



UCLOUVAIN

Mathématiques discrètes I : Théorie et algorithmique des
graphes

LINMA1691

Devoir 4 - Rapport

Étudiant.e.s

AUGUSTIN WEZEL (79592300)
LOUIS-FRÉDÉRIC PETITJEAN (7092301)

Enseignant.e.s

BLONDEL VINCENT
DELOGNE RÉMI
DELVENNE JEAN-CHARLES;

1 Introduction

Nous avons décidé de travailler sur le livre « L'Idiot » de Fiodor Dostoïevski, avec une version anglaise traduite par Eva Martin [1]. Le graphe que nous en avons tiré concerne les personnages, l'auteur étant connu pour le nombre élevé de personnages dans ses romans.

Nous avons ensuite décidé d'y appliquer un algorithme de détection de communauté, plus précisément la variante Leiden de l'algorithme de Louvain. L'objectif est d'étudier la corrélation entre les communautés détectées et les familles de personnages, et ce à plusieurs niveaux de résolution.

2 Méthode

Pour construire notre graphe, nous avons procédé de la manière suivante. Chaque nœud du graphe est un personnage et le poids de l'arête entre deux nœuds est le nombre de fois où les deux personnages apparaissent dans le même paragraphe. Nous avons considéré les dialogues comme un seul paragraphe.

Nous avons ensuite utilisé le logiciel *Gephi* [2] pour traiter nos données, appliquer les algorithmes et visualiser le graphe résultant.

2.1 Extraction du graphe

Nous avons utilisé le script `extract.py`, qui fourni une liste d'arêtes à partir de `book.txt`, le fichier contenant le livre, et `character_info.csv`, une base de données des personnages de l'Idiot que nous avons manuellement constituée. Cela pose une limite d'exhaustivité et de précision sur le graphe considéré dans ce travail. Nous importons ensuite les 2 fichiers `edge_list.csv` et `character_info.csv` dans *Gephi*, qui constitue pour nous le graphe pondéré des interactions entre les personnages.

2.2 Algorithme de Louvain

L'algorithme que nous appliquons au graphe est la méthode de Louvain [3], dans sa variante Leiden [4].

La méthode de Louvain est un algorithme d'optimisation greedy de la modularité d'une partition $c = \{c_1, c_2, \dots c_n\}$ du graphe (c_i est la communauté du noeud i), définie par :

$$Q = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(C_i, C_j)$$

La modularité mesure la qualité de communautés ainsi déterminées. Comme la méthode de Louvain itère sur des niveaux d'aggrégations des noeuds jusqu'à trouver une situation optimale, la variante de Leiden rajoute un coefficient de résolution γ afin d'influencer l'échelle à laquelle la détection des communautés se termine. Plus γ est grand, plus la taille des communautés détectées a tendance à augmenter.

$$Q_{\text{Leiden}} = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \gamma \frac{k_i k_j}{2m} \right) \delta(C_i, C_j)$$

Cette métrique s'appelle le *Reichardt Bornholdt Potts Model* [4].

2.3 Métrique de qualité : ARI

Afin de juger de la qualité des partitions obtenues pour chaque valeur du paramètre γ , nous utilisons la métrique *Adjusted Rand Index* (ARI) [5], qui mesure la similarité entre 2

partitions d'un même ensemble, ici entre les communautés détectées et la partition en familles des personnages.

2.4 Visualisation

Finalement, afin de visualiser les résultats, nous avons utilisé l'algorithme Force Atlas 2, qui est basé sur une simulation des noeuds dans le plan comme des corps reliés par des ressorts (dont la force dépend du poids de l'arête). Il permet de situer les noeuds spatialement et de mettre visuellement en évidence les liens forts dans le graphe. Nous avons aussi calculé l'*eigenvector centrality* de chaque noeud, que nous visualisons par la taille des noeuds afin d'améliorer la lisibilité du graphe.

3 Résultats

Nous obtenons un graphe pondéré non-dirigé avec les dimensions suivantes :

Nbre de noeuds	Nbre d'arêtes	Somme des poids	Degré moyen	Degré pondéré moyen
52	495	5796	19,1	222.9

Tableau 1. – Dimensions du graphe extrait

Au moment d'appliquer l'algorithme de Louvain, nous avons testé différentes valeurs pour le paramètre de résolution entre 0.05 et 2.

Valeur de la résolution γ	0.05	0.5	1.0	2.0
Modularité	-0.033	0	0.074	0.0
Nombre de communautés détectées	17	6	2	1
Taille de la plus grande communauté	7	17	41	52
Score ARI de la partition	0.11	0.21	0.03	0

Tableau 2. – Résultats de l'application de l'algorithme de Louvain au graphe, pour différentes valeurs du paramètre de résolution.

Les 4 partitions obtenues sont illustrées sur la Fig. 1, avec à chaque fois une couleur par communauté. La partition sur base des familles réelles est visualisée à la Fig. 1. Le layout du graphe est obtenu avec l'algorithme Force Atlas 2 (`scaling=500.0, gravity=0.2`)

Fig. 1. – Visualisation des communautés pour différentes valeurs du paramètre de résolution

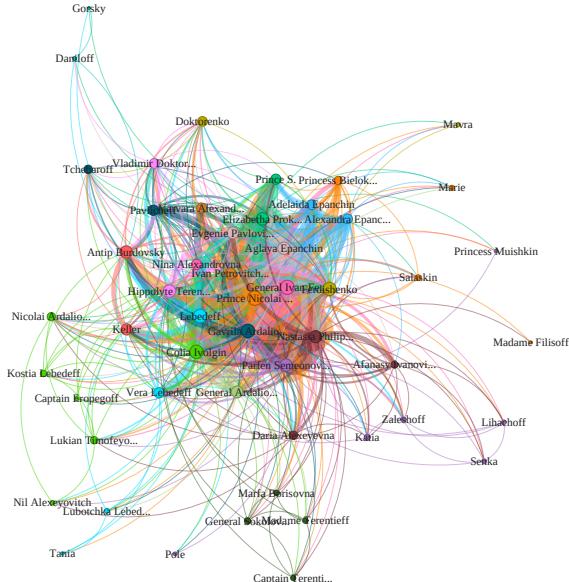


Fig. (a). – Communautés pour $\gamma = 0.05$

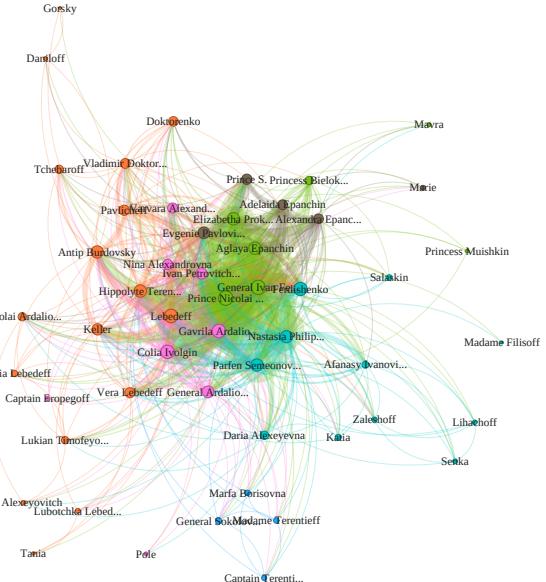


Fig. (b). – Communautés pour $\gamma = 0.5$

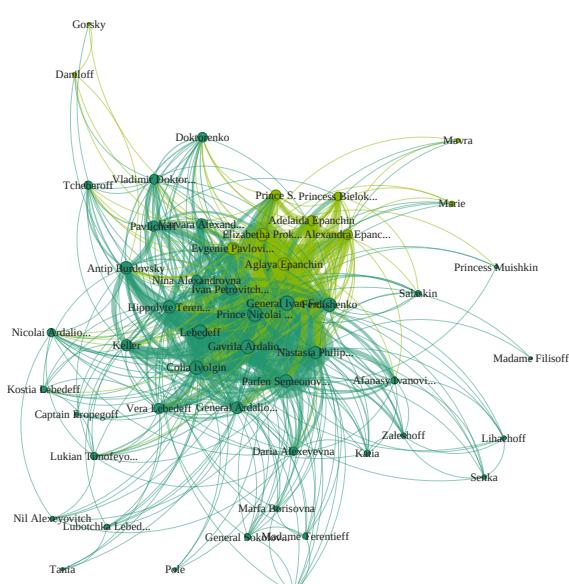


Fig. (c). – Communautés pour $\gamma = 1$

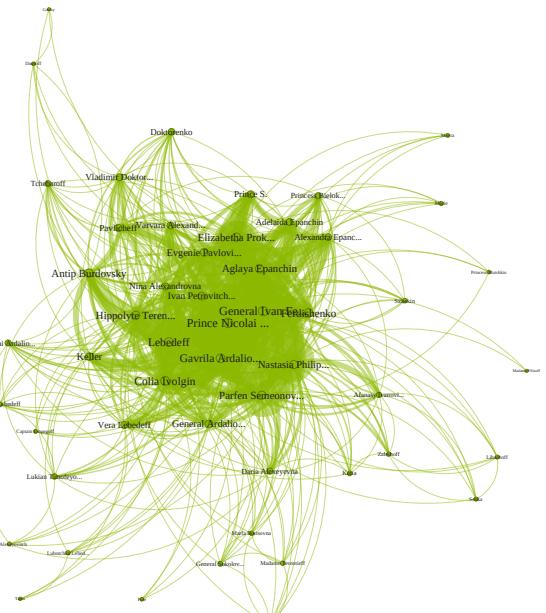


Fig. (d). – Communautés pour $\gamma = 2$

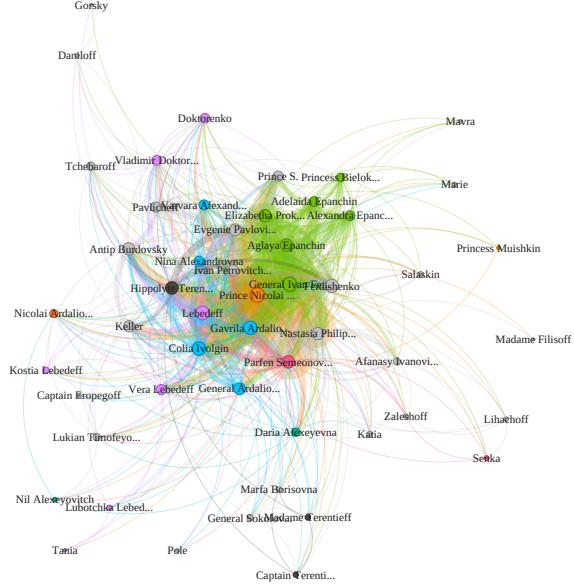


Fig. 1. – Visualisation avec l'algorithme Force Atlas 2 des différentes familles,

4 Discussion

4.1 Effets de la résolution sur les communautés

On voit dans le Tableau 2 que plus la résolution est grande, plus le nombre de communautés détectées diminue, et plus la taille des communautés augmente, comme attendu. On constate également que la modularité la plus élevée est atteinte avec $\gamma = 1$, c'est-à-dire lorsque l'algorithme optimise directement la modularité sans biais de résolution.

Si le graphe exhibe une structure communautaire par familles, on devrait s'attendre à ce que le score ARI soit corrélé avec la modularité. Or on constate que ce n'est pas le cas. L'échelle des communautés préférée par la méthode de Louvain n'est pas celle des familles. Nous avons également un score faible de modularité, les communautés détectées sont donc peu significatives. Il est possible que les imprécisions dans la méthode d'extraction créent des liens parasites qui réduisent la structure communautaire du graphe. Malgré cela, les communautés épousent relativement bien les familles, surtout pour $\gamma = 0.5$ (score ARI=0.21 maximal). Il est donc possible d'interpréter certains résultats à la lumière de ce que nous connaissons du roman.

Nous avons décidé d'analyser l'impact du paramètre de résolution sur comment se comportent les communautés des familles Epanchin et Ivolgin parce que nous les considérons représentatives du comportement des communautés quand le paramètre γ varie.

Pour le paramètre $\gamma = 0.05$, les membres de la famille Epanchin sont classifiés dans 4 communautés différentes, et que des membres d'une même génération ont tendance à être dans la même communauté. Pour la famille Ivolgin, on observe que chaque membre de la

famille est classifié différemment. Ce niveau de résolution est donc, en accord avec les métriques ARI et de modularité, une mauvaise échelle pour détecter les familles.

Avec $\gamma = 0.5$, les membres de la famille Epanchin ne sont plus que classifiés qu'en deux catégories. Et les générations ne sont plus très significative. Par exemple la grand-mère, sa fille et ses petites filles sont dans des communauté distinctes. En ce qui concerne les membres de la famille Ivolgin, ils n'appartiennent plus qu'à une communauté. Les communautés sont donc de taille semblable à celle des familles, et elles concordent bien, ce qui explique le score de ARI élevé.

Pour $\gamma = 1$ on obtient 2 communautés. La première contient presque exclusivement les membres de la famille Epanchin pendant que tous les autres noeuds appartiennent à l'autre communauté. Cette communauté résiste aux tentatives de l'algorithme de la fusionner avec une autre, ce qui démontre la forte relation entre les membres du clusters par rapport à l'extérieur.

Pour $\gamma = 2$, on obtient une seule et unique communauté. C'est un paramètre de résolution trop élevé qui ne nous fournit pas d'information pertinente sur la structure du graphe.

4.2 Analyse qualitative du graphe

L'objectif de cette partie est de réunir différents liens que nous pouvons faire entre le graphe obtenu (layout, communautés détectées) et l'histoire du roman.

- Nastasia Philipovna a décidé de se marier avec Parfen Semeonivitche contre une importante somme d'argent et ils sont dans la même communauté.
- Nastasia Philipovna et Afanasy Ivanovitch qui est tuteur et amant sont dans la même communauté.
- Daria Alexvnya est la meilleure amie de Nastasia Philipovna et elles sont dans la même communauté.
- Pendant un moment, le mariage entre Aglaya Epanchin et le Prince Nicolai Muishkin est envisagé, et Aglaya est dans la même communauté que le prince mais elle n'est pas dans la même communauté que ses sœurs.
- Varvara Alexandrovna se marie avec Ivan Petrovitch pour sauver sa famille de la banqueroute et ils sont dans la même communauté.

On peut aussi constater que le layout met en évidence les relations hiérarchiques entre les personnages. Les noeuds centraux sont les personnages principaux, entourés des personnages secondaires qui leur sont proches.

La Princesse Bielokonski est la grand-mère des trois sœurs Epanchin, et elle a peu de contacts avec l'extérieur du foyer. Cela peut se voir sur la visualisation, où son noeud est excentré et les noeuds des Soeurs Epanchin semblent faire écran entre elle et le reste des personnages.

5 Conclusion

En conclusion, notre projet avait pour objectif de mettre en évidence les familles des différents protagonistes à partir du graphe de leurs interactions dans le récit. Les résultats montrent que la structure communautaire du graphe n'est pas fortement significative (faible modularité). On constate également que le niveau de résolution adapté à la détection des familles n'est pas celui par défaut de la méthode de Louvain, et qu'utiliser des résolutions différentes pouvait améliorer le score ARI.

A partir d'observations qualitatives, nous avons mis en évidence la pertinence du layout Force Atlas 2 dans l'analyse de la proximité et de la relation des personnages, et aussi pu constater la pertinence de la partition du graphe en communauté au regard du récit.

Une piste d'amélioration de ce projet serait de tester plus de valeurs du paramètre de résolution, afin d'affiner les résultats et de mieux comprendre la réaction de l'algorithme au différentes valeurs de γ . Il pourrait aussi être intéressant de comparer les performances d'autres algorithmes de détection de communauté.

Il serait également possible d'améliorer le processus d'extraction afin d'avoir plus de personnages et d'être plus exhaustif sur les différents surnoms rencontrés dans le texte pour une même personne.

Bibliographie

- [1] F. Dostoïevsky, *The Idiot*. Projet Gutenberg. Consulté le 8 décembre 2025, de <https://www.gutenberg.org/ebooks/2638>, 1869.
- [2] M. Bastian, S. Heymann, et M. Jacomy, « Gephi: an open source software for exploring and manipulating networks », in *Third international AAAI conference on weblogs and social media*, 2009.
- [3] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, et E. Lefebvre, « Fast unfolding of communities in large networks », *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, n° 10, p. P10008, oct. 2008, doi: 10.1088/1742-5468/2008/10/p10008.
- [4] Wikipedia contributors, « Leiden algorithm — Wikipedia, The Free Encyclopedia ». 2025.
- [5] Wikipedia contributors, « Rand index — Wikipedia, The Free Encyclopedia ». [En ligne]. Disponible sur: https://en.wikipedia.org/w/index.php?title=Rand_index&oldid=1318214634