Exploration des données arXiv par le NLP et l'analyse de graphes

Issa KA et Clara GARCIA PEREZ

Mai 2025

1 Introduction

La production scientifique mondiale ne cesse de croître à un rythme exponentiel, avec des milliers d'articles publiés en libre accès chaque jour sur des plateformes comme arXiv. Si cette profusion constitue une richesse indéniable pour la recherche, elle engendre aussi des difficultés croissantes en termes de tri, de classification, et d'accès pertinent à l'information. Par ailleurs, la collaboration scientifique est de plus en plus globale et interconnectée, ce qui soulève de nouvelles questions sur la structure des réseaux de chercheurs et les dynamiques de co-publication. Dans ce contexte, les méthodes de Data Science, en particulier les outils issus du traitement automatique du langage naturel (NLP) et de l'analyse de graphes, apparaissent comme des leviers puissants pour automatiser l'analyse de ces données textuelles massives et pour modéliser les interactions scientifiques.

Notre attachement à ce sujet est double. D'une part, notre intérêt personnel pour le Machine Learning et le NLP nous a naturellement amenés à nous tourner vers arXiv, une véritable mine d'or de textes scientifiques exploitables. D'autre part, dans le cadre de notre recherche de stage en laboratoire de recherche, nous avons longuement exploré cette plateforme, ce qui a renforcé notre envie d'en comprendre la structure et les dynamiques.

L'objectif global de notre projet est donc d'analyser et de traiter les publications scientifiques d'arXiv et les collaborations scientifiques à l'aide de méthodes de Data Science, en combinant approches supervisées, non supervisées, et analyse de graphes. Dans ce rapport, nous présenterons d'abord le jeu de données constitué et notre méthode de collecte. Nous détaillerons ensuite les différentes problématiques posées et les méthodes choisies pour y répondre. Une section technique explicitera nos choix d'implémentation, avant de laisser place à l'analyse des résultats obtenus et à une réflexion sur les limites du projet. Nous conclurons enfin par un bilan et des perspectives d'amélioration ou de prolongement.

2 Jeu de données et collecte

Comme mentionné précédemment, Arxiv est une gigantesque source d'information. Cette plateforme permet, à tous, d'accéder gratuitement au contenu intégrale de millions d'articles scientifiques de thèmes divers ainsi qu'à leurs métadonnées. Pour récupérer cette masse d'information, nous avons développé un scraper basé sur l'API officielle d'arXiv. Nous l'avons conçu de manière modulaire et réutilisable, afin qu'il puisse facilement servir à d'autres cas d'usage. Son fonctionnement sera détaillé plus bas. Grâce à cet outil, nous avons constitué une base de données de plus de 1,2 million d'articles, chacun étant associé à un identifiant unique, un titre, un résumé (abstract), une catégorie (ex : mathématiques), une sous-catégorie (ex : algèbre), une liste d'auteurs, ainsi qu'aux dates de publication et de dernière mise à jour. L'ensemble représente plus de 2 Go de données exploitables. Le volume et la diversité de cette base de données nous ont offert un terrain d'expérimentation idéal pour mettre en œuvre les différentes méthodes d'analyse vues en cours et en explorer d'autres. Tout en garantissant une représentativité du monde de la recherche actuel.

3 Problématique et méthodes

Plutôt que de suivre une unique problématique centrale, nous avons fait le choix d'adopter une approche transversale. L'objectif de ce projet est de proposer un **tour d'horizon assez large** des

possibilités offertes par l'analyse des données issues d'arXiv.

Le premier enjeu que nous avons souhaité explorer est celui de la **classification automatique** des articles scientifiques. Face à la quantité massive de textes disponibles sur arXiv, il est légitime de se demander s'il est possible de **prédire automatiquement la catégorie d'un article et/ou sa sous catégorie** (par exemple mathématiques, physique, informatique ou analyse, mécanique, machine learning) à partir de son **titre et/ou de son résumé**. Pour cela, nous avons mobilisé des modèles de classification supervisée, allant de réseaux de neurones denses simples à des architectures plus avancées comme les LSTM et les LSTM combinés à des mécanismes d'attention.

Dans un second temps, nous nous sommes intéressés à la structure de **l'espace vectoriel associé aux articles**. Pour représenter numériquement le contenu textuel, nous avons commencé par **embedder le vocabulaire** : chaque mot du corpus est projeté dans un espace vectoriel de dimension arbitraire, de manière à capturer ses relations sémantiques avec les autres termes. Étant donné qu'un titre ou un résumé est une simple séquence de mots, nous pouvons alors en déduire une **représentation vectorielle de l'article lui-même**, obtenue à partir de ces embeddings de mots. Ces représentations vectorielles ne sont pas statiques : elles sont **apprises et ajustées dynamiquement pendant l'entraînement des modèles de classification** évoqués précédemment. L'objectif est que, progressivement, ces espaces deviennent les plus pertinents possible pour accomplir la tâche de classification. Une fois les modèles entraînés, nous disposons donc d'espaces vectoriels organisés de manière à séparer au mieux les catégories. Cela soulève naturellement plusieurs questions : **à quoi ressemblent ces espaces ?** L'organisation des mots reflète-t-elle des proximités sémantiques cohérentes ? Les articles proches dans cet espace appartiennent-ils aux mêmes domaines ? **Le regroupement non supervisé** des articles dans cet espace reproduit-il, ou même améliore-t-il, la classification officielle proposée par arXiv ? Pour explorer ces pistes, nous avons appliqué **l'algorithme des K-means** aux embeddings, combiné à la **réduction de dimension par UMAP** pour en faciliter la visualisation et l'interprétation.

Enfin, une autre dimension essentielle de la base de données d'arXiv est la structure des collaborations scientifiques. C'est pourquoi, nous avons modélisé les données de co-signature d'article sous forme d'un **graphe pondéré non dirigé**, où chaque nœud représente un auteur, et chaque arête une collaboration entre chercheurs. De là, nous nous sommes posé plusieurs questions : quelle forme prend le réseau ainsi constitué ? Observe-t-on la formation de communautés structurées ? Ces groupes présentent-ils des caractéristiques communes, comme un domaine de recherche ou une proximité géographique ? Ce type de réseau a la même structure qu'un réseau social ? Peut-on le considérer comme un **petit monde ?** Son degré de séparation est-il comparable à celui des humains (6 degrés) ?

Pour y répondre, nous avons appliqué plusieurs outils d'analyse dont **l'algorithme de Louvain** pour identifier des communautés dans le réseau, les **plus courts chemins** pour évaluer la connectivité globale, et la **centralité intermédiaire** pour détecter les auteurs les plus influents ou les mieux connectés. Une analyse descriptive complémentaire du graphe nous a permis de mieux comprendre sa structure générale (nature des collaborations, densité, degrés, etc.).

4 Détails techniques et implémentations

4.1 Scrappeur

ArXiv ne propose pas d'API au sens classique du terme, mais fournit un endpoint (`http://export.arxiv.org/api/query`) permettant de récupérer des articles au format XML en fonction d'une requête personnalisée. Nous avons donc conçu une classe `ArxivScraper` qui prend en charge la construction dynamique de ces requêtes selon les besoins de l'utilisateur. Elle intègre une méthode `scrape` qui gère l'ensemble des appels à l'endpoint. L'utilisateur peut ainsi contrôler les paramètres techniques des requêtes (temps de pause entre les appels, nombre d'articles par requête, fréquence d'interrogation, etc.) afin d'éviter de surcharger le serveur, tout en configurant les critères de filtrage des articles (catégories, sous-catégories, mots-clés, tri chronologique, nombre maximal d'articles par catégorie, etc.). Les réponses XML sont traitées au fil de l'eau et transformées en un fichier CSV structuré selon les champs sélectionnés.

4.2 Classification

4.2.1 Prétraitement

Chaque mot d'un titre ou d'un résumé est indexé par un entier dans un vocabulaire de taille Ω créé à partir des données, et chaque catégorie (ou sous-catégorie) est également mappée à un entier. La classe `ArticleDataset` que nous avons implémentée prend en entrée un CSV issu de notre scraper, supprime les doublons ainsi que les articles dont les catégories ou sous-catégories ne figurent plus dans la taxonomie actuelle d'arXiv, puis construit automatiquement le vocabulaire et les mappings mots $\rightarrow$ indices et catégories $\rightarrow$ indices associés. Elle permet également d'appliquer divers filtres (seuil d'occurrence minimale des mots, nombre d'articles par catégorie, etc.). Sa méthode `__getitem__` renvoie à chaque itération un tuple (liste d'indices de mots, indice de la catégorie), prêt à alimenter l'entraînement de nos modèles.

4.2.2 Modèles

Réseau de neurones dense (MLP) Le premier modèle repose sur une architecture de type perceptron multicouche (MLP). Chaque séquence de mots en entrée est d'abord projetée dans un espace dense via une couche `nn.Embedding`. On obtient alors une représentation vectorielle de dimension `emb_dim` pour chaque mot du titre ou résumé de l'article. Pour que chaque article soit représenté par un vecteur de taille fixe, nous appliquons une opération d'**average pooling**, consistant à faire la moyenne (coordonnée par coordonnée) des vecteurs de tous les mots présents dans la séquence. Ce vecteur moyen est ensuite passé à travers une ou plusieurs couches linéaires avec activation ReLU et dropout, avant d'atteindre la couche de sortie qui prédit la classe.

LSTM bidirectionnel (BiLSTM) Le second modèle introduit une structure séquentielle avec un LSTM bidirectionnel [1]. La séquence embeddée est traitée par une couche LSTM capable de modéliser les dépendances à long terme dans les deux directions du texte. Pour obtenir une représentation fixe de l'article, nous concaténons les derniers états cachés de la lecture avant et arrière (de gauche à droite, resp. de droite à gauche). Cette représentation est ensuite transmise à une couche linéaire produisant la prédiction.

BiLSTM avec mécanisme d'attention (BiLSTMATN) Notre troisième modèle reprend l'architecture BiLSTM, à laquelle nous ajoutons un mécanisme d'attention de type *scaled dot-product*, tel qu'introduit dans [2]. Ce mécanisme permet au modèle de pondérer dynamiquement l'importance relative des différentes positions dans la séquence, en fonction de leur contribution potentielle à la tâche de classification. L'état caché final (concaténation des deux directions) est utilisé comme requête Q, tandis que les sorties intermédiaires du LSTM constituent les clés K et les valeurs V. Les scores d'attention sont calculés par :

$$\text{score}_i = \frac{Q \cdot K_i^T}{\sqrt{d_k}} \tag{1}$$

avec d_k la dimension des vecteurs. Ces scores sont normalisés par une fonction softmax pour obtenir les poids d'attention α_i, puis la combinaison pondérée est calculée par :

$$\text{context} = \sum_i \alpha_i V_i \tag{2}$$

La représentation contextuelle ainsi obtenue est passée dans une couche de dropout puis dans une couche linéaire pour obtenir les logits finaux.

4.2.3 Entraînement

Pour l'entraînement des modèles, nous utilisons un `DataLoader` qui permet d'itérer efficacement sur le `ArticleDataset` en mini-batchs (taille 64 ou 128). Les différences de longueur entre les titres ou résumés sont gérées par un padding. Nous construisons une boucle d'entraînement basée sur la descente de gradient stochastique avec l'optimiseur Adam, et la fonction de perte `CrossEntropyLoss`. Cette

boucle est générique, chaque modèle définissant sa propre méthode `forward`. Nous avons également mis en place une validation croisée à k folds avec `GridSearch` ou `RandomSearch` pour la sélection des hyperparamètres.

4.3 Embedding

Chacun des modèles apprend un espace d'embedding à partir des données d'entraînement. La couche `nn.Embedding` correspond à une table de correspondance de taille (`vocab_size`, `emb_dim`) initialisée aléatoirement ou via des poids pré-entraînés, puis optimisée. L'accès à ces vecteurs se fait en temps constant $\mathcal{O}(1)$.

Pour représenter chaque article sous forme d'un vecteur unique, nous utilisons un **average pooling** sur les vecteurs de mots après masquage du padding, ce qui génère un nuage de points dans $\mathbb{R}^{\texttt{emb\_dim}}$. Ainsi, chaque article est le barycentre des vecteurs qui composent son titre.

Nous utilisons ensuite UMAP (*Uniform Manifold Approximation and Projection*) [3] pour projeter ces représentations dans $\mathbb{R}^2$ pour visualisation. Nous utilisons l'implémentation `umap-learn` avec les hyperparamètres : n_neighbors = 15, min_dist = 0.1, métrique euclidienne. UMAP construit un graphe de voisinage local dans l'espace initial ($\mathbb{R}^{\texttt{emb\_dim}}$) et tente de le reconstruire dans l'espace projeté ($\mathbb{R}^2$) en minimisant une perte de type cross-entropy entre distributions de voisinage. UMAP préserve les structures locales et globales, est rapide, et scalable sur des jeux de données volumineux.

4.4 Graphe

Pour l'analyse du graphe de collaboration scientifique, nous avons restreint l'ensemble initial de 1,2 millions d'articles arXiv aux domaines des mathématiques, économie, computer science et statistiques afin de limiter la taille du dataset à 694 925 articles. Pour créer un fichier CSV utilisable par NetworkX, nous avons extrait, à partir du dataset initial, toutes les paires d'auteurs ayant collaboré. Le CSV final comporte quatre colonnes : auteur1, auteur2, poids (nombre d'articles co-écrits) et catégorie (catégorie la plus fréquente des articles co-écrits).

Le graphe ainsi généré avec NetworkX comporte 581 712 nœuds (auteurs) et 3 195 437 arêtes (collaborations). Des attributs spécifiques ont été ajoutés aux nœuds et aux arêtes. Chaque nœud possède un nom, une **catégorie primaire**, ainsi qu'un identifiant de **communauté** obtenu par l'algorithme de **Louvain** à une résolution de 1.0, résultant en 30 890 communautés distinctes. Chaque arête dispose d'un poids indiquant le nombre de collaborations et d'une catégorie dominante.

Afin d'évaluer si notre réseau constitue un « **petit monde** », nous avons regardé la structure de **la plus grande composante connexe** parce que la complexité du coefficient de clustering moyen et de la longueur moyenne des plus courts chemins était trop élevée pour tourner en local. Le **degré de séparation** du réseau a ensuite été déterminé par l'algorithme du **plus court chemin**. Enfin, **Pyvis** a été utilisé pour une visualisation interactive du réseau, facilitant l'exploration des communautés et des collaborations.

5 Analyse et interprétation des résultats

5.1 Classification

Pour déterminer s'il est possible de prédire automatiquement la catégorie ou la sous-catégorie d'un article à partir de son titre ou de son résumé, nous avons testé toutes les combinaisons possibles entre type d'entrée, niveau de granularité et architecture. L'évaluation repose sur l'accuracy et le F1-score macro, ce dernier étant particulièrement adapté à notre problème en raison du déséquilibre entre classes. Afin de limiter le coût computationnel, les résultats reportés ne proviennent pas d'une validation croisée, sauf pour le MLP, sur lequel une cross-validation à 5 folds a permis de sélectionner les hyperparamètres optimisant l'accuracy. Ces mêmes hyperparamètres ont ensuite été utilisés pour les modèles séquentiels. Pour la classification fine, nous avons restreint l'analyse aux sous-catégories mathématiques, suffisamment riches pour évaluer la difficulté de la tâche, tout en restant plus léger en termes de calcul.

Entrée → Niveau	MLP (Acc / F1)	BiLSTM (Acc / F1)	BiLSTMATN (Acc / F1)
Titre → Catégorie	0.76 / 0.59	0.76 / 0.58	**0.77 / 0.62**
Résumé → Catégorie	0.82 / 0.69	**0.82 / 0.70**	0.82 / 0.70
Titre → Sous-cat. maths	0.61 / 0.57	**0.63 / 0.59**	0.62 / 0.59
Résumé → Sous-cat. maths	0.69 / 0.65	**0.72 / 0.68**	0.71 / 0.68

TABLE 1 – Performances des modèles selon l'entrée et le niveau de classification

Les résultats sont synthétisés dans le tableau suivant. Ils confirment que le résumé est systématiquement plus informatif que le titre, avec un gain de 6 à 10 points en F1-macro selon les cas. La prédiction des sous-catégories s'avère plus difficile que celle des grandes catégories, en raison notamment des recouvrements lexicaux entre certaines disciplines proches, comme par exemple "math.CA" (analyse classique et ODEs) et "math.FA" (analyse fonctionnelle). L'ajout d'un mécanisme d'attention ne permet pas d'amélioration globale des performances, mais il apporte un bénéfice net et régulier pour les classes sous-représentées. En mettant davantage en valeur les mots rares ou discriminants, il renforce la détection de signaux faibles dans les résumés longs. Enfin, l'utilisation d'un average pooling pour agréger les représentations de mots ne semble pas entraîner de perte d'information majeure : les résultats obtenus par le MLP montrent qu'une telle agrégation suffit, en particulier sur les titres, pour capter les tendances sémantiques dominantes.

L'analyse des erreurs met en évidence plusieurs motifs récurrents. Certaines disciplines voisines sont régulièrement confondues, comme "math" et "math-ph" (physique mathématiques), dont les résumés partagent un lexique presque identique. Le recall de "math-ph" est de 0.01 ce qui indique que 99% des articles de cette catégories sont classés autre part, bien souvent en mathématiques. Les titres très génériques — par exemple "New approach to multi-agent systems" — entraînent des prédictions incertaines, en raison d'un manque de contexte. Les articles interdisciplinaires ou à portée très générale posent également des difficultés, de même que les sous-catégories très peu représentées dans les données d'entraînement, telles que "HO" (History and Overview) avec un F1 de 0.15 ou "GM" (General Mathematics) 0.35, qui souffrent à la fois d'un faible effectif et d'un vocabulaire trop peu discriminant.

5.2 Exploration des espaces d'embedding

L'analyse de l'espace des mots révèle des regroupements thématiques nets. Certains clusters correspondent à des disciplines bien identifiées, d'autres à des structures linguistiques peu informatives pour la classification mais importantes pour le langage. La figure 1(a) montre que ces thématiques sont spatialement bien séparées dans la projection UMAP. Pour rendre compte de cette diversité, nous résumons dans le tableau 2 quelques exemples de clusters particulièrement cohérents.

Cluster	Thématique	Mots caractéristiques
0	ML / algorithmique	unveiling, compression, overfitting, algorithms, generalization
1	Physique expérimentale	vibrational, mechanics, photonic, electrons, ultrasound
2	Mathématiques pures	moduli, operator, topology, asymptotics, convex
4	Mots syntaxiques	in, the, of, by, at, to
7	Physique statistique	quasiperiodic, disorder, glass, localization, hubbard
12	Deep learning	transformer, generative, interpretability, neural, training

TABLE 2 – Extraits de clusters représentatifs dans l'espace des mots.

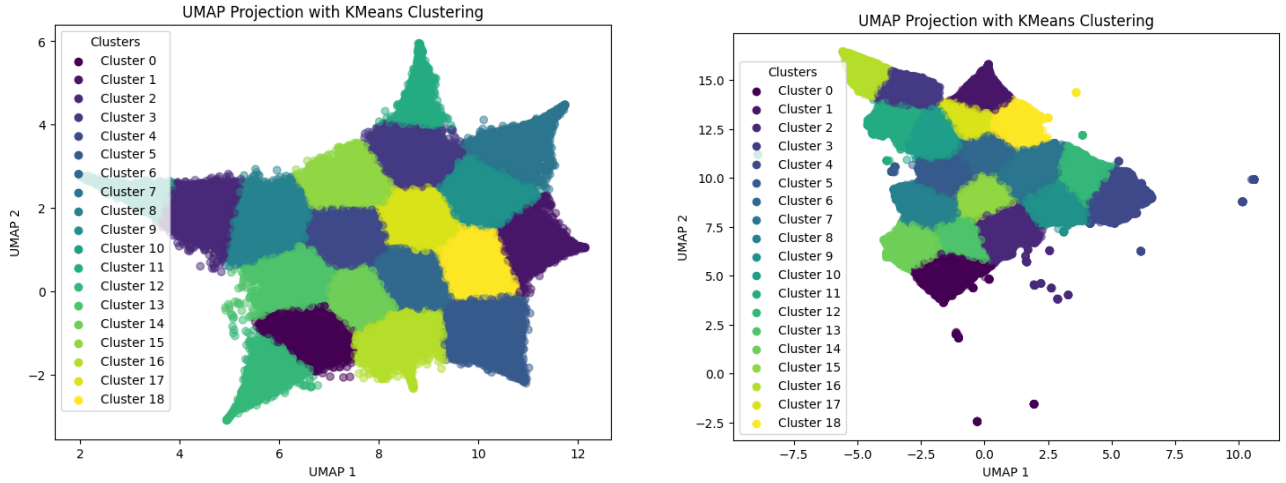
L'organisation de l'espace des mots, révélée par UMAP et confirmée par le clustering, met en évidence une structuration sémantique fine et cohérente. Les mots ayant des proximités thématiques fortes — qu'ils relèvent de la physique, des mathématiques, du machine learning ou encore du langage courant — se retrouvent regroupés spatialement. Aussi, certains mots font le pont entre différentes classes comme "information" qui est à la frontière entre mathématiques, physique et économie. Cela reflète une forme de continuité spatial dans cet espace. L'espace vectoriel ainsi appris ne se limite pas à encoder des formes superficielles, mais traduit une réelle compréhension du langage, où la proximité dans l'espace reflète une affinité sémantique ou fonctionnelle.

Cette propriété est d'autant plus précieuse que ces embeddings servent de base à la représentation

des articles eux-mêmes. La qualité de l'espace des mots conditionne donc directement la qualité des représentations d'articles, et par conséquent, l'efficacité des modèles de classification en aval. Le fait que des groupes disciplinaires se dégagent aussi nettement dans cet espace valide empiriquement sa pertinence pour la tâche de classification supervisée.

Notons que les clusters obtenus dans l'espace des titres ne coïncident pas avec ceux de l'espace des mots. Le cluster 0 des mots, par exemple, regroupe principalement des termes liés à l'apprentissage automatique et à l'algorithmique, tandis que le cluster 0 des titres rassemble des articles à l'intersection de la chimie quantique et du deep learning, comme en témoignent les titres *Efficient optimization of neural network backflow for ab-initio quantum chemistry* et *Deep learning-based holography for T-linear resistivity*. Mais, bien que les numéros de clusters ne soient pas comparables d'un espace à l'autre, les structures émergentes dans l'espace des titres traduisent également une organisation sémantique robuste.

L'agrégation des vecteurs de mots en représentations d'articles ne se contente pas de lisser l'information : elle préserve des signaux thématiques suffisamment riches pour faire émerger des proximités conceptuelles entre articles traitant de sujets similaires. Les clusters obtenus dans l'espace des titres traduisent une structuration sémantique cohérente avec les grandes familles disciplinaires d'arXiv. Toutefois, une analyse plus fine de la *distribution réelle des catégories arXiv* à l'intérieur de ces clusters révèle un décalage non négligeable : les regroupements issus du clustering non supervisé, bien que souvent homogènes sur le plan lexical, ne recouvrent pas les frontières officielles de classification. Un même cluster peut ainsi rassembler des articles de plusieurs disciplines, reflétant une organisation conceptuelle plus fluide et interdisciplinaire, là où la taxonomie d'arXiv repose sur un découpage plus rigide.



(a) Espace UMAP des mots (embedding).

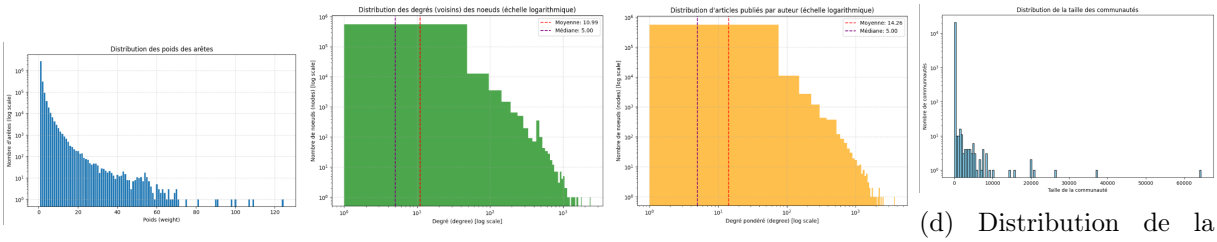
(b) Espace UMAP des titres (moyenne des embeddings).

FIGURE 1 – Visualisation des espaces vectoriels projetés en 2D par UMAP, segmentés en 19 clusters par K-means.

5.3 Graphe

L'analyse du réseau de co-auteurs révèle une structure fortement dispersée et **dominée par des liens faibles**.

La **distribution des poids des arêtes** 2(a)confirme cette impression, avec une écrasante majorité d'arêtes de poids 1 ou 2 (1.5 millions sur les 3 millions totales), indiquant que la moitié des auteurs ne co-publient ensemble qu'une ou deux fois. **La distribution du nombre de voisins** 2(b) par nœud, reflète que 93% des auteurs ont très peu de collaborations (entre 1 et quelques dizaines de voisins) avec une médiane à 5 collaborations. Par contre, les auteurs de **data science** sont ceux qui collaborent le plus fréquemment entre eux. Puis la **distribution des degrés pondérés** 2(b), qui quantifie l'intensité totale de collaboration des auteurs, est très asymétrique : environ le 90% des auteurs ont très peu publié (degré <10), tandis qu'un petit nombre de **hubs** présentent un degré extrêmement élevé, rendant le réseau sensible à la suppression ciblée de ces acteurs centraux.



(a) Distribution des poids des arêtes

(b) Distribution du nombre de voisins

(c) distribution des degrés pondérés

(d) Distribution de la taille des communautés crées par Louvain

FIGURE 2

L'application de l'algorithme de Louvain pour détecter les communautés montre que le réseau est **constitué principalement de micro-communautés** (moins de 10 auteurs) 2(d). Comme la moitié des auteurs dans ces communautés ont publié plus d'une fois avec leur voisins et le 98% de celles-ci sont denses (densité >0.99), on en déduit que cela correspond à des équipes fermées avec peu d'ouvertures vers l'extérieur. **Deux grandes communautés regroupent à elles seules près d'un quart du réseau**. Leur composition, avec un poids moyen des arêtes à 1.2 et 1.4 respectivement suggère plutôt des ensembles de co-auteurs n'ayant collaboré que ponctuellement : par exemple, des doctorants en début de carrière ayant participé à quelques publications, ou des co-auteurs d'articles très collaboratifs avec plusieurs dizaines de signataires n'ayant que peu de liens réels entre eux.

D'autre part, l'analyse des communautés montre, sans surprise, que les auteurs appartenant à une même communauté partagent généralement la **même discipline, la même nationalité et la même affiliation universitaire**. Lorsque les communautés mélangent plusieurs universités, celles-ci partagent souvent la même nationalité, tandis que les communautés multinationales regroupent généralement des auteurs issus d'une même institution, confirmant une logique claire dans la structuration des collaborations scientifiques.

Enfin, la **composante géante**, i.e. la plus grande sous-partie du graphe dans laquelle chaque paire de nœuds est reliée par au moins un chemin, comporte 87,26% des noeuds totaux avec un poids moyen des arêtes de 1.304. La prépondérance des liens faibles joue un rôle crucial dans l'existence d'une éventuelle structure de **"petit monde"**. De plus, à partir de n'importe quel auteur, on atteint presque tous les autres en moins de 10 étapes, avec une croissance rapide autour des degrés 4 à 7. Cette structure sigmoïde, stable sur plusieurs profils, confirme un **degré de séparation maximal autour de 10**. Cela suggère que la plus grande composante connexe du graphe semble présenter les caractéristiques attendues d'un tel réseau, combinant une forte cohésion locale avec des distances courtes entre auteurs éloignés.

6 Conclusion

Nos résultats montrent qu'il est possible de prédire correctement la catégorie d'un article arXiv à partir de son résumé, avec des performances satisfaisantes, en particulier pour les grandes catégories. Les sous-catégories sont plus difficiles à distinguer, en raison de leur proximité lexicale et de la présence d'articles interdisciplinaires. L'analyse des erreurs met en évidence les limites du système actuel de classification, qui ne reflète pas toujours la réalité des contenus.

Du côté des embeddings, nous avons observé que les représentations vectorielles apprises permettent de regrouper les mots et les articles selon leur signification. Ces regroupements sont souvent cohérents avec les thématiques scientifiques, mais ne recouvrent pas exactement les catégories arXiv. Cela suggère que l'organisation naturelle des textes repose sur des continuités sémantiques, plus souples que les catégories définies par arXiv. Nos résultats encouragent donc à envisager d'autres formes de structuration des connaissances scientifiques, basées sur la similarité réelle des contenus.

Le réseau de co-auteurs d'arxiv a une structure fragmentée, dominée par les collaborations faibles et ponctuelles, souvent limitées à une ou deux publications. Cela reflète la nature même d'arXiv, une bibliothèque digitale libre d'accès où les chercheurs — notamment en mathématiques et en machine learning — diffusent rapidement des prépublications pour accroître leur visibilité ou recueillir des retours. Dans des domaines à évolution rapide comme le machine learning, cette diffusion immédiate

permet de contourner les délais des revues traditionnelles, souvent trop longs. Certaines publications y restent temporairement, disparaissant une fois acceptées en revue. Ce fonctionnement explique la structure de « petit monde » propice à la circulation rapide des idées.

Références

[1] Sepp Hochreiter and Jürgen Schmidhuber.
Long short-term memory.
Neural computation, 9(8) :1735–1780, 1997. https://www.bioinf.jku.at/publications/older/2604.pdf

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin.
Attention is all you need.
Advances in neural information processing systems, 30, 2017.

[3] Leland McInnes, John Healy, James Melville.
UMAP : Uniform Manifold Approximation and Projection for Dimension Reduction.
https://arxiv.org/abs/1802.03426, 2018.