorithme (voir Section 3.8).

3.6.2 Mécanisme de Split

Si un cluster C_k a un score de cohérence faible (score$(C_k) < \tau_{split}$) et une taille suffisante ($|C_k| \geq \text{size}_{min}$), nous appliquons une procédure de split pour le diviser en deux sous-clusters. Nous sélectionnons les N_{top} documents les plus proches du centroïde et les N_{bottom} documents les plus éloignés. Pour chaque groupe, le LLM génère un nouveau titre spécifique :

$$t_{k,1} = f_{llm}(\text{“Titre pour : ”} + D^k_{top}) \quad (8)$$

$$t_{k,2} = f_{llm}(\text{“Titre pour : ”} + D^k_{bottom}) \quad (9)$$

Les nouveaux centroïdes sont $\mu_{k,1} = f_{emb}(t_{k,1})$ et $\mu_{k,2} = f_{emb}(t_{k,2})$. Tous les documents de C_k sont ensuite réassignés aux deux nouveaux clusters selon leur affinité. Ce mécanisme permet d'explorer de nouvelles configurations de clusters et de découvrir des sous-catégories sémantiques, palliant ainsi les minima locaux causés par des clusters trop génériques.

3.6.3 Mécanisme de Merge

Un mécanisme de merge est instancié pour éviter les optimums locaux, notamment les clusters de taille 1. En effet, si un cluster ne contient qu'un seul document, le titre généré sera parfaitement adapté à cette unique donnée. Ce cluster ne bougera jamais lors des itérations suivantes, créant ainsi un optimum local qui empêche toute évolution.

De plus, le mécanisme d'exploration engendré par le mécanisme de subdivision évoqué dans la section précédente peut entraîner la création de nombreux clusters de petites tailles. Il faut donc une procédure inverse pour compenser cette fragmentation excessive.

De ce fait, nous supprimons automatiquement les clusters d'une taille inférieur à un seuil (Ce seuil étant un paramètre de l'algorithme voir 4.2). Les documents de ces clusters sont réassignés aux clusters restants selon leur affinité, permettant ainsi de maintenir un équilibre entre exploration (via le split) et consolidation (via le merge).

3.7 Optimisation des Appels LLM

Pour réduire les coûts computationnels liés à l'usage de LLM, nous nous sommes concentré sur quelques leviers d'optimisations :

Assignation au cluster sur une sélection de chunks. Plutôt que de traiter chaque document dans son intégralité pour l'assignation à un cluster, nous calculons le score d'affinité basé sur les $M_{\text{top|chunks}}$ chunks les plus proches d'un centroïde en réalisant la somme des similarités cosinus par cluster et en affectant un document au cluster ayant la plus grande somme sur ces chunks représentatifs. Cette approche capture le cœur sémantique du document sans nécessiter le traitement de l'intégralité du texte, même si en pratique, le paramètre $M_{\text{top|chunks}}$ est réglé de sorte que seuls les documents les plus volumineux d'un dataset soient concernés par cette approximation.

Reconstruction ciblée. Lors de la génération de titres de clusters, nous sélectionnons uniquement les $M_{\text{top|docs}}$ chunks ayant la plus haute similarité cosinus avec le centroïde. Seul le contenu le plus représentatif est envoyé au LLM, réduisant drastiquement le nombre de tokens d'entrée. De façon similaire, ce paramètre intervient uniquement dans le traitement de documents volumineux.

Ces optimisations combinées réduisent significativement les coûts LLM tout en préservant la qualité du clustering.

3.8 Critère d'Arrêt

K-Means ou K-LLMeans sont des approches qui s'arrêtent lorsque le nombre de modifications entre deux itérations est faible (stabilité des assignations), ce qui n'est pas adaptée dans notre cas. En effet, nos mécanismes de split et merge augmentent artificiellement le nombre de modifications effectuées à chaque itération, rendant ce critère d'arrêt inefficace.

Ainsi, nous proposons une méthode plus adaptée qui consiste à réutiliser l'évaluation faite par le LLM as a Judge comme un indicateur de la performance actuelle du modèle.

Nous utilisons cette métrique comme critère d'arrêt en calculant le score de cohérence moyen global :

$$\text{score}_{global} = \frac{1}{K} \sum_{k=1}^{K} \text{score}(C_k) \tag{10}$$

L'algorithme a une patience de 3 itérations (auquel cas nous conservons l'itération avec le meilleur score) ou peut également s'arrêter si aucun split n'est effectué. Dans nos expériences, nous avons utilisé un nombre d'itérations fixes ($N_{\text{iterations}} = 5$) pour garantir la reproductibilité et permettre une comparaison équitable avec les approches antérieures.

4 Expérimentations

4.1 Datasets et Configuration

Nous évaluons notre approche sur trois datasets de référence : 20 Newsgroups [7] (10 classes), AG News [20] (4 classes), et Reuters-21578 [8] (79 classes). Nous utilisons Gemini 2.5 Flash comme LLM et text-embedding-3-large (3072 dimensions) comme modèle d'embedding.

Intentions utilisateur. Pour chaque dataset, nous définissons une intention spécifique guidant le clustering : **20 Newsgroups** ("Classification de groupes de discussion"), **AG News** ("Classification d'articles de presse"), et **Reuters-21578** ("Classification d'articles"). Ces intentions sont intégrées dans tous les prompts LLM (génération de titres initiaux, raffinement, évaluation) pour guider la formation des clusters.

Variantes comparées. Nous comparons deux approches. (1) **K-LLMeans** : K-LLMeans avec prompts intentionnels et sélection des $M_{\text{top|docs}}$ documents les plus proches du centroïde (Section 3.2) pour la génération de titres et initialisé avec K titres avec l'aide d'un LLM. (2) **JK-Means** : Notre approche complète ajoutant l'initialisation gloutonne (Section 3.4), l'évaluation LLM as a Judge (Section 3.6), et les mécanismes de split/merge.

Pour les deux approches, le nombre initial de titres est fixé au nombre de classes du dataset. Cela permet à la première approche d'obtenir les meilleures performances. D'un autre côté, une création trop intensive de nouveaux clusters risque d'impacter négativement la seconde approche en augmentant l'écart avec le nombre de classes réelles.

4.2 Paramètres Expérimentaux

Le Tableau 1 présente les hyperparamètres utilisés pour toutes les expériences. Ces paramètres sont identiques pour les trois datasets afin de garantir une comparaison équitable.

Assignation granulaire. Le paramètre $M_{\text{top|chunks}} = 5$ limite le nombre de chunks les plus pertinents utilisés pour calculer le score d'affinité de chaque document.

Raffinement des titres. Le paramètre $M_{\text{top|docs}} = 5$ limite le nombre de documents envoyés au LLM pour raffiner les titres de clusters. (Dans **K-LLMeans** et **JK-Means**)

Initialisation intentionnelle. Le paramètre $N_{\text{candidates}} = 50$ génère 50 titres candidats via le LLM, parmi lesquels l'heuristique gloutonne sélectionne les K titres les plus diversifiés (où K est le nombre de classes du dataset).

TABLE 1 – Hyperparamètres Expérimentaux

Paramètre	Valeur	Description
$M_{\text{top\|chunks}}$	5	Nombre de chunks les plus pertinents pour l'assignation
$M_{\text{top\|docs}}$	5	Nombre de documents représentatifs pour la génération de titres
$N_{\text{iterations}}$	5	Nombre maximum d'itérations
$N_{\text{max\|chars}}$	10000	Limite de caractères pour les textes envoyés au LLM
$N_{\text{candidates}}$	50	Nombre de titres candidats générés lors de l'initialisation
N_{refine}	1	Fréquence de raffinement des titres via LLM (à chaque itération)
size_{min}	4	Taille minimale d'un cluster avant fusion (merge)

Robustesse aux hyperparamètres. L'utilisation d'un ensemble unique de paramètres pour les trois datasets démontre la robustesse de notre approche. Cette stabilité met en avant la résilience de notre approche face à la perturbation d'un paramètre isolé.

4.3 Métriques d'Évaluation

Nous évaluons notre approche selon cinq métriques complémentaires, permettant l'évaluation de la qualité du clustering ainsi que de la qualité sémantique des titres générés. Nous utilisons des métriques robustes pour l'évaluation de solutions avec un nombre de clusters différents. C'est pourquoi nous prenons une accuracy bijective, ainsi que la NMI et le F1-score.

1. Accuracy Bijective. Cette métrique mesure l'alignement bijectif optimal entre clusters et classes via l'algorithme hongrois [6]. Contrairement à l'accuracy standard, elle pénalise fortement les écarts entre le nombre de clusters générés et le nombre de classes réelles. Ce choix est délibéré : nous souhaitons pénaliser un mécanisme de génération de clusters excessif, s'ils n'apportent que peu d'intérêt.

2. NMI (Normalized Mutual Information). Le NMI [16] mesure l'information mutuelle normalisée entre les clusters prédits et les vraies classes. Cette métrique est robuste aux déséquilibres de classes et quantifie combien d'information sur les vraies classes peut être obtenue en connaissant les clusters.

3. F1-Score. Le F1-Score est la moyenne harmonique de la précision et du rappel, calculée sur les paires de documents. Il équilibre la capacité à créer des clusters homogènes (précision) et à regrouper tous les documents d'une même classe (rappel).

4. Score de Cohérence LLM (Critère d'Arrêt). Comme évoqué en Section 3.8, nous utilisons le score de cohérence moyen calculé par le LLM as a Judge comme indicateur de la qualité du clustering. Ce score mesure le pourcentage de documents validés par le LLM parmi les documents les plus éloignés de chaque centroïde.

5. Similarité Cosinus Maximale (Alignement Sémantique). Pour chaque classe réelle l_j, nous calculons la similarité cosinus maximale avec les titres de clusters générés.

Cette métrique vérifie si nos titres générés sont sémantiquement proches des vraies classes du dataset. Elle illustre la capacité de l'algorithme à découvrir automatiquement des catégories cohérentes avec les labels réels, sans jamais les avoir vus.

5 Résultats et Discussions

5.1 Vue d'Ensemble des Résultats

La Figure 2 présente l'évolution de l'accuracy bijective au fil des itérations pour les trois datasets (AG News, 20 Newsgroups, Reuters). Cette vue globale révèle une tendance claire : JK-Means (courbe bleue) surpasse systématiquement K-LLMeans (courbe orange) sur les trois jeux de données. L'amélioration est particulièrement marquée après quelques itérations, démontrant l'efficacité de la couche exploratoire. Des résultats exhaustifs incluant toutes les métriques pour chaque itération sont présentés en Annexe A (Tableau 5).

Le reste de cette section détaille ces résultats en trois volets complémentaires : (1) une analyse quantitative des performances finales via des métriques de clustering, (2) une évaluation de la qualité de l'initialisation gloutonne, et (3) une mesure de l'alignement sémantique entre les titres générés et les vraies classes. Nous discuterons ensuite les comportements spécifiques observés sur chaque dataset.

5.2 Performances Globales de Clustering

Le Tableau 2 présente les meilleures performances obtenues sur 5 itérations pour les deux approches sur les trois datasets. Les métriques retenues sont l'accuracy bijective, le F1-Score et le NMI.

TABLE 2 – Comparaison des Performances Maximales (Meilleures sur 5 Itérations)

Dataset	Algorithme	Acc. Bij.	F1-Score	NMI
AG News	K-LLMeans	0.6260	0.6430	0.6354
AG News	JK-Means	**0.9600**	**0.9357**	**0.9102**
20NG	K-LLMeans	0.6450	0.5654	0.7563
20NG	JK-Means	**0.7830**	**0.7801**	**0.8821**
Reuters	K-LLMeans	0.6510	0.8252	0.7745
Reuters	JK-Means	**0.7527**	**0.9068**	**0.8258**

Les résultats démontrent des gains significatifs pour l'approche exploratoire sur tous les datasets. Sur AG News, l'amélioration est significative : +53.4% d'accuracy bijective, +45.5% de F1-Score, et +43.3% de NMI. Sur 20 Newsgroups, les gains sont de +21.4% en accuracy, +38.0% en F1-Score, et +16.6% en NMI. Sur Reuters, les améliorations sont de +15.6% en accuracy, +9.9% en F1-Score, et +6.6% en NMI.

Ces résultats montrent l'efficacité de notre approche exploratoire permettant la découverte d'un plus grand nombre de clusters candidats.

5.3 Qualité de l'Initialisation Gloutonne

L'heuristique gloutonne de sélection des titres initiaux vise à maximiser la diversité sémantique et à produire une distribution équilibrée des documents entre les clusters dès la première itération. Le Tableau 3 compare les métriques de distribution initiale entre l'approche standard et l'approche exploratoire.

TABLE 3 – Qualité de la Distribution Initiale (Itération 1)

Dataset	Algorithme	Acc. Bij.	Équilibre
AG News	K-LLMeans	0.4080	0.6961
AG News	JK-Means	**0.5840**	**0.8563**
20 Newsgroups	K-LLMeans	0.2350	0.7413
20 Newsgroups	JK-Means	**0.2840**	**0.8285**
Reuters	K-LLMeans	**0.1343**	**0.7031**
Reuters	JK-Means	0.1103	0.6231

Équilibre : Entropie normalisée [0,1]. Plus élevé = meilleure distribution.

Sur AG News et 20 Newsgroups, l'heuristique gloutonne améliore la qualité de l'initialisation avec des gains respectifs en accuracy de +43.1% et +20.9%. Concernant la mesure d'équilibre, des gains significatifs sur ces mêmes datasets sont observés avec une amélioration de 0.696 à 0.856 sur AG News et de 0.741 à 0.829 pour 20 Newsgroup. Cette stabilité permet de reduire la probabilité d'obtenir un super-cluster (optimum local) lors de l'initialisation.

Le cas de Reuters est plus nuancé : l'accuracy initiale est légèrement inférieure (11.0% vs 13.4%), il en est de même l'équilibre de distribution. Cette apparente contradiction s'explique par la nature du dataset : avec 79 classes, la probabilité d'obtenir un super-cluster générique est très faible, et l'approche standard bénéficie d'une initialisation favorable. Nous reviendrons sur ce point dans la discussion sur Reuters (Section 5.7).

5.4 Alignement Sémantique avec les Vraies Classes

Une métrique cruciale pour évaluer la qualité de notre approche est la similarité cosinus maximale entre les titres de clusters générés et les noms des vraies classes. Cette métrique démontre que l'intention utilisateur guide efficacement la formation des clusters vers des catégories alignées avec les concepts métier, tout en préservant le caractère non supervisé de l'approche. L'évolution détaillée de cette métrique pour chaque itération est présentée dans le Tableau 5 en Annexe A.

JK-Means génère systématiquement des titres plus proches sémantiquement des vraies classes. Sur 20 Newsgroups, à l'itération 4, la similarité cosinus atteint 0.412 (contre 0.324 pour K-LLMeans), soit une amélioration de +27.2%. Les titres générés capturent bien les thématiques réelles (sports, politique, technologie) sans avoir jamais vu les labels. Sur AG News, l'amélioration est encore plus spectaculaire : 0.376 à l'itération 2 (contre 0.273), soit +37.7%, confirmant la capacité de l'algorithme à découvrir des catégories cohérentes (World, Sports, Business, Sci/Tech). Sur Reuters et ces 79 classes, la similarité atteint 0.269 à l'itération 5 (contre 0.202 pour K-LLMeans), soit +33.2%, prouvant que notre approche génère des titres plus représentatifs.

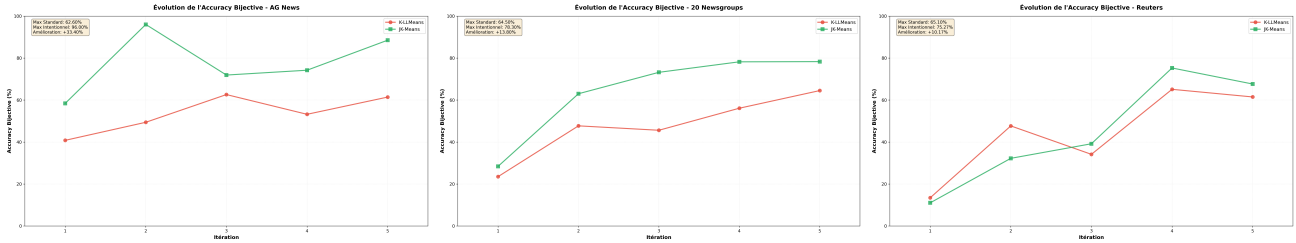

FIGURE 2 – Évolution de l'accuracy bijective au fil des itérations pour les trois datasets (AG News, 20 Newsgroups, Reuters). La courbe orange représente K-LLMeans, tandis que la courbe bleue représente JK-Means. La seconde approche converge rapidement vers des accuracy significativement supérieures, démontrant l'efficacité de l'initialisation et des mécanismes de split/merge.

5.5 Discussion : 20 Newsgroups – Cas Classique de Convergence

Le dataset 20 Newsgroups représente un cas classique où l'algorithme fonctionne de manière optimale. L'algorithme JK-Means surpasse systématiquement l'autre approche avec des gains significatifs : +21.4% en accuracy bijective (78.3% vs 64.5%), +38.0% en F1-Score (0.780 vs 0.565), et +16.6% en NMI (0.882 vs 0.756). Cette convergence rapide et stable valide l'efficacité de notre approche sur des données avec une structure sémantique claire et un nombre modéré de classes.

5.6 Discussion : AG News – Sur-Fragmentation et Critère d'Arrêt

Le dataset AG News présente un comportement particulier qui révèle une limitation de notre brique exploratoire : la tendance à la sur-fragmentation sur des datasets avec un nombre réduit de classes. AG News ne contient que 4 classes (World, Sports, Business, Sci/Tech), ce qui rend l'algorithme vulnérable à la création excessive de clusters via le mécanisme de split.

5.6.1 Évolution du Nombre de Clusters

Le Tableau 4 montre l'évolution du nombre de clusters de notre approche au fil des itérations. L'algorithme démarre avec 4 clusters, puis augmente progressivement jusqu'à 13 clusters à l'itération 4, avant de redescendre à 10 à l'itération 5.

TABLE 4 – Évolution du Nombre de Clusters sur AG News, pour la variante exploratoire

Itération	Nombre de Clusters	Acc. Bijective
1	4	0.5840
2	5	**0.9600**
3	9	0.7190
4	13	0.7420
5	10	0.8850

Cette sur-fragmentation explique la chute d'accuracy observée à partir de l'itération 3. L'accuracy bijective pénalise les écarts entre le nombre de clusters générés et le nombre de classes réelles. À l'itération 2, avec 5 clusters, l'accuracy atteint un pic optimal de 96.0%. Lors de l'itération suivante, l'augmentation du nombre de clusters entraine une chute de l'accuracy (71.9%).

5.6.2 Utilisation du Score de Cohérence comme Critère d'Arrêt

Le Tableau 5 en annexe présente l'évolution du score de cohérence LLM as a Judge en parallèle des métriques de performance. En s'arrêtant à l'itération où le score de cohérence est maximal, on éviterait la chute d'accuracy des itérations suivantes. Cette observation se vérifie sur les trois datasets : sur 20 Newsgroups et Reuters, l'itération avec le score de cohérence maximal correspond effectivement à la meilleure accuracy obtenue, tandis que sur AG News, elle correspond au second meilleur score, très proche de l'optimum. Ce critère d'arrêt met en avant un second intérêt de notre évaluation LLM as a Judge.

5.7 Discussion : Reuters – Initialisation Défavorable et Rattrapage

Le dataset Reuters présente un comportement inverse à celui d'AG News : l'approche K-LLMeans surperforme initialement, mais JK-Means rattrape et dépasse sur le long terme. Une explication se base sur le haut nombre de classes (79) dans ce dataset.

Avec une initialisation à 79 clusters, l'approche K-LLMeans a peu de chance d'avoir une initialisation défavorable. La chute de performance à l'itération suivante peut même présumer d'une initialisation favorable. D'un autre côté, notre heuristique gloutonne présente des faiblesses dans ce cas de figure. Sur une base de 250 titres générés (uniquement sur ce jeu de données) via un LLM pour n'en sélectionner que 79, il est possible qu'une aussi haute génération de titres basée sur une intention puisse générer de nombreux titres peu informatifs. De ce fait, notre heuristique gloutonne pourrait prioriser des titres avec une richesse sémantique trop faible.

Cependant, dès l'itération 2, JK-Means rattrape (32.2% vs 47.7%), puis dépasse progressivement K-LLMeans. À l'itération 4, nous atteignons 75.3% d'accuracy, contre 65.1%. Cette évolution démontre que les mécanismes d'exploration (split/merge) de notre approche permettent de compenser un mauvais point de départ, prouvant la robustesse de l'algorithme face aux conditions initiales défavorables.

6 Conclusion et Perspectives

Nous avons proposé JK-Means, une approche de clustering non supervisé guidé par l'intention utilisateur exprimée en langage naturel. Notre méthode repose sur trois contributions principales : (1) une initialisation gloutonne maximisant la diversité sémantique des titres initiaux, (2) une évaluation LLM as a Judge en temps constant favorisant l'exploration de nouveaux clusters, et (3) des mécanismes de split/merge permettant d'adapter dynamiquement le nombre de clusters.

Les expérimentations sur trois datasets de référence démontrent des gains significatifs en accuracy bijective, F1-Score et NMI sur tous les datasets. Plus important, notre métrique de similarité sémantique prouve que l'algorithme découvre automatiquement des catégories cohérentes avec les vraies classes, sans jamais les avoir vues. Cette capacité ouvre la voie à des applications métier où l'expertise utilisateur peut guider la découverte de structures dans les données.

Perspectives. Nos expérimentations révèlent deux écueils liés au nombre de classes. Sur des datasets avec un nombre très faible de classes (AG News, 4 classes), le mécanisme de split tend à créer une sur-fragmentation. Bien que notre critère d'arrêt basé sur le score de cohérence LLM permette d'éviter ces problèmes en détectant le point optimal avant la dégradation, une régulation plus fine des mécanismes de split/merge pourrait améliorer la stabilité. À l'inverse, sur des datasets avec un très grand nombre de classes (Reuters, 79 classes), la probabilité d'obtenir un super-cluster générique lors de l'initialisation devient très faible, rendant notre heuristique gloutonne moins déterminante.

Une perspective actuellement en cours est d'intégrer de l'exploitation dans la phase de split. Plutôt que de simplement diviser un cluster hétérogène en deux sous-clusters basés sur les documents les plus proches et les plus éloignés du centroïde, nous pourrions réeffectuer un algorithme de type K-LLMeans à petite échelle entre toutes les données les plus éloignées des clusters entre eux. Cette approche favoriserait un rapprochement de données similaires et permettrait d'optimiser le mécanisme de subdivision en réduisant la génération de nouveaux clusters de taille réduite.

Remerciements

Les auteurs remercient chaleureusement Talan pour son soutien constant et son engagement dans ces travaux de recherche. L'ensemble du code source est disponible en open source à l'adresse suivante : https://github.com/ArnaudDeleruyelle/JKMeans.

A Résultats Détaillés par Itération

Le Tableau 5 présente l'évolution complète de toutes les métriques pour les deux approches sur les trois datasets au fil des 5 itérations.

B Prompts utilisés

Cette annexe présente les trois prompts principaux utilisés dans notre approche K-LLMeans et JK-Means. Les variables sont indiquées entre crochets (ex : `[INTENTION]`, `[NUM_CANDIDATES]`).

B.1 Prompt 1 : Génération des Titres Initiaux

Prompt de Génération des Titres Initiaux

Tu es un expert en analyse sémantique de documents. Ta tâche est de proposer une liste très diverse de `[NUM_CANDIDATES]` sous-catégories spécifiques liées au thème principal : '`[INTENTION]`'.
Suis ces règles strictes pour chaque titre :

- Sois très spécifique et distinct des autres titres.
- Doit être concis (3 à 6 mots).
- **N'utilise pas de termes génériques** comme 'Système', 'Erreur', 'Problème', ou 'Incident'. Concentre-toi sur des sujets techniques ou fonctionnels concrets.
- Varie au maximum les domaines fonctionnels et techniques.

Retourne uniquement la liste des titres, un par ligne, sans numéros ni puces.

B.2 Prompt 2 : Régénération des Titres de Clusters

Prompt de Régénération des Titres

Tu es un expert en analyse sémantique de documents. Analyse les documents suivants pour en extraire le thème commun et créer un titre de cluster.
— Règles Impératives —

1. **Sois Unique** : Le nouveau titre doit être sémantiquement distinct des titres existants listés ci-dessous.
2. **Sois Spécifique** : Évite les termes trop génériques comme 'Problème', 'Erreur', 'Système'.
3. **Sois Concis** : Le titre doit faire entre 3 et 6 mots.
4. **Respecte le Thème** : Le titre DOIT s'aligner avec le thème général : '`[INTENTION]`'.

— Documents à Analyser —
Document 1 (Titre Original : '`[DOC_TITLE_1]`') : — Début du Document — `[CONTENU_1]` — Fin du Document —
Document 2 (Titre Original : '`[DOC_TITLE_2]`') : — Début du Document — `[CONTENU_2]` — Fin du Document —
...
— Titres Existants (à ne pas imiter) — - `[TITRE_EXISTANT_1]` - `[TITRE_EXISTANT_2]` - ...
IMPORTANT : Ne retourne STRICTEMENT que le nouveau titre commun et unique, et rien d'autre.
Nouveau Titre :

TABLE 5 – Évolution complète des métriques (sans ARI) par itération

Dataset	Algorithme	It.	Acc.	F1	NMI	Coh.	Cos.	Dist.
AG News	K-LLMeans	1	0.408	0.366	0.143	-	0.115	1.330
AG News	K-LLMeans	2	0.494	0.467	0.375	-	0.273	1.203
AG News	K-LLMeans	3	0.626	0.584	0.487	-	0.229	1.241
AG News	K-LLMeans	4	0.532	0.558	0.493	-	0.242	1.230
AG News	K-LLMeans	5	0.614	0.643	0.635	-	0.269	1.208
AG News	JK-Means	1	0.584	0.480	0.286	50.7%	0.115	1.331
AG News	JK-Means	2	**0.960**	**0.936**	**0.910**	**69.8%**	**0.376**	**1.116**
AG News	JK-Means	3	0.719	0.743	0.735	43.7%	0.361	1.128
AG News	JK-Means	4	0.742	0.818	0.812	55.3%	0.344	1.145
AG News	JK-Means	5	0.885	0.844	0.793	59.5%	0.333	1.154
20 News	K-LLMeans	1	0.235	0.279	0.308	-	0.100	1.341
20 News	K-LLMeans	2	0.477	0.471	0.632	-	0.258	1.210
20 News	K-LLMeans	3	0.456	0.414	0.657	-	0.286	1.185
20 News	K-LLMeans	4	0.561	0.533	0.743	-	0.324	1.153
20 News	K-LLMeans	5	0.645	0.565	0.756	-	0.294	1.180
20 News	JK-Means	1	0.284	0.307	0.371	18.4%	0.113	1.332
20 News	JK-Means	2	0.630	0.573	0.730	43.0%	0.365	1.122
20 News	JK-Means	3	0.732	0.752	0.863	65.2%	0.392	1.098
20 News	JK-Means	4	0.782	0.721	0.862	**85.2%**	**0.412**	**1.079**
20 News	JK-Means	5	**0.783**	**0.780**	**0.882**	76.8%	0.393	1.096
Reuters	K-LLMeans	1	0.134	0.336	0.442	-	0.128	1.320
Reuters	K-LLMeans	2	0.477	0.640	0.712	-	0.185	1.273
Reuters	K-LLMeans	3	0.341	0.619	0.700	-	0.191	1.269
Reuters	K-LLMeans	4	0.651	0.825	0.775	-	0.196	1.264
Reuters	K-LLMeans	5	0.615	0.753	0.766	-	0.202	1.258
Reuters	JK-Means	1	0.110	0.561	0.536	43.6%	0.160	1.296
Reuters	JK-Means	2	0.322	0.558	0.659	57.2%	0.253	1.219
Reuters	JK-Means	3	0.392	0.637	0.726	53.7%	**0.279**	**1.198**
Reuters	JK-Means	4	**0.753**	**0.907**	**0.826**	**70.2%**	0.263	1.210
Reuters	JK-Means	5	0.676	0.816	0.796	66.4%	0.269	1.206

It. : Itération. **Acc.** : Accuracy bijective. **F1** : F1-Score. **NMI** : Normalized Mutual Information. **Coh.** : Score de cohérence LLM (critère d'arrêt). **Cos.** : Similarité cosinus max. **Dist.** : Distance euclidienne min.

B.3 Prompt 3 : Évaluation LLM as a Judge

Prompt d'Évaluation LLM as a Judge

L'objectif est de regrouper les documents selon le thème suivant : `[INTENTION]`.
Contenu du document : `[DOCUMENT]`
Titre du cluster : `[TITRE_CLUSTER]`
Le titre du cluster correspond à une catégorie générale ou fonctionnelle. Votre objectif est d'évaluer si ce titre est **pertinent** pour regrouper ce document, même s'il ne couvre pas tous les détails du résumé, en tenant compte du thème général défini ci-dessus.
Considérez le titre comme **représentatif** si :

- Il reflète le thème principal du document (même partiellement).
- Il désigne un module, un processus, ou une activité en lien clair avec le résumé.
- Il est suffisamment proche pour qu'un regroupement de documents proches soit possible.
- Il respecte l'intention de clustering définie.

Répondez uniquement par 'OUI' si le lien est **raisonnable et justifié**, même s'il n'est pas strict ou exhaustif. Sinon, répondez 'NON'.

Références

[1] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. On the surprising behavior of distance metrics in high dimensional space. In *Proceedings of the 8th International Conference on Database Theory (ICDT)*, volume 1973 of *Lecture Notes in Computer Science*, pages 420–434. Springer, 2001.

[2] David Arthur and Sergei Vassilvitskii. k-means++ : The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1027–1035, 2007.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL-HLT)*, pages 4171–4186, 2019.

[4] Jairo Diaz-Rodriguez. Summaries as centroids for interpretable and scalable text clustering. *arXiv preprint arXiv :2502.09667*, 2025.

[5] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 226–231, 1996.

[6] Harold W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2) :83–97, 1955.

[7] Ken Lang. Newsweeder : Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning (ICML)*, pages 331–339, 1995.

[8] David D. Lewis. Reuters-21578 text categorization test collection, distribution 1.0. http://www.daviddlewis.com/resources/testcollections/reuters21578/, 1997.

[9] Stuart P. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2) :129–137, 1982.

[10] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, 1967.

[11] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *ICLR*, 2013.

[12] Justin K. Miller and Tristram J. Alexander. Human-interpretable clustering of short text using large language models. *Royal Society Open Science*, 12(1), 2025.

[13] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove : Global vectors for word representation. In *EMNLP*, 2014.

[14] Chau Minh Pham, Alexander Hoyle, Simran Srinivasan, and Nils Reimers. Topicgpt : A prompt-based topic modeling framework. In *NAACL*, 2024.

[15] Nils Reimers and Iryna Gurevych. Sentence-bert : Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 3982–3992, 2019.

[16] Andrew Rosenberg and Julia Hirschberg. V-measure : A conditional entropy-based external cluster evaluation measure. pages 410–420, 2007.

[17] Vijay Viswanathan, Kiril Tomkins, Arjun Mahajan, Ellie Pavlick, and Chris Callison-Burch. Large language models enable few-shot clustering. *arXiv preprint arXiv :2307.00524*, 2023.

[18] Hongtao Wang, Renchi Yang, and Wenqing Lin. Nilc : Discovering new intents with llm-assisted clustering. In *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining*, pages 671–680, 2026.

[19] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm : Deep self-attention distillation for task-agnostic compression. In *NeurIPS*, 2020.

[20] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28 (NIPS)*, pages 649–657, 2015.

[21] Yuwei Zhang, Zihan Meng, Preslav Nakov, and Sophia Ananiadou. Clusterllm : Large language models as a guide for text clustering. *arXiv preprint arXiv :2305.14871*, 2023.

[22] Hui Zheng, Yongfeng Liu, Xuanli Yang, Kexun Chen, and Yu-Ping Tan. Systematic evaluation of llm-as-a-judge in llm alignment tasks. *arXiv :2408.13006*, 2024.