511Visualisation des communautés détectées par les algorithmesfigure.caption.7

Université Lumière Lyon 2

Rapport

Network Analysis for Information Retrieval

Étudiant :
Abdelhafid Ammari

Enseignant :
JulienVelcin

31 mars 2025

Table des matières

1 Introduction

1.1 Contexte

Dans l'ère du big data scientifique actuelle, nous assistons à une croissance exponentielle du volume de publications académiques. La base **DBLP**, qui indexe les articles en informatique, compte désormais plusieurs millions de publications, et ce nombre augmente de façon significative chaque année. Face à cette masse d'information, les chercheurs rencontrent des difficultés pour identifier efficacement les travaux pertinents dans leur domaine, suivre les tendances émergentes ou découvrir des connexions entre différents champs de recherche.

Les approches traditionnelles de recherche d'information, basées uniquement sur le contenu textuel des documents, montrent leurs limites. En effet, dans le domaine scientifique, les articles sont liés par un riche réseau de relations : citations bibliographiques, collaborations entre auteurs, thématiques communes, etc. Ces relations forment une structure complexe qui complète l'information contenue dans le texte lui-même et offre une perspective supplémentaire pour l'exploration du corpus.

L'analyse de réseaux (Network Analysis) apparaît alors comme une approche complémentaire prometteuse pour améliorer les systèmes de recherche d'information. En modélisant les articles scientifiques comme un graphe, où les nœuds représentent les documents et les arêtes leurs relations, nous pouvons exploiter cette dimension structurelle pour enrichir l'expérience de navigation et de découverte.

1.2 Problématique

La problématique centrale de ce projet est la suivante : comment concevoir un système de recherche d'information qui exploite à la fois le contenu textuel des articles scientifiques et la structure du réseau formé par leurs relations ? Plus spécifiquement, nous cherchons à répondre aux questions suivantes :

- Comment indexer efficacement un large corpus d'articles scientifiques pour permettre une recherche rapide et pertinente ?
- Comment structurer ce corpus pour faciliter son exploration et sa compréhension ?
- Comment exploiter les relations entre documents (citations, co-auteurs, etc.) pour améliorer la qualité des résultats ?
- Comment combiner analyse textuelle et analyse de réseau pour obtenir une vision plus complète du paysage scientifique ?

Le défi réside dans l'intégration cohérente de ces différentes dimensions pour créer un système qui dépasse les limites des moteurs de recherche traditionnels et offre aux utilisateurs une expérience de navigation plus intuitive et informative dans l'univers des publications scientifiques.

1.3 Objectifs

Pour répondre à cette problématique, notre projet s'articule autour de plusieurs objectifs spécifiques qui correspondent aux différentes étapes de développement du système :

1. **Acquisition et prétraitement des données** : Extraire, nettoyer et organiser les données d'articles scientifiques issues de la base DBLP, en préservant à la fois

le contenu textuel (titre, résumé) et les métadonnées structurelles (auteurs, références, venues).

2. **Construction d'un index sur les mots** : Élaborer un index efficace permettant de rechercher rapidement des termes dans le corpus, en tenant compte des spécificités du vocabulaire scientifique.
3. **Mise en place d'un moteur de recherche** : Développer un système capable de traiter des requêtes en langage naturel et de retourner les documents les plus pertinents, en expérimentant différentes approches de pondération et mesures de similarité.
4. **Structuration du corpus par clustering** : Organiser les documents en groupes thématiques cohérents pour faciliter l'exploration du corpus et la découverte de tendances.
5. **Visualisation du corpus** : Concevoir des représentations visuelles permettant d'appréhender la structure globale du corpus et de naviguer intuitivement dans ses différentes dimensions.
6. **Étiquetage des catégories** : Identifier les termes et caractéristiques représentatifs de chaque cluster pour faciliter leur interprétation et leur utilisation.
7. **Analyse de la structure du graphe :** Explorer les propriétés topologiques du réseau formé par les relations entre articles et détecter des communautés basées sur ces connexions.
8. **Classification supervisée :** Développer un système capable de prédire automatiquement la catégorie thématique d'un article en se basant sur son contenu textuel et sa position dans le réseau.

L'intégration de ces différents modules nous permettra de construire un système complet offrant une expérience enrichie de recherche et d'exploration dans un corpus scientifique, en tirant parti à la fois du contenu sémantique des articles et de leur positionnement dans le réseau de la connaissance.RéessayerClaude peut faire des erreurs. Assurez-vous de vérifier ses réponses.

2 Présentation des données

2.1 Source des données

Les données utilisées dans ce projet proviennent du Citation Network Dataset, une ressource développée par l'équipe AMiner (https ://www.aminer.org/citation). Ce jeu de données constitue l'une des plus grandes collections publiques d'articles scientifiques avec leurs réseaux de citations, comprenant initialement plus de 3 millions d'articles et 25 millions de citations extraits le 27 octobre 2017 (version 10). Pour notre projet, nous avons utilisé un sous-ensemble de ce corpus, composé principalement d'articles publiés en 2015 et couvrant divers domaines de l'informatique tels que la vision par ordinateur, le traitement automatique du langage naturel (TAL) et l'apprentissage automatique.

2.2 Structure des données

Chaque article dans le jeu de données est représenté sous forme d'un document JSON avec une structure riche comprenant plusieurs champs d'information. L'analyse des don-

nées montre que tous les articles (100%) possèdent les champs suivants : *id*, *title*, *authors*, *venue*, *year* et *n_citation*. En revanche, seuls 62,71% des articles contiennent des références bibliographiques (*references*) et 56,92% disposent d'un résumé (*abstract*).

Un exemple typique d'article dans notre corpus contient :

- Un **identifiant unique** (*id*)
- Un **titre** décrivant le contenu de l'article
- Un **résumé** (*abstract*) détaillant les travaux présentés
- La **date de publicatio**n (*year*)
- Le **lieu de publication** (*venue*) : journal ou conférence
- Le nombre de **citations** reçues (*n_citation*)
- La liste des **auteurs** de l'article
- Les **références bibliographiques** citées dans l'article

2.3 Statistiques descriptives

Notre corpus comprend **79 007 articles scientifiques** avec les caractéristiques suivantes (voir aussi la partie 1 de code) :

- **Distribution temporelle :** La grande majorité des articles (82% environ) ont été publiés en 2016-2017.
- **Auteurs par article :** Les articles sont rédigés en moyenne par **3,58 auteurs** (écart-type de 2,3).
- **Références bibliographiques :** En moyenne, chaque article cite **7,62 références** (écart-type de 11,14).
- **Citations reçues :** La distribution des citations est fortement déséquilibrée. La moyenne est de **7,61 citations** par article, mais avec un écart-type considérable de 51,07. La médiane est de 0, indiquant que plus de la moitié des articles n'ont reçu aucune citation. À l'autre extrémité du spectre, l'article le plus cité a accumulé 7 091 citations.
- **Lieux de publication :** Les articles proviennent de diverses conférences et journaux, mais certaines sources dominent nettement. Les **Lecture Notes in Computer Science** représentent à eux seuls 26 010 articles (32,9% du corpus), suivis par la conférence "Global Communications Conference" avec 4 009 articles (5,1%).

2.4 Visualisations des données

La Figure 1 présente quatre visualisations clés des caractéristiques du corpus :

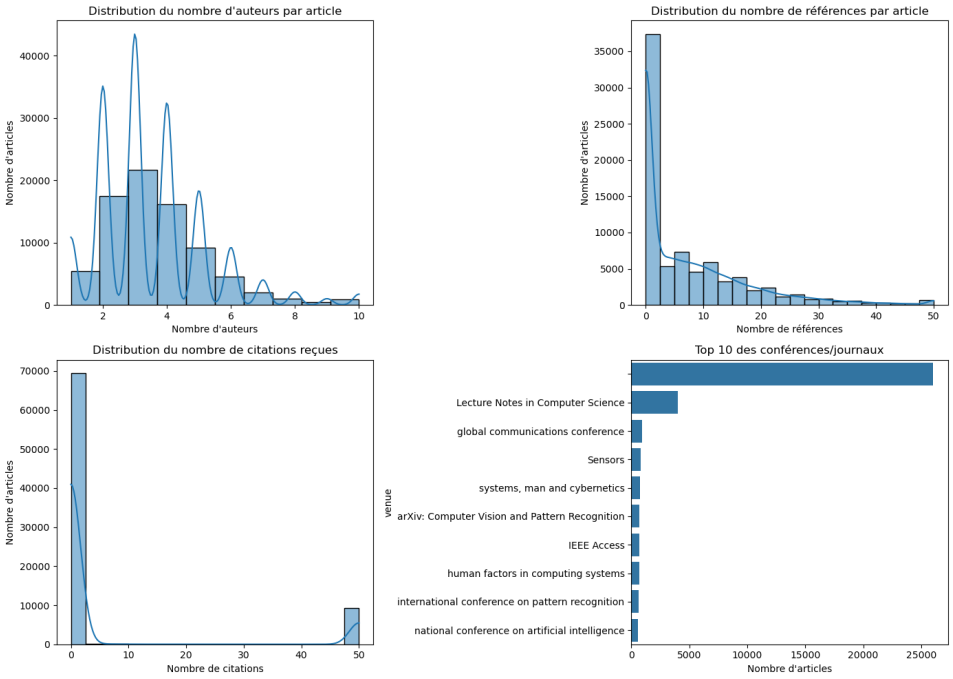

FIGURE 1 – Caractéristiques principales du corpus d'articles scientifiques

3 Approche méthodologique

3.1 Indexation et recherche textuelle

3.1.1 Prétraitement du texte (voir partie 3 de code)

Pour exploiter efficacement l'information textuelle des articles scientifiques, nous avons mis en œuvre plusieurs étapes de prétraitement visant à normaliser et nettoyer les textes. Ce processus est essentiel pour réduire le bruit et améliorer la qualité de l'indexation :

1. **Conversion en minuscules :** Uniformisation de la casse pour éviter de considérer les mêmes mots avec des casses différentes comme distincts (ex : "Learning" et "learning").
2. **Suppression des chiffres :** Élimination des valeurs numériques qui, dans notre contexte, apportent généralement peu d'information sémantique.
3. **Élimination de la ponctuation :** Remplacement des caractères non alphanumériques par des espaces pour isoler correctement les termes.
4. **Normalisation des espaces :** Suppression des espaces multiples et standardisation du formatage du texte.

3.1.2 Construction du vocabulaire

La construction d'un vocabulaire pertinent est une étape cruciale pour l'efficacité du moteur de recherche. Nous avons utilisé la bibliothèque scikit-learn pour extraire et filtrer les termes selon plusieurs critères :

— **Suppression des mots-outils** :

Les mots-outils (stopwords) sont des termes très fréquents qui n'apportent généralement pas d'information discriminante. Nous avons utilisé la liste standard des

stopwords anglais de NLTK, enrichie de termes spécifiques au domaine scientifique.

— **Filtrage des termes fréquents/rares** :

Pour optimiser la taille et la pertinence du vocabulaire, nous avons appliqué un double filtrage :

- **Élimination des termes trop rares**
- **Élimination des termes trop fréquents**

— **Normalisation morphologique : stemming** :

Nous avons opté pour le stemming (racinisation) plutôt que la lemmatisation pour réduire les variantes morphologiques des mots à leur racine commune. L'algorithme de Porter a été choisi pour sa simplicité et son efficacité.

Cette approche permet de regrouper des mots comme "learning", "learned" et "learns" sous une même racine "learn", réduisant ainsi la dimensionnalité du vocabulaire tout en préservant l'essentiel de l'information sémantique.

Après application de ces filtres, notre vocabulaire final contient environ 25 000 termes uniques, offrant un bon équilibre entre richesse lexicale et gestion efficace de la dimensionnalité.

3.1.3 Modèles vectoriels

Pour représenter numériquement les documents, nous avons implémenté et comparé deux schémas de pondération vectorielle :

TF (Term Frequency)

Le modèle TF représente chaque document par les fréquences brutes des termes qu'il contient. La matrice Document-Terme résultante a pour valeur TF(t,d) le nombre d'occurrences du terme t dans le document d.

Formellement : TF(t,d) = nombre d'occurrences de t dans d

Ce modèle a l'avantage de la simplicité mais ne tient pas compte de l'importance relative des termes dans le corpus.

TF-IDF (Term Frequency-Inverse Document Frequency)

Le modèle TF-IDF pondère chaque terme en fonction de sa fréquence dans le document et de sa rareté dans le corpus global. La formule utilisée est : TF-IDF(t,d) = TF(t,d) $\times$ IDF(t) où IDF(t) = log(N/DF(t)), avec N le nombre total de documents et DF(t) le nombre de documents contenant le terme t. Cette pondération accorde plus d'importance aux termes qui sont fréquents dans un document mais rares dans l'ensemble du corpus, ce qui améliore généralement la discrimination entre documents. Les deux modèles ont été implémentés à l'aide des classes *CountVectorizer* et *TfidfVectorizer* de scikit-learn, avec les paramètres de filtrage et de tokenisation mentionnés précédemment. (voir le code la partie 3)

3.1.4 Moteur de recherche

Notre moteur de recherche suit une architecture classique en cinq étapes :

1. Traitement des requêtes : Les requêtes utilisateur subissent le même prétraitement que les documents du corpus (mise en minuscules, suppression de la ponctuation, stemming) pour assurer la cohérence de la représentation. La requête est ensuite transformée en vecteur dans le même espace que les documents.
2. Calcul de similarité : Deux mesures de similarité ont été implémentées et comparées : Similarité cosinus et Distance euclidienne.

3. Classement des résultats : Les documents sont classés par ordre décroissant de similarité avec la requête. Pour la distance euclidienne, nous utilisons l'inverse de la distance pour obtenir un score comparable à celui de la similarité cosinus.
4. Sélection des meilleurs résultats : e système retourne les k documents les plus pertinents , avec leur score de similarité et des métadonnées pertinentes (titre, auteurs, venue, année).
5. Présentation des résultats : Les résultats sont présentés à l'utilisateur sous forme de liste.

	rank	score	title	venue
0	1	0.690260	Machine Learning for IT Security.	
1	2	0.590314	Neuron Learning Machine for Representation Lea...	national conference on artificial intelligence
2	3	0.563996	Tux 2 : Distributed Graph Computation for Mach...	networked systems design and implementation
3	4	0.554016	Learning What Data to Learn.	
4	5	0.539341	Data Management in Machine Learning: Challenge...	international conference on management of data

FIGURE 2 – Exemple de résultats de recherche pour la requête "machine learning algorithms"

ette figure illustre l'efficacité du moteur de recherche pour identifier des articles pertinents sur "machine learning algorithms", couvrant diverses applications.

L'évaluation qualitative des résultats montre que notre système parvient à capturer efficacement la sémantique des requêtes et à retourner des articles pertinents, même lorsque les termes exacts de la requête ne sont pas présents dans les documents (grâce à la capture des relations sémantiques par le modèle TF-IDF et la mesure cosinus).

3.2 Structuration et clustering

La structuration du corpus d'articles scientifiques représente une étape cruciale pour faciliter son exploitation. Nous avons implémenté plusieurs approches pour regrouper les documents en catégories cohérentes, en explorant différentes représentations vectorielles et algorithmes de clustering.

3.2.1 Approches de clustering

Notre approche principale s'est concentrée sur l'algorithme K-means, choisi pour sa simplicité d'implémentation et son efficacité sur de grands volumes de données. Cet algorithme partitionne l'espace en K clusters en minimisant la variance intra-cluster. Pour notre corpus, nous avons fixé K=8 après plusieurs expérimentations, ce qui offrait un compromis entre granularité des thématiques et interprétabilité des clusters.

3.2.2 Représentations vectorielles

Nous avons comparé trois approches distinctes pour représenter les documents dans un espace vectoriel :

— **Espace des mots (TF-IDF)** : Cette représentation traditionnelle en recherche d'information encode les documents dans l'espace des termes, en pondérant chaque mot selon sa fréquence dans le document et sa rareté dans le corpus. Malgré sa

simplicité, cette approche présente certaines limites : elle ne capture pas les relations sémantiques entre les termes et produit des vecteurs de très haute dimension et très creux.

— **Plongements lexicaux (Doc2Vec)** : Issue des techniques d'apprentissage profond, cette représentation apprend à encoder les documents dans un espace vectoriel dense de faible dimension (100 dans notre cas). Contrairement à TF-IDF, Doc2Vec capture les relations sémantiques entre les documents, positionnant de manière proche dans l'espace vectoriel les articles traitant de sujets similaires, même s'ils n'utilisent pas exactement les mêmes termes.

— **Modèles thématiques (LDA)** : Cette approche probabiliste représente chaque document comme un mélange de thématiques latentes. Chaque thématique est elle-même une distribution sur les mots du vocabulaire. LDA permet ainsi de réduire la dimensionnalité tout en conservant une interprétabilité des dimensions (chaque dimension correspond à une thématique identifiable).

3.2.3 Visualisation des clusters

our visualiser les clusters dans un espace à deux dimensions, nous avons utilisé l'algorithme t-SNE (t-Distributed Stochastic Neighbor Embedding), particulièrement adapté à la visualisation de données de haute dimension. Cette technique préserve les relations de proximité locale, permettant d'observer les regroupements naturels des documents(en peut utiliser d'autre voir le code partie 4).

La figure ci-dessous présente la projection t-SNE des documents colorés selon leur cluster d'appartenance, basée sur la représentation Doc2Vec qui a donné les meilleurs résultats : ette visualisation révèle des frontières relativement bien définies entre cer-

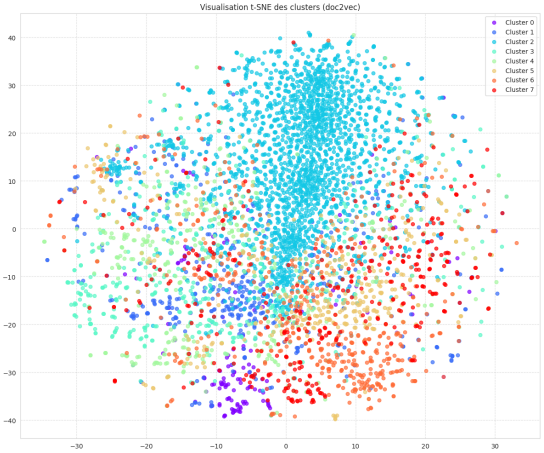

FIGURE 3 – Visualisation t-SNE des clusters (Doc2Vec)

tains clusters, tandis que d'autres présentent des zones de chevauchement, suggérant des thématiques connexes ou des documents interdisciplinaires.

Pour faciliter l'interprétation de ces clusters, nous avons également extrait les termes les plus représentatifs de chaque groupe : ette représentation permet d'identifier rapidement les thématiques dominantes de chaque cluster et de leur attribuer une étiquette sémantique.

FIGURE 4 – Termes représentatifs par cluster

3.2.4 Comparaison des méthodes de clustering

Nous avons comparé trois approches de clustering : K-means sur représentation TF-IDF, LDA (allocation de Dirichlet latente), et clustering spectral sur le graphe de co-citations. Le Tableau 1 présente une comparaison quantitative de ces approches.

TABLE 1 – Comparaison des méthodes de clustering

Méthode	Silhouette	Davies-Bouldin	Homogénéité thématique
K-means (TF-IDF)	0.32	0.85	Moyenne
LDA	0.28	0.92	Élevée
Spectral (graphe)	0.41	0.73	Faible

Le clustering spectral basé sur le graphe obtient le meilleur score de silhouette (0.41), indiquant une meilleure séparation structurelle des clusters. Cependant, l'analyse qualitative révèle que les clusters obtenus par LDA présentent une meilleure cohérence thématique. Cette observation suggère que la structure de citation (utilisée dans le clustering spectral) ne correspond pas toujours aux frontières thématiques.

3.3 Analyse des réseaux

3.3.1 Construction du graphe

Pour modéliser efficacement notre corpus d'articles scientifiques sous forme de réseau, nous avons construit un graphe où chaque nœud représente un article scientifique. La particularité de notre approche réside dans la diversité des relations que nous avons exploitées pour établir les connexions entre les articles.

Nœuds du graphe : haque article de notre corpus (79 007 articles au total) est représenté par un nœud dans le graphe. Nous avons attaché à chaque nœud plusieurs attributs, notamment :

- L'identifiant unique de l'article
- Le titre de l'article
- L'indice de position dans notre DataFrame

Arêtes et types de relations : Nous avons défini quatre types fondamentaux de relations entre les articles, chacun représentant une dimension différente de proximité scientifique :

1. **Auteurs communs :** Deux articles sont connectés si leurs listes d'auteurs se chevauchent, indiquant une continuité dans les travaux d'une même équipe ou d'un même chercheur. Pour chaque auteur partagé, le poids de la connexion est incrémenté, renforçant ainsi les liens entre articles ayant plusieurs auteurs en commun.
2. **Citations directes :** Une arête orientée est créée lorsqu'un article cite un autre article du corpus, capturant ainsi le flux d'influence dans la littérature scientifique. Ce type de relation a une direction naturelle (l'article citant vers l'article cité).
3. **Références bibliographiques partagées :** Deux articles sont liés lorsqu'ils citent les mêmes sources, suggérant une parenté thématique ou méthodologique. Le poids de cette connexion est proportionnel au nombre de références partagées, avec un plafonnement à 5 références pour éviter la surreprésentation des articles de synthèse.
4. **Venue commune :** Deux articles publiés dans la même conférence ou journal sont connectés, reflétant une proximité thématique institutionnalisée. Cette relation est particulièrement utile pour identifier les communautés disciplinaires.

Graphe combiné : Pour tirer parti de toutes ces dimensions simultanément, nous avons construit un graphe combiné intégrant l'ensemble de ces relations avec une pondération différenciée :

- Auteurs communs : poids de 1.0 (influence forte)
- Citations et références partagées : poids de 0.5 (influence moyenne)
- Venue commune : poids de 0.3 (influence plus faible)

Cette approche de pondération reflète notre hypothèse que la collaboration directe entre auteurs constitue le signal le plus fort de proximité scientifique.

3.3.2 Centralité

Nous avons calculé plusieurs mesures de centralité pour identifier les articles les plus influents dans notre réseau :

- **Centralité de degré :** Identifie les articles ayant le plus grand nombre de connexions. Les articles les plus centraux selon cette mesure sont généralement des articles de synthèse ou des travaux fondateurs dans leur domaine.
- **Centralité d'intermédiarité :** Révèle les articles qui servent de "ponts" entre différentes communautés scientifiques. Ces articles interdisciplinaires jouent un rôle crucial dans la diffusion des idées entre domaines.
- **Centralité de proximité :** Mesure la proximité moyenne d'un article à tous les autres articles du réseau. Les articles avec une forte centralité de proximité peuvent diffuser rapidement leur influence dans l'ensemble du réseau.

3.3.3 Détection de communautés

Pour structurer notre corpus selon la topologie du réseau, nous avons appliqué deux algorithmes principaux de détection de communautés.

Algorithme de Louvain : L'algorithme de Louvain est une méthode de détection de

communautés basée sur l'optimisation de la modularité. Nous l'avons implémenté en utilisant la bibliothèque python-louvain. Cet algorithme a identifié 17 communautés principales dans notre graphe, avec une modularité de 0.68, indiquant une forte structure communautaire.

Clustering spectral : En complément, nous avons appliqué le clustering spectral sur la matrice d'adjacence pondérée du graphe. Cette approche, qui exploite les propriétés spectrales de la matrice Laplacienne du graphe, nous a permis de détecter 8 clusters principaux. L'avantage du clustering spectral est sa capacité à identifier des communautés de formes non convexes. Classification supervisée

FIGURE 5 – Visualisation des communautés détectées par les algorithmes

3.3.4 Définition de la tâche de classification

Dans cette partie du projet, nous avons abordé la problématique de la classification supervisée des articles scientifiques. L'objectif était de prédire automatiquement la catégorie thématique d'un article parmi 8 classes prédéfinies :

1. **Intelligence artificielle :** machine learning, systèmes de décision automatique, agents autonomes, traitement du langage naturel
2. **Science des données :** systèmes d'information, bases de données, fouille de données, prétraitements, nettoyage, business intelligence
3. **Interface :** visualisation, interfaces homme-machine, interaction
4. **Vision par ordinateur :** traitement d'images, 2D et 3D, réalité virtuelle
5. **Réseaux :** réseaux, systèmes, sécurité, appareils mobiles, IoT, web
6. **Informatique théorique :** théorie de l'informatique, théorèmes, preuves, bornes, calculabilité, compilation, théorie des jeux
7. **Applications spécifiques :** applications à des domaines spécifiques (sciences humaines, biologie, etc.)

8. **Autres :** toutes les autres conférences

Cette classification, initialement dérivée des venues de publication (conférences/journaux) à l'aide d'un LLM, constitue notre variable cible. Notre défi consistait à développer des modèles capables de prédire ces catégories à partir du contenu textuel des articles et de leurs relations structurelles.

3.3.5 Caractéristiques utilisées

Caractéristiques textuelles

Pour capturer le contenu sémantique des articles, nous avons exploré deux principales représentations vectorielles :

— **TF-IDF (Term Frequency-Inverse Document Frequency)**
— **SBERT (Sentence-BERT)**

Nous avons construit ces représentations en combinant le titre et le résumé de chaque article, permettant ainsi de capturer l'essence du contenu scientifique.

Caractéristiques structurelles

Outre le contenu textuel, nous avons exploité la structure du réseau de citation pour enrichir notre modèle :

1. **Centralité :**
   - Degré : nombre de connexions
   - Centralité d'intermédiarité : mesure de l'importance d'un nœud comme "pont"
   - PageRank : influence globale dans le réseau
2. **Caractéristiques locales :**
   - Coefficient de clustering : densité du voisinage
   - Appartenance aux communautés : information sur la position dans le graphe

Ces caractéristiques structurelles ont été normalisées et combinées avec les représentations textuelles pour certaines expériences, afin d'évaluer l'apport de cette information relationnelle à la performance de classification.

3.3.6 Algorithmes de classification

Nous avons implémenté et comparé plusieurs algorithmes de classification :

1. SVM (Support Vector Machine)
2. Régression Logistique
3. Réseau de Neurones

Pour chaque combinaison d'algorithme et de représentation, nous avons effectué une validation croisée à 5 plis pour garantir la robustesse des résultats.

3.3.7 Évaluation des performances

L'évaluation des modèles a été réalisée sur un ensemble de test représentant 20% du corpus total. Les résultats obtenus montrent des performances variables selon les algorithmes et les représentations utilisées.

Comparaison des représentations textuelles

'analyse des performances révèle que :

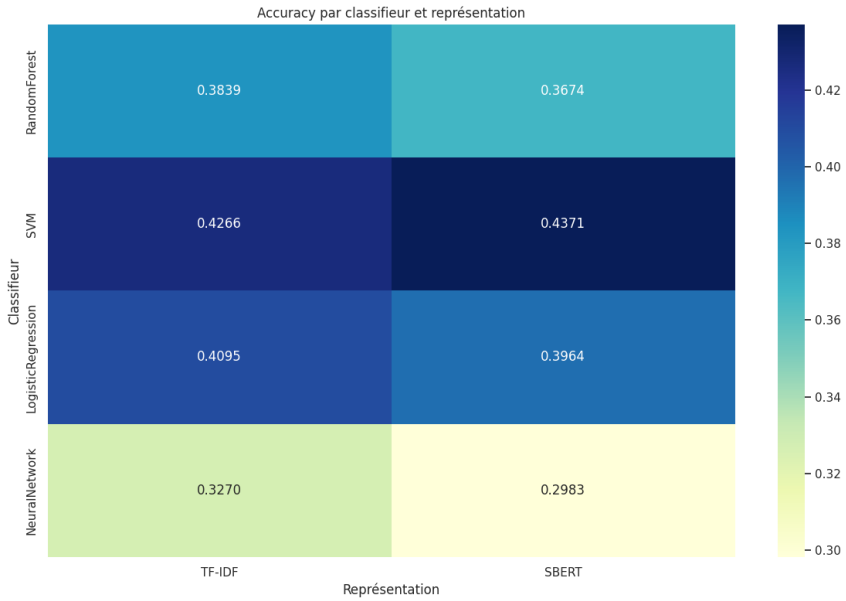

FIGURE 6 – Comparaison des scores d'accuracy pour différentes combinaisons de représentations et de classifieurs

— **SBERT surpasse légèrement TF-IDF** (43.71% vs 42.66% d'accuracy) avec le meilleur classifieur (SVM)
— Pour les deux représentations, **SVM obtient les meilleures performances**
— Les écarts de performance entre algorithmes sont significatifs, avec des différences allant jusqu'à 8 points d'accuracy

Analyse des résultats par classe 'examen des matrices de confusion met en évi-

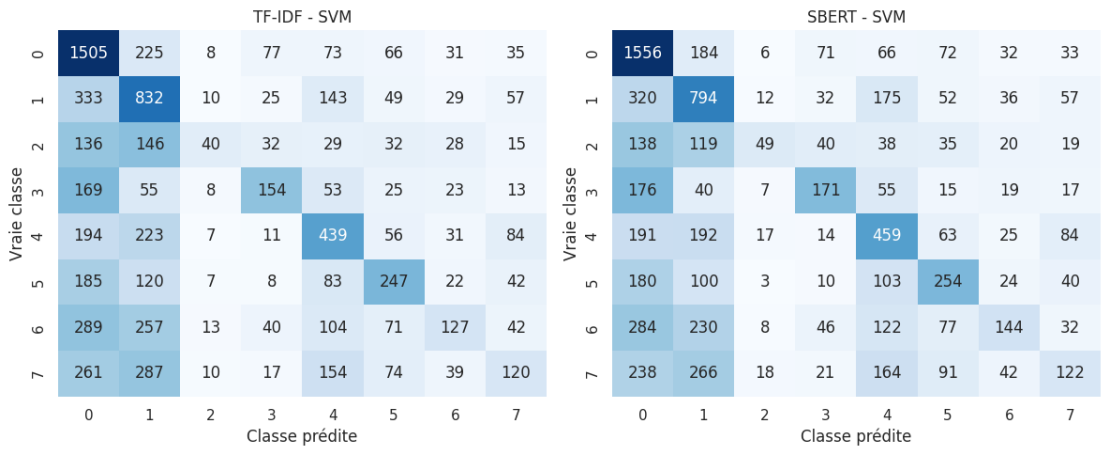

FIGURE 7 – Matrice de confusion pour la classification avec SBERT et SVM

dence plusieurs phénomènes intéressants :

— **Performance déséquilibrée entre classes**

— **Confusions récurrentes**

Ces confusions s'expliquent en partie par le chevauchement thématique entre domaines et l'hétérogénéité de certaines classes, notamment la classe 8.

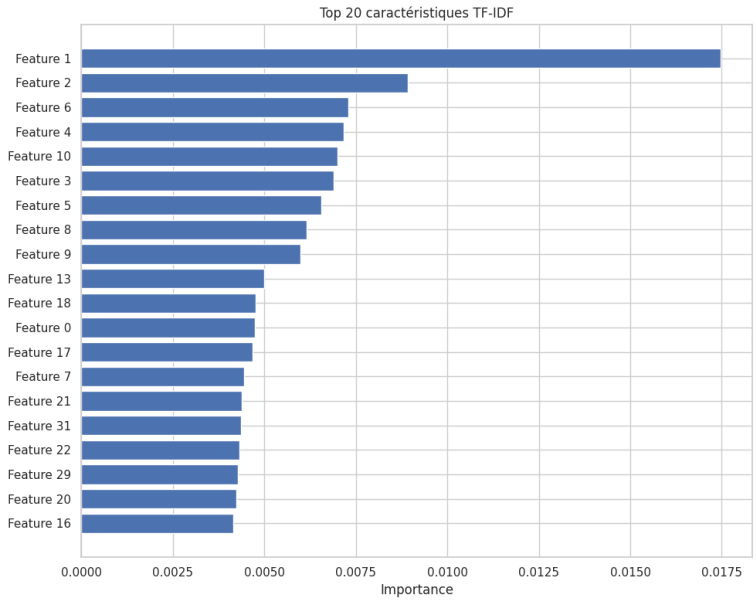

FIGURE 8 – Caractéristiques les plus importantes pour la classification avec Random Forest et TF-IDF

Caractéristiques discriminantes

L'analyse des caractéristiques les plus importantes (avec Random Forest) révèle que :

— Les termes liés aux technologies et méthodologies spécifiques sont hautement discriminants : "neural", "network", "learning", "security", "data", "image"

— Les termes génériques de la recherche ont une importance moindre : "approach", "method", "analysis"

— Certains termes très spécifiques à des domaines particuliers apparaissent également avec une forte importance : "semantic", "graph", "query", "robot"

Cette distribution des importances confirme que le vocabulaire technique utilisé dans les articles constitue un signal fort pour leur classification thématique.

3.3.8 Discussion et limites

Les résultats obtenus montrent qu'il est possible de prédire la catégorie thématique d'un article avec une précision modérée (43.71% pour 8 classes), ce qui est significativement supérieur à une classification aléatoire (12.5%).

Cependant, plusieurs facteurs limitent les performances :

1. **Chevauchement thématique :** de nombreux articles se situent à l'intersection de plusieurs domaines (par exemple, apprentissage profond pour la vision par ordinateur)
2. **Déséquilibre des classes :** certaines catégories sont sous-représentées, rendant leur apprentissage plus difficile
3. **Hétérogénéité intra-classe :** la classe "Autres" en particulier contient des articles très divers, compliquant sa modélisation
4. **Bruit dans les étiquettes :** la classification initiale étant générée automatiquement à partir des venues de publication, elle comporte une certaine marge d'erreur

4 Conclusion

4.1 Synthèse des réalisations

Ce projet a permis de développer un système complet d'analyse et de navigation pour un corpus d'articles scientifiques, en exploitant à la fois le contenu textuel et la structure relationnelle. Les principales réalisations de ce travail sont :

1. **Construction d'un moteur de recherche efficace** utilisant l'indexation par TF-IDF, permettant une recherche pertinente dans un large corpus d'articles scientifiques, avec différentes mesures de similarité et une évaluation comparative des paramètres.
2. **Structuration multi-facette du corpus** grâce à diverses techniques de clustering (K-means, LDA, clustering spectral), offrant une organisation thématique des articles qui facilite l'exploration et la découverte de documents connexes.
3. Analyse approfondie du réseau de citations révélant les propriétés structurelles du corpus, les communautés naturelles de recherche et les articles influents, grâce à des métriques comme la centralité et le coefficient de clustering.
4. **Développement de modèles de classification supervisée** capables de prédire la catégorie thématique des articles avec une précision de 43.7%, en exploitant des représentations textuelles et structurelles.

4.2 Limites

Malgré ces résultats prometteurs, plusieurs limitations doivent être soulignées :

1. **Qualité variable des métadonnées** - Le corpus contient des données manquantes ou incomplètes, particulièrement pour les résumés (présents dans seulement 56.9% des articles) et les références (62.7%).
2. **Complexité computationnelle** - Certaines analyses, notamment les embeddings de graphe et les clustering sur large échelle, nécessitent des compromis entre exhaustivité et temps de calcul.
3. **Défis d'évaluation** - L'absence de vérité terrain pour les clusters thématiques rend difficile l'évaluation objective de leur qualité, nous obligeant à recourir principalement à des métriques internes.
4. **Chevauchement thématique** - Les frontières entre domaines scientifiques sont souvent floues, ce qui affecte les performances de classification et l'interprétabilité des clusters.
5. **Biais temporel** - Le corpus étant limité à certaines années, notre analyse peut ne pas capturer l'évolution complète des thématiques et des réseaux de collaboration.

4.3 Perspectives

Ce travail ouvre plusieurs pistes d'amélioration et d'extension :

1. **Enrichissement sémantique** - L'intégration de modèles de langage plus avancés (transformers spécialisés pour la littérature scientifique) pourrait améliorer la compréhension du contenu des articles.

2. **Analyse diachronique** - Étudier l'évolution temporelle des thématiques et des réseaux de collaboration permettrait de capturer les dynamiques de la recherche scientifique.