L'espace sémantique du "Dictionnaire électronique des synonymes" (DES) et les méthodes de regroupement de sens : l'exemple de "sec"

Laurette Chardon

L'espace sémantique du *Dictionnaire électronique des synonymes* (*DES*) et les méthodes de regroupement de sens : l'exemple de *sec*

Laurette Chardon

Université de Caen Normandie

Centre de recherches inter-langues sur la signification en contexte (CRISCO)

laurette.chardon@unicaen.fr

Résumé : Le *Dictionnaire électronique des synonymes* (*DES*) du laboratoire CRISCO (Centre de recherches inter-langues sur la signification en contexte) de l'université de Caen a été initié dans les années 1990. Il a été créé à partir de sept dictionnaires et régulièrement enrichi (semi-manuellement) depuis : il contient à ce jour 50 000 entrées et 209 000 liaisons synonymiques. Le projet de recherche à l'origine de cet outil se fonde sur les cliques et la notion d'espace sémantique.

Nous nous proposons dans cet article d'approfondir à l'aide de l'adjectif *sec* cette notion d'espace sémantique au travers de différentes méthodes de regroupement que nous comparerons entre elles et à la méthode actuelle en ligne.

Nous terminerons en exposant des pistes d'améliorations pour nettoyer ou corriger les données de départ de façon à permettre que l'espace sémantique devienne un outil utile pour comprendre la polysémie du lexique du français contemporain pour tous les utilisateurs du *DES*.

Abstract: *The* Dictionnaire électronique des synonymes *(*DES*) of the CRISCO laboratory (Centre de recherches inter-langues sur la signification en contexte) at the University of Caen was initiated in the 1990s. It was created from seven dictionaries and has been regularly updated since then: it now contains 50,000 entries and 209,000 synonymic links. The research project behind this tool is based on cliques and the notion of semantic space.*

In this article, we propose to deepen this notion of semantic space with the help of the adjective sec *in French through different grouping methods that we will compare to each other and to the current online method.*

We will conclude by outlining improvements to clean up or correct the source data so that the semantic space becomes a useful tool for representing the polysemy in French for all DES *users.*

1. Introduction

Le *Dictionnaire électronique des synonymes* (*DES*) du laboratoire CRISCO (Centre de recherches inter-langues sur la signification en contexte) de l'université de Caen créé dans les années 1990 est un outil plébiscité par le public avec plus de 150 000 requêtes par jour[1]. Il a été créé à partir de sept dictionnaires et régulièrement enrichi de façon semi-manuelle depuis : il contient à ce jour 50 000 entrées et 209 000 liaisons synonymiques[2].

La partie recherche fondée sur les cliques (sous-ensemble de synonymes d'une vedette donnée tous synonymes entre eux) et la notion d'espace sémantique est moins connue des chercheurs en linguistique probablement à cause de l'aspect mathématique sous-jacent. Ces calculs (la recherche des cliques maximales ou les réductions de dimension pour l'espace sémantique) sont pourtant des opérations qui nous permettent de représenter de façon automatique, claire et synthétique les différents sens principaux d'un mot vedette.

Le calcul des cliques est une méthode de calcul parmi d'autres. Il existe d'autres méthodes qui, à partir de la matrice d'adjacence, vont donner divers résultats de regroupements de sens, en particulier en programmation python avec la librairie igraph.

Au travers d'un exemple, l'adjectif et le substantif *sec*, nous allons dans cet article revenir sur le calcul de l'espace sémantique avec la création des cliques et la réduction de dimension utilisée. Puis, nous nous pencherons sur les méthodes de regroupement (ou de *clustering*) les plus courantes que nous propose le langage python avec une librairie de « *machine learning* » intitulée igraph en étudiant et comparant leurs résultats mathématiques avec les regroupements de sens lexicographiques que nous présenterons en 2.2.

2. L'étude de *sec*

Nous allons commencer par préciser la notion de synonymie que nous utiliserons dans cet article. Comme le précisent Ploux et Victorri (1998 : 162), il existe deux types de synonymie : « pure » et « partielle ». La synonymie « pure » considérée comme trop restrictive par les auteurs, la

1. Les statistiques sont issues du logiciel de mesure de statistiques Web Matomo installé par la direction des systèmes d'information de l'université de Caen (voir « Matomo (logiciel) », page de présentation sur Wikipédia : https://fr.wikipedia.org/wiki/Matomo_(logiciel)).
2. Voir « Présentation du *DES* », site du *DES* : http://crisco.unicaen.fr/dictionnaire-electronique-des-synonymes/presentation-du-des/.

définition de synonymie « partielle » sera donc utilisée ici. La définition donnée est la suivante :

> Deux unités lexicales sont en relation de synonymie si toute occurrence de l'une peut être remplacée par une occurrence de l'autre dans un certain nombre d'environnements sans modifier notablement le sens de l'énoncé dans lequel elle se trouve.

C'est cette définition que les auteurs ont utilisée lors de la création du *DES*. Nous nous proposons donc d'étudier l'adjectif et substantif *sec* avec l'ensemble de ses synonymes « partiels » présents dans le *DES*[3]. Il possède à ce jour dans le *DES* 67 synonymes, listés par ordre alphabétique ci-dessous, 41 antonymes, et apparaît dans 96 cliques :

> *âpre*, *abrupt*, *acerbe*, *aigre*, *anhydre*, *aride*, *asséché*, *austère*, *autoritaire*, *blessant*, *bourru*, *bref*, *brusque*, *brutal*, *cassant*, *concis*, *cru*, *décharné*, *déplaisant*, *désagréable*, *désargenté*, *désertique*, *désobligeant*, *desséchant*, *desséché*, *dur*, *efflanqué*, *égoïste*, *émacié*, *endurci*, *essuyé*, *étique*, *étriqué*, *fauché*, *ferme*, *froid*, *glacé*, *glacial*, *impécunieux*, *improductif*, *indifférent*, *ingrat*, *insensible*, *maigre*, *maigrelet*, *osseux*, *pauvre*, *pincé*, *pur*, *racorni*, *raide*, *rébarbatif*, *rebutant*, *revêche*, *rigide*, *rogue*, *rude*, *sans-cœur*, *séché*, *sécot*, *seul*, *sévère*, *simple*, *squelettique*, *stérile*, *tranchant*, *vide*

C'est une unité très polysémique. Son étude a fait l'objet de plusieurs publications (Jacquet *et al.* 2005 ; Venant 2004) et en cela il constitue un cas pertinent à observer d'un point de vue de la modélisation informatique. Réaliser un graphe d'adjacence constitue l'un des moyens de visualiser rapidement l'ensemble des synonymes et d'appréhender ceux qui paraissent visuellement les plus significatifs. C'est cette méthode que nous allons présenter en premier.

2.1. Graphe d'adjacence

Ce graphe est représenté sur la figure 1 (ci-après). Les graphes sont des « modèles abstraits de dessins de réseaux reliant des objets » ; « Ces modèles sont constitués par la donnée de sommets (aussi appelés *nœuds* ou *points*, en référence aux polyèdres), et d'arêtes (aussi appelées *liens* ou *lignes*) entre ces sommets »[4].

---

3. https://crisco2.unicaen.fr/des/synonymes/sec
4. « Théorie des graphes », page de présentation sur Wikipédia : https://fr.wikipedia.org/wiki/Th%C3%A9orie_des_graphes.

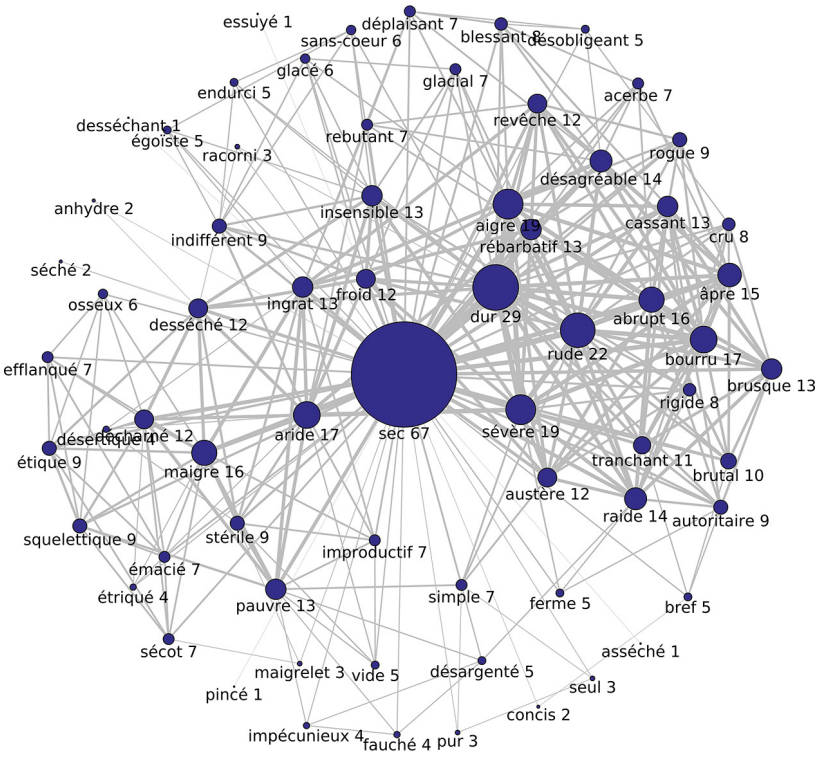


Figure 1. Graphe d'adjacence de *sec*

La vedette et chacun de ses synonymes sont représentés par un cercle bleu avec un diamètre plus ou moins grand en fonction du nombre de synonymes que chacun possède : c'est la valeur affichée à côté du cercle. Le rond bleu représentant *sec* est central et a le plus grand diamètre puisque *sec* a 67 synonymes. Nous observons aussi que *dur*, *rude* et *sévère* ont respectivement un diamètre de cercle proportionnel aux nombres 29, 22 et 19. Ces nombres représentent le nombre de liens synonymiques que le sommet de la vedette a avec les autres sommets.

Ceci nous permet déjà de déduire que, dans le *DES*, *sec* est très fortement connecté avec certains synonymes qui sont liés entre eux et qui ont par conséquent beaucoup de synonymes en commun avec la vedette : outre *dur*, *rude* et *sévère* que nous avons cités, nous pouvons facilement déduire aussi que *bourru*, *aride*, *maigre*, *désagréable*, *insensible*, *ingrat* et *pauvre* sont fortement reliés à *sec* puisque la densité d'arêtes qui les entourent est importante.

L'emplacement des sommets entre eux nous renseigne sur leur relation. Le calcul automatique réalisé par le système pour placer les sommets va rapprocher ceux qui ont beaucoup de liens et éloigner ceux qui en ont moins de façon à rendre le graphique le plus lisible possible. En effet, *dur*, *rude*, *sévère*, *désagréable* et *bourru* sont disposés à droite de la vedette, alors que *aride*, *maigre*, *ingrat* et *pauvre* sont à gauche : ils n'ont donc pas ou peu de liens entre eux. *Insensible* occupe une position médiane en haut.

De plus, nous remarquons que les traits de couleur grise ont des largeurs différentes : ces dernières reflètent la force de la relation entre les synonymes, deux à deux.

La force d'une relation est proportionnelle au calcul du nombre minimum de liens à supprimer dans le graphe pour que l'un des sommets ne soit plus accessible à partir de l'autre et inversement, autrement dit c'est le nombre de liens à supprimer pour obtenir deux graphes disjoints[5]. Plus le nombre de liens à supprimer est important, plus cela signifie que ces deux sommets ont de nombreux autres synonymes en commun. L'exemple simple de la figure 1b (ci-après) va nous permettre de comprendre. Pour ne plus pouvoir atteindre le sommet U2 à partir du sommet U7, nous devons enlever quatre arêtes (en pointillé bleu). De même entre les sommets U7 et U8, nous ne devons enlever que deux arêtes (tirets verts). Nous voyons ainsi que la relation entre U7 et U2 est plus forte que celle entre U7 et U8[6].

Cela nous permet d'ajouter, en complément des déductions précédentes, que *dur*, *rude*, *sévère*, *abrupt* sont très liés puisqu'ils sont proches (sur la droite du graphe) et les liens entre eux sont forts. De même *maigre*, *aride*, *pauvre* et *stérile* à gauche du graphe sont assez proches les uns des autres.

2.2. Vision lexicographique

Sec est répertorié dans le *Trésor de la langue française* (*TLF*) avec plus de 30 sens différents. Mais nous pouvons regrouper les sens en six acceptions principales comme le proposent Jacquet *et al.* (2005) et Venant (2004) :

1. qui ne contient pas d'eau : « la route était sèche » ;
2. maigre, décharné : « un vieil homme sec et ridé » ;

5. Voir « Edge Connectivity », in *R igraph Manual Pages* : https://igraph.org/r/doc/edge_connectivity.html.
6. Voir le programme ExempleEdgeConnectivity.py du serveur Git de l'université de Caen, « étude-SEC » : https://git.unicaen.fr/crisco-des-public/etude-sec.

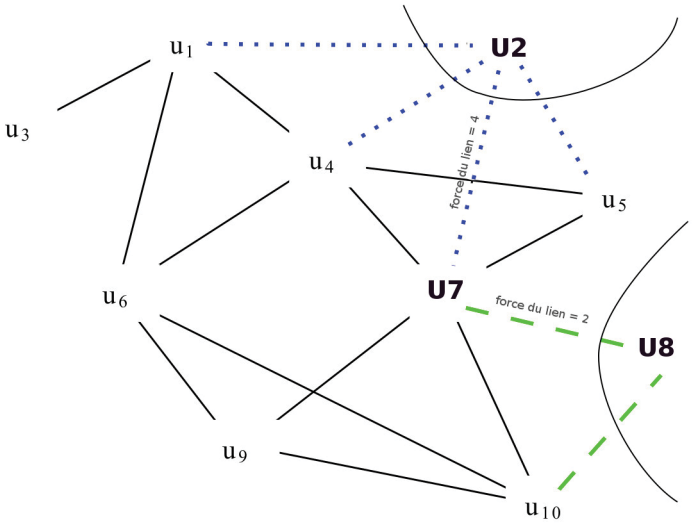


Figure 1b. Force d'une relation selon la méthode *Edge_connectivity* de la librairie igraph

3. stérile, improductif : « rester sec à un examen » ;
4. insensible, sévère, égoïste : « un homme au cœur sec » ;
5. brusque, abrupt : « donner un coup sec » ;
6. simple, seul : « avoir un atout sec dans son jeu ».

L'ensemble des sens est décrit comme une « ressemblance de famille » avec six sens qui se recouvrent en partie. Nous voyons bien que l'étude du graphe d'adjacence nous apporte des informations qui confortent la représentation lexicographique donnée dans le dictionnaire puisque nous y retrouvons comme synonymes les plus représentatifs : *maigre*, *abrupt*, *insensible* et *sévère*. *Simple* et *seul* en revanche apparaissent en bas du graphique avec peu de synonymes : ils ne sont donc pas visibles et c'est pourtant une signification de *sec* à souligner.

En comparaison, *Le Grand Robert*[7] propose quatre catégories : I avec la notion de concret, II avec la notion d'abstrait, III en tant que nom et IV en tant qu'adverbe. Dans la catégorie I (concret), nous avons neuf subdivisions :

7. Version en ligne du *Grand Robert de la langue française*.

- qui n'est pas ou peu imprégné de liquide ;
- qui n'est plus mouillé ou humide ;
- déshydraté, séché (suite à un traitement) : « fruits secs » ;
- qui n'est pas accompagné du liquide auquel il est généralement associé : « mur de pierres sèches », « toux sèche » ;
- non accompagné d'un autre élément : « pain sec » ;
- qui est peu charnu, qui a peu de graisse : « corps sec » ;
- qui est peu étoffé, manque d'ampleur : « contours très précis, secs » ;
- tissu, lainage sec, à tissage bien marqué ;
- vins secs, peu sucrés.

Dans la catégorie II (abstrait) :

- froid, indifférent, insensible, pincé ;
- aigre, désobligeant, glacial ;
- rébarbatif, rebutant, revêche.

Dans la catégorie III (en tant que nom) :

- sécheresse : le sec et l'humide.

Dans la catégorie IV (en tant qu'adverbe) :

- brutalement, rapidement, rudement, sèchement : « qui claque sec », « laisser tomber tout sec ».

Dans le graphe d'adjacence, nous remarquons que *sec* en tant qu'adverbe n'est pas représenté mais les autres catégories du *Grand Robert* le sont. Cela peut sembler surprenant étant donné que le *DES* ne fait pas de distinction de catégorie grammaticale.

Nous allons maintenant étudier l'espace sémantique de *sec* tel qu'il se présente à ce jour dans le *DES*.

2.3. La création des cliques et l'espace sémantique

Comprendre l'espace sémantique en 2D accessible en ligne[8] peut paraître difficile dans un premier temps. Pour cela, nous avons réalisé une présentation détaillée du calcul (voir Chardon 2020) et un tutoriel

8. Voir la visualisation 2D de l'espace sémantique d'un mot, site du *DES* : https://crisco2.unicaen.fr/espsem/.

proposant une méthode d'investigation en prenant comme exemple l'adjectif *curieux*[9].

Le calcul se fonde sur la publication de Ploux & Victorri (1998) qui détaille la métrique du χ^2. En effet, nous partons de la *matrice d'appartenance* des mots aux cliques. Les cliques sont des sous-ensembles du graphe dont les sommets sont tous reliés entre eux. L'algorithme utilisé dans le cadre de l'espace sémantique du *DES* est celui de Bron-Kerbosch (Bernard & Seba 2018).

Cet algorithme récursif[10] se présente ainsi :

```
algorithm BronKerbosch1(R, P, X) is
    if P and X are both empty then
        report R as a maximal clique
    for each vertex v in P do
        BronKerbosch1(R ∪ {v}, P ∩ N(v), X ∩ N(v))
        P := P \ {v}
        X := X ∪ {v}
```

- L'ensemble R contient la clique partielle à un instant t.
- L'ensemble P contient les nœuds candidats pour agrandir la clique partielle.
- L'ensemble X contient les nœuds ayant déjà été visités pour la construction de la clique.

Tant que P et X ne sont pas vides, un candidat potentiel (v) est sélectionné dans P et l'algorithme est relancé avec R auquel on ajoute l'élément v et avec P et X tenant compte des voisins de v (N(v)). Une fois l'ensemble des cliques calculé, la matrice d'appartenance des mots aux cliques se présente comme nous le montre la figure 2 (ci-contre).

Cette matrice M est composée des coefficients C_iM_j égaux à 0 ou 1 selon si le mot M_j appartient à la clique C_i. Ensuite une nouvelle matrice Q composée des coefficients X_{ij} est calculée en fonction de ces coefficients C_iM_j divisés par la somme totale, X, et les racines carrées des sommes marginales selon la formule de la figure 3 (ci-contre).

9. Voir L. Chardon, « Espace sémantique avec CURIEUX : tutoriel pas à pas », 23 avril 2019, site du *DES* : http://crisco.unicaen.fr/dictionnaire-electronique-des-synonymes/actualites-des/espace-semantique-de-curieux-tutoriel-pas-a-pas-964401.kjsp?RH=1530619460865.
10. Voir « Bron-Kerbosh Algorithm », page de présentation sur Wikipédia : https://en.wikipedia.org/wiki/Bron%E2%80%93Kerbosch_algorithm.

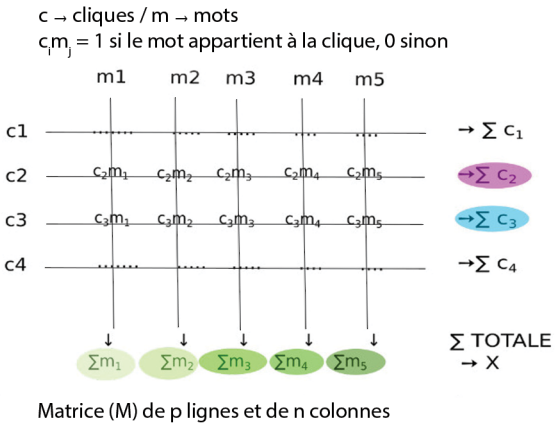


Figure 2. Matrice d'appartenance des mots aux cliques

$$x_{ij} = \frac{c_i m_j}{X.\sqrt{\sum_{i=1}^{n} C_i}\sqrt{\sum_{j=1}^{p} M_j}}$$

Figure 3. Calcul des coefficients de la matrice Q selon la métrique du χ^2 [11]

Cette matrice Q est ensuite décomposée en valeurs singulières [12] correspondant aux produits de trois matrices qui vont servir à calculer les coordonnées des mots et des cliques dans un espace à deux dimensions (voir Chardon 2020).

Un espace sémantique (Poudat & Landragin 2017 : 103-121) est une représentation graphique issue d'une réduction des données de

11. La formule du χ^2 est un test statistique pour vérifier l'adéquation d'une série de données à une famille de lois de probabilité. À la base de ce test, il y a la formulation d'une hypothèse appelée hypothèse nulle, notée H0. Elle suppose que les données considérées proviennent de variables aléatoires qui suivent une loi de probabilité donnée, et l'on souhaite tester la validité de cette hypothèse. Dans notre cas, nous allons donc pondérer chaque valeur de chaque point (C_i, M_j) de la matrice par le nombre de cliques dans lesquelles se trouve le mot M_j et par le nombre de mots que contient la clique C_i. Pour plus d'information, consulter « Test du χ^2 », page de présentation sur Wikipédia : https://fr.wikipedia.org/wiki/Test_du_%CF%87%C2%B2
12. Voir « Décomposition en valeurs singulières », page de présentation sur Wikipédia : https://fr.wikipedia.org/wiki/D%C3%A9composition_en_valeurs_singuli%C3%A8res.

départ. L'objectif est de déformer le moins possible ces dernières de façon à mieux rendre compte de leur structure. Dans notre exemple, les données de départ sont regroupées dans la matrice d'appartenance montrant l'appartenance de *sec* et de ses synonymes aux différentes cliques comme indiqué ci-dessus. Elles sont réduites en plusieurs facteurs dont les deux principaux, qui vont donner le maximum d'information des données initiales, sont représentés par les axes X et Y.

Pour chaque axe, on va en premier lieu repérer les points qui s'opposent le plus, c'est-à-dire ceux qui sont les plus éloignés du centre, parce qu'ils vont permettre au mieux d'apprécier les contrastes. Dans la figure 4 (ci-contre), nous avons d'un côté de l'axe Y mais assez proches du centre *rude*, *dur*, *bourru* et *sévère* et de l'autre, plus éloignés, *maigre*, *osseux* et *squelettique*. Nous voyons également que, partant de *maigre* vers le centre, nous avons *aride*, *pauvre*, *ingrat*. Cet axe Y met en opposition les sens concrets et les sens abstraits de *sec*.

L'axe X, de son côté, met en évidence un autre sens concret de *sec* : *simple*, *seul* (vin ou vol sec). Les points les plus proches du centre (*rude*, *dur*, *bourru* et *sévère*) sont les plus proches de la « moyenne » : ils s'attirent et sont fortement corrélés. Ils représentent les synonymes les plus nombreux, fortement reliés entre eux par les cliques.

Ce qui compte donc pour comprendre un espace sémantique est d'observer l'emplacement des points par rapport aux axes (éloignés, proches du centre) et les regroupements de points qui vont mettre en évidence des rapprochements de sens.

Poudat & Landragin (2017 : 111) signalent toutefois que la représentation en 2D pose un problème, qui peut être partiellement réglé par une représentation en 3D :

> C'est donc en observant la position des points sur les axes qu'on peut commencer à interpréter les facteurs. Pourtant un problème majeur se pose à ce stade : la position des points sur un plan factoriel peut être trompeuse.
>
> En effet, on doit garder à l'esprit que la localisation des points sur un plan ne permet pas d'évaluer la distance réelle entre ces points. La visualisation des variables sur les axes linéaires de l'ACP [analyse en composantes principales] entraîne potentiellement des biais : deux variables proches, voire presque recouvertes, quand elles sont projetées sur deux axes peuvent en effet s'avérer très éloignées l'une de l'autre dans l'espace.
>
> Observer les descripteurs dans un espace à trois dimensions peut visuellement apporter davantage d'information sans toutefois résoudre le problème.

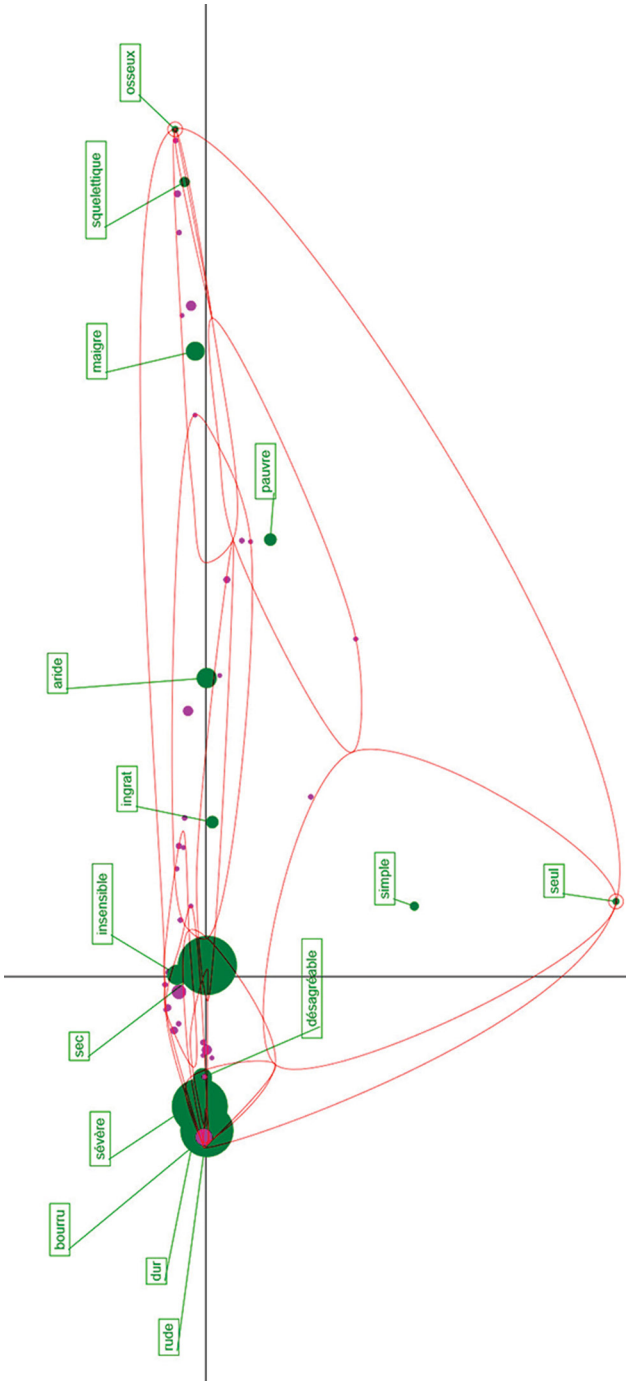


Figure 4. Espace sémantique de *sec*

Nous avons récemment mis en place une version test 3D[13] dans le *DES*. Celle-ci apporte une vision enrichie et affiche quatre « groupes de sens » :

1. *sans-cœur*, *indifférent*, *insensible* ;
2. *blessant*, *abrupt*, *désagréable* ;
3. *désertique*, *stérile*, *aride* ;
4. *maigre*, *squelettique*, *émacié*.

On constate cependant que le groupe représenté par *simple* et *seul* n'est pas aussi clairement visible : cet espace sémantique apparaît entre celui de *blessant* d'un côté et celui de *désertique* de l'autre sans toutefois être distinctement délimité.

Nous souhaitons maintenant passer en revue d'autres méthodes de regroupement (ou *clustering*) réalisées directement sur le graphe d'adjacence afin de les comparer avec celle de l'espace sémantique du *DES*.

3. Autres méthodes de regroupement

Ces méthodes font partie des traitements disponibles dans le cadre des études en « *machine learning* ». Les méthodes de regroupement exposées ci-dessous sont issues de la librairie igraph[14]. La liste n'est pas exhaustive car il existe en réalité plusieurs dizaines d'algorithmes de détection de communauté (Fortunato 2010). Le code python qui génère les graphiques ci-dessous est disponible en public sur le serveur Git de l'université de Caen[15].

3.1. L'algorithme Multilevel

Cet algorithme est issu de la publication de Blondel *et al.* (2008). Il est également appelé l'algorithme de Louvain (les auteurs étant rattachés à l'université de Louvain en Belgique).

---

13. Voir la visualisation 3D de l'espace sémantique de *sec*, site du *DES* : https://crisco2.unicaen.fr/des3d/sec.html. Il est possible de faire tourner la vue manuellement en cliquant au centre du graphique et en faisant glisser la souris sur l'axe rouge, vert ou bleu selon que l'on souhaite faire une rotation par rapport à l'axe X, Y ou Z. Il est également possible de cliquer sur les rectangles de couleur à gauche pour obtenir la vue décrite à côté de ce rectangle (rectangle magenta = vue générale 3D, etc.). Enfin il est possible d'activer une rotation automatique *via* les paramètres disponibles en bas à droite du graphique, en cliquant sur la roue crantée puis sur la flèche circulaire.
14. https://igraph.org
15. « étude-SEC », serveur Git de l'université de Caen : https://git.unicaen.fr/crisco-des-public/etude-sec.

> La méthode permet d'effectuer le partitionnement d'un réseau en optimisant la modularité. La modularité est une valeur comprise entre -1 et 1 qui mesure la densité d'arêtes à l'intérieur des communautés (regroupements de sens) comparée à celle des arêtes reliant les communautés entre elles[16].

Il s'agit d'un algorithme ascendant : au départ, chaque sommet appartient à une communauté distincte, et les sommets sont déplacés entre les communautés de manière itérative de façon à maximiser la contribution locale des sommets au score global de modularité. Lorsqu'un consensus est atteint (c'est-à-dire qu'aucun mouvement n'augmenterait le score de modularité), chaque communauté dans le graphique original est réduite à un seul sommet (tout en conservant le poids total des arêtes adjacentes) et le processus se poursuit au niveau suivant. L'algorithme s'arrête lorsqu'il n'est plus possible d'augmenter la modularité après avoir réduit les communautés aux sommets.

On dit que cet algorithme fonctionne presque en temps linéaire sur des graphiques clairsemés.

Sur *sec*, cela donne le résultat de la figure 5 (ci-après).

Nous remarquons six clusters différents :

1. en vert autour de *dur*, *sévère*, *rude*, *aigre*, *bourru*, *brusque* ;
2. en bleu foncé avec *maigre*, *décharné*, *squelettique* ;
3. en bleu clair représenté par *aride*, *pauvre*, *ingrat*, *stérile*, *asséché* ;
4. en rouge avec *désagréable*, *blessant*, *rebutant*, *déplaisant*, *acerbe* ;
5. en jaune avec *insensible*, *froid*, *indifférent* ;
6. en magenta avec *pur*, *simple* et *seul*.

On retrouve de façon assez prononcée les mêmes catégories de sens que celles citées à la section 2.2 « Vision lexicographique ». Deux remarques s'imposent toutefois :

- l'absence d'eau (« la route était sèche ») apparaît dans le cluster 3 mais elle est peu représentative de ce dernier (*asséché* et *séché* ont peu de relations) ;
- le cluster 4 (*désagréable*, *blessant*…) et le cluster 5 (*insensible*, *froid*, *indifférent*) se retrouvent dans la catégorie 4 de Jacquet *et al.* (2005) et Venant (2004), laquelle est représentée par l'exemple « un homme au cœur sec » (voir section 2.2).

16. « Algorithme de Louvain », page de présentation sur Wikipédia : https://fr.wikipedia.org/wiki/M%C3%A9thode_de_Louvain.

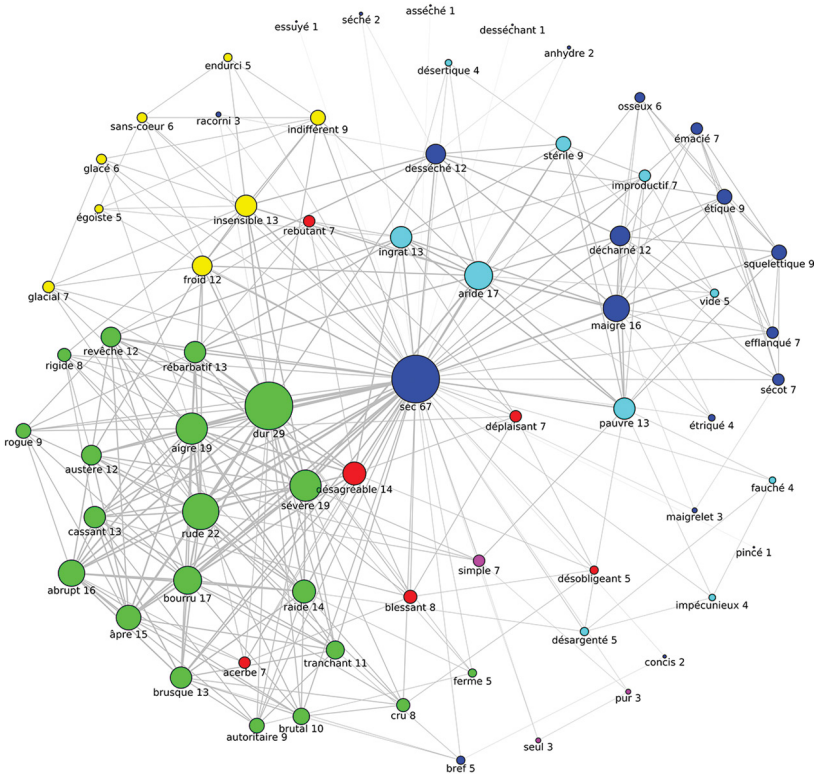


Figure 5. Regroupements des nœuds du graphe de *sec* avec l'algorithme *Multilevel*

3.2. L'algorithme Infomap

Cet algorithme se fonde sur deux publications, l'une de Rosvall & Bergstrom (2008) et l'autre de Rosvall *et al.* (2009), et le lien sur son implémentation dans la librairie igraph est donné dans la sitographie.

L'idée de base derrière l'algorithme *Infomap*[17] est d'utiliser l'image d'une personne se déplaçant aléatoirement d'un sommet à un autre. Par analogie avec une carte routière, la probabilité de rester longtemps,

17. Voir L. Beauguitte, « Sur la détection de communautés en général et avec R en particulier », 10 avril 2019, publication dans le carnet de recherche du groupement de recherche « Analyse de réseaux en sciences humaines et sociales » : https://arshs.hypotheses.org/1314#more-1314 ; et « Infomap community detection understanding », réponse à une question posée sur le site *Stack Overflow* : https://

après un certain nombre de choix, dans une communauté sera d'autant plus grande que les sommets dans cette communauté sont fortement connectés. De même que, sur un chemin ou une route, un marcheur aventureux aura plus de chance de voyager à l'intérieur d'une région où les villes et villages sont très connectés entre eux. Si le graphe est aléatoire, aucune communauté ne sera détectée puisque le marcheur se déplacera au hasard des chemins rencontrés qui ne seront pas plus nombreux à un endroit qu'à un autre.

Chaque sommet est codé à différents niveaux à l'aide de préfixes. Un préfixe est un terme utilisé dans plusieurs domaines[18]. Nous l'utilisons ici dans le domaine des réseaux informatiques dans le sens de routage[19]. Par analogie à une adresse postale, un numéro de rue n'a de sens que par rapport au nom de cette rue qui elle-même n'a de sens que par rapport à un arrondissement (dans les grandes villes) lui-même identifié par le nom de la ville dans laquelle il est, ville qui elle-même se situe dans un pays donné.

Une optimisation est nécessaire : lorsque les sommets ne sont codés qu'à un seul niveau, alors il y a trop peu de communautés. Si, au contraire, les sommets sont codés avec de nombreux préfixes, les communautés sont trop nombreuses. Nous devons donc trouver une partition optimale qui assigne des nœuds aux communautés de façon à réduire au minimum l'information nécessaire pour comprimer le mouvement de nos marcheurs aléatoires.

L'algorithme *Infomap* et celui fondé sur la modularité (*Multilevel*) sont deux exemples de méthodes optimales de détection de communauté : ils ont chacun une fonction de qualité et recherchent ensuite dans l'espace des partitions graphiques pour trouver la partition qui optimise cette fonction de qualité. La différence réside dans la fonction de qualité : *Infomap* se concentre sur les informations nécessaires pour

---

stackoverflow.com/questions/48528648/infomap-community-detection-understanding/54292999.

18. Voir « Préfixe », page de présentation sur Wikipédia : https://fr.wikipedia.org/wiki/Pr%C3%A9fixe.
19. Voir « Routage », page de présentation sur Wikipédia : https://fr.wikipedia.org/wiki/Routage : « Le routage est le mécanisme par lequel des chemins sont sélectionnés dans un réseau pour acheminer les données d'un expéditeur jusqu'à un ou plusieurs destinataires. Le routage est une tâche exécutée dans de nombreux réseaux, tels que le réseau téléphonique, les réseaux de données électroniques comme Internet, et les réseaux de transports. Sa performance est importante dans les réseaux décentralisés, c'est-à-dire où l'information n'est pas distribuée par une seule source, mais échangée entre des agents indépendants. C'est grâce à ça que par exemple les mails sont envoyés aux bons destinataires ».

comprimer le mouvement de la marche aléatoire, tandis que *Multilevel* définit les communautés en fonction de la densité des arêtes (une communauté contient un nombre d'arêtes supérieur au nombre dû au hasard[20]).

À nouveau, dans la figure 6 (ci-contre), nous obtenons six clusters :

1. *dur*, *rude*, *sévère*, *aigre*, *insensible*, *froid*, *indifférent* en rouge ;
2. *aride*, *pauvre*, *ingrat*, *stérile* en bleu foncé ;
3. *maigre*, *décharné*, *squelettique* en vert ;
4. *fauché*, *désargenté*, *impécunieux* en bleu clair ;
5. *séché*, *desséché*, *racorni* en jaune ;
6. *simple*, *seul*, *pur* en magenta.

Le cluster 4 avec *désagréable*, *blessant*, *rebutant*, *déplaisant*, *acerbe* de l'algorithme *Multilevel* a laissé sa place à *fauché*, *désargenté et impécunieux*. Les autres clusters sont similaires entre les deux méthodes.

3.3. L'algorithme Fastgreedy

Cet algorithme se fonde sur la publication de Clauset *et al.* (2004) et le lien sur son implémentation dans la librairie igraph est donné dans la sitographie.

Il fusionnera deux communautés actuelles de manière itérative, dans le but d'obtenir le maximum de gain de modularité au niveau local optimal. Il regroupe donc les nœuds individuels en communautés d'une manière qui maximise le plus (*greedily* = « goulûment ») le score de modularité du graphique[21].

Cela donne le dendrogramme (c'est-à-dire une représentation sous forme d'arbre) de la figure 7 (ci-après).

Les rectangles colorés montrent bien les six clusters qui sont relativement proches des deux précédemment étudiés.

---

20. Voir « What are the differences between community detection algorithms in igraph ? », réponse à une question posée sur le site *Stack Overflow* : https://stackoverflow.com/questions/9471906/what-are-the-differences-between-community-detection-algorithms-in-igraph.
21. Voir « Fastgreedy », in *Python-igraph Manual* : https://igraph.org/python/doc/igraph.Graph-class.html#community_fastgreedy ; et « Community Detection in Python », site *Yoyo in Wanderland* : https://yoyoinwanderland.github.io/Community-Detection.

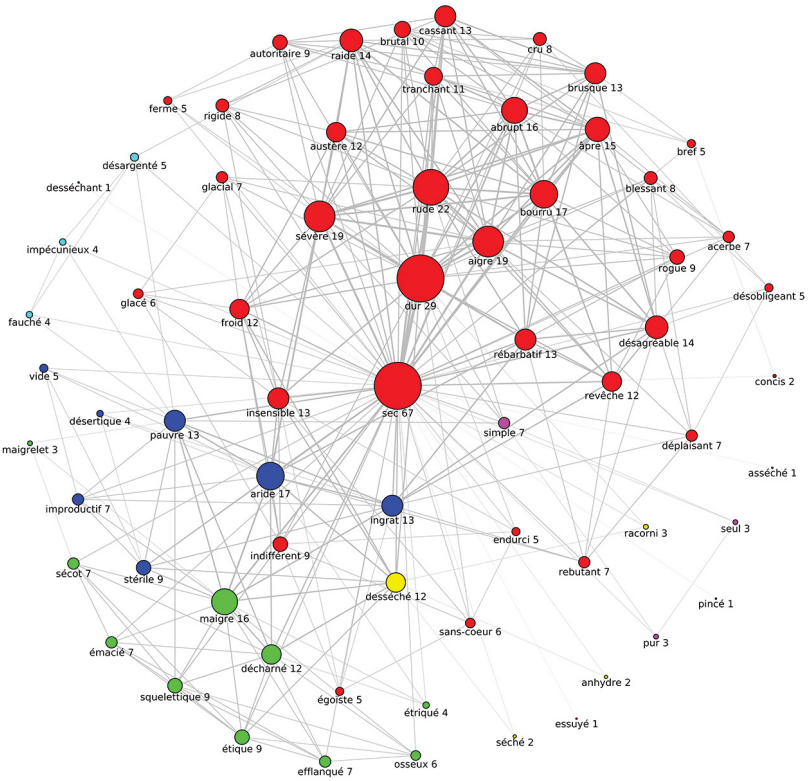


Figure 6. Regroupement des nœuds du graphe de *sec* avec l'algorithme *Infomap*

L'inconvénient de cette méthode est que les petites communautés vont être regroupées avec les plus grandes même si elles sont bien formées. Et c'est effectivement ce que l'on voit : l'algorithme renvoie un nombre optimal de clusters de trois. Or les deux clusters, *pur-seul-simple* d'un côté et *impécunieux-fauché-désargenté-pauvre* de l'autre, sont très vite « absorbés » dans les autres clusters et passent quasiment inaperçus.

D'autre part, on remarque que les synonymes avec peu de relations (*pincé* : 1 ; *séché* : 2 ; et aussi *asséché*, *essuyé* et *desséchant* : 1…) ne sont pas traités : ils sont raccordés bien en aval, vers la fin du dendrogramme de façon incohérente.

Si nous souhaitons avoir une vue sous forme de six regroupements et non de dendrogramme hiérarchique, nous obtenons le graphique de la figure 8 (ci-après).

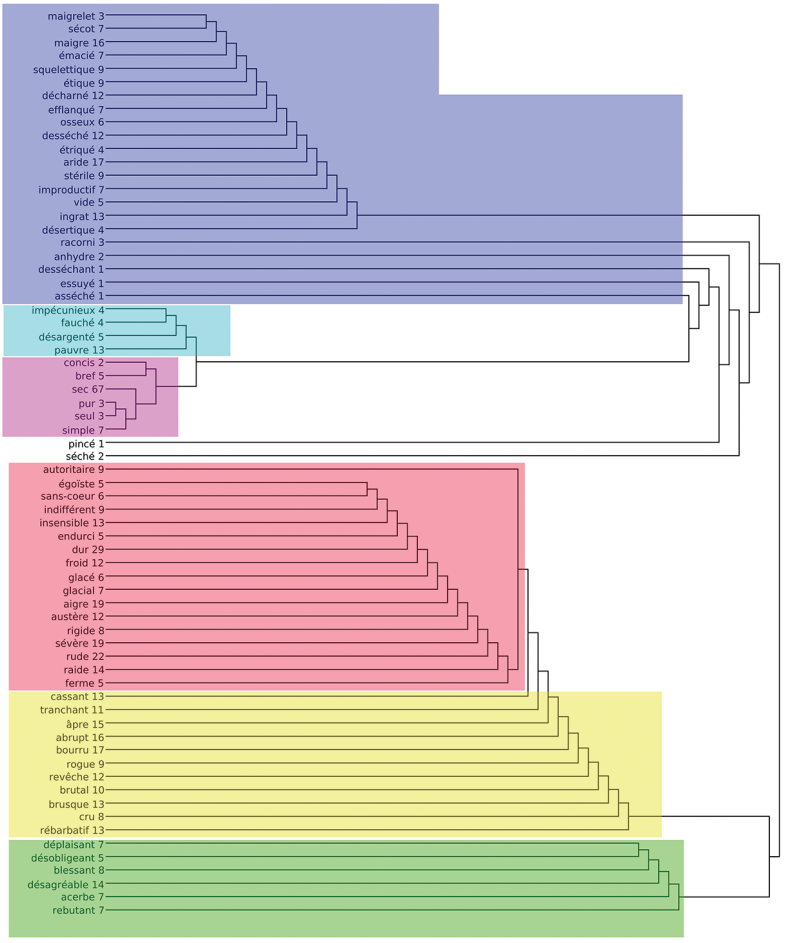


Figure 7. Regroupement des nœuds du graphe de *sec* avec l'algorithme *Fastgreedy*

Nous y voyons bien les quatre principaux regroupements :

1. bleu : *désagréable*, *blessant*, *rebutant* ;
2. vert : *dur*, *sévère*, *rude* ;
3. rouge : *pauvre*, *simple*, *désargenté* ;
4. jaune : *aride*, *desséché*, *maigre*.

Les deux derniers regroupements sont plus difficiles à déceler puisqu'ils contiennent un seul sommet :

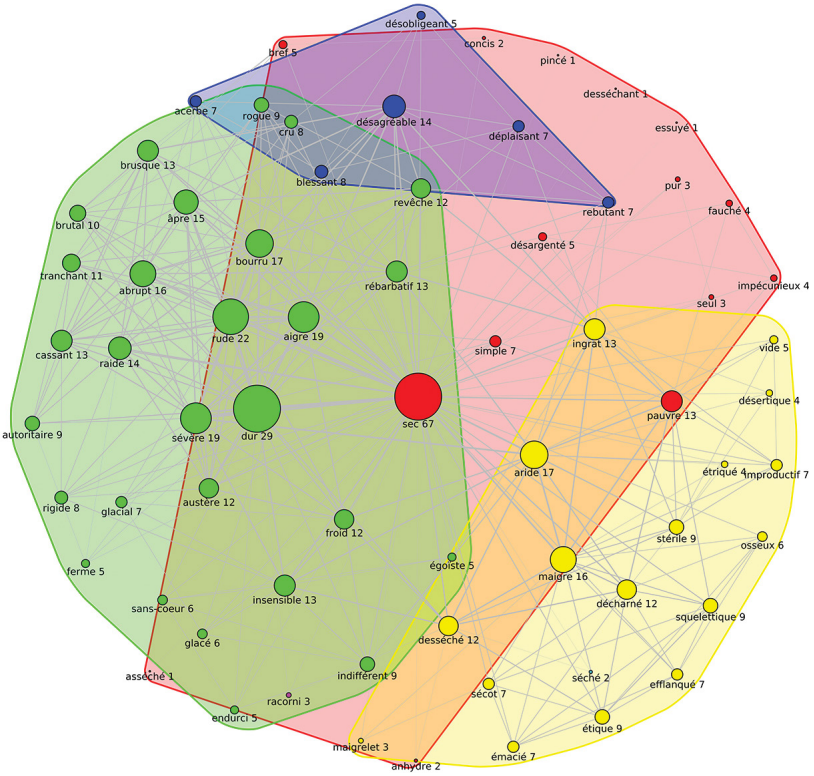


Figure 8. Graphe de *sec* avec l'algorithme *Fastgreedy* avec six clusters

5. *racorni* en magenta (en bas et légèrement à gauche dans le graphique) ;
6. *séché* en bleu clair (en bas et légèrement à droite dans le graphique)[22].

Cela correspond parfaitement à une coupe droite verticale dans le graphique du dendrogramme à l'endroit où l'on coupe six lignes horizontales.

22. On note que *séché* n'est relié dans la figure 8 qu'à *sec* et *desséché* selon la méthode *Fastgreedy*. Comme le *DES* évolue tous les mois, dans l'interface actuelle, *séché* et *sec* ont en commun non seulement *desséché* mais aussi *asséché* et *desséchant*. *Fastgreedy* donnerait peut-être autre chose avec ces nouvelles données.

3.4. L'algorithme Walktrap

Cette méthode issue de la publication de Pons & Latapy (2005) est une approche fondée sur une marche aléatoire. L'idée générale[23] est que si vous effectuez une marche aléatoire dans le graphique, en choisissant sur chaque sommet un des chemins vers un autre sommet de façon arbitraire, alors votre marche est plus susceptible de rester dans la même communauté parce qu'il n'y a que quelques chemins qui mènent en dehors de celle-ci.

Cet algorithme est donc fondé sur le principe de la marche aléatoire comme l'algorithme *Infomap*. Mais ici on va se focaliser sur la probabilité pour chaque sommet d'être le passage obligé du marcheur s'il est très connecté aux autres sommets (son degré est élevé). Au départ, chaque sommet est une communauté, qui sont ensuite regroupées étape par étape jusqu'à former une seule communauté. Puis la mesure de la modularité permet de couper le dendrogramme pour obtenir la partition la plus adaptée.

Walktrap fait de courtes marches aléatoires de 3-4-5 pas (selon l'un de ses paramètres) et utilise les résultats de ces marches aléatoires pour fusionner des communautés séparées d'une manière ascendante comme *Fastgreedy*.

L'algorithme détecte de façon optimale six clusters (champ « optimal_count ») représentés sur la figure 9 (ci-contre) assez représentatifs de la vision lexicographique développée précédemment (voir section 2.2) :

1. qui ne contient pas d'eau (« la route était sèche ») : cluster bleu foncé ;
2. *maigre*, *décharné* (« un vieil homme sec et ridé ») : cluster vert ;
3. *stérile*, *improductif* (« rester sec à un examen ») : à nouveau le cluster bleu foncé ;
4. *insensible*, *sévère*, *égoïste* (« un homme au cœur sec ») : cluster jaune (sauf *sévère*) ;
5. *brusque*, *abrupt* (« donner un coup sec ») : cluster rose ;
6. *simple*, *seul* (« avoir un atout sec dans son jeu ») : cluster magenta.

Les 1er et 3e groupes de sens sont réunis dans le cluster bleu foncé. Le sens de *impécunieux*, *fauché*, *désargenté* en bleu clair qui est aussi un sens de *sec* (manquer d'argent = « être sec ») n'apparaît pas dans l'analyse lexicographique.

---

23. Voir « What are the differences between community detection algorithms in igraph ? » ; et L. Beauguitte, « Sur la détection de communautés en général et avec R en particulier ».

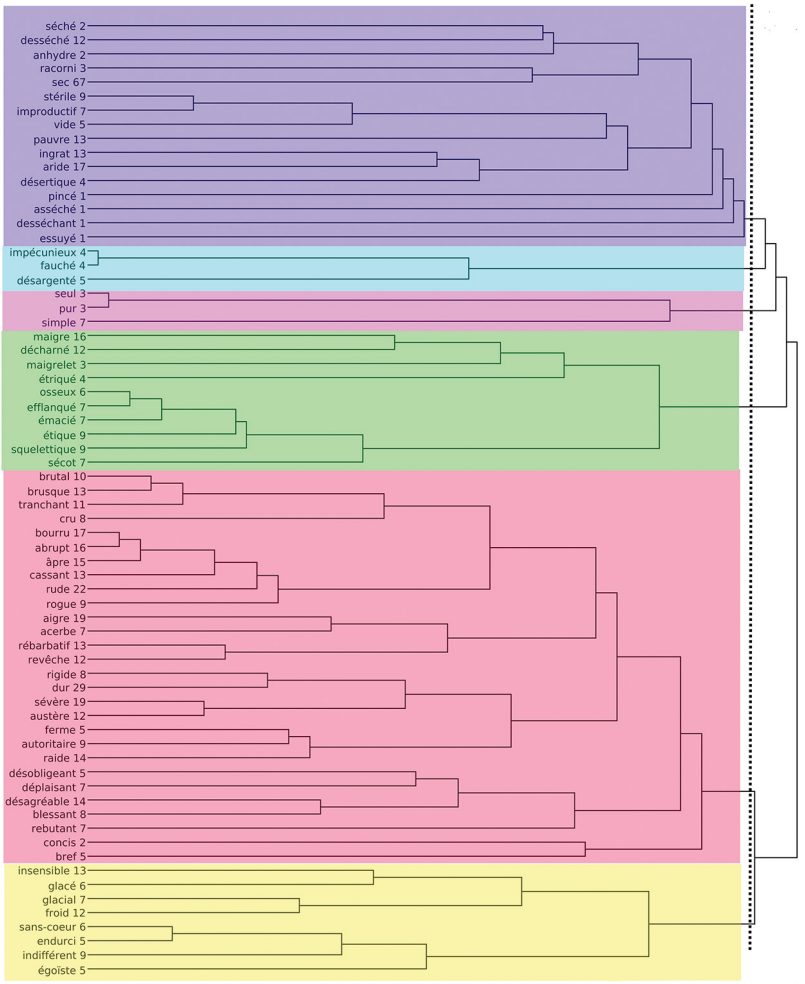


Figure 9. Regroupement des nœuds du graphe de *sec* avec l'algorithme *Walktrap*

Nous pouvons mieux visualiser les six regroupements dans le graphique de la figure 10 (ci-après).

3.5. L'algorithme Spinglass

Cette méthode exposée dans la publication de Reichardt & Bornholdt (2006) est fondée sur la recherche des graphes hamiltoniens, du nom de son concepteur William Rowan Hamilton, astronome royal en Irlande au XIX^{e} siècle. « […] un chemin hamiltonien d'un graphe orienté ou non

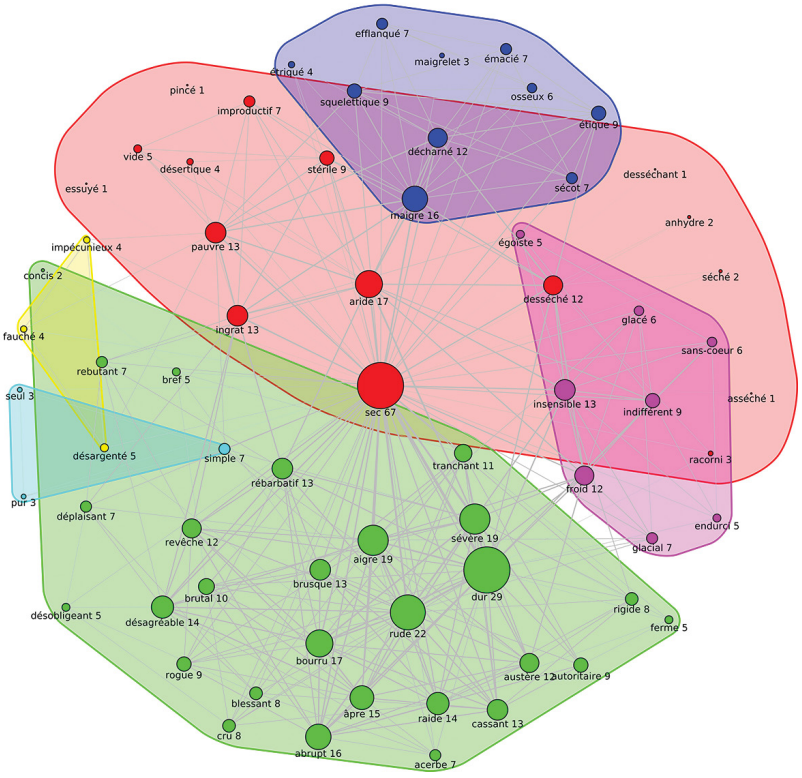


Figure 10. Graphe de *sec* avec les six clusters de l'algorithme *Walktrap*

orienté est un chemin qui passe par tous les sommets une fois et une seule »[24]. Une analogie peut être proposée avec la tournée d'un facteur pour distribuer le courrier qui s'arrête une et une seule fois devant chaque boîte aux lettres. « Un cycle hamiltonien est un chemin hamiltonien qui est un cycle », c'est-à-dire que le sommet d'arrivée est celui de départ. « Un graphe hamiltonien est un graphe qui possède un cycle hamiltonien »[25].

On va donc rechercher tous les sous-graphes dans lesquels il existe un chemin qui passe par tous les nœuds une seule fois avant de revenir à son point de départ. Autrement dit, ces sous-graphes représentent des communautés, des groupes de nœuds fortement interconnectés qui ne

24. « Graphe hamiltonien », page de présentation sur Wikipédia : https://fr.wikipedia.org/wiki/Graphe_hamiltonien.
25. *Ibid.*

sont que faiblement connectés au reste du réseau. Le calcul part donc du principe qu'il faut regrouper ce qui est lié et séparer ce qui ne l'est pas. Il doit donc : a) récompenser les bords internes entre les nœuds d'un même groupe (dans le même état de rotation) et b) pénaliser les bords manquants (les non-liens) entre les nœuds d'un même groupe. En outre, il devrait c) pénaliser les arêtes existantes entre différents groupes (nœuds dans différents états de rotation) et d) récompenser les non-liens entre différents groupes.

Dans notre exemple, nous obtenons le résultat de la figure 11 :

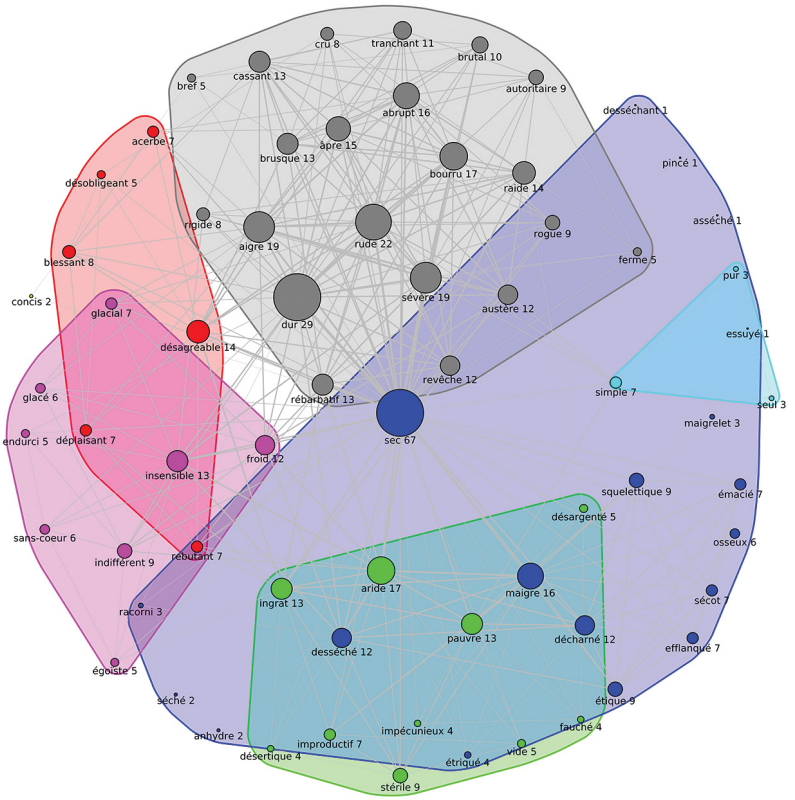


Figure 11. Regroupement des nœuds du graphe de *sec* avec l'algorithme *Spinglass*

Nous obtenons six clusters sensiblement identiques aux méthodes précédentes.

1. *simple*, *pur* et *seul* en bleu clair ;
2. *sévère*, *dur*, *rude* en gris ;

3. *ingrat*, *aride*, *pauvre*, *stérile* en vert ;
4. *désagréable*, *blessant*, *rebutant*, *acerbe* en rouge ;
5. *insensible*, *froid*, *glacial* en magenta ;
6. *maigre*, *décharné*, *desséché* en bleu foncé.

On remarque toutefois un minuscule cluster en jaune avec *concis* difficilement compréhensible.

3.6. L'algorithme Edge_betweenness

Il s'agit d'un partitionnement issu de Girvan & Newman (2002) et fondé sur l'intermédiarité des liens. À chaque itération, l'intermédiarité des liens est calculée. Ceux ayant le score le plus élevé sont supprimés. L'opération est répétée jusqu'à rendre le graphe non connexe[26].

L'intermédiarité est « égale au nombre de fois que ce sommet est sur le chemin le plus court entre deux autres nœuds quelconques du graphe. Un nœud possède une grande intermédiarité s'il a une grande influence sur les transferts de données dans le réseau, sous l'hypothèse que ces transferts se font uniquement par les chemins les plus courts »[27].

C'est un processus de décomposition hiérarchique où les arêtes sont supprimées dans l'ordre décroissant de leur intermédiarité. Ceci est motivé par le fait que les arêtes reliant les différents groupes sont plus susceptibles d'être contenues dans de multiples chemins les plus courts simplement parce que, dans de nombreux cas, elles sont la seule option pour passer d'un groupe à un autre. Cette méthode donne de bons résultats, mais elle est très lente en raison de la complexité des calculs d'entre-distance et parce que les scores d'entre-distance doivent être recalculés après chaque enlèvement d'arête. Les graphes à traiter avec cette méthode ne doivent pas dépasser environ 700 sommets et 3 500 arêtes : c'est la limite supérieure de la taille des graphiques qu'il est possible d'analyser avec cette approche[28].

Avec cette méthode, nous obtenons le graphique de la figure 12 :

26. Voir « What are the differences between community detection algorithms in igraph ? ».
27. « Centralité intermédiaire », page de présentation sur Wikipédia : https://fr.wikipedia.org/wiki/Centralit%C3%A9_interm%C3%A9diaire.
28. Voir « What are the differences between community detection algorithms in igraph ? ».

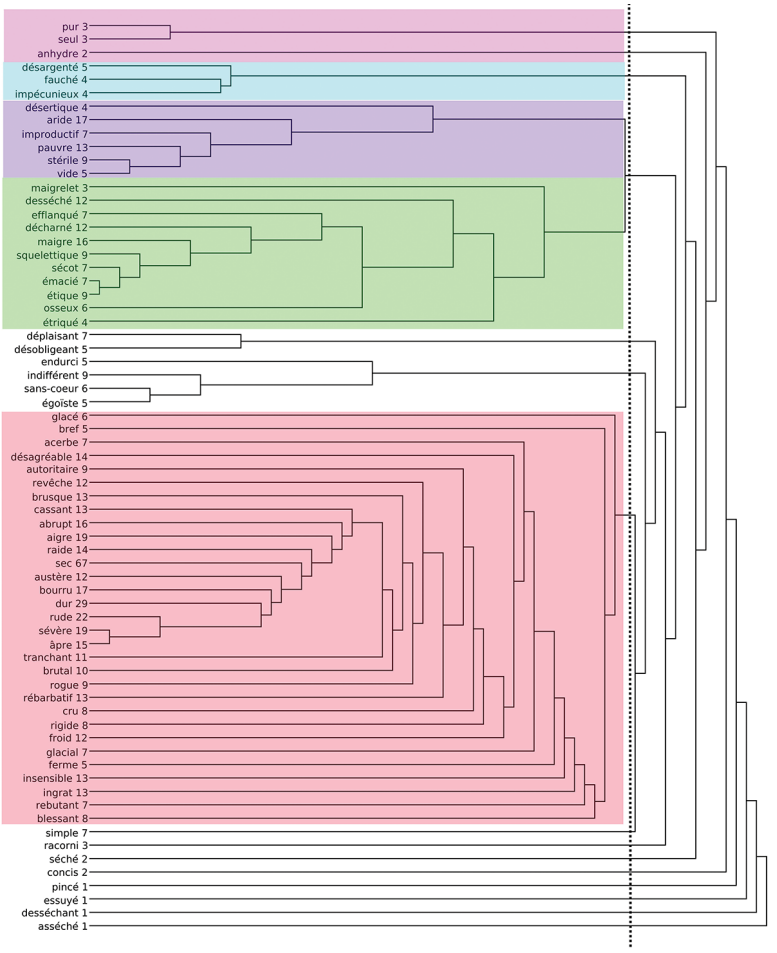


Figure 12. Regroupement des nœuds du graphe de *sec* avec l'algorithme *Edge_betweenness*

De façon optimale, quinze clusters sont détectés. En réalité, en étudiant ce résultat, on se rend compte que cinq de ces clusters, colorés sur l'image, correspondent à la classification des sens décrite dans la section 2.2 telle que proposée par Venant (2004) et Jacquet *et al.* (2005). Les deux groupes de synonymes non colorés n'ont pas été pris en compte correctement :

- d'une part, *déplaisant*, *désobligeant*, *endurci*, *indifférent*, *sans-cœur*, *égoïste* devraient plutôt rejoindre le cluster rose et ce dernier aurait dû être séparé en deux, *insensible*, *sévère*, *égoïste* d'une part et *brusque*, *abrupt* de l'autre ;

- d'autre part, en bas du graphique :
  - *simple* aurait dû rejoindre le cluster magenta,
  - *racorni*, *séché*, *essuyé*, *desséchant* et *asséché* auraient dû rejoindre le cluster bleu foncé,
  - enfin, *concis* et *pincé* le cluster rose (*insensible*, *sévère*…).

Parmi ces synonymes en bas du graphique, certains n'ont qu'une, voire deux, liaisons synonymiques (sur la figure 12, « séché 2 » veut dire que *séché* n'a que deux liaisons synonymiques, « asséché 1 » veut dire qu'*asséché* n'en a qu'une seule). Cela explique en partie du moins le traitement moins efficace réalisé par cette méthode.

Comme pour les dendrogrammes des algorithmes *Fastgreedy* et *Walktrap*, si nous coupons le dendrogramme *Edge_betweenness* avec une ligne verticale de façon à produire six clusters, nous obtenons le graphique de la figure 13 (ci-contre) :

Le graphique montre :

- un regroupement principal en rouge qui comprend quasiment tous les mots et tous les sens de *sec* : *rude*, *maigre*, *bref*, *indifférent* sans distinction ;
- cinq autres regroupements représentés par *un seul mot* chacun : *asséché*, *desséchant*, *essuyé*, *pincé* et *concis*.

Ce découpage ne peut pas ainsi être représentatif des différentes définitions de *sec* de la section 2.2.

3.7. L'algorithme Eigenvector

Cette méthode se fonde sur la publication de Newman (2006). Elle utilise la notion de vecteurs propres pour détecter la structure de la communauté. Cependant, elle ne fonctionne pas bien sur des *graphes dégénérés*. « Un graphe est *k*-dégénéré si tout sous-graphe contient un nœud de degré inférieur ou égal à *k*, et la dégénérescence d'un graphe est le plus petit *k* tel qu'il est *k*-dégénéré »[29]. Dans les graphes d'adjacence du *DES* pour une vedette, nous avons une proportion non négligeable de graphes 1-dégénérés c'est-à-dire qu'un synonyme est relié uniquement à la vedette.

29. « Dégénérescence (théorie des graphes) », page de présentation sur Wikipédia : https://fr.wikipedia.org/wiki/D%C3%A9g%C3%A9n%C3%A9rescence_(th%C3%A9orie_des_graphes).

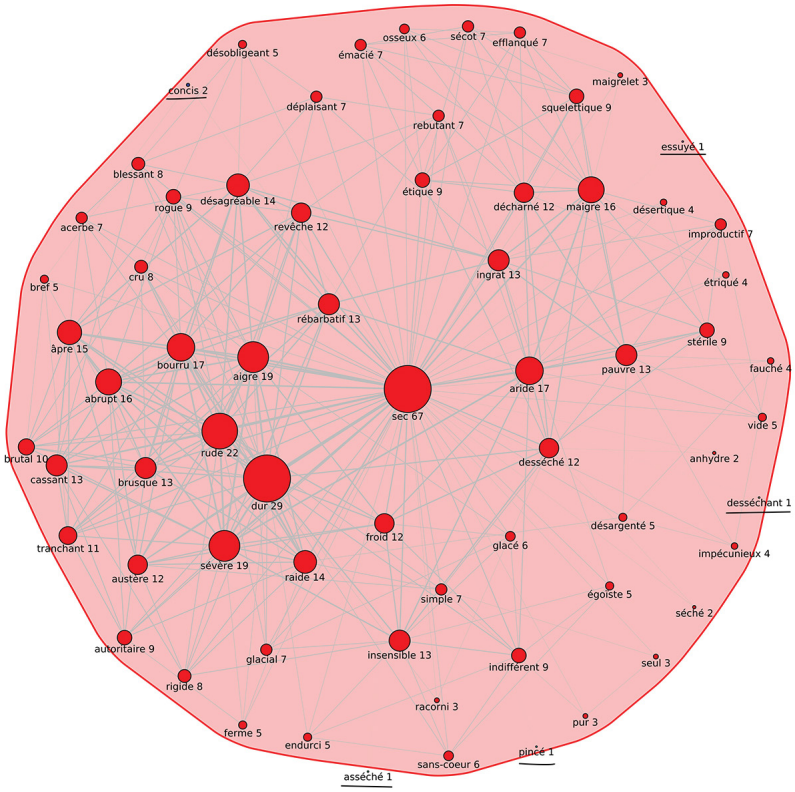


Figure 13. Graphe de *sec* avec les six clusters de l'algorithme *Edge_betweenness*

Effectivement, comme nous pouvons le constater dans la figure 14 (ci-après), même si nous précisons un nombre de clusters de six, nous n'en obtenons que deux, pas suffisamment représentatifs de la classification lexicographique de Venant (2004) et Jacquet *et al.* (2005).

3.8. L'algorithme de modularité optimale

Cette fonction, qui s'appuie sur Good *et al.* (2010), utilise le « GNU Linear Programming Kit » (GLPK) pour résoudre un problème d'optimisation de grands entiers afin de trouver le score de modularité optimal et la structure de communauté correspondante. Le GLPK est un solveur libre et *open source* constitué d'un ensemble d'outils et de méthodes utilisables à partir d'un langage de programmation (C, Java, Python)

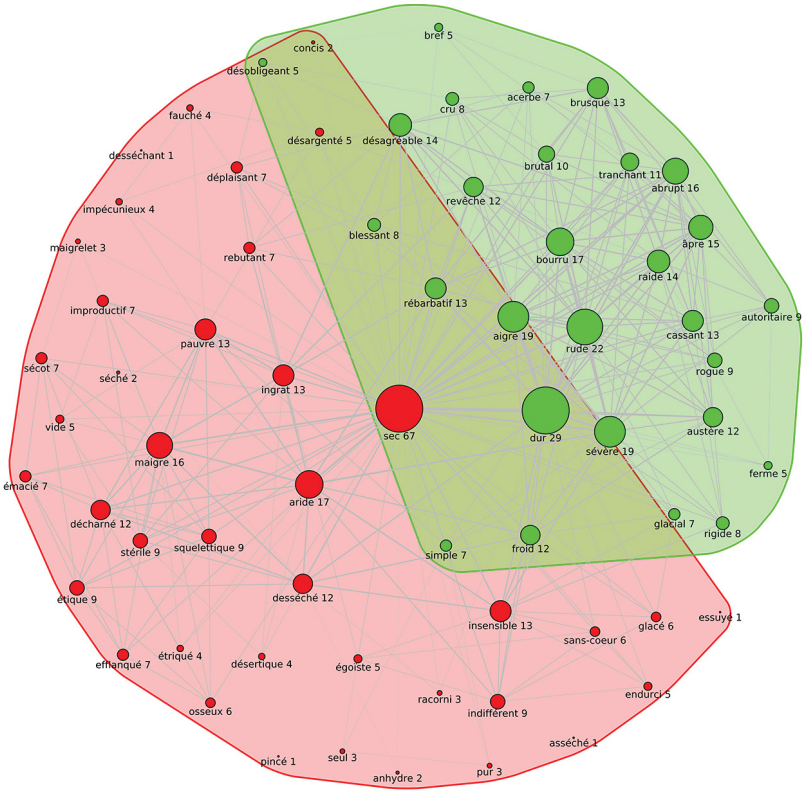


Figure 14. Regroupements des nœuds du graphe de *sec* avec l'algorithme *Eigenvector*

pour trouver la solution de problèmes complexes. Dans cet algorithme, nous cherchons à calculer les clusters par lesquels la modularité, qui mesure la qualité du partitionnement, sera optimale[30]. Cet algorithme fonctionne pour des graphiques de moins de 100 sommets car son exécution est longue[31].

Sur notre exemple qui possède moins de 100 sommets et après plusieurs minutes d'exécution, nous obtenons le résultat de la figure 15 :

30. Voir « Modularité (réseaux) », page de présentation sur Wikipédia : https://fr.wikipedia.org/wiki/Modularit%C3%A9_(r%C3%A9seaux).
31. Voir « Optimal Modularity », in *Python-igraph Manual* : https://igraph.org/python/doc/igraph.Graph-class.html#community_optimal_modularity.

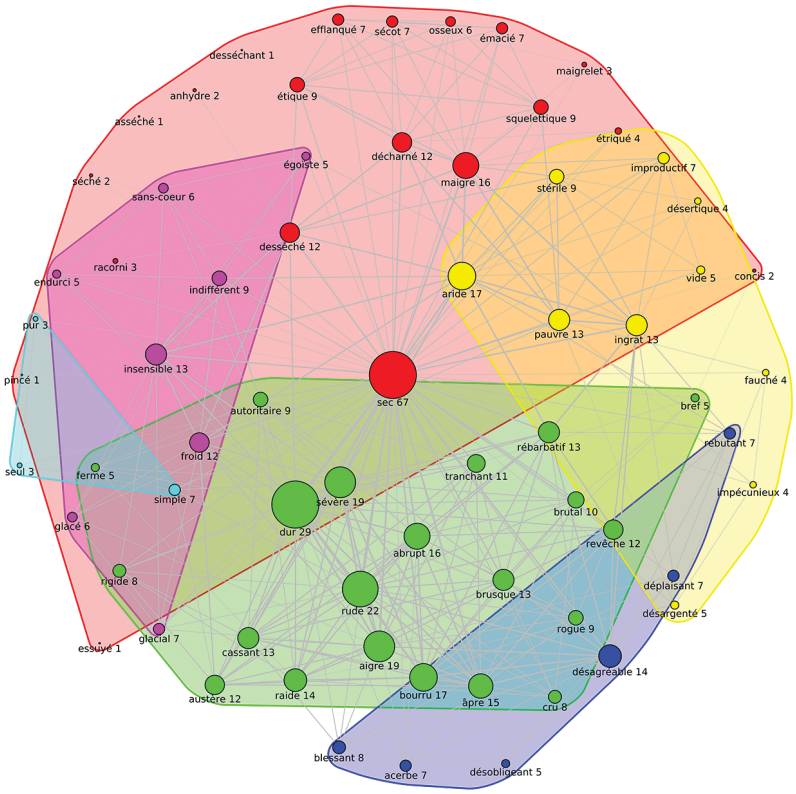


Figure 15. Regroupement des nœuds du graphe de *sec* avec l'algorithme de modularité maximale

Nous obtenons un résultat assez proche de l'algorithme de Louvain :

1. *dur*, *sévère*, *rude* de couleur verte ;
2. *maigre*, *décharné*, *squelettique* en rouge ;
3. *aride*, *pauvre*, *ingrat*, *stérile* en jaune ;
4. *désagréable*, *blessant*, *rebutant* en bleu foncé ;
5. *insensible*, *froid*, *indifférent* en magenta ;
6. *simple*, *seul* et *pur* en bleu ciel.

Ce résultat caractérise dans son ensemble les différents sens de *sec* donnés dans le dictionnaire *TLF*.

4. Comparaison des algorithmes

Si nous résumons l'essentiel des différentes méthodes[32], et de façon générale : a) si les ressources de calcul sont suffisantes avec des graphes ne dépassant pas 700 sommets et 3 500 arêtes, *Edge_betweenness* donne le meilleur résultat ; b) si la modularité est importante, alors il est préférable d'utiliser un de ces algorithmes :

- si le graphique est particulièrement petit (inférieur à 100 sommets), alors il est préférable d'utiliser la modularité optimale ;
- si on veut obtenir une première vision, les algorithmes *Fastgreedy* ou *Walktrap* seront appropriés ;
- si le graphique a plus de 100 sommets et n'est pas un graphique dégénéré, et que l'on souhaite quelque chose de plus précis que les méthodes *Fastgreedy* ou *Walktrap*, il est judicieux d'utiliser la méthode *Eigenvector* ;
- si on recherche une solution similaire au *clustering* K-means[33], la méthode *Spinglass* est plus pertinente.

Enfin, c) *Infomap* s'applique également sur les graphes orientés. Cela peut être intéressant si nous décidons de rendre les arêtes de notre graphe non symétriques, c'est-à-dire qu'un mot pourra être synonyme d'un autre sans que la réciproque soit vraie. C'est une éventualité envisageable pour les liaisons à caractère insultant, comme *noir* et *nègre* ou des liaisons d'hyponymie.

En ce qui concerne notre exemple avec *sec*, l'essentiel des sens de cet adjectif transparaît de façon relativement conforme à la vision lexicographique : c'est le cas dans l'espace sémantique du *DES* et dans les méthodes de regroupements *Multilevel*, *Infomap*, *Walktrap*, *Spinglass* et modularité optimale. Pour les méthodes *Edge_betweenness*, *Fastgreedy* et *Eigenvector*, le résultat n'est pas représentatif des classements de sens sur lesquels nous nous sommes basés puisqu'il est trop éloigné des différentes définitions présentes dans les dictionnaires.

Qu'en est-il des fonctions qui permettent de comparer ces diverses méthodes ? Nous donnent-elles des résultats qui confirment ce que nous venons de déduire manuellement ?

La librairie igraph nous propose cinq fonctions de comparaison[34] :

---

32. Voir « Community Detection in Python ».
33. Voir « K-moyennes », page de présentation sur Wikipédia : https://fr.wikipedia.org/wiki/K-moyennes.
34. Voir « Compare Communities », in *Python-igraph Manual* : https://igraph.org/python/doc/igraph.clustering-module.html#compare_communities.

- La variation de la métrique de l'information qui se réfère à Meilă (2003). Elle tient compte des sommets communs aux deux clusters pondérés de la somme des logarithmes des poids de chacun des clusters[35]. Elle est assez proche de l'information mutuelle décrite ci-dessous.

- L'information mutuelle normalisée selon Danon *et al.* (2005) : ce concept[36] est fortement lié à l'entropie qui mesure l'incertitude sur l'état d'un système. Un exemple très simple, illustrant l'entropie sur le site *Le webinet des curiosités*, explique clairement le principe[37].

- La distance de séparation (*split-join*) selon Van Dongen (2000) : chaque ensemble de la partition A est évalué par rapport à tous les ensembles de la partition B. Ensuite, pour chaque ensemble de la partition A, on trouve l'ensemble le mieux adapté dans la partition B et on calcule la taille du chevauchement. La concordance est quantifiée par la taille du chevauchement entre les deux ensembles. Enfin, les tailles maximales de chevauchement pour chaque ensemble de la partition A sont additionnées et soustraites du nombre d'éléments de la partition A.

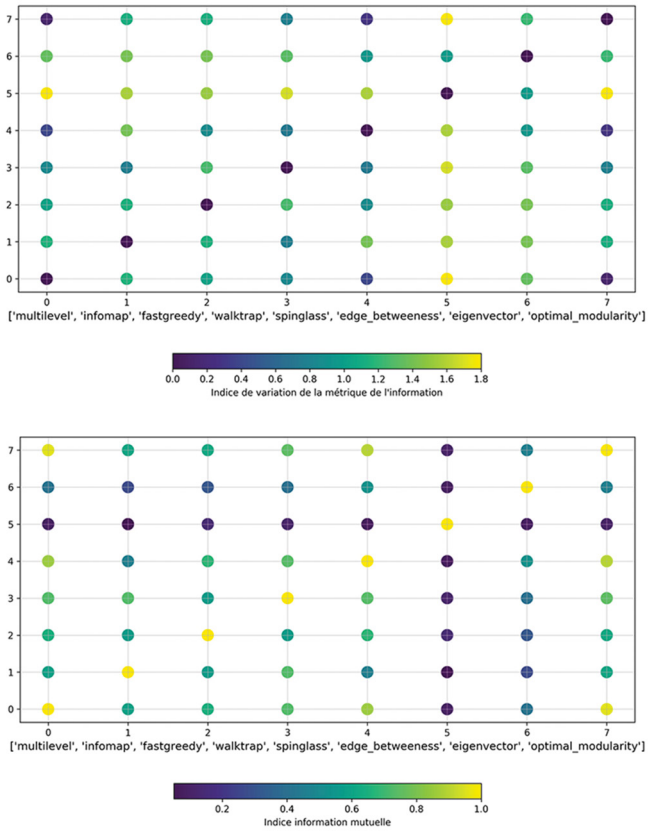
- L'indice de Rand d'après Rand (1971) mesure la consistance entre deux regroupements[38]. Le principe est assez simple : pour chaque paire de sommets du graphe (s1,s2) et deux regroupements g1 et g2, nous comptons si la paire (s1,s2) est dans g1 et g2, si elle est dans g1 mais absente de g2 et inversement, enfin si elle est absente de g1 et de g2. Le premier et le dernier cas de figure vont représenter un accord entre g1 et g2. Les deuxième et troisième cas vont exprimer un désaccord.

- L'indice de Rand ajusté décrit dans Hubert & Arabie (1985) reprend le principe ci-dessus en y insérant un regroupement réalisé par un modèle aléatoire.

---

35. Voir « Variation of Information », page de présentation sur Wikipédia : https://en.wikipedia.org/wiki/Variation_of_information, qui détaille simplement la formule.
36. Également expliqué dans la page de présentation de Wikipédia, « Mutual Information » : https://en.wikipedia.org/wiki/Mutual_information.
37. Voir Xochipilli, « Au fait c'est quoi l'entropie », 14 avril 2013, billet publié sur le blog *Le webinet des curiosités* : https://webinet.cafe-sciences.org/articles/au-fait-cest-quoi-lentropie/.
38. Voir « Rand Index » et « Adjusted Rand Index », page de présentation sur Wikipédia : https://en.wikipedia.org/wiki/Rand_index.

L'application de ces cinq fonctions de comparaison décrites brièvement ci-dessus sur les huit méthodes de regroupement étudiées est représentée par la figure 16. En ligne et en colonne, numérotées de 0 à 7, nous avons les huit méthodes de regroupement étudiées et le point de couleur à l'intersection représente le résultat de la fonction de comparaison désignée dans la légende. La diagonale qui compare chaque cluster avec lui-même est pour certaines fonctions à 0 (concordance parfaite) comme l'indice de la variation de la métrique de l'information et la fonction *split-join*. Pour d'autres cette concordance parfaite se traduit par le chiffre 1 sur la diagonale pour l'information mutuelle, l'index de Rand et l'index de Rand ajusté. Dans le premier cas, un indice de comparaison entre deux clusters proches de 0 marquera une similitude importante. Dans le second cas, ce sera un chiffre proche de 1 qui marquera cette similitude.



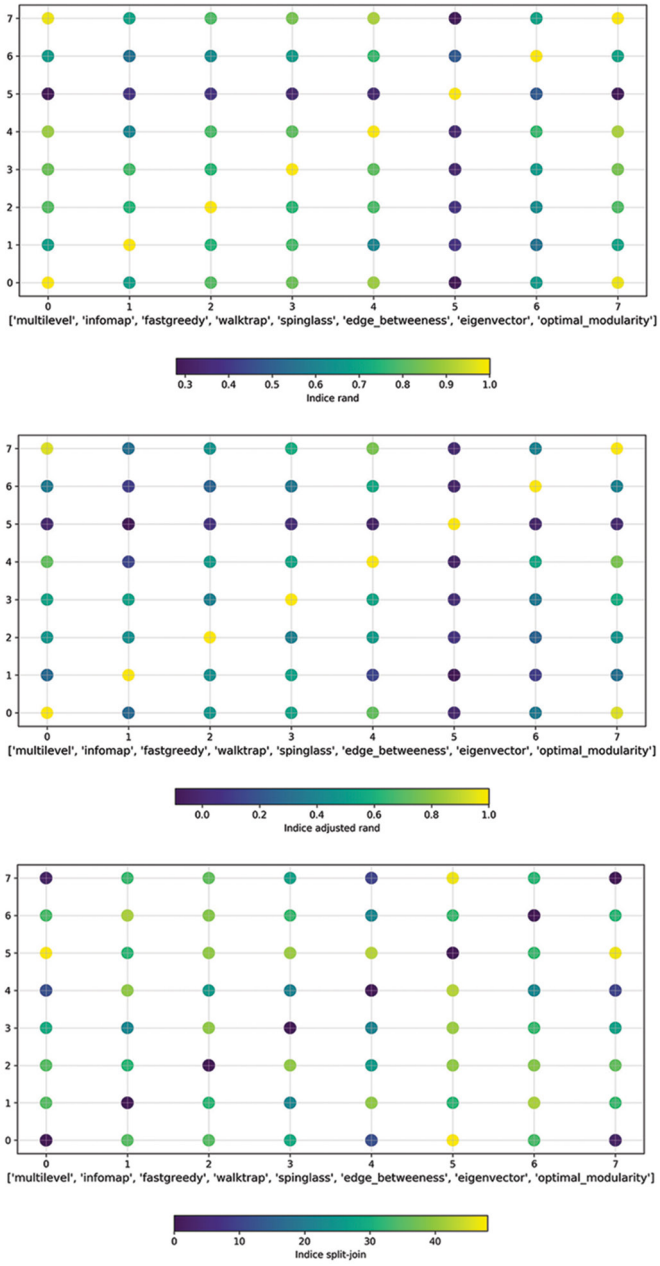


Figure 16. Comparaison des huit algorithmes de regroupement

Ces fonctions de comparaison nous indiquent que, d'une part, les clusters *Multilevel*, *Spinglass* et *Optimal modularity* et, d'autre part *Infomap*, *Walktrap* sont proches. Cette conclusion conforte une partie de ce que nous avons déduit précédemment à partir des graphiques. En effet, nous avons conclu que ces cinq regroupements sont les plus représentatifs et le calcul de comparaison nous confirme que deux catégories d'algorithmes les incluant sont similaires. Si les fonctions de comparaison avaient donné *Eigenvector* proche de *Multilevel* par exemple, nous nous serions posé la question de savoir pourquoi *Eigenvector* donne un résultat inacceptable d'un point de vue lexicographique.

Notre déduction manuelle précédente est donc en partie vérifiée : il est logique que ces cinq méthodes de regroupement ensemble satisfassent notre déduction manuelle et lexicographique.

Néanmoins, les fonctions de comparaison ne nous expliquent pas pourquoi ces cinq algorithmes sont ceux qui se rapprochent le plus de la vision lexicographique issue des dictionnaires.

5. Conclusion

Les méthodes de calcul utilisées dans cette étude préliminaire reposent sur la thèse de Ploux & Victorri (1998) selon laquelle il est possible de représenter le sens géométriquement. Nous avons ici cherché à tester plusieurs de ces méthodes pour voir en quoi les représentations obtenues coïncident avec le découpage lexicographique, et dans quelle mesure ces résultats sont homogènes ou bien hétérogènes.

Dans notre étude, nous avons vu que cinq des huit méthodes de regroupement proposées par la librairie igraph apportent une visualisation qui est homogène avec la représentation lexicographique de la polysémie de *sec*. En effet, elles donnent des résultats similaires aux différentes catégories de sens issues du *TLF*. Nous concluons que les fonctions automatiques de comparaison de clusters fournis par igraph confirment la convergence des résultats obtenus dans la mesure où on retrouve une classification relativement homogène. Il serait nécessaire de procéder à une analyse de plus grande échelle pour voir d'une part si l'homogénéité se confirme ou non, et d'autre part si les méthodes qui donnent des résultats divergents se comportent de la même façon sur d'autres mots vedettes. Cela permettrait de déterminer de manière plus sûre si ces méthodes sont inadaptées et en quoi. Nous avons par exemple remarqué que la méthode *Eigenvector* produit une représentation divergente non fiable car elle ne fonctionne pas sur des graphes dégénérés, comme cela est souvent le cas dans les graphes représentant les synonymes d'une vedette dans le *DES*.

Deux autres pistes peuvent être étudiées en rapport avec l'aspect mathématique. Ces deux pistes pourront s'appuyer sur une interface en ligne affichant le graphe d'adjacence et le choix des différentes méthodes de regroupement pour permettre aux linguistes intéressés de réaliser ces tests de façon autonome. Une première piste consiste à analyser la structure de départ du graphe. En effet, on constate la présence de plusieurs relations synonymiques « solitaires », c'est-à-dire qu'un synonyme n'est relié qu'à la vedette mais à aucun autre synonyme du graphe : ce sont les sommets qui n'ont qu'un lien. Soit cette liaison doit être enrichie par d'autres liaisons synonymiques avec les autres sommets du graphe, soit elle doit être supprimée, n'étant pas vraiment à propos. Pour cela, le calcul automatique des liens probables existants[39] pourrait être adapté pour traiter spécifiquement ces relations « solitaires » afin d'intégrer le synonyme isolé à un cluster existant. Une seconde piste consiste à tenir compte des dernières avancées dans le domaine de l'apprentissage automatique. En effet, les projets camemBERT[40] et FlauBERT[41] permettent d'obtenir des indicateurs de similitude entre les mots suite au traitement d'énormes corpus. La valeur de cet indicateur de similitude appliqué aux entrées du *DES* pourrait être utilisée pour confirmer ou infirmer des liens synonymiques.

Enfin, pour exploiter de manière linguistique cette étude mathématique, il serait pertinent d'envisager une perspective psycholinguistique et une perspective de linguistique de corpus. Il est possible notamment d'envisager de réaliser des tests d'ordre psycholinguistique pouvant apporter une dimension pratique à la question du regroupement cognitif des sens. En effet, comme la conclusion de Polguère (2014) l'indique, il s'agirait d'interroger les locuteurs pour dégager les regroupements de sens validés par des usagers de la langue, puis de les confronter aux résultats de regroupement obtenus grâce aux différents calculs. Autrement dit, on peut se demander dans quelle mesure les perceptions linguistiques de la polysémie chez les usagers coïncident avec les regroupements et visualisations proposés par les calculs mathématiques. Une seconde piste pourrait porter sur des tests avec des corpus pour dégager le comportement en usage de *sec* ou d'autres mots. D'une part, le dictionnaire ne peut pas tout nous apprendre puisqu'il est statique, et

39. Voir « Calcul des liens manquants probables », site du *DES* : http://crisco.unicaen.fr/dictionnaire-electronique-des-synonymes/presentation-du-des/calcul-des-liens-manquants-probables/.
40. https://camembert-model.fr/
41. https://ins2i.cnrs.fr/fr/cnrsinfo/flaubert-la-rescousse-du-traitement-automatique-du-francais

d'autre part les relations synonymiques du *DES* provenant des dictionnaires à l'origine reflète un usage à un moment t et dans un contexte particulier. Dans ce domaine, nous pourrons nous appuyer sur Venant (2007) qui prend en compte l'influence des autres éléments présents dans un corpus autour de la vedette étudiée, autrement dit il s'agit de prendre en compte un espace cotextuel couplé à l'espace sémantique.

Références

Sitographie

(dernière consultation : le 9 octobre 2020)

« Algorithme de Louvain », page de présentation sur Wikipédia : https://fr.wikipedia.org/wiki/M%C3%A9thode_de_Louvain.

Beauguitte L. (2019), « Sur la détection de communautés en général et avec R en particulier », 10 avril 2019, publication dans le carnet de recherche du groupement de recherche « Analyse de réseaux en sciences humaines et sociales » : https://arshs.hypotheses.org/1314#more-1314.

« Bron-Kerbosh Algorithm », page de présentation sur Wikipédia : https://en.wikipedia.org/wiki/Bron%E2%80%93Kerbosch_algorithm.

« Calcul des liens manquants probables », site du *DES* : http://crisco.unicaen.fr/dictionnaire-electronique-des-synonymes/presentation-du-des/calcul-des-liens-manquants-probables/.

« Centralité intermédiaire », page de présentation sur Wikipédia : https://fr.wikipedia.org/wiki/Centralit%C3%A9_interm%C3%A9diaire.

Chardon L., « Espace sémantique avec CURIEUX : tutoriel pas à pas », 23 avril 2019, site du *DES* : http://crisco.unicaen.fr/dictionnaire-electronique-des-synonymes/actualites-des/espace-semantique-de-curieux-tutoriel-pas-a-pas-964401.kjsp?RH=1530619460865.

« Community Detection in Python », site *Yoyo in Wanderland* : https://yoyoinwanderland.github.io/Community-Detection.

« Community Infomap », in *Python-igraph Manual* : https://igraph.org/python/doc/igraph.Graph-class.html#community_infomap.

« Compare Communities », in *Python-igraph Manual* : https://igraph.org/python/doc/igraph.clustering-module.html#compare_communities.

« Décomposition en valeurs singulières », page de présentation sur Wikipédia : https://fr.wikipedia.org/wiki/D%C3%A9composition_en_valeurs_singuli%C3%A8res.

« Dégénérescence (théorie des graphes) », page de présentation sur Wikipédia : https://fr.wikipedia.org/wiki/D%C3%A9g%C3%A9n%C3%A9rescence_(th%C3%A9orie_des_graphes).

« Edge Connectivity », in *R igraph Manual Pages* : https://igraph.org/r/doc/edge_connectivity.html.

« étude-SEC », serveur GIT de l'université de Caen : https://git.unicaen.fr/crisco-des-public/etude-sec.

« Fastgreedy », in *Python-igraph Manual* : https://igraph.org/python/doc/igraph.Graph-class.html#community_fastgreedy.

« Graphe hamiltonien », page de présentation sur Wikipédia : https://fr.wikipedia.org/wiki/Graphe_hamiltonien.

« Infomap community detection understanding », réponse à une question posée sur le site *Stack Overflow* : https://stackoverflow.com/questions/48528648/infomap-community-detection-understanding/54292999.

« K-moyennes », page de présentation sur Wikipédia : https://fr.wikipedia.org/wiki/K-moyennes.

Librairie fanalysis : https://github.com/OlivierGarciaDev/fanalysis.

Librairie igraph : https://igraph.org.

« Matomo (logiciel) », page de présentation sur Wikipédia : https://fr.wikipedia.org/wiki/Matomo_(logiciel).

« Modularité (réseaux) », page de présentation sur Wikipédia : https://fr.wikipedia.org/wiki/Modularit%C3%A9_(r%C3%A9seaux).

« Mutual Information », page de présentation sur Wikipédia : https://en.wikipedia.org/wiki/Mutual_information.

« Optimal Modularity », in *Python-igraph Manual* : https://igraph.org/python/doc/igraph.Graph-class.html#community_optimal_modularity.

« Préfixe », page de présentation sur Wikipédia : https://fr.wikipedia.org/wiki/Pr%C3%A9fixe.

« Présentation du *DES* », site du *DES* : http://crisco.unicaen.fr/dictionnaire-electronique-des-synonymes/presentation-du-des/.

« Rand Index » et « Adjusted Rand Index », page de présentation sur Wikipédia : https://en.wikipedia.org/wiki/Rand_index#Adjusted_Rand_index.

« Routage », page de présentation sur Wikipédia : https://fr.wikipedia.org/wiki/Routage.

« Théorie des graphes », page de présentation sur Wikipédia : https://fr.wikipedia.org/wiki/Th%C3%A9orie_des_graphes.

« Variation of Information », page de présentation sur Wikipédia : https://en.wikipedia.org/wiki/Variation_of_information.

Visualisation 2D de l'espace sémantique d'un mot, site du *DES* : https://crisco2.unicaen.fr/espsem/.

Visualisation 3D de l'espace sémantique de *sec*, site du *DES* : https://crisco2.unicaen.fr/des3d/sec.html.

« What are the differences between community detection algorithms in igraph ? », réponse à une question posée sur le site *Stack Overflow* : https://stackoverflow.com/questions/9471906/what-are-the-differences-between-community-detection-algorithms-in-igraph.

Xochipilli, « Au fait c'est quoi l'entropie », 14 avril 2013, billet publié sur le blog *Le webinet des curiosités* : https://webinet.cafe-sciences.org/articles/au-fait-cest-quoi-lentropie/.

Bibliographie

Bernard J., Seba H. (2018), « Résolution de problèmes de cliques dans les grands graphes », communication au 15e atelier sur la fouille de données complexes (FDC) extraction et gestion des connaissances (EGC 2018), en ligne : https://hal.archives-ouvertes.fr/hal-01886724/document.

Blondel V., Guillaume J.-L., Lambiotte R., Lefebvre E. (2008), « Fast Unfolding of Communities in Large Networks », *Journal of Statistical Mechanics*, nº 10, prépublication en ligne : https://arxiv.org/pdf/0803.0476.pdf.

Chardon L. (2020), « Présentation du *Dictionnaire électronique des synonymes* (*DES*) », en ligne : https://halshs.archives-ouvertes.fr/halshs-02489368/file/DESPresentationExistant.pdf.

Clauset A., Newman M. E. J., Moore C. (2004), « Finding Community Structure in Very Large Networks », *Physical Review E*, vol. 70, nº 6, prépublication en ligne : https://arxiv.org/pdf/cond-mat/0408187.pdf.

Danon L., Díaz-Guilera A., Duch J., Arenas A. (2005), « Comparing Community Structure Identification », *Journal of Statistical Mechanics*, nº 9, prépublication en ligne : https://arxiv.org/pdf/cond-mat/0505245.pdf.

Fortunato S. (2010), « Community Detection in Graphs », *Physics Reports*, vol. 486, nº 3-5, p. 75-174, prépublication en ligne : https://arxiv.org/pdf/0906.0612.pdf.

Girvan M., Newman M. E. J. (2002), « Community Structure in Social and Biological Networks », *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, nº 12, p. 7821-7826, en ligne : https://www.pnas.org/content/pnas/99/12/7821.full.pdf.

Good B. H., De Montjoye Y.-A., Clauset A. (2010), « The Performance of Modularity Maximization in Practical Contexts », *Physical Review E*, vol. 81, nº 4, prépublication en ligne : https://arxiv.org/pdf/0910.0165.pdf.

Hubert L., Arabie P. (1985), « Comparing Partitions », *Journal of Classification*, vol. 2, nº 1, p. 193-218.

Jacquet G., Venant F., Victorri B. (2005), « Polysémie lexicale », in *Sémantique et traitement automatique du langage naturel*, P. Enjalbert (dir.), Paris, Lavoisier – Hermes, p. 99-129.

Meilă M. (2003), « Comparing Clusterings by the Variation of Information », in *Learning Theory and Kernel Machines : 16th Annual Conference on Computational Learning Theory and 7th Kernel Workshop, COLT/ Kernel 2003, Washington, DC, USA*, B. Schoelkopf, M. K. Warmuth (dir.), Berlin, Springer (Lecture Notes in Computer Science ; 2777), p. 173-187.

Newman M. E. J. (2006), « Finding Community Structure in Networks Using the Eigenvectors of Matrices », *Physical Review E*, vol. 74, nº 3, prépublication en ligne : https://arxiv.org/pdf/physics/0605087.pdf.

Ploux S., Victorri B. (1998), « Construction d'espaces sémantiques à l'aide de dictionnaires informatisés de synonymes », *TAL*, vol. 39, nº 1, p. 161-182.

Polguère A. (2014), « Principes de modélisation systémique des réseaux lexicaux », communication à la 21e conférence sur le traitement automatique des langues naturelles – TALN 2014, Marseille, 1er-4 juillet 2014, p. 79-90, en ligne : https://www.aclweb.org/anthology/F14-1008.pdf.

Pons P., Latapy M. (2005), « Computing Communities in Large Networks Using Random Walks », in *Computer and Information Sciences*, P. Yolum, T. Güngör, F. Gürgen, C. Özturan (dir.), Berlin, Springer (Lecture Notes in Computer Science ; 3733), p. 284-293.

Poudat C., Landragin F. (2017), *Explorer un corpus textuel*, Louvain-la-Neuve, De Boeck Supérieur (Champs linguistiques).

Rand W. M. (1971), « Objective Criteria for the Evaluation of Clustering Methods », *Journal of the American Statistical Association*, vol. 66, nº 336, p. 846-850.

Reichardt J., Bornholdt S. (2006), « Statistical Mechanics of Community Detection », *Physical Review E*, vol. 74, nº 1, prépublication en ligne : https://arxiv.org/pdf/cond-mat/0603718.pdf.

Rosvall M., Bergstrom C. T. (2008), « Maps of Random Walks on Complex Networks Reveal Community Structure », *Proceedings of the National Academy of Sciences of the United States of America*, vol. 105, nº 4, p. 1118-1123, en ligne : https://www.pnas.org/content/pnas/105/4/1118.full.pdf.

Rosvall M., Axelsson D., Bergstrom C. T. (2009), « The Map Equation », *The European Physical Journal Special Topics*, vol. 178, p. 13-23, en ligne : https://www.mapequation.org/assets/publications/EurPhysJ2010Rosvall.pdf.

Van Dongen D. (2000), « Performance Criteria for Graph Clustering and Markov Cluster Experiments », Technical Report INS-R0012, National Research Institute for Mathematics and Computer Science in the Netherlands, Amsterdam, mai 2000.

Venant F. (2004), « Polysémie et calcul du sens », in *Le poids des mots* (Actes des 7es journées internationales d'analyse statistique des données textuelles, Louvain-la-Neuve, 10-12 mars 2004), Louvain-la-Neuve, Presses universitaires de Louvain, p. 1145-1156.

Venant F. (2007), « La construction du sens : un système complexe dynamique », in *Acta cognitica : cognition, complexité, collectif* (Actes du colloque de l'Association pour la recherche cognitive, Nancy, 28-30 novembre 2007), Vandœuvre-lès-Nancy, Institut national de polytechnique de Lorraine, p. 251-264.