Les communautés dans les graphes et l'accès à l'information par la visualisation du contenu

Bilal YALAOUI
Division D&R en Science de l'Information - CERIST
Rue des 3 frère AISSOU BEN-AKNOUN 16030, Alger
yalaoui@mail.cerist.dz

Résumé : Le présent papier est structuré en deux parties. Dans la première partie, quelques techniques pour la structuration des graphes en communautés sont présentées. La deuxième partie, consiste en l'exploitation des concepts de visualisation et d'analyse structurelle des graphes pour l'accès à l'information modélisés par ces mêmes graphes.

Mot clés : Communautés dans les graphes, visualisation du contenu, Graphes

Introduction

Une communauté est une notion difficile à définir formellement, une signification intuitive est souvent utilisée : Une communauté est un ensemble d'objets partageant un intérêt commun. Ces éléments sont similaires entre eux et dissimilaires aux objets des autres groupes vis-à-vis de cet intérêt.

Dans un graphe, elle correspond à l'existence de groupes de sommets fortement connectés entre eux, par la nature des liens qui les relient, que vers les autres sommets. C'est-à-dire que les éléments d'une communauté doivent être aussi « homogènes » que possible (par la structure du graphe induite), et qu'ils doivent être aussi « différents » que possible (le sous graphe induit comme une unité authentifiable) avec les éléments d'autres communautés.

Ce type de structure intervient dans de nombreux réseaux d'interactions et apporte de l'information sur leur organisation. Elle peut avoir des interprétations différentes suivant le type de réseau à considérer.

Dans les réseaux d'association de termes d'un corpus textuel, le terme communauté est synonyme de « thème ». Nous entendons par thème un des sujets abordés dans le corpus textuel. Ainsi, dans ce type de réseau la structuration en communautés revient à ré- organiser le contenu du corpus en thématiques.

La détermination des communautés revient donc à trouver une partition de l'ensemble des sommets du réseau selon un certain critère prédéfinit. Ce critère doit permettre d'obtenir des classes cohérentes ou homogènes.

1. Quelques notions de base sur les graphes

Dans cette partie du papier, nous rappelons quelques principaux fondements mathématiques issus de la théorie des graphes utiles à la compréhension de la suite de notre travail. Une présentation complète et exhaustive peut être consultée dans les ouvrages de : (C. Berge, 1973), (A. Gibbons, 1985), (Evans J. T. et E. Mineka, 1992), (Jungnickel, 1999) et (R. Diestel, 2000).

Définition 1. Un graphe orienté (directed graph en anglais) G=(V,U) est défini par un ensemble fini dénombrable de sommets (ou nœuds) $V=\{v_i|i=1,\ldots n\}$, noté V(G), et un ensemble d'arcs (appelées aussi liens) $U=\{u_k|k=1,\ldots m\}$, noté U(G). Tout arc u_k est associé à un couple (ensemble ordonné) de sommets (v,w) où v est le sommet de départ et w celui d'arrivée.

Définition 2. Un graphe non orienté (undirected graph en anglais) g=(V,E) est défini par un ensemble de sommets (ou nœuds) $V=\{v_i|i=1,\ldots n\}$, noté également V(G), et un ensemble d'arêtes (ou liens) $E=\{v_k|k=1,\ldots m\}$, noté E(G). Chaque arête e_k est caractérisée par une paire {v,w} de sommets appelés extrémités. Quand aucune confusion n'est possible, l'arête sera dénotée vw.

Définition 3. Une boucle est une arête liant un même sommet.

Définition 4. Deux arêtes sont dites parallèles si elles ont le même sommet de départ et le même sommet d'arrivée (mêmes extrémités pour un graphe non orienté).

Définition 5. Un graphe est dit simple s'il ne contient pas de boucles ni d'arêtes parallèles.

NB. Dans toute la suite on ne considère que les graphes simples.

Définition 6. Un graphe valué (dit aussi arêtes-valuées) est un graphe pour lequel on associe à chaque arête une valeur positive ou nulle appelée poids.

Définition 7. Un graphe sommets-valués est un graphe pour lequel on associe à chaque sommet une valeur positive ou nulle appelée poids.

Définition 8. Dans un graphe non orienté, une chaîne est constituée d'une séquence de sommets adjacents. Elle est dite simple si elle ne contient pas deux fois la même arête. Elle est dite élémentaire si elle ne contient pas (sauf pour ses extrémités) deux fois le même sommet. On notera par uv-chaîne une chaîne élémentaire entre u et vde longueur minimum (c.-à-d., une plus courte chaîne de u vers v).

Définition 9. Dans un graphe orienté on définit un chemin comme une chaîne orientée (dans le sens de la séquence constituant la chaîne). Il est dit simple s'il ne contient pas deux fois le même arc. Il est dit élémentaire s'il ne contient pas (sauf pour ses extrémités) deux fois le même sommet. On notera par (u,v)-chemin un chemin élémentaire de u vers v de longueur minimum (c.-à-d., un plus court chemin de u vers v).

Définition 10. Soit un graphe non oriente, la relation « être liés par une chaîne » est une relation d'équivalence dont les classes sont dites les composantes connexes de G.

Définition 11. G=(V,E) est dit connexe s'il ne contient qu'une seule composante connexe.

Définition 12. Un graphe orienté G est dit fortement connexe si pour tout couple de sommets, il existe un chemin d'un sommet vers l'autre.

Remarque 13. La relation « être liés par un chemin » n'est pas une relation d'équivalence car elle n'est pas symétrique. C'est une relation d'ordre où chaque classe induite est dite composante fortement connexe de G.

Définition 14. Un sommet v d'un graphe G est dit point d'articulation si sa suppression augmente le nombre de composantes connexes de G.

Définition 15. Un ensemble d'articulation est un ensemble de sommet $A \subset V$ tel que ; G_{V-A} à plus de composantes connexes que G. Les composantes connexes de G_{V-A} sont appelées pièces relatives à A.

Définition 16. Une arête e d'un graphe G est appelée isthme si sa suppression augmente le nombre de composantes connexes de G.

Définition 17. Dans un graphe un ensemble $E' \subset E$ tel que G[E-E'] à plus de composantes connexes que G est appelé ensemble déconnectant.

Définition 18. Un graphe d'ordre n est dit complet (noté K_n) si toute paire de sommets du graphe est adjacente (c.-à-d., tous les sommets sont connectés, le graphe est (n-1)-régulier et m=n(n-1)/2 .

Définition 19. Un sous-graphe complet d'un graphe est appelé clique.

2. La recherche des communautés par la structuration du graphe

2.1. Première approche : Connexité et forte-connexité

Les composantes connexes constituent une première structuration du graphe en communautés, même si elle est très large, elle est nécessaire pour repérer l'existence d'ensembles disjoints (non reliés par des arêtes dans le graphe) puis effectuer dans chaque composante connexe une structuration en sous-communautés appropriés.

Deux algorithmes existent pour la recherche des composantes connexes dans un graphe (Lacomme & al., 2003) :

Algorithme 1 :

Cet algorithme est basé sur une exploration en largeur ou en profondeur en démarrant d'un sommet s, qui va nous fournir les descendants de s, c'est-à-dire, les sommets reliés par une chaîne. Ces sommets forment donc la composante connexe de s.

Pour obtenir toutes les composantes, nous initialisons une fois seulement les marques au début, puis nous lançons l'exploration à partir d'un sommet s quelconque, ce qui donnerait la première composante connexe. Tant qu'il reste des sommets non marqués, nous relançons une exploration à partir de l'un d'entre eux, ce qui donnerait les composantes suivantes.

Algorithme 2 :

Cet algorithme construit une liste d'arbre Ai associé à chaque composante connexe. Il fait un parcours en profondeur des sommets du graphe. Chaque successeur non marqué x est ajouté dans Ai ainsi qu'une arête {s,x}. La construction de l'arbre se poursuit récursivement avec les successeurs de s.

Partant d'un sommet du graphe, on va numéroter les sommets dans l'ordre d'une exploration en profondeur d'abord, c.-à-d. que l'on descendra dans le graphe le plus loin possible sans former de cycle puis on remontera jusqu'à la dernière bifurcation laissée de coté et ainsi de suite jusqu'au retour au sommet de départ. L'ensemble des sommets ainsi rencontrés forme une première composante connexe.

Si tous les sommets du graphe ont été rencontrés, le graphe est connexe. Sinon, on recommence l'exploration précédente à partir d'un sommet non encore rencontré, jusqu'à ce qu'il ne reste aucun sommet non exploré.

Remarque 2. Nous proposons d'utiliser le deuxième algorithme, car il donne non seulement les composantes connexes, mais aussi il propose un arbre couvrant qui peut être considéré comme une structuration (en termes d'ordre) de la composante connexe. Cet algorithme construit des composantes connexes, basé sur un parcours en profondeur des sommets du graphe à une complexité de $O(m) \approx O(n^2)$ pour une représentation en matrice d'adjacence.

Algorithme 3 : Composantes fortement-connexes

Tarjan (1972) a conçu un algorithme astucieux pour extraire en seulement $O(m) \approx O(n^2)$ toutes les composantes fortement connexes d'un graphe. Cet algorithme est basé sur une exploration en profondeur complète du graphe comme dans l'algorithme 2. On lance l'exploration à partir d'un sommet non encore visité, et on recommence jusqu'à ce que tous les sommets soient visités. La forêt de visité est l'ensemble des arborescences de visites résultant de ces explorations élémentaires, les numéros de visite de chaque sommet sont stockés.

Les deux Figures ci-dessous illustres l'effet de structuration d'une composante connexe du graphe de glossaire du droit des TIC « ELMOUGHITH » en composantes fortement-connexes :

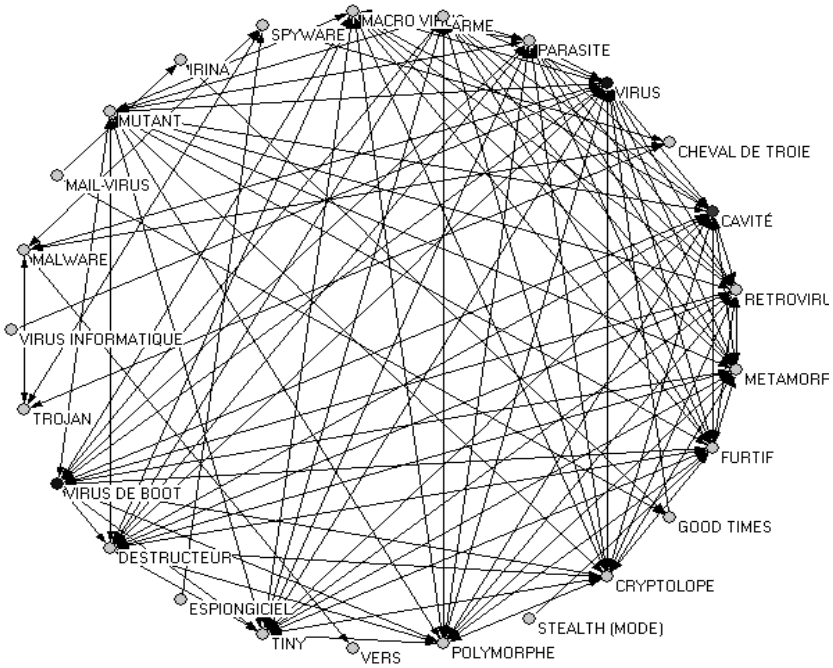

Figure 1. Représentation 2D circulaire de la composante connexe du terme « virus »

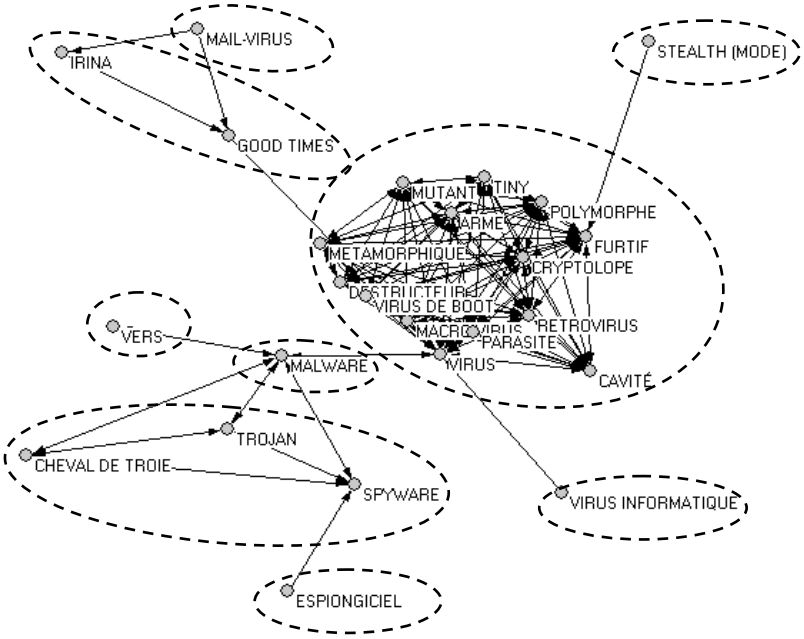

Figure 2 : Représentation 2D Kamada-Kawai libre de la composante connexe du terme « virus », les composantes fortement connexes sont entourées par des cercles

2.2. Deuxième approche : Hiérarchie des clusters

Par définition d'une communauté dans un graphe, tout Clustering du graphe induit une structuration du graphe en communautés (graphe quotient associé dans le cas du Clustering par partition, et arbre de Clustering hiérarchique ou mixte).

Notons que le Clustering hiérarchique d'un graphe constitue une structuration en communautés remarquable comme l'illustre la Figure ci-dessous :

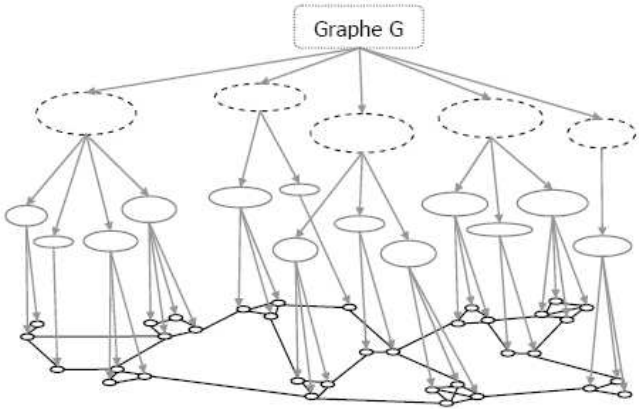

Figure 3 : Représentation 3D d'un graphe clustérisé hiérarchique

D'autres représentations sont possibles :

1) Graphes composés (Sugiyama et Misue 1991) ;
2) Arbre d'ensembles ;
3) Arbre composé simples (Boutin et Hascoët, 2004) ;
4) Arbre composé multi niveau (Boutin, Thièvre et al, 2005).

Si les sommets de deux clusters distincts sont reliés par une ou plusieurs arêtes, on relie ces deux clusters par une arête dont le poids et le maximum des poids des arêtes les reliant (la moyennes des poids peut être également prie).

3.3-Troisième approche : Hiérarchie basée sur l'arête-connexité

Soit G=(V,E) un graphe connexe.

Définition 20. L'arête-connexité du graphe G, notée k(G), est le nombre minimum d'arêtes du qu'il faut supprimer pour déconnecter le graphe G.

Définition 21. Une coupe minimum dans le graphe G est un ensemble d'arrête dont suppression déconnecte le graphe G. Une coupe minimum, est une coupe avec un nombre minimum d'arêtes.

Propriété 1. Une coupe $S \subset E$ est une coupe minimum dans le graphe G, si et seulement si, $|S|=k(G)$

Définition 3. Si $k(G>n/2)$, le graphe G est dit très connecté (Hartuv & al., 1999).

Le principe de cette approche est d'identifier les sous-graphes très connectés par la recherche des coupes minimums pour définir une hiérarchisation du graphe.

Nous pouvons montrer facilement que :

Propriété 2. Chaque coupe minimum dans un graphe déconnecte le graphe en exactement deux composantes connexes.

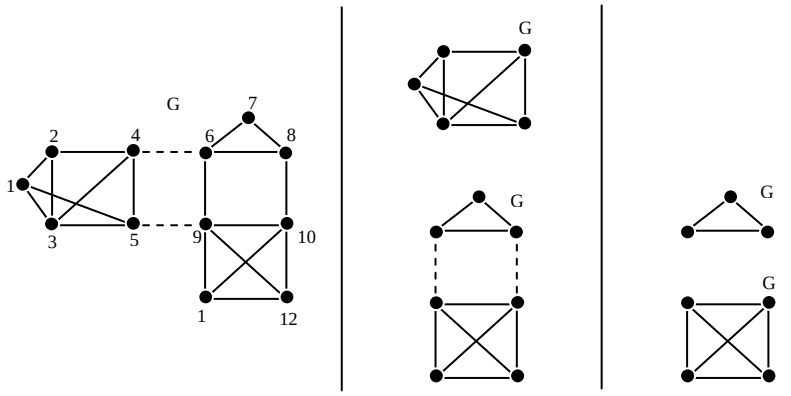

Figure 4 : Exemple de structuration d'un graphe en sous-graphes très connecté par la recherche de coupes minimums (les coupes minimums sont représentées par des lignes discontinues

L'hiérarchisation du graphe dans la Figure 4 basé sur l'arête-connexité est présentée dans la Figure 5 :

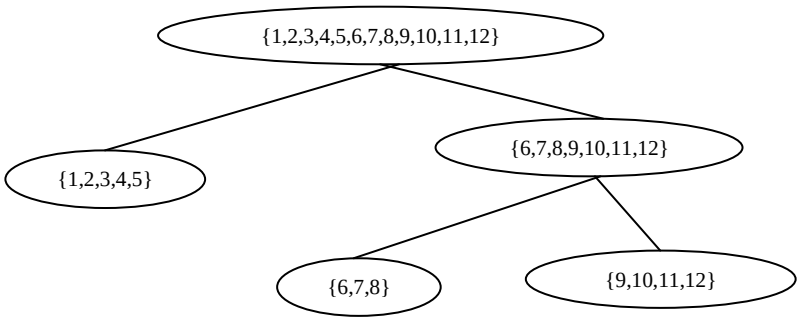

Figure 5 : Résultat de l'hiérarchisation du graphe de la Figure 4, en se basant sur l'arête connexité

2.4. Quatrième approche : Arborescence

L'idée est de rechercher un arbre couvrant (arbre maximal), pour cela nous pouvons utiliser l'algorithme de Kruskal qui utilise la propriété suivante :

Propriété 3. Un arbre est un graphe connexe sans cycle.

L'algorithme consiste d'abord à ranger par ordre de poids décroissant les arêtes d'un graphe, puis à retirer une à une les arêtes selon cet ordre et de les ajouter à l'arbre

tant que cet ajout ne fait pas apparaître un cycle dans l'arbre couvrant de poids maximum.

L'algorithme de Kruskal à une complexité O(m.log(n)) avec (m : le nombre d'arête et n : le nombre de sommet).

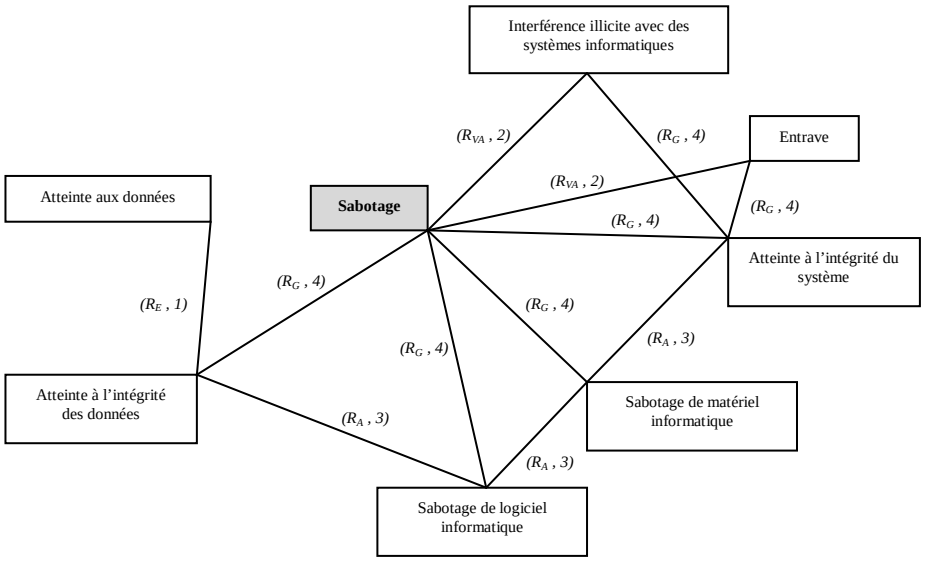

Figure 6 : Représentation de composante connexe « sabotage » du glossaire ELMOUGHITH sur les arêtes les relations et poids des relations (RG relation générique, RA relation d'association, RVA relation voir-aussi, RE relation d'équivalence)

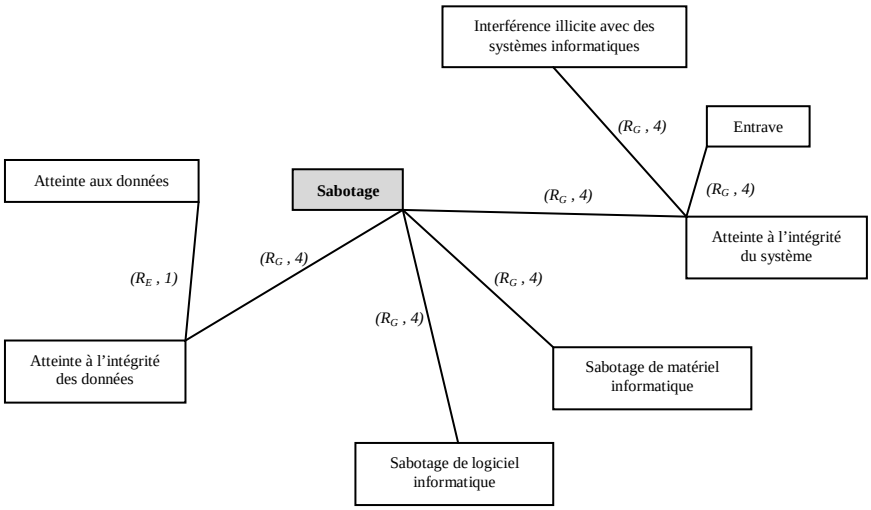

Figure 7 : Résultat d'application de l'algorithme de Kruskal sur la composante connexe « sabotage » du glossaire ELMOUGHITH

2.5. Cinquième approche : Bloques issues des points d'articulations

Les points d'articulation jouent un rôle important dans un graphe. Un point d'articulation est un sommet augmentant le nombre de composantes connexes du graphe si on l'enlève. Pour déterminer les points d'articulation on utilise l'algorithme d'exploration par profondeur d'un graphe. Dans une composante connexe du graphe la suppression d'un point d'articulation créera au moins deux composantes connexes.

Les sous groupes dans les quelles les points d'articulation divise le graphe sont dites bloques. L'intersection des bloques est les points d'articulations du graphe. Les bloques non séparables maximales sont dites bloques issues des points d'articulation du graphe (l'intersection des bloques ici est réduite à un seul point d'articulation).

Par exemple, la composante connexe « virus » du glossaire ELMOUGHITH constituée de 25 sommets (voir Figure 1) :

1 "ARME"
2 "CAVITÉ"
3 "CHEVAL DE TROIE"
4 "CRYPTOLOPE"
5 "DESTRUCTEUR"
6 "ESPIONGICIEL"
7 "FURTIF"
8 "GOOD TIMES"
9 "IRINA"
10 "MACRO VIRUS"
11 "MAIL-VIRUS"
12 "MALWARE"
13 "METAMORPHIQUE"
14 "MUTANT"
15 "PARASITE"
16 "POLYMORPHE"
17 "RETROVIRUS"
18 "SPYWARE"
19 "STEALTH (MODE)"
20 "TINY"
21 "TROJAN"
22 "VERS"
23 "VIRUS"
24 "VIRUS DE BOOT"
25 "VIRUS INFORMATIQUE"

Contient cinq (05) points d'articulations :

1) 7 "FURTIF"
2) 8 "GOOD TIMES"
3) 12 "MALWARE"
4) 18 "SPYWARE"
5) 23 "VIRUS"

Ces points d'articulation définissent neuf (09) bloques :

1) Bloque 1: 8, 9, 11
2) Bloque 2: 8, 23
3) Bloque 3: 6, 18
4) Bloque 4: 3, 12, 18, 21
5) Bloque 5: 12, 22
6) Bloque 6: 12, 23
7) Bloque 7: 23, 25
8) Bloque 8: 7, 19
9) Bloque 9: 1, 2, 4, 5, 7, 10, 13, 14, 15, 16, 17, 20, 23, 24

La décomposition obtenue en bloque peut être représenté comme illustré dans la Figure ci-dessous :

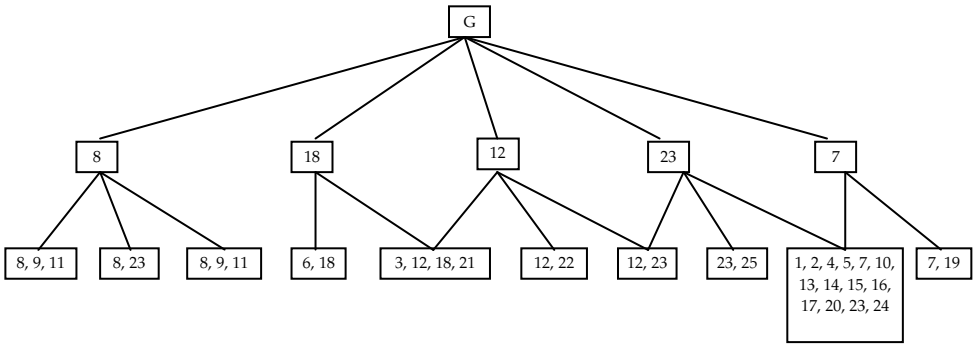

Figure 8 : Représentation hiérarchique des bloques issues des points d'articulation du composante connexe « virus » du glossaire ELMOUGHITH

2.6. Sixième approche : Structuration basé sur les cliques maximales

Une clique par sa définition est un ensemble de sommets du graphe intensément liés l'un à l'autre. L'idée est que l'ensemble des cliques possibles entre les sommets du graphe suggère des contextes, c.-à-d. des communautés.

Nous somme intéressé par le problème de recherche de toutes les cliques maximales dans chaque composante connexe d'un graphe. Le problème de recherche de cliques maximum étant NP-complet.

Nous proposant ici deux méthodes pour la structuration de graphe en communautés en utilisant les cliques maximale du graphe:

1) **1er Méthode :** Utiliser une heuristique pour trouver toutes les cliques maximales, construire la matrice des nombres coappartenance des paires de sommets à une même clique maximale et produire un Clustering hiérarchique du graphe basé sur cette matrice ;

2) **2^{e} Méthode :** Triangulé le graphe et construire l'arbre de cliques maximales par l'algorithme de Galinier *& al. (1995)* ;

1er Méthode

(1) La recherche de toutes les cliques maximales du graphe peut être effectuée par l'ancien algorithme « BK » proposé par *Bron-Kerbosh (1973)*, ou variante récente amélioré « IK » proposé par *Ina Koch (2001)*. Un autre algorithme proposé par *Frédiric Cazals et Chinmay Karande (2004)* en modifiant la stratégie du pivot utilisé par *I. Koch* ;

(2) Une fois les cliques maximales connues, on construit la matrice de coappartenance des paires de sommets dans les cliques maximales $CA=(CA_{ij})_{i=1..n,\ j=1..n}$ telle que :

- CA_{ii} est la cardinalité d'une clique maximale contenant le sommet $v_i \in V$;

- CA_{ij} , $i \neq j$ est le nombre de cliques maximales où les deux sommets $v_i, v_j \in V$ sont membre au même temps.

(3) Effectuer un Clustering hiérarchique du graphe basé sur la matrice $CA=(CA_{ij})_{i=1..n,\ j=1..n}$, nous suggérons ici d'utiliser la méthode *« single linkage agglomerative cluster analysis » (Zhang & al., 1996) (Manning et Schütze,1999)* pour une rapidité d'exécution de l'algorithme ;

Remarque : Soit $CC=(CC_{ij})_{i=1..q,\ j=1..q}$ avec q le nombre de cliques maximales trouvées. Tel que ;

- C_{ii} est la cardinalité de la ième clique maximale ;
- CC_{ij} , $i \neq j$ est le nombre de sommets communs entre la ième et jème cliques maximales.

Le Clustering hiérarchique du graphe des cliques basé sur la matrice $CC=(CCij)_{i=1..q,\ j=1..q}$ (avec la même méthode) est une autre structuration en communautés du graphe. Il suffit de définir les poids sur les arrêtes (métrique entre cliques) inter sommets-cliques du graphe.

2e Méthode

En 1974, Gavril a proposé une autre caractérisation très utile, de point de vue algorithmique, des graphes triangulés : « Un graphe est triangulé, si et seulement si, c'est le graphe d'intersection des sous-arbres d'un arbre. Cela signifie en fait que les cliques maximales peuvent être arrangées sous forme d'un arbre tel que les cliques contenant un sommet donné induisent un sous arbre. Un tel arbre est appelé un arbre de cliques.

Dans (Galinier & al., 1995), un algorithme en O(m+n) simple basé sur Lex-BFS pour construire un arbre de cliques a été présenté, sont principe est :

Considérons le diagramme composé des étiquettes des sommets lorsqu'ils sont numérotés. Lorsque le graphe est triangulé chaque séquence strictement croissante pour l'ordre d'inclusion correspond à une clique maximale. On obtient donc facilement l'ensemble des cliques maximales, il reste à les connecter entre elles. Pour cela, on retient pour chaque sommet, la clique qui l'a découvert ainsi que le dernier sommet qui l'a marqué. Lorsqu'une nouvelle clique est visitée, on la relie à la clique qui a découvert le dernier sommet ayant marqué l'ensemble de la séquence croissante. On assure ainsi que si une clique contient un sommet alors toutes les cliques sur le chemin de l'arbre entre cette clique et la clique qui découvert le sommet contient également ce sommet.

Exemple : Structuration basé sur les cliques maximales

Considérons l'exemple de la composante connexe « virus » du graphe de glossaire de droit des TIC ELMOUGHITH, on va appliquer la première méthode proposée :

1) L'algorithme de recherche de cliques maximale repères neuf (09) cliques :

1: 1, 2, 4, 5, 7, 10, 13, 14, 15, 16, 17, 20, 23, 24
2: 8, 23
3: 1,2, 23
4: 2,3, 25
5: 3, 12, 18, 21
6: 6, 18
7: 8, 9, 11
8: 7, 19
9: 12, 22

2) Matrice de coappartenance des paires de sommets dans les cliques maximales :

	1 1 1 1 1 1 1 1 1 1 2 2 2 2 2
	1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
	- -
1	1 1 0 1 1 0 1 0 0 1 0 0 1 1 1 1 1 0 0 1 0 0 1 1 0
2	1 1 0 1 1 0 1 0 0 1 0 0 1 1 1 1 1 0 0 1 0 0 1 1 0
3	0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0
4	1 1 0 1 1 0 1 0 0 1 0 0 1 1 1 1 1 0 0 1 0 0 1 1 0
5	1 1 0 1 1 0 1 0 0 1 0 0 1 1 1 1 1 0 0 1 0 0 1 1 0
6	0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
7	1 1 0 1 1 0 2 0 0 1 0 0 1 1 1 1 1 0 1 1 0 0 1 1 0
8	0 0 0 0 0 0 0 2 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0
9	0 0 0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
10	1 1 0 1 1 0 1 0 0 1 0 0 1 1 1 1 1 0 0 1 0 0 1 1 0
11	0 0 0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
12	0 0 1 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 1 0 0 1 1 0 0
13	1 1 0 1 1 0 1 0 0 1 0 0 1 1 1 1 1 0 0 1 0 0 1 1 0
14	1 1 0 1 1 0 1 0 0 1 0 0 1 1 1 1 1 0 0 1 0 0 1 1 0
15	1 1 0 1 1 0 1 0 0 1 0 0 1 1 1 1 1 0 0 1 0 0 1 1 0
16	1 1 0 1 1 0 1 0 0 1 0 0 1 1 1 1 1 0 0 1 0 0 1 1 0
17	1 1 0 1 1 0 1 0 0 1 0 0 1 1 1 1 1 0 0 1 0 0 1 1 0
18	0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 2 0 0 1 0 0 0 0
19	0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
20	1 1 0 1 1 0 1 0 0 1 0 0 1 1 1 1 1 0 0 1 0 0 1 1 0
21	0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0
22	0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0
23	1 1 0 1 1 0 1 1 0 1 0 1 1 1 1 1 1 0 0 1 0 0 4 1 1
24	1 1 0 1 1 0 1 0 0 1 0 0 1 1 1 1 1 0 0 1 0 0 1 1 0
25	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1

3) Clustering hiérarchique des sommets par la méthode *« single linkage agglomerative cluster analysis »* :

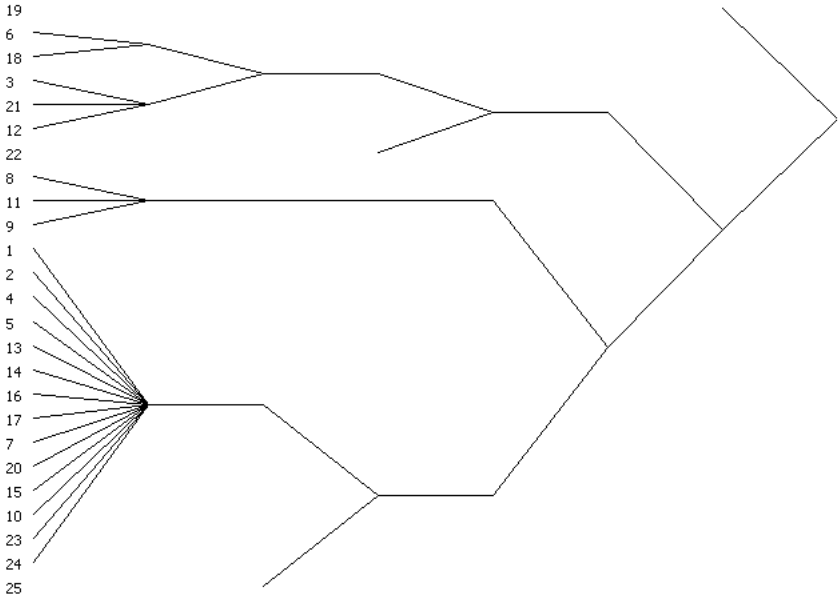

Figure 9 : Clustering hiérarchique de la composante connexe « virus » du glossaire ELMOUGHITH

4) Matrice du nombre de sommets dans l'intersection de deux cliques maximales :

	1	2	3	4	5	6	7	8	9
1	14	1	1	1	0	0	0	1	0
2	1	2	1	1	0	0	1	0	0
3	1	1	2	1	1	0	0	0	1
4	1	1	1	2	0	0	0	0	0
5	0	0	1	0	4	1	0	0	1
6	0	0	0	0	1	2	0	0	0
7	0	1	0	0	0	0	3	0	0
8	1	0	0	0	0	0	0	2	0
9	0	0	1	0	1	0	0	0	2

5) Clustering hiérarchique du graphe des cliques :

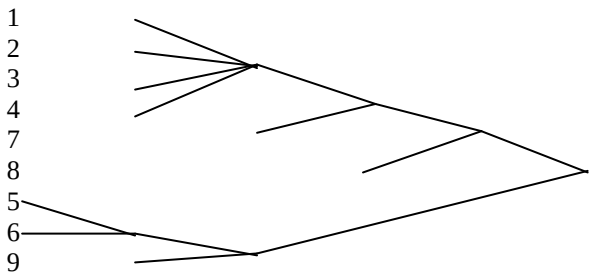

Figure 10 : Clustering hiérarchique de graphe des cliques maximales de la composante connexe « virus » du glossaire ELMOUGHITH

3. Élaboration d'une cartographie d'un corpus textuel à partir d'un réseau d'association de termes

La cartographie des données textuelles constitue actuellement l'un des moyens d'accès à l'information par la visualisation du contenu. Notre contribution, tout en s'inscrivant dans ce cadre, fait appel au traitement Automatique des Langues (TAL) et à quelques éléments structurels de la théorie des graphes afin de parvenir à l'élaboration d'une cartographie d'information spécifique à un corpus textuel donné.

Avant l'élaboration de la cartographie d'un corpus textuel Γ donnée, certaines hypothèses doivent être vérifiées :

- Γ est homogène et traite une certaine thématique donnée,
- Γ est subdivisé en un nombre connu d'entités textuelles,
- Un ensemble des candidats termes significatifs de Γ est connu et est filtré par des techniques de TAL, le résultat étant l'ensemble T,
- Les valeurs W_{ij} sont calculées.

On obtient ainsi le réseau R=(T,E,W) associé au corpus Γ.

Nous proposons, une démarche pour élaborer une cartographie à trois niveaux :

- Globale,
- Par communautés
- et Ponctuelle.

Ceci en exploitant les propriétés structurelles de notre modèle (i.e. les réseaux d'association de termes)

Global. Consiste à proposer une vue générale sur le contenu du corpus. Cette vue globale sera obtenue par la détermination d'un ensemble de termes génériques représentatifs dans le réseau d'association de termes de corpus. Elle constituer une information primale pour l'utilisateur. En plus, elle lui permet d'envisager une stratégie d'exploration du contenu.

Par communautés. Ce niveau est sollicité lorsque l'utilisateur veut approfondir un aspect quelconque du contenu d'un texte. Les connaissances dissimulées dans le corpus, se voient représentées par le réseau d'association de termes, celui-ci se présente spontanément sous forme de thématiques (i.e. de communautés). Les communautés s'obtiennent par la recherche d'une partition du réseau d'association de terme de corpus. La vue par communautés sera alors constituée du réseau induit par l'ensemble des termes génériques représentatifs de toutes les communautés. Elle fournira une cartographie où les communautés sont spatialement bien visibles. Cette représentation permet ainsi à l'utilisateur d'appréhender le contenu global du corpus en un ensemble réduit de thèmes, en même temps que les liens entre ces thèmes. Elle lui permettra d'avoir une vision à la fois plus globale et synthétique de contenu.

Ponctuelle. Cette dernière dimension consiste à donner une vision sélective et ponctuelle sur un ou plusieurs concepts utilisés dans le corpus. Ceci peut être obtenu en faisant un zoom (peut être assimilé à une fenêtre) dans le réseau d'association des

termes. L'opération concernera un ensemble de termes sélectionnés par l'utilisateur. La cartographie proposée est alors, le sous- réseau induit par ces sommets. Elle permettra à l'utilisateur d'aller vers les détails ou vers les précisions sémantiques sur les connaissances et informations exprimées dans le corpus. La représentation obtenue pourra être complétée par un renvoi vers les entités textuelles du corpus.

Conclusion

Dans ce papier, quelques techniques pour la recherche de communautés dans un graphe sont proposées. Comme l'on a illustré, l'étude de quelques problématiques concernant la structure combinatoire du réseau d'association de termes, permettent la mise au point de méthodes mathématiques pour l'étude sémantique du contenu d'un corpus textuel représenté par un modèle de graphe.

En perspective, on se donne pour objectif la mise à contribution nos travaux couplés aux cartes causales afin de construire un prototype de moteur de recherche avancé sur les textes. Celui-ci permettra de faire une analyse terminologique et sémantique profonde sur une thématique d'un corpus donnée.

Références

[1] BERGE. C. (1973). Graphes et hypergraphes. Dunod

[2] BORN, C., & KERBOSCH, J. (1973). Algorithm 457 : finding all cliques of an undirected graph. Comm. ACM , Vol. 16 (Num. 9), pp. pp 575-577.

[3] CAZALS, F., & KARANDE, C. (2005). Reporting maximal cliques : new insights into an old problem. Rapport de recherche, INRIA, Unité de recherche Sophia Antipolis, Sophia Antipolis - France.

[4] CHERFI, H. (2004). Etude et réalisation d'un système d'extraction de connaissances à partir de textes. Ecole doctorale IAEM Lorraine, Départment de formation doctorale en informatique. Lorrain: Laboratoire Lorrain de Recherche en Informatique et ses Applications.

[5] DIESTEL Reinhard (2000). Graph Theory. Springer-Verlag New York.

[6] EVANS, J. T. et E. Mineka (1992). Optimization algorithms for networks and graphs. New York. Marcel Dekker inc.

[7] GAVRIL, F. (1974). The intersection graphs of path in tree are exactly the chordal graphs. Journ. Comb. Theory (Num. 16), pp 47-56.

[8] GIBBONS, A. (1985). Algorithm graph theory. Cambridge university press

[9] GUENOCHE, A., COLOMBO, T., QUENTIN., Y. (2003) Recherche de zones denses dans un graphe: Mathematics, Actes des Journées Informatiques de Metz, INRIA 203-212

[10] HEARST M., (1999). Untangling Text Data Mining, In Proc of the 37th Annual Meeting of the Association for Computational Linguisitics (ACL'99), University of Maryland,.

[11] JUNGNICKEL, D. (1999). Graphs, Networks and Algorithms. Berlin.

[12] LACOMME, P., PRINS, C., & SEVAUX, M. (2003). Algorithmes de graphes. Paris, France: EYROLLES.

[13] MOKRANE, A,. PONCELET, P., AREZKI, R., DRAY, G. (2004) Cartographie automatique du contenu d'un corpus de documents textuels. In proceeding of the 7 th International Conférence on the Statistical Analysis of Textual Data JADT, vol 2, Louvain-la-Neuve : Presses Universitaires de Louvain. 816-823

[14] NAKACHE, J.-P., & CONFAIS, J. (2005). Approche pragmatique de la classification. Paris, France: TECHNIP.

[15] TARJAN, R, E. (1972)Depth first search and linear graph algorithms. SIAM Journal on Computing, 1(2). 146-160

[16] YALAOUI, B., & AÏT HADDADENE, H. (2006). Sur le problème de partitionnement d'un graphe d'association de termes. 13ème Rencontre de la Société Française de Classification, Campus Universitaire de l'île de Saulcy, du 5 au 8 Septembre. Metz – France.

[17] YALAOUI, B., & HARIK, H. (2006). Un algorithme pour la recherche d'un ensemble générique de termes dans un réseau de termes associés. 7ème congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision, du 6 au 8 Février. Lille – France.

[18] YALAOUI, B., HARIK, H., & DAHMANE, M. (2008). Une démarche pour l'exploration et l'aide à l'analyse de corpus textuel. In I. éditions (Ed.), 1er Conférence Internationale sue les Système d'Information & Intelligence Economique, 14-16 Février, Tome I, pp. pp 410-423. Hammamet – Tunisie.

[19] ZHANG, T., RAMAKRISHNAN, R., and LIVNY, M. BIRCH(1996). An efficient data clustering method for very large databases. Proceedings of ACM SIGMOD Conference, Montreal, Canada, pp. 103–114.