THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par
Université Toulouse 1 Capitole (UT1 Capitole)

Discipline ou spécialité

INFORMATIQUE

Présenté et soutenue par :

Christian BELBÈZE

le 5 avril 2012

Titre :

Agrégats de mots sémantiquement cohérents
issus d'un grand graphe de terrain

Ecole doctorale :
Mathématiques Informatique Télécommunications de Toulouse (MITT)

Unité de recherche :
Institut de Recherche en Informatique de Toulouse (IRIT)

Directeur(s) de Thèse :
Mme Chantal Soulé-Dupuy, Professeur, Directrice des recherches, Univ. Toulouse 1 Capitole
M. Max Chevalier, Maître de Conférences, Co-encadrant, Univ. Toulouse 3 Paul Sabatier

Rapporteurs :
Mme Thérèse Libourel, Professeur des Universités, Univ. Montpellier 2 – Montpellier
Mme Christine Verdier, Professeur des Universités, Univ. Joseph Fourier – Grenoble

Examinateurs :
M. Claude Chrisment, Professeur des Universités, Univ. Toulouse 3 Paul Sabatier
Mme Frédérique Laforest, Professeur des Universités, Univ. Jean Monnet – Saint Etienne

Remerciements

Je remercie M. Claude Chrisment pour m'avoir accueilli au sein de l'équipe SIG de l'IRIT.

Je suis reconnaissant à Mme Thérèse Libourel, Professeur à l'Université Montpellier 2 et à Mme Christine Verdier, Professeur à l'Université Joseph Fourier à Grenoble d'avoir accepté d'être rapporteurs de ce mémoire, de leurs observations qui ont permis d'améliorer ce rapport de thèse ainsi que de l'honneur qu'elles me font en participant au jury de thèse.

Je remercie M. Claude Chrisment, Professeur à l'Université Toulouse 3 Paul Sabatier et Mme Frédérique Laforest, Professeur à l'Université Jean Monnet à Saint-Etienne de m'honorer de leur présence en tant qu'examinateurs à ce jury.

Toute ma gratitude va à Mme Chantal Soulé-Dupuy, Professeur à l'Université des Sciences Sociales Toulouse 1, pour avoir accepté d'être mon directeur de recherche me donnant ainsi une chance rare.

Je remercie également M. Max Chevalier, Maître de Conférences à l'Université Toulouse 3 Paul Sabatier, pour sa disponibilité, pour toute l'aide qu'il m'a prodiguée ainsi que pour sa participation à ce jury.

D'autre-part, je voudrais saluer l'intérêt que M. Matthieu Latapy, Directeur de Recherche au CNRS affecté au LIP6, a bien voulu porter à mon travail avec une généreuse disponibilité et remercier Jeremy Ozog et Claude Flottes pour leur aide en mathématiques.

J'ai aussi une pensée pour mes « cobayes » Internautes : Annie, Marie, Jean, Yasmina, Lou, Paul, Guillaume, Céline et Georges qui sont à l'origine de ce travail.

Je me dois enfin de remercier plus particulièrement ma Directrice de Recherche Chantal Soulé Dupuy et ma femme Sylvie Flottes. Sans leurs qualités de cœur, leur intelligence, leur incroyable patience et enfin leur ancrage dans une réalité qui parfois m'échappe, rien de tout ceci n'aurait été possible.

Je vais avoir 53 ans dans quelques semaines. Cette thèse est aussi le résultat de toutes les rencontres et de toutes les expériences qui m'ont construit durant ces années. Il y a donc tant de personnes à remercier pour avoir contribué à l'accomplissement de ces travaux que je ne tente pas d'en faire une liste exhaustive. Mais que tous ceux qui m'ont respecté, encouragé, écouté, aidé et (ré) éduqué sachent que je les remercie sincèrement.

Résumé

L'observation d'internautes en situation de recherche d'informations a permis de mettre en évidence un besoin, celui d'échanges immédiats. Une telle relation instantanée peut, dans le cadre qui nous occupe, revêtir différents aspects et notamment l'aspect coopératif permettant à un internaute de bénéficier, à un instant critique, des recherches des autres utilisateurs par des recommandations dynamiques. Selon le principe des réseaux sociaux, une communauté est un ensemble d'internautes pouvant tirer parti de liens, prédéfinis ou non, sur la base de centres d'intérêts communs, de pratiques communes… Repérer ces liens dynamiquement et provoquer des rencontres entre internautes nous a semblé être un vrai défi à relever.

Il s'agit donc de faire en sorte que se créent dynamiquement des communautés d'internautes à partir de recherches en cours via des moteurs de recherche (fichiers de log par exemple). Le processus de génération dynamique de communautés repose en grande partie sur l'extraction des thèmes de recherche (centres d'intérêts) des internautes présents sur le réseau à un instant donné (ou pendant un laps de temps donné). Les thèmes de recherche permettant la connexion entre internautes constituent le noyau de la communauté dynamique. L'ensemble des communautés se présente alors comme un graphe de termes (extraits des thèmes) s'apparentant à un grand graphe de terrain dans lequel les connexions représentent les cooccurrences.

Dans cette thèse, nous proposons une démarche de création et de validation du graphe communautaire. Cette démarche consiste à agréger les nœuds du graphe pour que chaque agrégat présente la plus forte cohérence sémantique possible. Les problématiques suivantes doivent être résolues:

- créer des agrégats de mots pouvant contenir des parties en recouvrement (une orthographe peut appartenir à plusieurs thématiques) ;
- choisir ou définir une technique de regroupement garantissant une forte cohérence sémantique ;
- caractériser les agrégats pour comprendre les différences de cohérence sémantique ;
- proposer des techniques de validation de la cohérence sémantique des agrégats.

Dans une première partie constituant un état de l'art, nous étudions de nombreuses méthodes de création de communautés au sein des graphes. Cependant aucune ne satisfait totalement à l'ensemble des critères nécessaires.

Dans une deuxième partie nous présentons notre contribution. Celle-ci est constituée de plusieurs méthodes d'agrégation et de plusieurs méthodes de validation sémantiques.

Nous proposons quatre méthodes d'agrégation : Détection de Cliques (agglomération de clique), Rigidification Simple (recherche de points de rupture dans le graphe), Rigidification Régulée (recherche de points de rupture en s'appuyant sur l'étude de populations spécifiques, mots vides et monosémique) et une Méthode d'Enrichissement d'Agrégat par Gravité (la méthode détermine un coefficient d'attraction pour chaque mot vers chaque agrégat).

Nous proposons, ensuite, trois méthodes de validation de la cohérence sémantique des agrégats : la Méthode de Coefficient de Validation Sémantique Comparé (estimation de la valeur sémantique des agrégats par comparaison du comportement de moteur de recherche sur Internet en utilisant différents jeux de test et les agrégats), la Méthode Trec-Eval par enrichissement de requête (les agrégats sont utilisé pour préciser des requêtes utilisateurs) et une Méthode de Comparaison de Cohérence de Documents Retournés (comparaison de la cohérence sémantique des documents retournés par des requêtes provenant de jeux de test spécifiques et des agrégats). Nous utiliserons aussi des validations manuelles réalisées par des experts du domaine des espaces sémantiques manipulés incluant la comparaison avec d'autres méthodes.

Les différentes propositions et méthodes d'expérimentations apportent la preuve de l'importance de pondérer les nœuds et les liaisons, ainsi que de diriger les graphes. La limitation de la taille des agrégats de mots est aussi un élément majeur de leur cohérence sémantique. Les différentes méthodes de regroupement peuvent encore évoluer. La combinaison de plusieurs types de liaisons au sein d'un même graphe, par exemple, permettrait d'affiner le contenu des agrégats.

Mots-clés

Graphes, Agrégats de termes, Communautés et communautés d'utilisateurs, Graphes de terrain, Petits Mondes.

Summary

The observation of internet users in a situation of information research has helped to highlight a general need to immediate exchange. The immediacy of the exchanges may take different aspects and in particular the fact for a surfer, at a given moment, to be able to benefit from the research of other surfers by dynamic recommendation. According to the principle of social networks, a community is a set of Internet surfers who can take advantage of links, predefined or not, on the basis common interests, common practices… Identifying these links dynamically and causing meetings between Surfers seemed to be a true challenge.

Then we have to dynamically create communities of internet users from ongoing research via search engines (log files for example). The process of dynamic generation of communities is largely based on the extraction of the research themes (centers of interests) of Internet users present on the network at a given moment (or during a given period of time). The themes of research allowing the connection between Internet users constitute the core of the Community dynamic. The community is then presented as a Large complex network graph of words (extracts of themes) in which the connections represent the cooccurrences.

In this thesis, we propose an approach for creation and validation of the graph community. This approach involves the aggregation of the nodes of the graph so that each aggregate has the highest semantics consistency possible. The following issues must be resolved:

- creating clusters of words that can contain overlap (a spelling may belong to several thematic);
- choosing or defining a grouping technique that guarantees a high degree of semantics consistency;
- characterizing the aggregates to understand the differences of semantics consistency;
- proposing techniques to validate semantics consistency of aggregates.

In a first part constituting a state of the art, we are studying many methods of creating communities in the graphs. However no one fulfills all of the necessary criteria.

In a second part we present our contribution. The latter is constituted of several methods of aggregation and several methods of semantic validations.

We offer 4 methods of aggregation: cliques Detection (agglomeration of clique), Simple Ratification (search for points of rupture in the graph), Regulated Regasification (search for points of rupture in relying on the study of specific populations, empty words and monosemic) and a method of Enrichment of Aggregate by Gravity (the method determines a coefficient of attraction for each word toward each aggregate).

We then propose three methods to validate the semantic consistency of aggregates : Method of Compared Coefficient of Semantics Validation (estimate of the value semantics of aggregates by comparing the behavior of search engine on the Internet by using different test sets and aggregates), Trec-Eval method for requests enrichment (the aggregates are used to specify user requests) and a method of consistency comparison of documents returned (comparison of the semantics consistency of documents returned by queries from test specific sets and aggregate). We will also use the manual validation by experts in the field of semantic spaces handled including comparison with other methods.

The various proposals and methods of experiments provide evidence of the importance of weighted nodes and links, as well as to direct the graphs. Limiting the size of the aggregates of words is also a major element of semantics consistency. The different clustering methods can still evolve. The combination of several types of links in a graph, for example, would refine the content of the aggregates.

Key-words

Graphs, Term aggregates, Communities and user communities, Complex networks, Small words.

Table des matières

« *Tout le monde savait que c'était impossible. Il est venu un imbécile qui ne le savait pas et qui l'a fait.* »

Marcel Pagnol

Ce travail est dédicacé aux exclus du système scolaire et à tous ceux qui n'ont pas, ou pas eu, accès à l'éducation.

Avant-propos

« Ou comment créer des communautés dynamiques en utilisant des agrégats de mots dans les grands graphes de terrain. »

Les graphes représentent un espace d'étude passionnant dont l'intérêt suffirait à justifier ce travail. Mais telle ne fût pas notre motivation. Ce travail a commencé par l'observation d'internautes recherchant des informations. De cette observation est né le désir de trouver des solutions pouvant aider les internautes à acquérir, comprendre et assimiler l'information. Internet est devenu un conteneur d'informations dont la dimension et la richesse semblent donner à ceux qui en dominent les accès un don d'omniscience. Cependant, Internet est aussi vécu comme une zone anxiogène. L'observateur ne trouvera pas surprenant que la manipulation d'outils comme les moteurs de recherche soit difficile pour les débutants ou les utilisateurs occasionnels. D'ailleurs, certaines informations retournées, comme les url, sont codées et absconses pour un grand nombre d'internautes. En fait, les mécanismes de blocage sont étrangement semblables chez ceux-ci quelle que soit leur expérience.

I. La solitude du chercheur d'informations

Suivant un protocole précis mélangeant des recherches choisies et imposées, cinq adultes et quatre enfants ont été observés.

Les utilisateurs présentaient des niveaux de connaissance sur l'usage d'Internet très divers. Le matériel disponible pour cette expérimentation était un PC connecté à Internet, une interface homme-machine adaptée à la personne, une caméra vidéo filmant le sujet (Webcam) et un logiciel enregistrant les écrans permettant l'incrustation de la caméra dans l'enregistrement. Ceci a permis de garder une trace complète des observations.

Chaque observation présentée est une recherche sur Internet ayant eu un but bien précis : soit la réponse à une question soit la recherche d'un document. Elle a donné le plus

souvent lieu à une ou plusieurs requêtes (une requête est comptabilisée chaque fois qu'un internaute clique sur « rechercher » dans un moteur de recherche).

Dans cette expérience réalisée en 2006, nous avons essayé de choisir un échantillon représentatif de la société, en âge, genre et niveau social.

Prénom		Âge	Usage d'Internet	Exp. sur Internet	Temps max. théorique	Situation perso.	Commentaire
– Les adultes							
Jean		16 ans	Quotidien	> 5 ans	15 minutes	Lycéen	Utilisateur référent, Jean utilise Internet plusieurs heures par jour (chat, recherche, …), il a grandi avec Internet. Il s'y connecte depuis 1995 ! Il est de la génération Internet !
Annie		72 ans	Mensuel	> 5 ans	25 minutes	PDG retraitée	Annie n'utilisait Internet que coachée .... Ce sont ici ses premiers pas en solo.
Georges		49 ans	Jamais	Nulle	15 minutes	Artisan	Georges est vraiment un débutant complet. Internet lui fait peur, mais son intérêt pour la moto est un bon moteur.
Marie		37 ans	Rare	< 1 an	15 minutes	Infirmière	Marie à une double personnalité elle est artiste par passion et scientifique de profession. Internet la fascine, mais elle n'est qu'une internaute occasionnelle.
Yasmina		20 ans	Mensuel	> 5 ans	15 minutes	Employée	Yasmina est une jeune fille qui n'utilise Internet que pour faire des recherches sur des paroles de chanson. Yasmina malgré ses engagements ne poursuivra pas le protocole.
– Les enfants							
Guillaume		11 ans	Hebdo	Entre 1 et 5 ans	15 minutes	Collégien	Guillaume est passionné d'informatique. Il est à l'aise avec la machine. Patient, posé, il va trouver ce qu'il cherche.
Céline		11 ans	Rare	< 1 an	15 minutes	Collégienne	Céline est une enfant rêveuse qui ne manque pourtant pas de ténacité.
Lou		7 ans	Hebdo	Entre 1 et 5 ans	15 minutes	Scolaire	Lou est une enfant appliquée et bonne élève.
Paul		7 ans	Rare	Jamais	15 minutes	Scolaire	Paul est un enfant dont on dit qu'il est souvent « dans la lune ». Il devra faire beaucoup d'efforts pour se concentrer sur ses recherches.

Tableau AVP.1 : Les participants au protocole d'observation. NOTE : Une présentation vidéo des internautes ayant participé à l'étude est consultable sur le site http://sissiprojet.free.fr.

II. L’observation d’internautes en recherche d’informations

Deux protocoles d’observation sont définis : un pour l’observation des adultes, un autre concernant les enfants. Certains enfants étant très jeunes et sans expérience, un protocole spécifique, plus léger est préférable.

Le protocole des adultes

Il est composé de cinq recherches (deux libres et trois imposées) :

Recherches imposées :

1 Un texte : les paroles de la chanson « All Blues »
2 La distance entre Toulouse et Rodez
3 Une partition de musique : la partition de l’hymne national roumain

Recherches libres : L’adulte doit fournir deux sujets sur lesquels il devra, autant que possible rechercher une réponse à une question ou un document dans un format particulier.

Le protocole des enfants

Le protocole des enfants est composé de deux recherches (une libre et une imposée) :

Recherche imposée : La durée de vie d’un dauphin.

Recherche libre : L’enfant doit fournir un sujet de recherche sur lequel il devra, autant que possible, trouver une réponse à une question ou un document dans un format particulier.

Mesures

Pour chacune des observations des marqueurs sont choisis. Ces marqueurs ont essentiellement pour but de mesurer le sentiment global de l’utilisateur et de nous permettre de rattacher ce sentiment à des éléments mesurables. On recherche ainsi, par le typage des difficultés rencontrées et des comportements, à mieux comprendre quelles situations peuvent créer un sentiment d’échec ou de stress.

Cinq valeurs sont mesurées pour toutes les observations.

Valeur mesurée pour chaque recherche	Règles
Le temps complet de l'observation	Ce temps comptabilisera seulement le temps (format h : min : sec) de recherche effective pendant l'observation. Les temps de présentation de la recherche et d'auto-notation ne sont donc pas comptabilisés (En cas de dépassement notable (100%) du temps maximal donné par l'utilisateur, l'observateur lui propose de mettre un terme à la séance).
Le nombre de sites visités	On comptabilise le nombre de fois où l'internaute clique sur un lien dans le moteur de recherche ou clique sur un lien dans un site envoyant sur un autre site. Si le site à déjà été visité préalablement il est quand même comptabilisé.
Le nombre de requêtes	On comptabilise le nombre de fois où l'internaute envoie une requête au moteur de recherche. Les requêtes renvoyées plusieurs fois sont comptabilisées plusieurs fois.
Le nombre de pages consultées dans les moteurs de recherche	On comptabilise le nombre de fois où l'internaute demande à afficher les sites retournés par le moteur de recherche.
Si l'information, le document ou la réponse à la question recherchée ont été trouvés	Valeurs possibles : • oui, • certains éléments ++, • certains éléments --, • non.

Quatre marqueurs sont enregistrés pour chaque requête effectuée pendant l'observation.

Marqueurs enregistrés pour chaque requête	Explication
Les mots clés utilisés	La requête est archivée à l'identique, les fautes d'orthographe et les caractères spéciaux sont conservés.
Le nombre de mots clés utilisés	Chaque mot est comptabilisé. Les expressions entre guillemets sont comptabilisées comme mots-clés.
Le nombre de sites « retournés »	On note le nombre de sites théoriques retournés par le moteur de recherche. C'est en fait le nombre de sites affiché comme le nombre de sites trouvés sur internet pour cette requête.
Le caractère multi-langue des mots clés utilisés	Si une requête contient des mots de langues différentes elle est notée comme une requête multi-langues.

Pour chacune des observations (résolues ou pas) le sujet s'exprimera sur 5 caractéristiques subjectives qu'il notera de 0 à 10 : 0 signifiant « très mauvais », 10 « excellent ».

Objet de l'auto-notation	Explication
Capacité à comprendre l'information	Ressenti sur la clarté des documents parcourus.
Intérêt des sites rencontrés	Ressenti sur la qualité informative des documents.
Longueur subjective de la recherche	Ressenti du temps passé sur Internet.
Ressenti général	Ressenti sur le plaisir éprouvé à surfer sur le Web.

En cas de blocage ou de découragement, si le temps maximum donné par l'utilisateur n'est pas dépassé, l'intervention d'une tierce personne est possible de façon à ne pas arrêter l'expérience. Ces interventions qui ont le plus souvent pour but de guider l'utilisateur (replacer un utilisateur dans un moteur de recherche, lui apprendre la notion de lien hyper texte, répondre à une question technique, …) sont toutes notifiées.

Résultats et exploitation

Les difficultés rencontrées par les utilisateurs se situent au niveau de chacune des tâches élémentaires qui composent la tâche globale de recherche d'information sur le Web. Ces difficultés sont principalement de quatre ordres :

- Trouver les mots-clés efficaces
- Faire un choix dans une liste longue et hétérogène
- Extraire de l'information des sites web proposés
- Gérer le temps (temps réel de recherche et de perception)

Ces difficultés sont détaillées dans les sections qui suivent.

Difficultés pour trouver des mots-clés efficaces

Avec un nombre moyen de 7 millions de sites Web trouvés par recherche, nous pouvons certifier que les 3,44 mots-clés employés dans les requêtes ne sont pas assez efficaces pour filtrer le Web. Les utilisateurs ont des difficultés pour employer davantage de mots-clés et pour choisir ces mots. Lorsqu'ils le font, cela devient même contre-productif : le taux de satisfaction décroît. Cet état de fait provient d'un problème de compétences des utilisateurs sur la nature même de la recherche. En effet, en général les utilisateurs ne connaissent pas assez le sujet pour enrichir la demande. Ceci provient également d'un mélange entre atonie et manque de compétence sur le fonctionnement du moteur de recherche. Le moteur de recherche est un outil avec lequel l'interaction est limitée. On l'interroge et comme par magie, la réponse est retournée. D'autre part, l'utilisateur ne manie pas de mots dans une langue inconnue, par exemple, pour rechercher de l'information sur la musique roumaine, personne n'a employé de mots-clés roumains (même si des outils de traduction étaient connus). Un seul des utilisateurs observés (Jean), a utilisé des mots de la chanson anglaise à retrouver. Bilingue, il a su mélanger le français et l'anglais. Ainsi on peut remarquer que l'utilisateur choisit les mots-clés de sa requête en fonction de sa connaissance du sujet et de sa connaissance du moteur de recherche utilisé. Il faut également prendre en considération que les mots-clés peuvent être mal orthographiés. De fait, les jeunes utilisent de plus en plus le langage SMS. On a pu observer lors des expérimentations que les enfants trouvaient (par erreur ?) l'information en formulant leurs requêtes en langage SMS.

Difficultés pour faire un choix dans une liste longue et hétérogène

La liste des informations retournées par un moteur de recherche en réponse à une requête comprend un certain nombre d'éléments. Avec Google, par exemple, comme le montre la figure AVP.1, chaque item de la liste retournée intègre :

- Le *titre du site*. Mais comme tous les sites n'ont pas un titre, dans ce cas, les moteurs, comme Google, utilisent d'autres métadonnées (metatags) telles que

le titre de la page, l'auteur, l'URL,… pour construire une sorte de titre. De plus, lorsque ce titre est trop long, il est tronqué.

- Un *extrait (« snippet »)*. Il s'agit en fait de bouts de phrases (quelques mots), entourant les mots-clés, extraits du site. L'extrait constitué en assemblant ces bouts de phrases n'a globalement pas de sens.

- L' *URL (Uniform Resource Locator)* de la page ou du site.

On peut facilement imaginer un débutant déstabilisé face à une telle liste. Par exemple, Annie, 70 ans, a soumis à Google la requête suivante « Natura 2000 marais de Gabarret » où « Natura 2000 » est une organisation écologiste française et Gabarret est un petit village français. Elle a obtenu la réponse suivante…

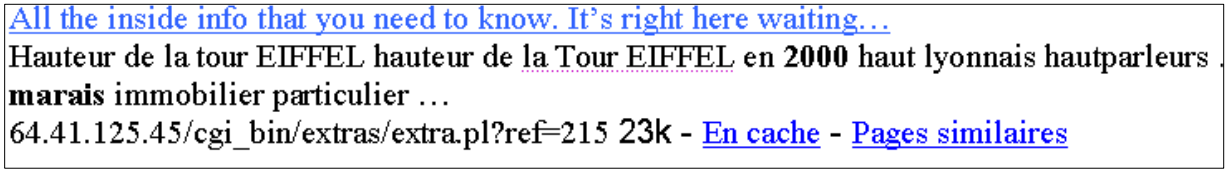

All the inside info that you need to know. It's right here waiting…
Hauteur de la tour EIFFEL hauteur de la Tour EIFFEL en 2000 haut lyonnais hautparleurs **marais** immobilier particulier …
64.41.125.45/cgi_bin/extras/extra.pl?ref=215 23k - En cache - Pages similaires

Figure AVP. 1 : Snippet et Information d'un site retourné par Google dans une liste de résultats.

Le titre est en anglais, l'extrait est en français et parle de la Tour Eiffel, l'URL est une adresse IP (Internet Protocol).

Difficultés pour extraire de l'information des sites web proposés

A partir d'une page HTML : il est difficile pour les utilisateurs d'extraire l'information présente dans la page. Tous les utilisateurs peuvent « rater » des informations sur une page. Ceci pour plusieurs raisons :

- ils cessent de lire la page avant que l'information « pertinente », ou du moins intéressante, ait été atteinte, la page leur paraît trop longue ;

- ils ne lisent que la partie affichée de la page (effet fenêtre) ;

- la page est trop chargée, comme par exemple par une présence excessive de publicité ;

- Ils confondent pages retournées et moteur de recherche. Ils utilisent une fonction de recherche sur un site (Quid, Amazon...) au lieu d'utiliser le moteur de recherche choisi ;

- ils « tournent en rond sur un document », le scénario est alors le suivant : l'utilisateur fait défiler un document, il trouve un lien qui pointe en réalité sur ce même document clique et recommence l'opération (cela peut se produite quatre ou cinq fois sans que l'utilisateur ne remarque que le document parcouru est toujours le même) ;

- ils cessent de lire si la page ou une partie de cette page est dans une langue inconnue.

À partir d'un fichier : l'information à extraire peut se trouver dans un fichier et non dans une page Web. Il faut alors savoir exploiter ce fichier, qu'il s'agisse d'une image (comme pour les partitions musicales par exemple) ou de tout autre type de fichier. Par exemple, Jean recherchait une partition de musique ; il a trouvé un fichier MIDI (son). Mais il ne savait pas qu'il était possible d'extraire la partition à partir d'un fichier MIDI (Musical Instrument Digital Interface) et a abandonné déçu.

Difficultés pour gérer le temps de recherche et sa perception

La majorité des utilisateurs a déclaré que le temps passé sur une recherche n'excédait pas 15 minutes. Or il se trouve que le temps moyen calculé pour une recherche, est de 18 minutes et 45 secondes. De plus, 50% des recherches ont largement dépassé les 15 minutes (Annie cherchera 1 heure 5 minutes et 30 secondes la partition de l'hymne national roumain). Si le temps consacré à la recherche est moins important que le succès ou l'échec dans l'attribution de la note générale moyenne, au-delà de 15 minutes, l'appréciation fait systématiquement apparaître une certaine déception (seulement 5,5/10).

Conclusion sur les observations

Les utilisateurs d'Internet sont déçus par le processus de recherche d'informations. Ils ont des difficultés pour trouver les mots-clés efficaces, pour utiliser un moteur de recherche et pour extraire l'information pertinente à partir de la liste restituée, comme à partir des sites eux-mêmes. Au regard des observations, des questions et des difficultés des internautes, il apparaît que :

- le choix des mots-clés dans un référentiel sémantique ne semble pas suffisant pour retourner un nombre raisonnable de sites et pour garantir un ordonnancement lié à ce champ sémantique ;

- les internautes ne savent pas formuler des requêtes dans des langues étrangères, et lorsque les informations retournées sont dans une langue qui leur est inconnue, ils se considèrent en situation d'échec ;

- les internautes ne comprennent pas forcément les textes retournés par les moteurs de recherche ;

- les formats des informations ou des fichiers retournés ne leur permettent pas toujours d'extraire ni d'exploiter l'information pertinente.

Ce qui frappe l'observateur d'internautes en situation de recherche d'informations est évidemment le besoin de réponses à une multitude de questions posées par la recherche elle-même, par les outils et leurs usages. La méconnaissance de la structure du réseau Internet est

elle aussi un élément qui transforme Internet et ses outils en un environnement mystérieux et donc ressenti comme potentiellement dangereux.

Les moteurs de recherche sont des outils qui ne guident pas l'internaute. En fonction de sa connaissance des mécanismes de ces moteurs et de ses propres compétences (à la fois sur le sujet recherché et sur le format de l'information), l'internaute fournit un référentiel de mots clés qui lui est propre, le plus souvent différent de celui (ou de ceux) utilisé(s) par les auteurs des informations. Dans une requête sans résultat, par exemple, un internaute aguerri et parlant anglais va rapidement élargir sa recherche par l'usage de mots clés dans cette langue. Mais cela est rare dans un contexte grand public.

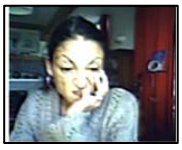
Les internautes sont confrontés aux limites de leurs compétences. Les questions qui doivent être exprimées en utilisant un vocabulaire spécifique au domaine de recherche ou une langue appropriée restent sans réponse. Le vocabulaire est donc une des clés. Mais comment trouver les mots quand précisément ce sont eux qui permettent de trouver les documents qui contiennent ces mots ?

De plus quand les questions sont complexes ou relatives à l'usage du moteur de recherche ou qu'elles se posent dans un domaine où les concepts de bases ne sont pas connus l'internaute interpelle l'observateur.

Dans la liste ci-dessous, nous présentons quelques situations où précisément, les internautes en recherche d'informations sont confrontés à des difficultés d'usage, de compréhension ou de situations appelant à un dialogue.

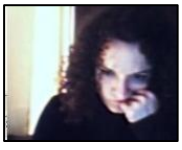
Annie pendant une recherche se demande soudain *« Comment savoir quelle crédibilité je peux donner à ces informations ? ».* http://www.mysissi.com/anniemarais.wmv

Georges dit, qu'il n'arrive pas à taper en anglais. À la question *« C'est anxiogène,* Georges *? »,* la réponse est immédiate *"Oui ! je tape des mots que je ne comprends pas, je ne comprends pas les réponses et je ne comprends pas comment marche le système. »* http://www.mysissi.com/georgesallblues.wmv



Marie est gênée par toute la codification. Elle essaie de cliquer sur tous les mots dans la page. Je devrai par la suite, lui expliquer la notion de lien hypertexte.
http://www.mysissi.com/marieallblues.wmv

Jean n'arrive pas à utiliser des mots en roumain, il revient plusieurs fois sur le même site exactement comme Georges notre débutant. L'expert de 16 ans a le même comportement que le débutant de 49 ans.
http://www.mysissi.com/jeanhymneroumain_media/jeanhymneroumain.wmv



Yasmina, après avoir écouté une chanson et en avoir noté quelques paroles, recherche le texte complet sur Internet. Elle n'a jamais utilisé les mots notés pendant l'écoute comme mots-clés dans le moteur de recherche : *« Je ne pensais pas que ce soit possible ».* http://www.mysissi.com/yasminaprez_media/yasminaprez.wmv

Lou est perdue à la lecture des sites retournés par Google.
http://www.mysissi.com/loulily_media\loulily.wmv

Guillaume recherche un clip vidéo en sachant que c'est une chanson très populaire. Il utilise des mots clés comme "mini clip" sans savoir vraiment pourquoi : « Une fois ça a marché ! ». Guillaume va battre le record du nombre de sites retournés avec 230 millions de sites pour la requête « clip vidéo ». Il va finir par trouver ce qu'il cherchait dans une publicité.
http://www.mysissi.com/guillaumepakito_media/guillaumepakito.wmv

Paul a beaucoup de mal pour retrouver une information dans les pages trop riches. Paul se perd aussi dans le moteur de recherche interne à un site. En fait, il a le réflexe de taper des mots clés dès qu'il voit un champ « recherche ».
http://www.mysissi.com/pauldauphin_media/pauldauphin.wmv

Céline utilise comme tous les autres enfants de l'observation des requêtes en langage « naturel » : « combien de temps vit un dauphin ? » (sic) ou encore « un dauphin vit-il longtemps? ». Elle va même jusqu'à taper le point d'interrogation. http://www.mysissi.com/celinedauphin_media/celinedauphin.wmv

Le lecteur peut avoir à priori le sentiment que seuls les débutants ou les gens d'un certain âge vont se trouver en proie à des difficultés. Il n'en est rien et des internautes ayant 50 ans de différence peuvent rencontrer les mêmes problèmes. Par exemple, Annie, une débutante âgée de 70 ans et Yasmina, une utilisatrice confirmée de 20 ans, vont se retrouver confrontées aux mêmes troubles dans le repérage. Jean, notre expert de 16 ans, va « bloquer » sur l'usage d'une langue inconnue (alors qu'il manipule les outils de traduction sans problème) exactement comme Georges. Ce dernier utilise un moteur de recherche pour la première fois dans cette observation et il a 49 ans.

Comment s'approprier l'information ?

Trouver une information est donc parfois facile, parfois difficile. Le vrai problème réside dans l'appropriation de l'information qui nécessite une phase d'échange et de confrontation. Or, les recherches se font le plus souvent dans une totale solitude. La question d'Annie « Comment savoir quelle crédibilité je peux donner à ces informations ? » restera sans réponse à moins de rompre l'isolement.

Si l'information est le plus souvent immédiatement disponible il n'en est pas de même pour la rencontre et l'échange avec d'autres internautes concernés par le même sujet. Nous avons choisi de nommer cette difficulté : « la solitude du chercheur d'informations ». Les vidéos et les conclusions de ce travail d'observation sont disponibles sur le site http://www.mysissi.com. Ces travaux sont également présentés dans l'article **[Belbeze&al-2007-3]**.

III. Pourquoi briser la solitude du chercheur d'informations ?

L'échange entre pairs est, dans le processus d'acquisition de l'information, étudié sous le nom de conflit-cognitif ou conflit sociocognitif. Un conflit sociocognitif est défini par Tania Zittoun comme *« conflit de points de vue socialement expérimenté et cognitivement résolu »* **[Zittoum-1997]**.

En partant des travaux de Piaget dans le domaine de la psychologie cognitive, Vygotsky a mené des études sur les interactions sociales (dans « *Psychologie et pédagogie »*, Piaget analyse comment les groupes d'enfants sont capables de résoudre collectivement des problèmes et, ce faisant, de progresser cognitivement **[Piaget-1969]**). Vygotsky a fortement contribué à l'élaboration du courant socioconstructiviste. En conférant une dimension sociale essentielle aux processus cognitifs régissant l'apprentissage, Vygotsky a ouvert une nouvelle voie. Pour lui, *« la vraie direction du développement ne va pas de l'individuel au social, mais du social à l'individuel »* **[Vygotsky-1932]**.

Aujourd'hui les modes d'apprentissage recourant aux échanges entre pairs sont fréquents. Ces échanges dans un but pédagogique sont vus par Christophe Gaignon comme de la réciprocité transformatrice ou transformation réciproque: *« Les identités aidant/aidé dialoguent entre elles et l'aide est reçue tantôt par l'un, tantôt par l'autre : la transformation sera réciproque grâce au mouvement circulaire de donner-recevoir. En effet, il n'y a pas d'aidant qui donne et un aidé qui reçoit, parfois nous recevons et parfois nous donnons. »* **[Gaignon-2006]**.

Philipe Meirieu est, quant à lui, très sceptique sur le fait que le travail de groupe puisse permettre une avancée individuelle. Il considère par exemple que *« les pratiques (pédagogiques) de groupe n'apparaissent pas vraiment comme une méthode capable de promouvoir des apprentissages repérables dans le domaine cognitif. ».* Ce que Meirieu veut signifier, c'est que dans un travail de groupe où l'objectif est commun, chacun va travailler dans son champ d'expertise. Il n'y a pas, dans ce cas, de nouveaux apprentissages individuels. Il y a même le risque que celui qui ne sait rien faire, ne fasse rien. Son jugement est différent s'il s'agit de groupes d'apprentissage tels que lui-même les définit. Pour lui, le groupe d'apprentissage a « *sa raison d'être qu'en tant qu'il est l'occasion pour chaque participant, d'atteindre un objectif nommé*. ». Il déclare que dans les groupes d'apprentissage *«… le sujet [y] acquiert la capacité de mettre en correspondance son point de vue ou son apport avec les effets qu'ils entraînent et de conserver, de modifier ou d'abandonner ses propositions à l'issue de l'échange. La confrontation extérieure joue le rôle de régulateur et permet les ajustements que la réflexion solitaire du sujet n'aurait pas toujours autorisés. »* **[Meirieu-1996].**

Effectivement, si chacun possède un but qui lui est propre, l'échange sera bénéfique pour tous. Les communautés d'internautes que nous nous proposons de créer, communautés « d'identités aidant/aidé », correspondraient alors à la définition du groupe de travail donnée

par Meirieu. En effet, dans une rencontre où chaque participant a déjà entamé une recherche préalable de manière indépendante sur un sujet proche voire commun, chaque participant, aurait bien pour but individuel *« d'atteindre un objectif nommé »*.

IV. Comment briser la solitude du chercheur d'informations ?

Devant les difficultés et la solitude du chercheur d'informations, la tentation pour un informaticien à proposer des outils technologiques est grande **[Belbeze&al-2007-1] [Belbeze&al-2007-2]**. Cependant la nature même de ces outils technologiques en fait des éléments supplémentaires à appréhender et à intégrer. Ils sont alors autant de nouvelles sources potentielles de difficultés pour les internautes.

Pour combattre cette solitude et atténuer cette tension qui peut aller jusqu'à l'angoisse et donc au renoncement, la solution la plus naturelle est la recherche d'un échange avec un pair. Cet usage permet à la fois d'exprimer son besoin ou sa difficulté, de mieux le ou la définir. L'échange est aussi propice à la réflexion et bien sûr à l'acquisition de nouveaux éléments.

À ce jour, il n'existe pas à notre connaissance, d'outil générique permettant de rencontrer et converser immédiatement avec des internautes concernés par une même thématique. Les outils de messagerie instantanée, voix sur IP et vidéoconférence ne permettent la connexion qu'avec des personnes déjà connues. Les réponses aux recherches d'aide principalement effectuées dans des forums sont données dans un temps décalé qui ne règle en rien cette « solitude » de l'instant. De plus, la durée de vie d'un thème de recherche peut être très courte. C'est le cas, par exemple, pour des thèmes liés à l'actualité ou à un problème professionnel ponctuel.

L'échange par le biais de nos réseaux sociaux actuels, le plus souvent en temps décalé, par mail ou forum, présente l'avantage d'un contenant forçant le plus souvent une forme de qualité (on réfléchit et on relit son message avant de le poster). Richard Faebert déclare : *« Indéniablement, la place que laissent ces outils de communication (Forum, listes de distribution) à une phase réflexive tire la teneur vers le haut. »* **[Faebert-2002]**. Certes, mais en « tirant vers le haut » le contenu ne va-t-on justement pas rendre l'utilisation de ces outils encore plus exigeante, notamment pour ceux qui ont des difficultés avec l'écrit ? Il manque aussi à ce type de messagerie, la force de l'échange dans le temps de la discussion orale. De plus, ils ont la forme d'une bouteille jetée à la mer. On n'a jamais la certitude d'avoir une réponse ni même d'être lu. Les « conflits cognitifs » sont donc moins nombreux et moins forts puisque tempérés par la relecture et le temps différé voire souvent éteints par une non réponse.

Dans le même article que celui cité plus haut Richard Faebert déclare à propos de la messagerie instantanée : *« Le temps de les taper (les mots) au clavier, vos interlocuteurs ont*

déjà dérivé sur un autre sujet. ». Le problème est que bien souvent, il n'y a en fait pas de sujet. Si l'on se retrouve autour de la messagerie instantanée comme autour d'un verre dans une conversation phatique, sans réel désir de partager une thématique commune, il n'est pas surprenant que les sujets glissent de l'un à l'autre. C'est alors l'absence de volonté de débattre ou de partager une thématique et non le média qui est à remettre en cause.

Il n'est pas question ici de juger de l'efficacité d'un type de communication par rapport à un autre. Remarquons cependant que si Internet propose de l'information immédiatement disponible, la rencontre et l'échange entre internautes autour de cette même information (sur des forums), ne possèdent pas, elles, cette « immédiateté » qui serait pourtant parfois souhaitable. Ainsi, la spontanéité du promeneur qui, se sentant perdu, interroge un passant pour retrouver son chemin, ou celle du voisin qui nous interpelle pour échanger un sentiment sur une actualité locale, ou encore celle qui conduit des étudiants à partager leurs interrogations à propos d'un travail scolaire, sont des attitudes impossibles sur le réseau Internet sans connaitre l'interlocuteur.

De plus, les outils de messagerie instantanée sont eux aussi en perpétuelle évolution : intégrant la vidéo et la voix sur IP ils permettent aujourd'hui des échanges en groupe de plus en plus simples.

Nous recherchons donc une solution permettant de trouver des pairs et de créer des espaces d'échange le plus rapidement possible. Cette solution permettra la création de ce que nous nommerons des « **Communautés Dynamiques** ».

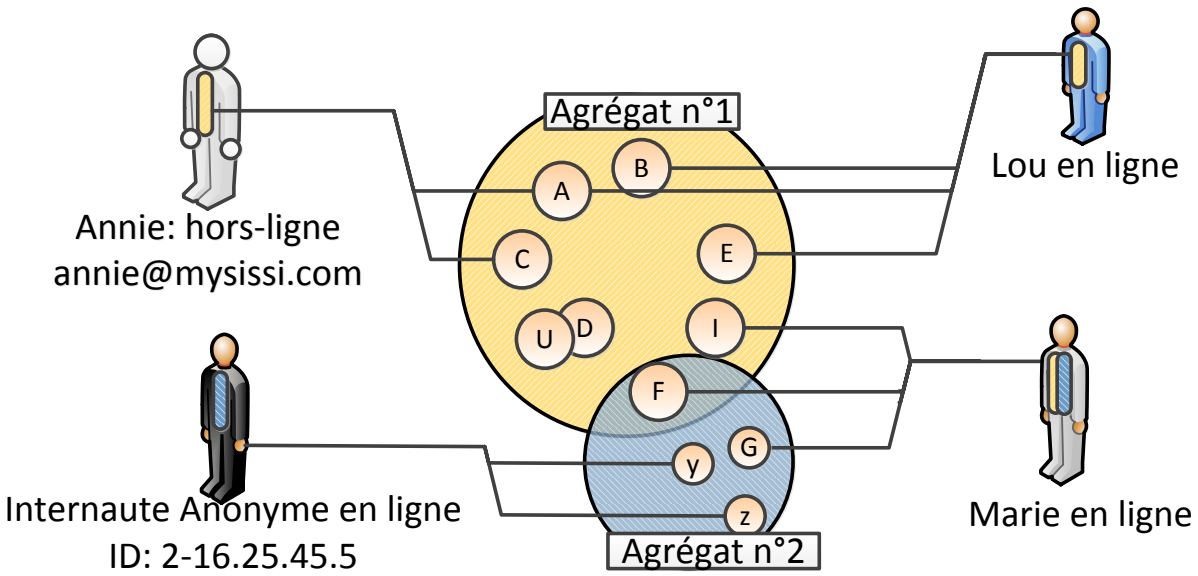


Figure AVP. 2 : Attachement des internautes à un agrégat en fonction de leurs recherches.

Les rencontres entre utilisateurs se feront sur la base d'un attachement à une thématique commune. Le repérage et la construction des thèmes seront automatisés pour permettre la création de communautés d'utilisateurs de type « génération spontanée » ou communautés dynamiques.

Les communautés dynamiques donneront aux internautes la capacité de communiquer entre eux de manière instantanée ou différée. Les utilisateurs emploieront comme outil de recherche de lien social le même moteur de recherche que celui servant à la recherche d'information. Ils n'auront pas d'opération supplémentaire ou nouvelle à apprendre.

Nous pouvons définir une **Communauté Dynamique** comme constituée de deux types d'éléments :

- Premièrement, un objet contenant un certain nombre de mots, l'**agrégat**. Celui-ci est construit à partir des usages conjoints de mots effectués par les utilisateurs dans une requête. Le principe de la construction d'un agrégat doit permettre de s'assurer d'une cohérence sémantique. L'évaluation de cette cohérence sémantique est un élément majeur.

- Deuxièmement, des **internautes** qui sont rattachés à un agrégat (les internautes auront utilisé suffisamment de mots de l'agrégat pour se voir rattachés à celui-ci).

La création de communautés dynamiques d'utilisateurs pourrait donc être exploitée afin de permettre à un utilisateur de coopérer avec d'autres sans avoir ni à s'authentifier, ni à se décrire, ni même à s'inscrire dans ces espaces. Il n'en reste pas moins que la signature ou un élément de communication permanent, comme une adresse de messagerie électronique, permettront un fonctionnement asynchrone du système.

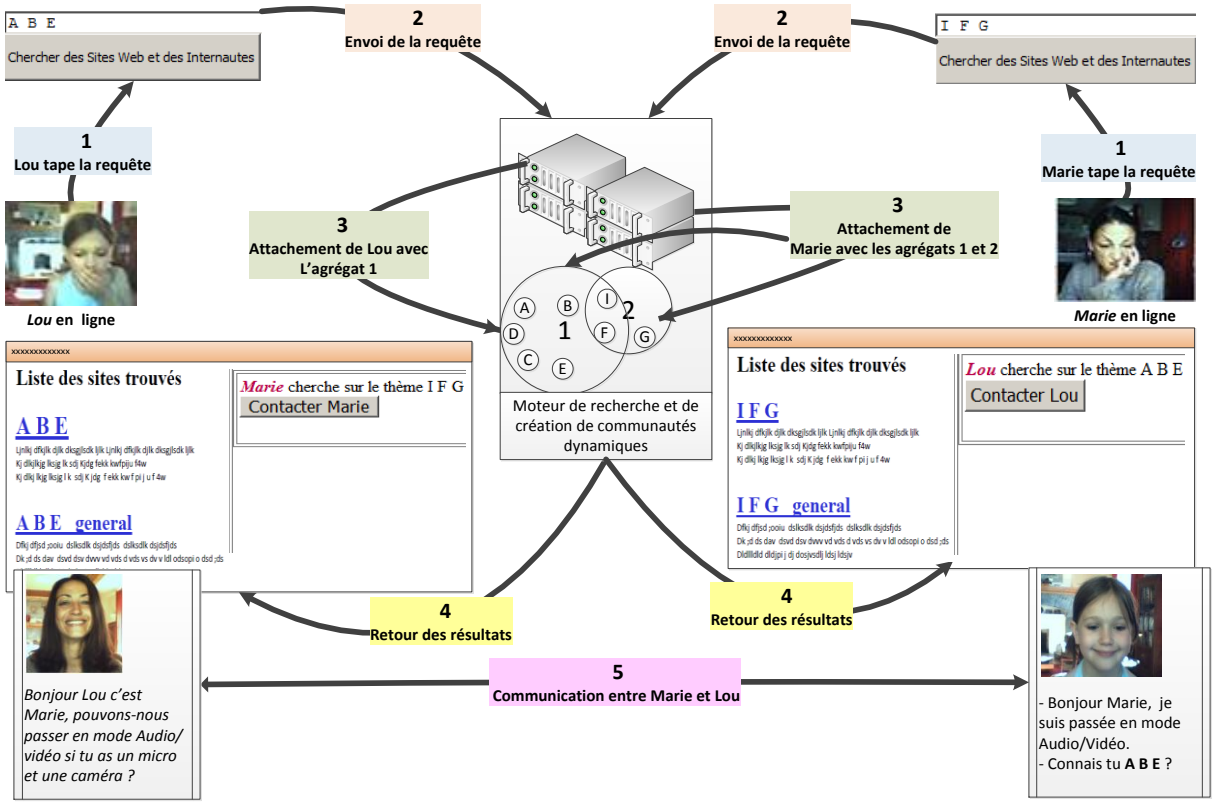


Figure AVP.3 : Exemple d'utilisation des nouveaux services de communication au sein de la communauté dynamique avec des utilisateurs en ligne.

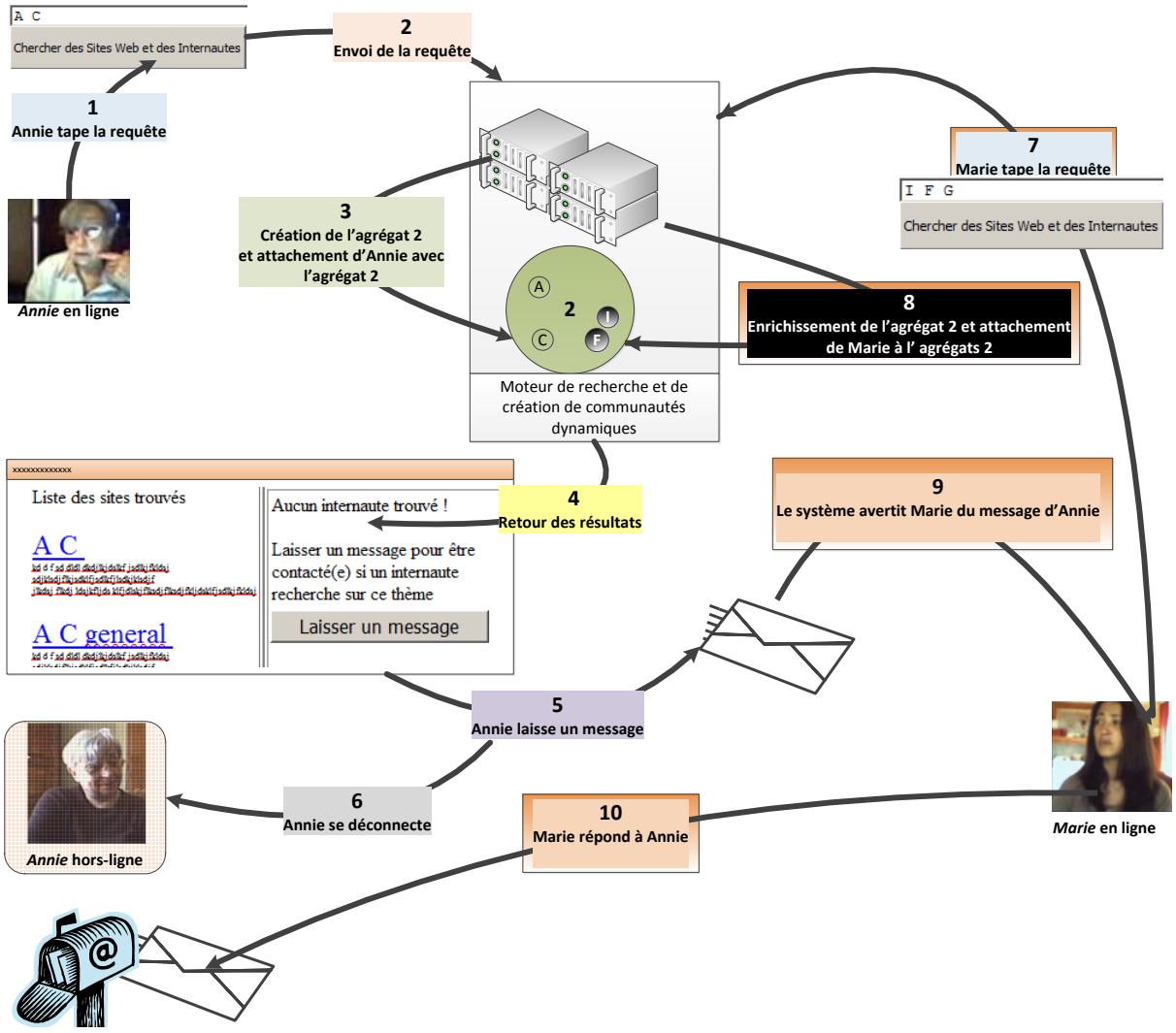


Figure AVP.4 : Exemple d'utilisation des nouveaux services de communication au sein de la communauté dynamique avec des utilisateurs hors-ligne.

Dans l'exemple présenté dans les figures AVP.2, AVP.3 et AVP.4, Marie recherche des sites web en utilisant les mots-clés I, F et G. Elle est donc connectée comme ayant des recherches sur les thématiques de l'agrégat 1 et 2. Lou est elle aussi connectée sur l'agrégat 1. Marie et Lou peuvent se voir proposer une connexion de façon dynamique telle que celle décrite dans la figure AVP.3.

Marie peut aussi se voir proposer d'échanger avec d'autres utilisateurs ayant des centres d'intérêts proches des siens. Elle peut aussi ouvrir un salon de discussion où seront invités automatiquement les internautes concernés par les mots-clés de l'agrégat N°1 ou N°2. Elle peut démarrer une conversation en messagerie instantanée avec l'utilisateur « Anonyme » en utilisant un identifiant temporaire (adresse IP + numéro de session). Les agrégats peuvent être créés et recalculés par un processus planifié. Ceux-ci pourront ensuite être mis à jour ponctuellement par de nouvelles recherches d'internautes s'il y a lieu.

Marie a aussi la possibilité de répondre au message d'Annie comme dans la figure AVP.4, le système ayant en quelque sorte créé un forum dynamique dont Annie et Marie sont les premiers membres.

L'affiliation d'un utilisateur à une communauté dynamique peut aussi être vue comme un typage temps réel de l'utilisateur du système. En effet, le profil de l'utilisateur est immédiatement modifié en fonction des actions de recherche.

L'objectif des travaux que nous présentons est de définir une méthode de création de ces agrégats de mots-clés auxquels un utilisateur pourra être rattaché. De fait, le regroupement ou la création d'agrégats a pour objectif, dans un ensemble de mots donné, de rassembler les éléments les plus proches possibles selon un ou plusieurs critères. Il a également pour but de créer des agrégats les plus éloignés possibles, sur ce ou ces critères. Le critère prédominant utilisé dans notre cas sera l'homogénéité sémantique.

Une fois l'appartenance de l'utilisateur à une communauté celui-ci peut se voir proposer un grand nombre de services. Des mots-clés supplémentaires dans une recherche ou la définition de contextes de recherche sont autant de services susceptibles d'aider les utilisateurs à accéder à toute information utile, voire à optimiser l'accès à cette information par un partage implicite de compétences.

V. Dernière justification…

La dernière justification à la mise en œuvre de tels outils de rencontre est simplement le plaisir d'échanger en instantané sur un intérêt commun. Le plaisir qui est celui de partager avec une intelligence humaine, vivante et interactive est aussi celui de la poésie de la rencontre d'un mot dit, entendu, écrit ou lu par l'autre et du moment présent, moment qui n'est déjà plus, et dont l'instantanéité rejoint le mystère de la vie.

Introduction générale

I. Contexte et motivation

En 2010 plus de 20% de la population mondiale avait accès à Internet (http://donnees.banquemondiale.org). Ce média ne cesse de prendre de l'importance. L'expérimentation montre, par ailleurs, qu'un nombre important d'internautes éprouve des difficultés d'utilisation, de repérage et d'accès à l'information. Or, des communications de pairs à pairs entre usagers pourraient aider à les résoudre.

La matière première dont nous disposons, pour rapprocher ces utilisateurs est constituée des requêtes soumises aux moteurs de recherche. Afin de proposer les liens les plus pertinents possibles entre usagers, nous devons avant tout cerner les thématiques partagées entre ces derniers. Nous nous appliquerons à détecter ces thématiques dans l'ensemble des mots-clés constituant les requêtes, de sorte qu'au cours de leurs recherches les internautes soient mis en relation automatiquement. Ainsi se constitueront ce que nous appellerons des « Communautés Dynamiques » (cf. Avant-propos).

La matière première constituée de mots et de leurs utilisations conjointes a permis la création de graphes. Les mots représentent les nœuds et les co-utilisations, au sein des requêtes, les liaisons. Le graphe de mots ainsi constitué est issu du monde réel. Il est dit graphe de terrain par opposition au graphe généré mathématiquement. Nous avons ainsi positionné notre espace de recherche comme faisant partie de l'étude des graphes de terrain.

L'étude des grands graphes de terrain et plus particulièrement l'aspect qui s'attache à la création de groupes appelés communément « communautés », est un espace de recherche suscitant un fort engouement. Un graphe est un modèle particulièrement efficace pour représenter des interactions entre des objets en très grand nombre. L'étude des grands graphes

de terrain a permis de relever des propriétés communes à ces réseaux que nous étudierons pour la construction de communautés dynamiques.

Les graphes obtenus à partir d'un fichier de log de requêtes issu d'un moteur de recherche ont pour des périodes de quelques semaines, un nombre de nœuds (de mots) supérieur à un million et plusieurs dizaines de millions de liaisons. En raison de la taille importante de ces graphes et de leur origine liée à un usage, ces graphes peuvent être considérés comme un Grand Graphe de Terrain.

Le noyau de la communauté dynamique sera un ensemble de mots permettant la connexion entre utilisateurs. Cet ensemble de mots devra représenter un espace sémantique cohérent autour d'une thématique précise.

Nous considérons que l'usage volontaire de mots associés dans un même texte (par exemple dans une requête utilisateur) par un auteur est le critère déterminant la cohérence sémantique entre ces mots. La cohérence sémantique est donc consécutive à l'intention d'un auteur.

Notre but est ainsi d'obtenir des agrégats de mots sémantiquement cohérents issus d'un Grand Graphe de Terrain.

II. Approche et principaux objectifs

L'essentiel de notre approche consiste à agréger les nœuds d'un graphe ; chaque agrégat obtenu devant correspondre à un ensemble présentant une cohérence sémantique. Notre approche se propose de traiter principalement les problématiques suivantes :

- *Créer des agrégats de mots pouvant contenir des parties en recouvrement*. Une orthographe peut appartenir à plusieurs thématiques. Pour cette raison nous étudions plus particulièrement les méthodes de regroupement avec recouvrements.

- *Définir une technique de regroupement garantissant une forte cohérence sémantique*. Pour cela nous proposons et utilisons plusieurs techniques de regroupement avec recouvrements ou de création de recouvrements et de validation sémantique dont nous comparerons les résultats.

- *Caractériser les agrégats pour comprendre les différences de cohérence sémantique.* Nous recherchons par une évaluation sémantique en fonction de caractéristiques et plus particulièrement de la taille des agrégats, à déterminer ce qui fait la différence entre des agrégats de forte et de faible homogénéité sémantique.

- *Créer des agrégats non pollués.* Les mots ne sont pas tous égaux entre eux en tant que signifiants. Les mots de liaisons ou les articles ne sont pas, par

exemple, porteurs de sens. Nous rechercherons une technique de regroupement qui a la capacité d'écarter ou de conserver ces mots en fonction de leurs usages dans la globalité du graphe et dans la relation locale aux mots de l'agrégat.

- *Proposer des techniques de validation de la cohérence sémantique des agrégats*. Nous proposons et mettons en œuvre plusieurs techniques de validation de la cohérence sémantique des agrégats, notamment une technique de validation basée sur la comparaison du « comportement » d'agrégats avec le comportement « des requêtes d'utilisateurs » et d'agrégats aléatoires lorsqu'ils sont utilisés comme élément de requêtes dans des moteurs de recherche. D'autres techniques automatiques, manuelles ou semi manuelles sont utilisées et comparées.

III. **Plan du mémoire**

Ce mémoire est constitué de deux parties.

La première partie présente le contexte de notre travail et l'état de l'art des travaux connexes. Cette première partie est divisée en deux chapitres :

- Dans le premier chapitre, nous introduisons le vocabulaire utilisé dans le mémoire.
- Dans le second chapitre, nous proposons un état de l'art des méthodes utilisées pour créer des communautés dans un graphe. Nous étudierons ces différentes propositions en fonction de notre objectif. Dans notre cas la nature des objets manipulés - des agrégats de mots représentant un thème - nous ont amenés à classer ces méthodes en deux familles principales : les méthodes sans recouvrements et les méthodes avec recouvrements.

Dans une deuxième partie nous décrivons notre contribution. Fondée sur une recherche orientée sur la création de regroupements de mots, elle ne prétend en aucun cas se positionner comme une technique universelle. Cette deuxième partie est partagée en deux chapitres.

- Dans le troisième chapitre, nous exposons plusieurs techniques de regroupement. Nous justifions l'usage d'une nouvelle technique fondée sur la résolution de contraintes ainsi que ses évolutions et des techniques complémentaires.
- Dans le quatrième chapitre, nous présentons plusieurs techniques d'évaluation de la validité sémantique des agrégats de mots obtenus par les méthodes du chapitre précédents.

Enfin, dans un cinquième chapitre, nous faisons partager au lecteur quelques réflexions, retours d'expériences et sentiments personnels sur notre expérience.

Première partie.

Définitions et état de l'art

La première partie a pour but de donner au lecteur les éléments nécessaires à la compréhension de ce mémoire et d'effectuer un état de l'art des technologies de regroupement.

Le premier chapitre introduit les graphes et leurs caractéristiques. Cette partie ne se veut en rien exhaustive. Au contraire, nous ne couvrons ici que les notions présentes dans ce travail. Il est conseillé au lecteur recherchant des informations plus complètes sur les graphes de se référer à d'autres ouvrages tels que « *Théorie des graphes et ses applications* » de Claude Berge **[Berge-1958]** ou encore du même auteur « *Graphes et hypergraphes* » **[Berge-1970]** et enfin de Béllé Bolllobas **«** *Modern Graph Theory »* **[Bollobas-1998]**. Dans ce chapitre, nous explicitons aussi, autant que faire se peut, en avant-propos, les termes spécifiques utilisés et tentons de les situer et d'en évaluer la pertinence.

Dans un second chapitre, nous effectuons un état de l'art des diverses méthodes de détection de communautés dans les graphes. Nous tentons de cerner leurs intérêts et leurs limites.

Chapitre 1.

État de l'art, notions, définitions et vocabulaire sur les graphes

1.1 Introduction

Manuel Castells, sociologue américain, définit Internet comme le *« ... produit d'une combinaison unique de stratégie militaire, de coopération scientifique et d'innovation contestataire »*. Ce qui est notable dans cette définition amusante est la diversité des composantes d'Internet. Cette diversité est un facteur de croissance. D'une manière plus générale, les réseaux créés à des fins d'utilisation, tels que les transports en commun, le mail, le téléphone, ou le « cloud computing », ont souvent des croissances d'usage exponentielles et ceci d'autant plus que leurs clients sont hétérogènes. Devenus populaires, les réseaux se mettent à porter des noms, quelquefois des noms de marques et parfois même des noms propres comme Internet. Identifiés, utilisés par tous, ces réseaux offrent l'attrait de nouveaux usages.

Pour symboliser ces réseaux constitués, par définition, d'objets en relation les uns avec les autres, on utilise le plus souvent une représentation sous forme de « graphes ». L'étude des graphes ou « Théorie des graphes » est en premier lieu une théorie mathématique. Mais l'importance de ces réseaux dans notre quotidien pousse de plus en plus les femmes et les hommes « de l'art » à les étudier. Ainsi, nombre d'informaticiens étudient Internet en passant par des représentations graphiques et nombre de sociologues utilisent les graphes dans des études de réseaux sociaux, par exemple. Les graphes ont leurs règles, leurs vocables et leur histoire. C'est de ces éléments dont nous allons traiter dans ce chapitre.

1.2 Historique

Les premières études sur la théorie des graphes sont celles effectuées par Leonhard Eulervdans avec sa recherche d'une solution au problème des ponts de Königsberg (Euler 1736). La ville de Königsberg située en Prusse est alors constituée de deux iles reliées par sept ponts (cf. figure 1.1). La ville se nomme aujourd'hui Kaliningrad.

1.2.1 Le problème

Le problème posé est de trouver un chemin permettant de passer sur chaque pont en n'empruntant chaque pont qu'une seule fois.

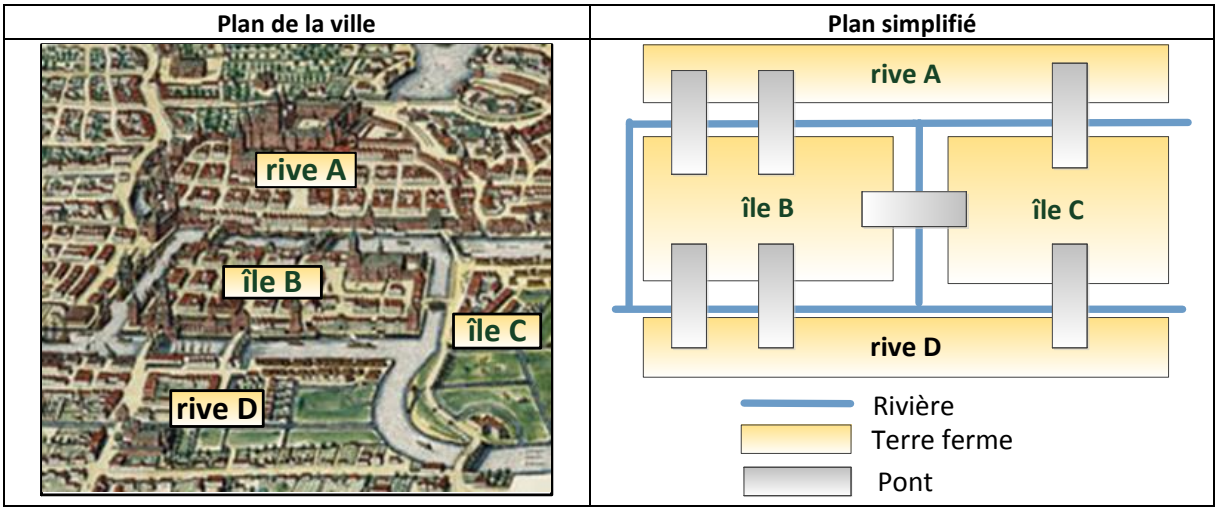


Figure 1.1 : Les sept ponts de Königsberg.

Leonhard Euler va dessiner un schéma où rives et îles seront représentées par des points et chaque pont comme des « fils » entre ces points, créant ainsi un graphe (cf. Figure 1.2).

1.2.2 La réponse par le graphe

Les points de terre ferme sont les nœuds ou sommets du graphe. Les nœuds et sommets représentent toujours les objets connectés du graphe. Habituellement un nœud (ou un sommet) représente un objet actif du graphe. Dans un réseau social, les nœuds représentent des personnes et par transposition, les connections leurs relations sociales, par exemple.

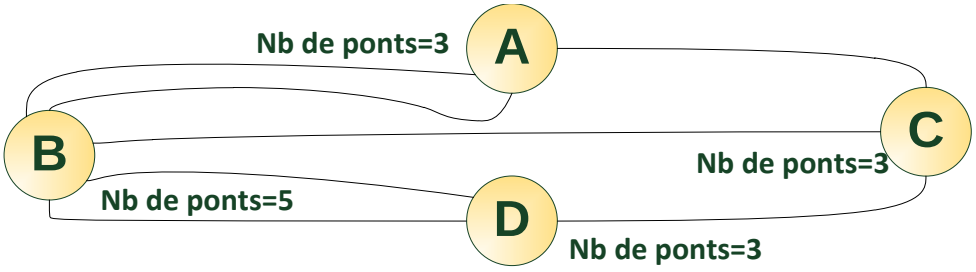


Figure 1.2 : Les sept ponts de Königsberg dans une représentation graphique.

Une fois la représentation graphique créée, la question : « Peut-on faire un parcours passant par les sept ponts en n'utilisant qu'une seule fois chaque pont ? » se résume à : « Existe-t-il un chemin pour revenir d'un point ferme à un autre, différent de celui pris pour aller ? ». Si nous notons à coté de chaque nœud (point de terre ferme) le nombre de ponts (cf. figure 1.2), il devient évident que ce nombre étant toujours impair, il ne sera pas possible depuis un point de terre ferme visité en « milieu » de promenade de revenir directement au point précédent sans réemprunter un pont déjà utilisé.

Cette caractéristique n'est pas nécessaire pour tous les nœuds. Elle l'est cependant pour au moins deux : celui de départ et celui de fin. Aucun point de terre ferme n'étant accessible par un nombre pair de pont, la réponse est finalement qu'il n'est pas possible d'effectuer la promenade demandée.

La représentation graphique nous permet donc d'affirmer qu'il n'existe pas de solution à ce problème.

Il est par ailleurs intéressant de noter certains enseignements fournis par ce travail fondateur :

- C'est la pondération des éléments de terre ferme par le nombre de ponts qui permet de trouver la réponse au problème.
- Une fois le graphe créé, il n'est plus nécessaire de le parcourir pour connaître les informations nous permettant de répondre à la question posée. La localisation des ponts et des points de terre ferme n'a plus d'importance. Et on pourrait tout à fait répondre à la question sans représenter les fils entre les points de terre ferme.

Comme on peut le voir, une représentation d'un réseau par un graphe permet de répondre à une question donnée. La représentation et les informations à représenter sont à choisir en fonction de la nature du graphe et de la question à résoudre. Dans notre travail nous aurons donc à rechercher une représentation graphique la plus efficace possible, pour répondre à nos questions de regroupement.

Nous nous devons aussi de souligner que cette étude porte sur un réseau d'usage (nos promeneurs utilisent les ponts) et de « terrain » au sens premier du mot.

1.3 Notions et définitions

La représentation mentale d'un graphe est généralement aisée et la notion de nœud et liaison est le plus souvent comprise de manière intuitive. Cependant, l'utilisation d'une terminologie précise se révèle nécessaire dès qu'il s'agit d'approfondir l'étude de ces ensembles.

Voici listées les notions utilisées dans notre contexte de travail ; à noter que certaines définitions données peuvent tenir compte de notre point de vue. Pour une information plus complète il est possible de consulter plusieurs ouvrages de référence tels que **[Berge-1958] [Berge-1970] [Bollobas-1998]**.

Arc

Un arc est le nom donné à une liaison ou à une arête dans un graphe dirigé.

Arête

Élément reliant deux points d'un graphe. Généralement représenté par un segment de droite. Dans la matrice d'adjacence du graphe la présence de l'arête est représentée par un 1 et son absence par un 0.

Arête orientée

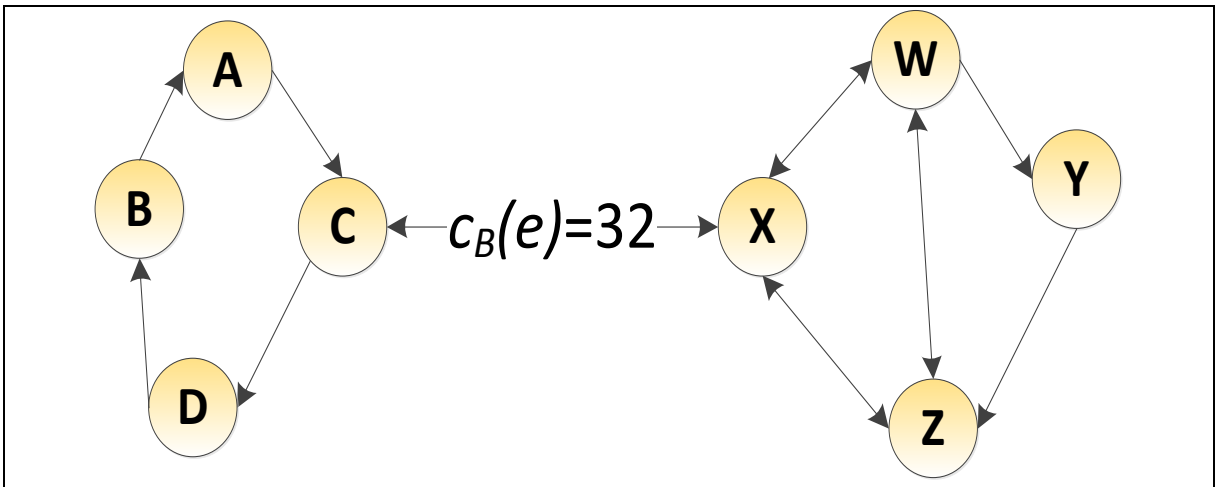
Une arête orientée est une arête présentant un sens. Un des pairs est un émetteur l'autre un récepteur. On parle aussi d'arc. Les arêtes orientées sont des éléments des graphes orientés.

Arête pondérée

Une arête pondérée est une arête présentant un poids. Ce poids est une valeur numérique permettant de comparer la validité des arêtes. Les arêtes pondérées sont des éléments des graphes pondérés.

Centralité

La centralité d'une arête e, notée $c_B(e)$, est définie comme le nombre de plus court(s) chemin(s) entre toutes paires de nœuds contenant e : ainsi, si la centralité d'une arête est grande, on peut s'attendre à ce qu'elle se trouve à l'interface entre deux communautés du graphe considéré. Cette notion peut facilement être étendue aux nœuds en considérant le nombre de plus court(s) chemin(s) passant par un nœud donné. Une arête à forte centralité est considérée comme un séparateur possible de communautés.



Liste des plus courts chemins passant par l'arête [C-X/X-C] en considérant le graphe comme dirigé							
A-W	A-X	A-Y	A-Z	W-A	X-A	Y-A	Z-A
B-W	B-X	B-Y	B-Z	W-B	X-B	Y-B	Z-B
C-W	C-X	C-Y	C-Z	W-C	W-C	Y-C	Z-C
D-W	D-X	D-Y	D-Z	W-D	W-D	Y-D	Z-D

Figure 1.3 : Exemple d'arête ayant une centralité élevée et étant susceptible de séparer deux communautés.

Chemin

Un chemin d'un nœud A à un nœud Z est une suite de nœuds reliés par des arêtes tel qu'il est possible de se déplacer du nœud A au nœud Z en parcourant les nœuds du chemin.

Clique

Une clique peut être définie comme une figure connectée de trois nœuds minimum d'un graphe non dirigé dans laquelle on ne peut rajouter ni lien ni nœud. En effet, chacun des nœuds doit être connecté à tous les autres nœuds et il ne doit pas exister de nœud connecté à tous ces nœuds qui ne seraient pris en compte. La clique est aussi définie comme un sous-graphe complet.

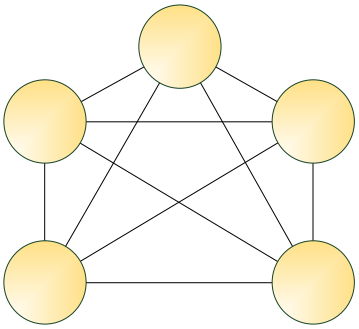


Figure 1.4 : Exemple de clique de 5 sommets.

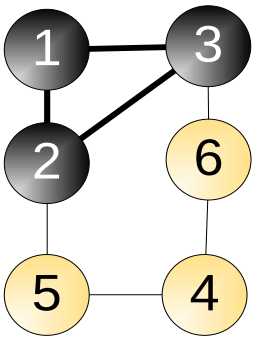


Figure 1.5 : En noir, un exemple de clique formée par les nœuds 1,2 et 3 dans un graphe.

Composante connexe

Une composante connexe est une partie du graphe où il existe au moins un chemin pour rejoindre tous les nœuds de la composante connexe. Si un graphe est seulement et totalement composante connexe, on utilise le terme de graphe connexe.

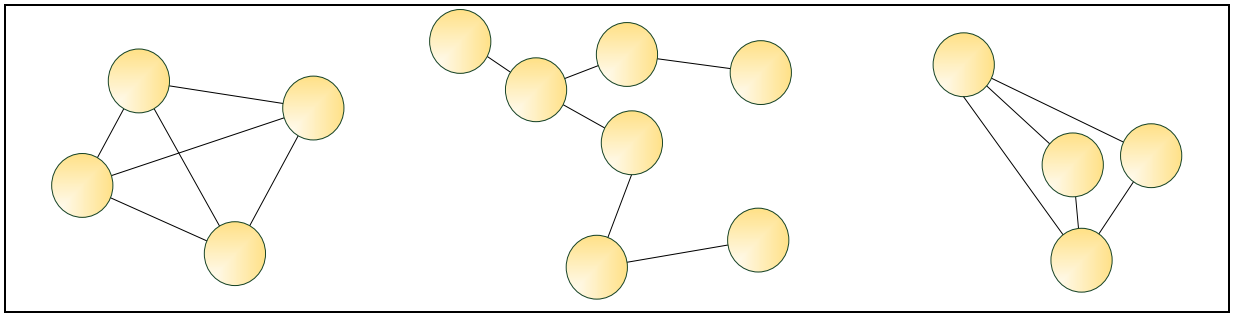


Figure 1.6 : Graphe à 3 composantes connexes.

Degré

Dans un graphe dans lequel un nœud ne peut pas recevoir de liaison avec lui-même, le degré d'un nœud est simplement le nombre de nœuds avec lequel il existe une liaison. On parle aussi du nombre de nœuds voisins.

Si le graphe est défini comme acceptant des liaisons autoportées, ses liaisons autoportées ont par convention un poids double.

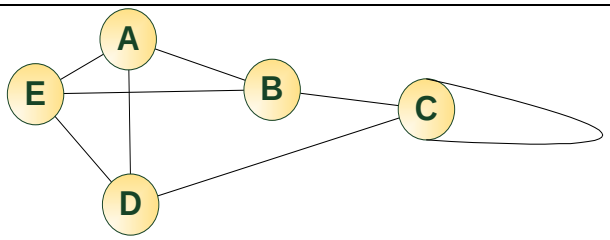
Degré des nœuds du graphe		Graphe
Nœud	Degré	
A	3	
B	3	
C	4	
D	3	
E	3	

Figure 1.7 : Exemple de valeur de degré pour les nœuds d'un graphe incluant une liaison autoportée.

Densité d'un graphe

La densité est le rapport entre le nombre d'arêtes présentes dans le graphe étudié sur le nombre maximal d'arêtes possible sur un graphe contenant le même nombre de nœuds. Dans le cas où le nombre de liaisons par nœud n'est pas limité, ce nombre maximal est, pour un graphe de n éléments, le nombre de paires possibles que l'on peut noter (combinaison de n éléments d'ordre 2) soit $C_n^2 = \frac{n!}{2(n-2)!}$.

Diade

Une diade est une paire de nœuds connectés.

Diamètre d'un graphe

Le diamètre d'un graphe est la distance la plus élevée entre deux nœuds en utilisant le chemin le plus direct. Une géodésique est l'un des plus courts chemins entre deux sommets donnés. Le diamètre d'un graphe peut être défini comme le plus long chemin géodésique du graphe.

Diamètre effectif

Le diamètre d'un graphe est défini comme une distance maximale et peut donc être une valeur non représentative car trop marginale. Pour éviter cette dérive, plusieurs auteurs proposent de mesurer le diamètre effectif ou *petit diamètre* [Leskovec&al-2005]. Le diamètre effectif ou *petit diamètre* est le nombre minimum de sauts ou liaisons dans lequel une fraction (ou quantile q, par exemple q = 90%) de toutes les paires de nœuds connectés sont présentes.

Graphe (et représentation)

Un graphe est, dans sa représentation dessinée, un ensemble de points dont certaines paires sont reliées par un lien. Le positionnement des points et la longueur des liaisons ne sont pas significatives. Ainsi, le graphe de la figure 1.8 est le même quelles qu'en soient les représentations.

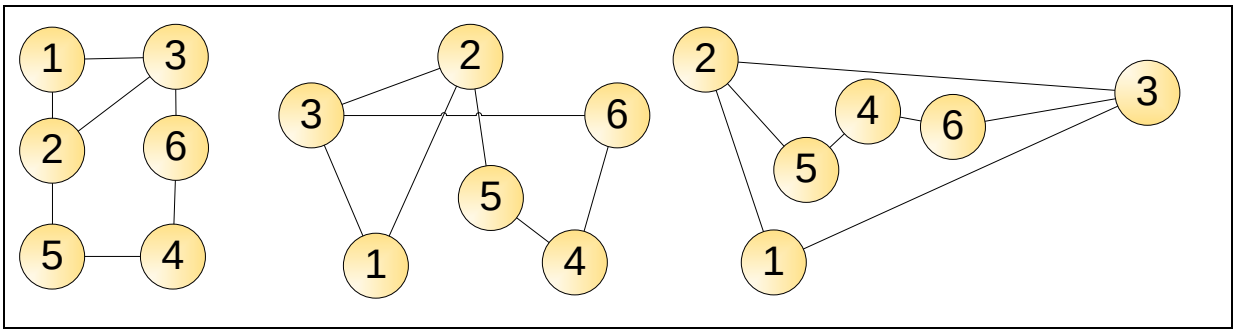


Figure 1.8– Plusieurs représentations dessinées d'un même graphe.

Il est aussi possible de représenter un graphe par une représentation matricielle. Plusieurs types de matrices [*matrix*] existent. La plus connue et usuelle est la matrice d'adjacence $\mathbf{M_A}$. Les nœuds sont présents en abscisses et ordonnées, la jonction de deux nœuds étant alors par convention représentée par un 1 si une liaison existe et un 0 si ce n'est pas le cas.

La matrice des degrés est aussi couramment utilisée. Elle donne, en plus de la matrice des liaisons, des informations sur la valeur des degrés des nœuds. La matrice Laplacienne non normalisée *L* est la matrice résultante de $\mathbf{M_D}$- $\mathbf{M_A}$.

On utilise aussi la matrice Laplacienne normalisée. Dans ce cas-là, la fonction de normalisation *N(x,y)* est égale à 0 si *x* et *y* ne partagent pas de liaison, égale à 1 si *x*= *y* et degré *(x)* > 0, et égale à $\frac{-1}{\sqrt{degré(x).degré(y)}}$ sinon.

M_A-Matrice d'adjacence	M_D-Matrice des degrés du graphe ou matrice diagonale.	M_L-Matrice Laplacienne non normalisée	Matrice Laplacienne normalisée
$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$	$\begin{bmatrix} 2 & -1 & - & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 \\ -1 & -1 & 3 & 0 & 0 & -1 \\ 0 & 0 & 0 & 2 & -1 & -1 \\ 0 & -1 & 0 & -1 & 2 & 0 \\ 0 & 0 & -1 & -1 & 0 & 2 \end{bmatrix}$	$\begin{bmatrix} 1 & \frac{-1}{\sqrt{6}} & \frac{-1}{\sqrt{6}} & 0 & 0 & 0 \\ \frac{-1}{\sqrt{6}} & 1 & \frac{-1}{3} & 0 & \frac{-1}{\sqrt{6}} & 0 \\ \frac{-1}{\sqrt{6}} & \frac{-1}{3} & 1 & 0 & 0 & \frac{-1}{\sqrt{6}} \\ 0 & 0 & 0 & 1 & \frac{-1}{2} & \frac{-1}{2} \\ 0 & \frac{-1}{\sqrt{6}} & 0 & \frac{-1}{2} & 1 & 0 \\ 0 & 0 & \frac{-1}{\sqrt{6}} & \frac{-1}{2} & 0 & 1 \end{bmatrix}$

Tableau 1.1 : Représentations matricielles du graphe de la figure 1.8.

Graphe de terrain

Ensemble d'objets naturels ou existants physiquement dans le monde réel dont les interactions sont exprimables par des arêtes entre paires d'objets. Les graphes de terrains sont ainsi constitués d'éléments aussi divers que des personnes humaines échangeant des mails, des ordinateurs échangeant des trames IP, des mots présents dans la même définition, etc.

Graphe dirigé

Un graphe dirigé est un graphe dont les liaisons appelées arcs ont un sens. Un des nœuds est un émetteur et l'autre un récepteur. Un exemple bien connu de graphe dirigé est l'arbre généalogique inversé. Se lisant du haut vers le bas, le sens de l'arc du haut vers le bas signifie « Parent de ».

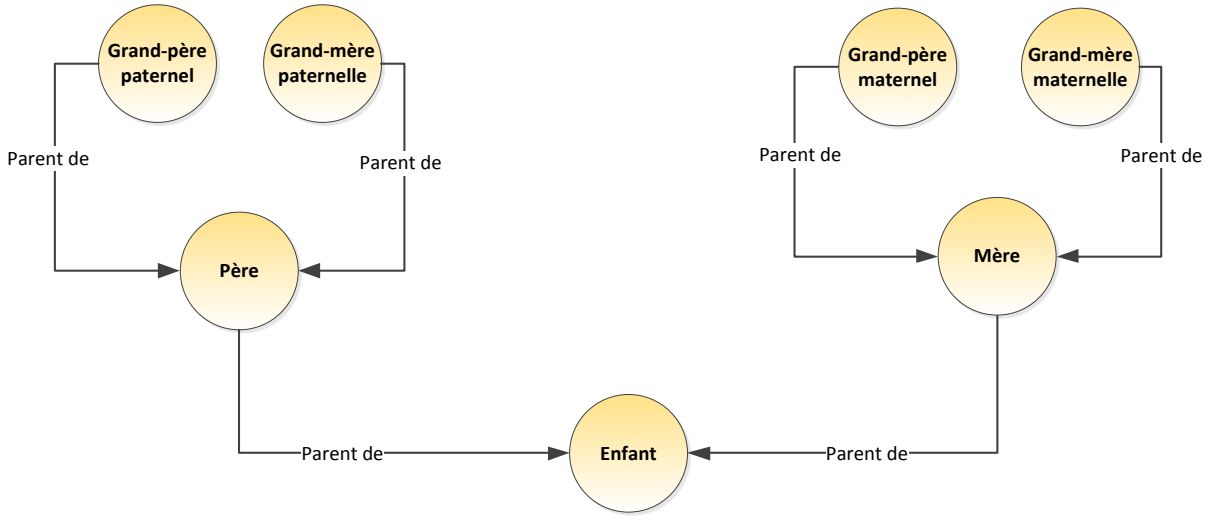


Figure 1.9 : Exemple de graphe dirigé.

Graphe pondéré

Un graphe pondéré est un graphe dans lequel les nœuds et les liaisons peuvent recevoir une valeur numérique. Ces valeurs peuvent être soit calculées par des informations sur le graphe lui-même soit être des informations complémentaires. Par exemple dans un réseau social d'échange par Emails, les acteurs peuvent être pondérés par la somme de tous les Emails reçus et chaque liaison par le nombre d'Emails échangés.

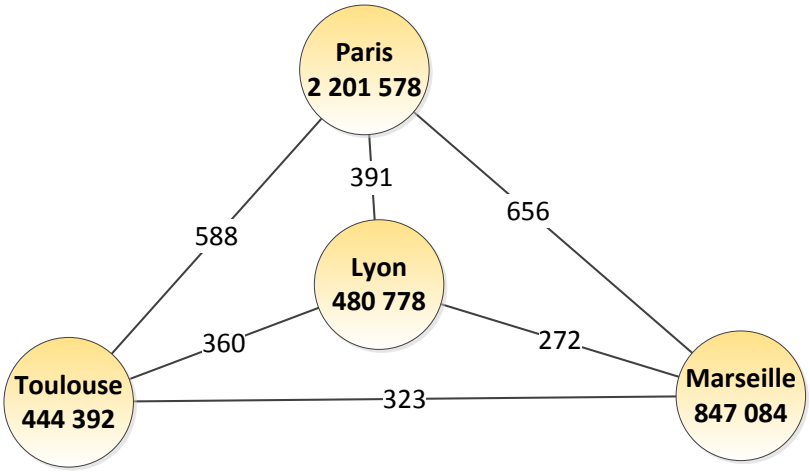


Figure 1.10 : Graphe de villes de France. Les villes sont pondérées par le nombre d'habitants et les liaisons par la distance à vol d'oiseau.

Graphe pondéré et dirigé

Un graphe pondéré et dirigé va cumuler des arcs et des pondérations. Le graphe d'un site web est naturellement un graphe dirigé et pondéré. Les pages du site représentent ici les nœuds et les arcs représentent les liens. La pondération des nœuds peut être donnée par le nombre de liens sortant de chaque page du site, les arêtes seront dirigées comme le sont les liens entre les pages et pondérées par le nombre de liens.

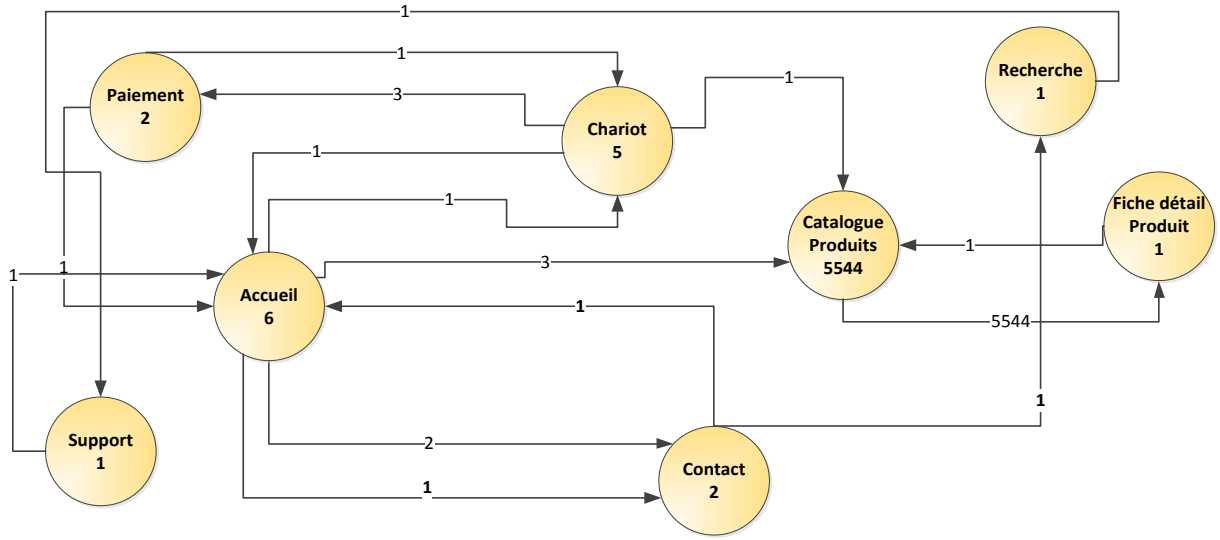


Figure 1.11 : Exemple de graphe dirigé et pondéré d'un site web d'E-commerce.

Graphe bi-connexe

Un graphe est dit bi-connexe s'il existe entre chaque nœud au moins deux chemins complètement distincts. Une autre définition possible est que la suppression de n'importe quel lien permet au graphe de rester une composante connexe.

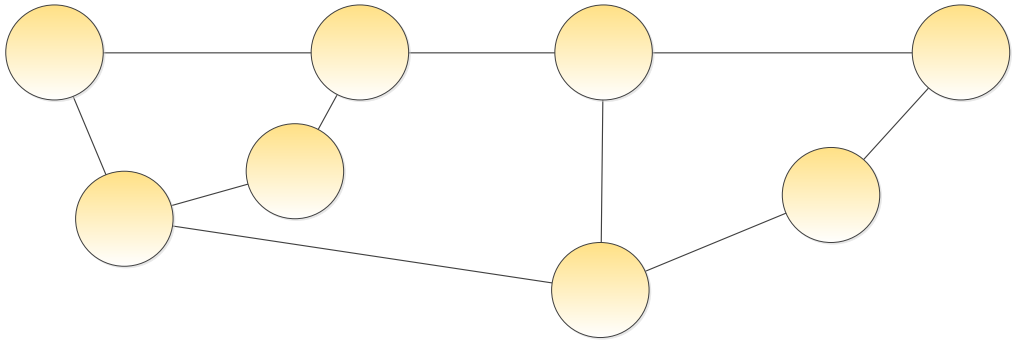


Figure 1.12 : Exemple de graphe bi-connexe.

K-clique

Une K-clique est une clique de K sommets. Chaque sommet possédant un degré de valeur *k-1*, le nombre de liaisons est donc de $k * (k-1)/2$.

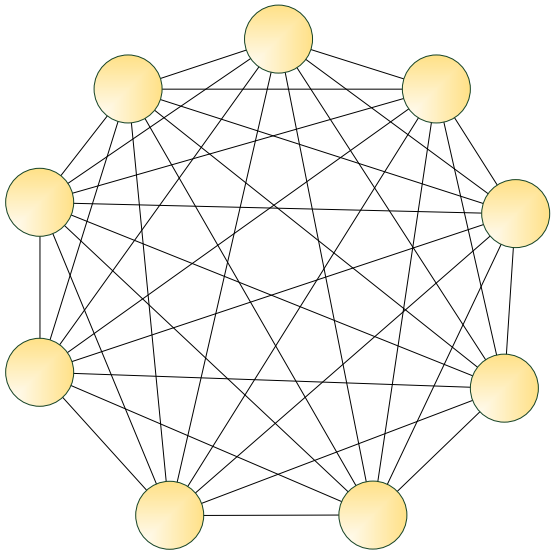


Figure 1.13 : Exemple de K-clique avec K= 9. Le graphe est constitué de 9 nœuds et 36 liaisons.

Liaison

Autre nom donné à une arête.

Méga-graphe

Afin d'indiquer rapidement un ordre de grandeur du nombre de nœuds inclus dans un (grand) graphe, on peut parler de Méga-graphe pour les graphes contenant plus d'un million d'objets.

Modularité et module

La notion de modularité a été développée par Newman **[Newman-2004-2]**. Son but est de comparer la proportion relative de liaisons d'un sous-graphe avec le nombre de liens d'un sous-graphe de même taille construit aléatoirement en respectant la valeur statistique de présence de liaisons de l'ensemble du graphe.

La valeur de modularité d'un sous-graphe est comprise entre [-1,1]. La valeur est supérieure à 0 si le nombre de liens intra sous-groupe est supérieur au nombre de liens du sous-graphe aléatoire créé en respectant la proportionnalité de liaisons présente dans le graphe complet.

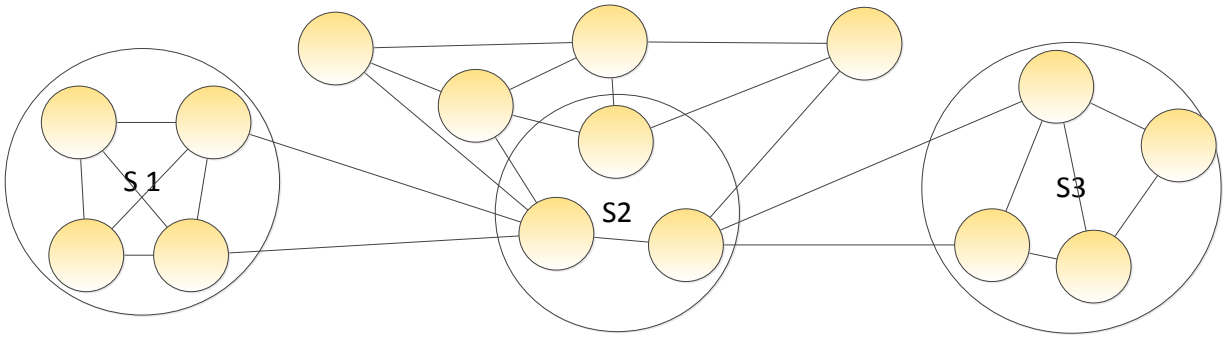


Figure 1.14 : Exemple de calcul de modularité.

Pour un sous-graphe S, le nombre de liens internes à S est noté L_S, le nombre de liens dans le graphe est noté L_G. La somme des degrés des nœuds présents dans le sous-graphe S est notée D_S. La modularité du sous-graphe S sera considérée comme valide si la modularité de S est supérieure à 0 soit : $M_S = (L_S / L_G) - (D_S / (2L_G))^2 > 0$

Dans l'exemple de la figure 1.14, nous sélectionnons arbitrairement 3 sous-graphes. Sachant que L_G=26,

Pour *S1*, L_{S1}= 6, D_{S1} =14 on a donc M_{S1}= 6/26 – $(14/52)^2$ = + 0.158

Pour *S2*, L_{S2}= 1, D_{S2} = 12 on a donc M_{S2}= 1/26 – $(12/52)^2$ = - 0.014

Pour *S3*, L_{S3}= 5, D_{S3} =12 on a donc M_{S3}= 5/26 – $(12/52)^2$ = + 0.139

Comme on peut le constater, les sous-graphes S1 et S3 sont bien des modules. S2 n'en est pas un, son score de modularité étant négatif. Ce critère de modularité est aujourd'hui très utilisé.

Partition

Une partition est un élément constitutif d'un graphe, structuré de telle sorte que chaque partition contient un nombre proche de nœuds et que l'ensemble des partitions contient tous les nœuds.

Taux de clustering ou d'agrégation

Le taux de clustering ou d'agrégation d'un graphe est la moyenne du rapport pour chaque nœud du nombre réel de liaisons existantes entre ses voisins et le nombre maximum théorique possible de ces liaisons.

Par exemple, pour un nœud *X*, il existe *K* nœuds avec lesquels il est connecté. Si le nombre de liaisons n'est pas limité pour chacun des nœuds, le nombre de liaisons maximales entre ces nœuds est *K(K-1) /2*. Le coefficient de clustering pour le nœud *X* dans un graphe non pondéré où le nombre de liaisons entre les nœuds voisins de *X* est égal à L_K sera alors de :

$$L_K \,/(\, K(K\text{-}1)\,/2$$

Le coefficient de clustering du graphe sera la moyenne de ces valeurs pour l'ensemble des nœuds *n* du graphe.

$$CC_G = \sum_{X=1}^{n} L_K \,/\, (\, K(K-1)\,/2)/n$$

Ce coefficient peut être vu comme une mesure statistique de la transitivité. Plus ce coefficient est élevé, plus la probabilité que les nœuds soient liés entre eux est forte. Autrement dit, pour rester dans un exemple des réseaux sociaux plus le taux de clustering est élevé plus il y de chance « que les amis de X soient amis entre eux ».

Triade

La triade, est une figure connectée de trois éléments comprenant trois sommets et trois liaisons. Les triades ont une importance particulière dans les réseaux sociaux, notamment car elles sont garantes de phénomènes impossibles aux diades, tels que la médiation et, de plus, sont porteuses de transitivité **[Faust-2010]**.

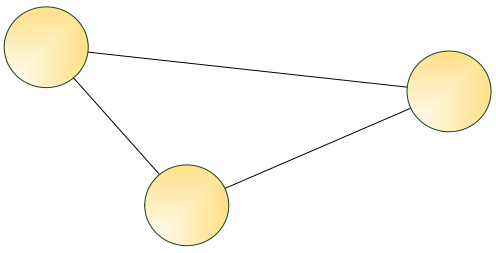


Figure 1.15 : Exemple de triade.

Voisins (nœuds et sommets voisins)

Les nœuds connectés au nœud X sont dits voisins de X.

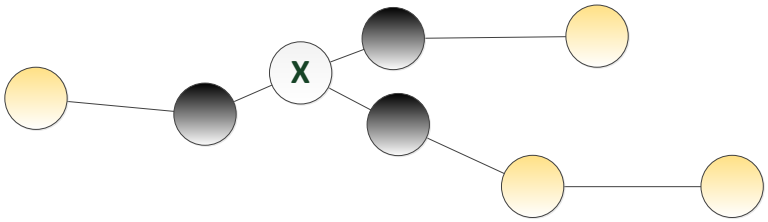


Figure 1.16 : En noir, les voisins du nœud X.

1.4 Grands graphes de terrain

La recherche sur les graphes est un domaine transversal. En effet, les graphes peuvent être constitués de nœuds représentant toutes sortes d'objets en relation. Cependant, il apparaît que la plupart des grands graphes de terrain partagent des caractéristiques communes.

1.4.1 Définition

Par définition, ces graphes venus du monde réel ne sont donc pas issus d'une formule mathématique. Ils existent sur le « terrain » et les nœuds se doivent d'avoir une existence physique.

Matthieu Latapy **[Latapy-2007]** considère la phrase de Watts et Strogatz, en 1998, **[Watts&al-1998]** *« la plupart des graphes de terrain ont des propriétés non-triviales en commun »* comme la consécration de leur domaine d'étude.

C'est à partir de ces propriétés communes que l'on définit les grands graphes de terrain.

Bruno Gaume qui nous apparaît comme l'inventeur de la formule « graphes de terrain » résume ces caractéristiques à quelques points essentiels **[Gaume-2004].**

Les graphes de terrain :

- Présentent un L faible où L représente la distance moyenne entre deux sommets. Autrement dit, on va généralement trouver un nombre de sommets faible dans le chemin d'un sommet à un autre.
- Présentent un C élevé où C est le coefficient de « clusterisation ». Ce qui signifie que deux sommets connectés à un troisième seront le plus souvent connectés entre eux créant ainsi une triade.

S'il est vrai que l'immense majorité des graphes de terrain répond à ces critères M. Latapy préfère plus prudemment définir ces objets par ce qu'ils ne « doivent pas posséder » **[Latapy-2007]**.

Selon ses travaux, les graphes de terrain ne doivent pas posséder :

- de structure apparente simple (comme des cliques ou des arbres) ;

- de structure comparable à des graphes aléatoires.

1.4.2 Caractéristiques

Certaines caractéristiques données dans la définition des « Grands Graphe de Terrain » ne sont précisées que par un ordre de grandeur ou une tendance. Il existe cependant un consensus autour de certaines de ces tendances. D'autres mériteraient d'être mieux précisées. Les exemples sont pris dans le domaine des réseaux sociaux dans le but de permettre au lecteur non spécialiste de mieux les visualiser.

Grands graphes

Les graphes de terrain que nous considérons ici sont dits « grands », notion toute relative. Historiquement la plupart des « grands graphes de terrain » étudiés sont constitués de quelques centaines à quelques centaines de milliers de sommets. L'étude récente de grands graphes de terrain, constitués de millions d'objets ou plus, pose de nouvelles difficultés. Ces difficultés se retrouvent dans la manipulation, l'étude et la visualisation de ces objets. C'est pourquoi, afin de marquer cette nouvelle étape, nous proposons de nommer les graphes constitués d'un à plusieurs millions de sommets « Méga-graphes de Terrain ». On pourra alors aller jusqu'à parler de « Giga-graphes » pour des objets d'études comme Internet **[Barabas&ali-2000]** vu comme un graphe de 10^9 sommets.

Une faible densité

Une faible densité correspond à une probabilité très faible que deux nœuds choisis aléatoirement soient directement connectés. Les valeurs rencontrées dans les graphes de cette étude (inférieur à .001) sont effectivement assez faibles. Il n'y a pas à notre connaissance d'ordre de grandeur fixé pour définir cette caractéristique comme « faible ».

Une composante connexe majoritaire

Cette composante est un ensemble de nœuds connectés présentant plus de 90% du nombre des sommets. De plus, cette composante connexe devra présenter un diamètre faible et donc fournir au graphe un diamètre effectif faible.

Une distance moyenne, un diamètre faible et un diamètre effectif faible

Il n'y a pas, à notre connaissance, d'ordre de grandeur fixé ni de seuil pour définir ces caractéristiques comme « faibles ». Cependant, d'une manière générale leurs évaluations relatives ne posent pas de problème. Par exemple, les diamètres effectifs et diamètres de Méga-graphes de terrain mesurés ici sont tous inférieurs à 15. Cette valeur comparée à un maximum théorique pouvant approcher le nombre de nœuds du graphe (supérieur à 10^6) est sans aucun doute faible.

Une distribution de degrés très hétérogène.

Cette caractéristique possède le plus souvent un écart type très important. Cependant, la nature des liaisons fournit parfois ses limites naturelles propres. Par exemple, dans un réseau social où la relation serait « est l'enfant de… », on comprend aisément que chaque nœud ne possédera que deux liens entrants et que le nombre de liens sortants « est le parent de … » ne peut pas atteindre de valeur très importante.

M. Latapy donne cependant comme système d'approximation de cette valeur une loi de puissance, telle que :

Pk = fraction des nœuds de degré k ;

k = degré, A est l'exposant de la loi : $Pk \sim K^{-A}$ où il estime A étant généralement entre 2 et 3.

La fraction de noeuds de degré k est proportionnelle (quand k varie) à une puissance négative de k. Le fait de suivre une telle loi (loi de puissance) est une marque de l'hétérogénéité. La constante A donne une indication de la force de cette hétérogénéité **[Latapy-2007].**

Un coefficient de clustering élevé

Cette caractéristique n'est pas citée par tous les auteurs comme déterminante des grands graphes de terrain. Elle est liée à la nature du graphe. Dans les réseaux sociaux, par exemple, il semble naturel que mes amis soit davantage amis entre eux que deux personnes choisies aléatoirement.

1.4.3 Contexte

Les mathématiciens et théoriciens travaillent la plupart du temps sur des graphes générés aléatoirement. Les graphes de terrain sont utilisés pour des modélisations d'espaces réels et davantage étudiés par des spécialistes du domaine. La théorie des graphes est ainsi utilisée dans de nombreuses disciplines, comme la biologie, la chimie, les réseaux d'ordinateurs, l'épidémiologie et la sociologie. Les linguistes utilisent aussi les graphes pour représenter les relations entre « termes » (relations sémantiques, proximités d'usage …)

Les contextes où l'on rencontre des graphes de terrain ne doivent pas être réduits à cette liste ; elle n'est en rien exhaustive : des bancs de dauphins aux interactions entre sites Web, les espaces où s'exerce l'étude de ces réseaux est en constante augmentation.

Nous proposons un panorama des travaux de recherche associés aux domaines cités précédemment :

- En biologie, les réseaux représentent des éléments du vivant comme par exemple les protéines. Les travaux de Mashaghi concluent à la classification en « petit monde » (cf. paragraphe 1.1.4) d'un réseau de protéines **[Mashaghi&al-2004]**. Il nous faut aussi citer les travaux de Palla et al. sur la clusterisation avec recouvrements de graphes. Nous reparlerons dans la section 2.3.1 de ces travaux qui sont eux aussi appliqués aux réseaux de protéines **[Palla&al-2005]**.

- En chimie, les éléments sont souvent des molécules ou des atomes. Nous pouvons citer le travail de Francesco Rao et al. qui, dans *« Structural Inhomogeneity of Water by Complex Network Analysis »*, appliquent les méthodes d'analyse des grands graphes de terrain à un ensemble de molécules d'eau à température ambiante **[Rao&al-2010**], afin de mieux comprendre la structure moléculaire de l'eau et les changements de niveau d'entropie.

- En informatique et en électronique, les exemples foisonnent. Pour n'en citer que quelques-uns, nous évoquerons, dans le domaine des réseaux d'ordinateurs, les travaux de Matthieu Latapy et de son équipe sur les réseaux points à points **[Aidouni-2009-1]** ainsi que ceux sur la topologie d'Internet **[Aidouni-2008]**. Citons également les travaux d'Estrada et al. qui portent sur la recherche de communautés de pixels à l'intérieur d'images **[Estrada&al-2010]** et les travaux de Hagen et al. sur la conception de circuits imprimés multicouches visant à limiter les connexions inter-couches **[Hagen&al-1992]**.

- Dans le domaine de l'épidémiologie, les réseaux représentent les relations entre des objets du monde du vivant. En cela, ces réseaux se rapprochent de ceux étudiés en biologie. Mais la finalité de l'étude est différente. Le but de ces recherches est la détection des mécanismes de transmission d'agents pathogènes. On peut mentionner les travaux de Romualdo Pastor-Satorras et Alessandro Vespignani sur la relation entre les structures des réseaux et la propagation des maladies **[Pastor-Satorras&al-2001]**. Faisant un parallèle entre virus informatiques et pathologies contagieuses humaines, les auteurs arrivent, par une modélisation informatique, à la conclusion qu'une infection peut se développer quelque soit son ou ses points de départ et son niveau de contagion, avec un risque de pandémie toujours faible.

- En sociologie, les réseaux sont constitués d'êtres humains. La première réflexion ayant mis en œuvre les grands graphes de terrain est sans doute celle menée par Travers, Jeffrey et Stanley Milgram, mieux connue sous le nom de l'expérience des « petits mondes » ou « des six poignées de main » **[Millgram&al-1969]**. Nous reparlerons en détail de cette expérience initiatrice et de ses limites dans le paragraphe suivant.

1.4.4 Des petits mondes ou la légende des six poignées de mains

En 1969, Travers Jeffrey et Stanley Milgram effectuent une expérience qui donnera à penser que « notre monde est tout petit ». L'expérience a donné naissance au lieu commun « nous sommes tous à une distance de six poignées de main de quiconque sur la planète ». Il convient de rappeler les faits : Millgram donne un paquet à 296 personnes avec la consigne de de l'envoyer à une personne qu'elles « connaissent », la définition de cette connaissance étant « vous l'appelez par son prénom » ; chaque destinataire devant à son tour envoyer le paquet à un destinataire « connu » d'eux et ainsi de suite jusqu'à ce que le paquet parvienne au destinataire final commun, un agent de change, vivant à une adresse fournie, dans la ville de Sharon dans le Massachusetts. Les personnes qui « connaissaient » le destinataire final

pouvaient le lui envoyer directement. Sur 296 paquets, 64 sont arrivés à destination. Pour ces 64 paquets, la moyenne du nombre de personnes ayant porté le paquet jusqu'à destination était alors de 5.5. La conclusion aurait pu être : dans la même composante connexe, la distance moyenne entre deux individus semble être de 5.5 sauts.

Depuis cette expérience **[Millgram&al-1969],** les grands graphes de terrain présentant un diamètre faible sont nommés « Small Word » ou *petit monde* **[Watts-1999]**. Le concept de petit monde est donc lié à la notion de diamètre faible en regard du nombre de nœuds. Cette caractéristique est accompagnée d'une faible distance entre deux sommets quelconques et une forte connectivité locale. Elle provient initialement de la célèbre expérience de Stanley Millgram. La notion de petit monde est peut-être réelle, mais construite en partie sur ce qu'il est convenu d'appeler une « légende urbaine ».

Toutefois, le coût en temps CPU d'un calcul exact du diamètre est prohibitif lorsqu'on utilise les algorithmes classiques sur des réseaux de plusieurs millions de nœuds. Nous pouvons citer ici la méthode proposée par C. Magnien **[Magnien-2009]** qui effectue une estimation du diamètre par un encadrement entre deux valeurs : une valeur haute obtenue par une simplification préalable du graphe en arbre et une valeur basse correspondant à la distance maximale parcourable depuis des nœuds sélectionnés aléatoirement. Nous avons aussi proposé une autre piste d'estimation basée sur la mesure de la distance maximale mesurable en prenant comme nœuds de départ des nœuds prédéterminés ayant une forte potentialité à être une des extrémités du diamètre du graphe **[Belbeze&al-2012]**.

1.5 Les communautés

1.5.1 Définition et choix de la terminologie : clusters, communautés ou agrégats ?

Le terme « communauté » provient d'une analogie avec les réseaux sociaux, un des champs d'étude au cœur duquel les graphes sont très présents.

La communauté au sein d'un graphe est définie par Santo Fortunato **[Fortunato-2010]** comme un *« ensemble autonome »*. Ceci est l'expression d'un nombre de nœuds connectés et regroupés (en communauté) de telle sorte que le nombre de liens intracommunautaires soit le plus élevé possible et le nombre de liens extracommunautaires soit le plus faible possible.

La définition de la communauté induite devient alors : *« Une communauté forte est telle que le degré de chaque nœud interne est supérieur à son degré externe »*. La représentation de la figure 1.17 illustre cette définition.

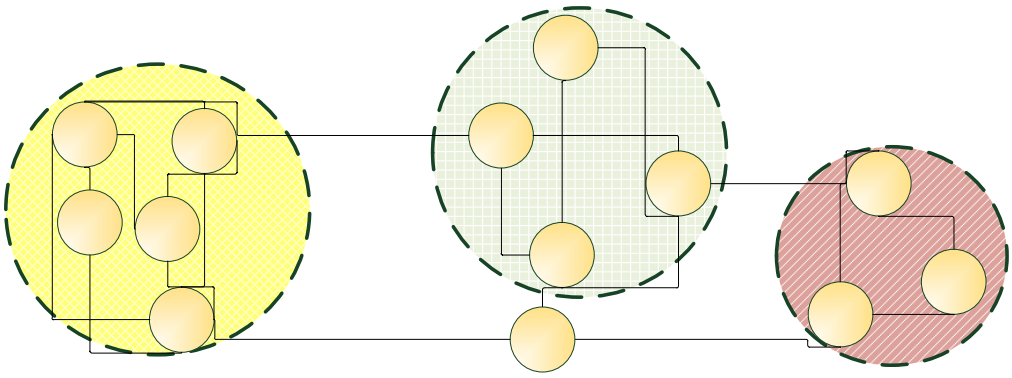


Figure 1.17 : Exemple de graphe où l'on peut intuitivement détecter trois communautés telles que définies par Santo Fortunato.

Pourtant, il n'existe pas de définition couramment admise de ces ensembles. Le regroupement intuitif des nœuds selon leurs connectivités, bien qu'ayant un grand succès, n'a pas démontré son universalité. Fortunato lui-même, déclare que *« le premier problème dans la clusterisation de graphe est la recherche d'une définition des caractéristiques quantitatives d'une communauté »* **[Fortunato-2010]**.

Dans ce contexte, le terme même de « communauté » peut sans doute être considéré comme abusif. En effet, hérité des réseaux sociaux, il sous-tend un lien sémantique d'appartenance à une unité et le partage d'éléments identitaires communs. Par exemple, parlerions-nous facilement de communautés d'ordinateurs pour nommer un ensemble de postes de travail sur un LAN au sein d'un WAN ?

La terminologie anglaise parfois utilisée, qui est celle de « Cluster », possède, elle, des définitions précises et contextuelles. Cependant, elle se réfère autant à la structure inter-sommets, qu'aux sommets eux-mêmes. Considérer l'ensemble à étudier (le cluster) comme un ensemble de sommets participant d'une entité ayant sa propre identité, nous semble abusif. C'est, pour utiliser une métaphore courante, comme si pour définir un être humain on nommait un ensemble d'organes liés par un réseau sanguin par le nom de ce réseau sanguin. De plus, dans les réseaux sociaux, une fois la communauté découverte, la structure porteuse n'a plus d'« usage ». Ainsi, un groupe d'amis est indépendant de la relation ayant servi à les repérer (SMS, emails, connexions téléphoniques ou autres).

De plus, si la définition des clusters ou des communautés retient comme caractéristique majeure la proximité la plus importante possible en interne et la plus faible possible en inter-communautés, l'appartenance d'objets à plusieurs communautés devient alors une source naturelle de baisse de la qualité. La communauté devant, par définition, être le moins en interaction avec l'extérieur, la dénomination « communauté avec recouvrements » devient un oxymore.

Il en est de même pour la terminologie de « super-communauté ». Cette terminologie, utilisée pour signifier des communautés de taille importante, associe alors le préfixe « super » à un objet dont la qualité peut être à priori jugé faible. La taille très importante d'une

communauté est le plus souvent l'expression d'une incapacité à déterminer des ensembles plus pertinents.

On peut aussi aisément concevoir que pour nommer les ensembles de mots destinés à servir de moteurs à des communautés dynamiques d'utilisateurs, le choix du terme « communauté » ne soit pas judicieux. Il convient d'utiliser un lexique différent pour nommer d'une part, les communautés sociales, de l'autre, les ensembles de mots.

On peut enfin remarquer que Gregory Palla, au fil de ses articles, a remplacé le mot « community » **[Palla&al-2005]** par « module » **[Palla&al-2007]**. Le terme de « module » ayant déjà en mathématiques (module de nombre complexe, vecteur) et en informatique (bloc de code) des définitions précises et différentes, il ne semble pas approprié.

Toutes ces raisons nous encouragent à l'utilisation d'un autre terme : celui d'**agrégat**. Bien qu'il soit rarement utilisé **[Botafogo&al-1991] [Cucala-2009], ce terme** semble pourtant le plus adapté.

Un agrégat est défini comme une *« réunion d'éléments matériels juxtaposés, généralement hétérogènes, présentant entre eux une certaine cohésion et formant un tout »* (Larousse 2001). Tant que la nature du « tout » n'est pas caractérisée comme ayant une identité propre, il ne nous semble pas judicieux d'employer d'autres termes pour nommer ce regroupement. Ainsi, tout travail de regroupement va-t-il créer un agrégat qui est éventuellement une communauté. Un agrégat est défini par Bayaly et Cunny, comme *« un ensemble de nœuds liés logiquement dans un graphe »* **[Bailey&al-1986].**

Pour conclure cette tentative de définition, je citerai Filippo Radicchi et al **[Radicchi&al-2004]** : *« Cependant, pour analyser un réseau, il est nécessaire de préciser quantitativement et sans ambiguïté ce qu'est une communauté. … Une communauté peut être vue comme un ensemble d'éléments qui répondent à certaines règles »*. Ainsi, par exemple, la communauté des sommets présentant les degrés les plus élevés devient possible. De telles communautés seraient alors à l'opposé des définitions données par Santo Fortunato **[Fortunato-2010]**.

Il nous faudra définir les règles de la communauté. Une fois l'agrégat validé comme respectant les règles caractérisant « notre définition » d'une communauté, il pourra être éventuellement nommé troupeau, ban, équipe, club, sous-réseau ou même communauté en fonction de sa nature et de la nature des objets regroupés. Cependant, pour respecter les terminologies habituelles et la cohérence avec certains travaux, nous continuerons à nommer « communautés » un ensemble de nœuds identifié comme groupe constitué dans l'état de l'art de ce travail. Nous réserverons le terme d'**agrégat** à la deuxième partie de ce travail.

La création de communautés dans des graphes est un sujet qui est de plus en plus abordé. Selon notre étude et notre approche, identifier les communautés dans les grands graphes revient à partitionner les grands graphes en sous-graphes et à se poser la question suivante : recouvrement ou non recouvrement ?

1. **Les communautés sans recouvrement**

Dans les communautés, les nœuds appartiennent au plus à une seule communauté. Ce sont celles-ci qui sont majoritairement étudiées. La figure 1.17 présente un exemple de découpage en communautés sans recouvrement.

Dans la liste des travaux présentés au paragraphe 3.4.3, les travaux Hagen et al. sont parmi les plus concrets **[Hagen&al-1992]**. Ils utilisent des algorithmes de partitionnement de graphes de façon à optimiser le regroupement des liaisons entre composants sur une même couche du circuit imprimé, afin de limiter le nombre de liaisons inter-couches, liaisons qui sont à la fois chères et sources de défaillance.

2. **Les communautés avec recouvrement**

Dans les communautés avec recouvrement, les nœuds peuvent appartenir à un nombre indéterminé de communautés. Bien que représentant souvent des découpages plus proches de la vie réelle, elles sont peu étudiées (cf. figure 1.18). Une des raisons en est la difficulté de validation et le caractère flou que peut présenter l'affectation d'un nœud à plusieurs communautés si celle-ci est pondérée ou relative.

Dans la liste des travaux présentés au paragraphe 3.4.3, ceux de Palla et al. sont les plus célèbres. Ils portent sur la création de communautés dans le domaine de la biologie mais aussi dans celui des réseaux sociaux **[Palla&al-2005]**.

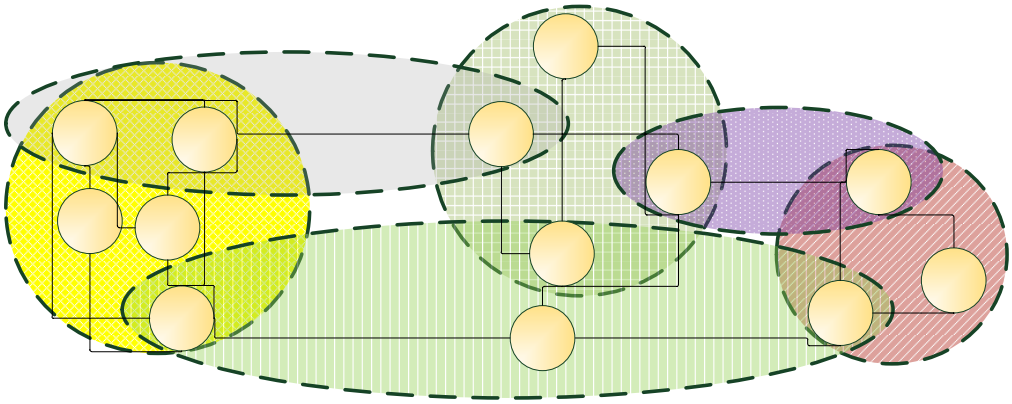


Figure 1.18 : Un exemple de graphe où l'on peut observer six communautés avec recouvrement.

1.5.2 Recherche et détection de communautés dans les graphes

Dans la littérature scientifique, les processus de recherche au sein d'un graphe pouvant créer un ensemble de nœuds cohérent sont habituellement nommés « clusterisation de graphe » ou « Détection de communautés ».

Obtenir la capacité à déterminer des communautés valides dans des grands graphes de terrain peut apporter des changements majeurs dans notre quotidien. Ainsi, la création de groupes d'utilisateurs ayant les mêmes centres d'intérêt permet d'améliorer le type de recommandations sur certains achats de produits comme chez « amazon.com ». On peut aussi

se servir de ces outils pour repérer des communautés spécifiques telles que les communautés pédophiles **[Belbeze&al-2009-1].** Il est également possible d'étudier ces communautés pour suivre leurs évolutions et les nouveaux comportements de leurs membres ; par exemple, par la découverte de nouveaux mots utilisés par une communauté identifiée **[Belbeze&al-2009-2]**. La détection de communautés peut aussi être un élément d'étude en épidémiologie pour mieux comprendre les processus de propagation **[Britton&al-2008]**. Les regroupements sont aussi manipulés de façon à rendre les grands graphes lisibles. En représentant uniquement les communautés des graphes de plusieurs milliers de points, ceux-ci peuvent alors devenir beaucoup plus lisibles **[Villa&al-2009]**. Dans les réseaux de mots, les communautés peuvent aussi être employées pour la représentation d'espaces ontologiques **[MikaVrije-2005]**.

1.6 Conclusion

La théorie des graphes est, du point de vue des mathématiques, un objet d'étude relativement ancien. Ce n'est que très récemment qu'elle a été mise à profit dans le cadre d'étude d'objets non génériques. Les experts des réseaux sociaux l'ont ainsi utilisée à partir de la deuxième moitié du XXème siècle seulement.

1.6.1 Vocabulaire et terminologie

Le vocabulaire très marqué par les recherches sur les réseaux sociaux est sans doute appelé à évoluer. Les termes doivent gagner en précision ou retourner à un contexte qui est le leur :

- le terme de communauté ne devrait être utilisé qu'avec précaution en dehors des réseaux sociaux ;
- les notions de petit monde et de « Grand Graphe de Terrain » doivent être plus nettement dissociées qu'elles ne le sont. Un graphe de terrain n'est jamais qu'un réseau existant dans la vie réelle, alors que le petit monde reste, lui, à définir plus précisément ;
- nous proposons l'utilisation du terme d'agrégat en remplacement de cluster ou de communauté tant que la nature du groupe n'a pas été définie.

1.6.2 Caractéristiques et valeurs

Les caractéristiques des petits mondes et des grands graphes doivent être précisées et définies. Un ordre de grandeur ou des seuils pour chacune des valeurs de ces caractéristiques serait des plus utiles de façon à pouvoir effectuer des classifications des différents graphes.

La notion de grand graphe recouvre aujourd'hui des objets de tailles extrêmement différentes. Afin de classer les réseaux par taille et de préciser la notion de grande taille, nous proposons d'utiliser le vocable de kilo-graphe pour les réseaux de milliers de points, de méga-graphe pour ceux constitués de millions de points et de giga-graphe pour ceux dépassant le milliard.

La notion de petit monde, qui est pourtant communément admise, souffre de l'imprécision de l'ordre de grandeur de ses caractéristiques. Par exemple, ne pourrions-nous pas définir une taxinomie ? Celle-ci serait composée de grands, moyens et petits mondes où les valeurs de diamètre, distance moyenne, écart type des degrés, poids de la composante connexe principale etc. se verraient proposer en comparaison avec des valeurs relevées sur des graphes de même taille obtenus aléatoirement ou existant dans le monde réel.

L'étude des graphes de terrain, à l'origine centrée sur les réseaux sociaux, doit aujourd'hui trouver ses propres règles et sa propre terminologie autour d'objets génériques pouvant composer les réseaux.

Les éléments et notions abordés dans ce premier chapitre permettront de mieux comprendre les enjeux et fonctionnements des algorithmes d'extraction de communautés présentés dans le chapitre suivant.

Chapitre 2.

Les algorithmes de création de communautés

2.1 Introduction

Un réseau ou un graphe est souvent constitué, dans le monde des grands graphes de terrain, d'une seule composante connexe ou d'une composante représentant plus de 90% du graphe. Son unité, sa taille et des zones de forte densité de liens suggèrent de le décomposer en ensembles de plus petites tailles. Les méthodes permettant de créer des sous-ensembles de nœuds à l'intérieur d'un graphe sont de plus en plus nombreuses.

Ces méthodes ont deux objectifs possibles: soit créer des ensembles disjoints soit, créer des ensembles avec recouvrements. Les approches purement mathématiques ont le plus souvent comme objet la création de communautés sans recouvrement. Au contraire, les approches davantage orientées « terrain » introduisent des méthodes plus souples.

Les communautés avec recouvrement font infiniment moins couler d'encre que celles qui n'en ont pas. Nommées parfois « overlapping communities » ou « fuzzy clusters », elles peuvent être regardées avec un certain dédain. Schaeffer déclare : *« Fuzzy clustering has not been established as a widely accepted approach for graph clustering … »* **[Schaffer-2007]**. La raison tient peut-être à la difficulté d'en définir les règles de création et de validation, position compréhensible eu égard à l'oxymore que suggère l'idée de communauté avec recouvrements (cf. 1.5.1). En effet, on peut tout à fait considérer que le cluster - ou la communauté de nœuds génériques - existe bien d'un point de vue mathématique. Il suffit pour cela de définir mathématiquement la notion de cluster ou de communauté. Par contre, dans le monde réel, le nœud perd son caractère mathématique. Le groupe de nœuds devient famille, groupe d'amis, troupeau, espace sémantique etc. L'étude des grands graphes de terrain appelle donc une autre approche et une autre terminologie. Les règles ne peuvent plus être générales. Elles doivent être adaptées aux caractéristiques du groupe à former. Par exemple, un ordinateur est en général dans un seul Lan, mais un individu appartient à plusieurs groupes d'amis. De plus, la définition de la communauté est le plus souvent implicite. Il est alors impossible de modéliser de manière générique un regroupement dont les caractéristiques dépendent de la nature des objets manipulés.

Pourtant, dans le monde réel, il est souvent impossible de placer des limites entre des communautés. Il est évident qu'un usager n'appartient pas qu'à une seule communauté d'usage de téléphones portables ou d'échange de méls. Il en est de même pour les pathologies humaines ou les catégories socioprofessionnelles. Un article ou un livre peut évoquer plusieurs sujets. Un sportif peut posséder plusieurs licences dans plusieurs clubs de sports différents. Un paysan peut travailler sur plusieurs fermes. Et enfin, un mot peut appartenir à plusieurs espaces sémantiques : il peut avoir plusieurs sens dans une même langue, mais aussi des sens différents dans des langues différentes. Le mot « car » est, par exemple, une conjonction de coordination en français et signifie « automobile » en langue anglaise. Notre travail portant sur le regroupement des mots, c'est donc bien sûr sur ces dernières méthodes permettant le recouvrement que notre attention va se porter en priorité.

Dans ce chapitre, nous nous questionnerons aussi sur les méthodes de validation des communautés avec recouvrement et sur l'opportunité d'une classification qualitative des méthodes selon le niveau de complexité de leur algorithme.

2.2 Les partitions ou communautés sans recouvrement

Face à un ensemble complexe de taille importante, il n'existe finalement que deux attitudes : soit l'ignorer, soit, chercher à l'ordonner en le décomposant en entités plus réduites. La création de communautés ou de sous-parties distinctes dans un graphe tient à un besoin d'ordre. Que ce soit pour effectuer des taxinomies ou simplement obtenir des sous-ensembles plus clairs, plus maniables ou plus faciles à étudier, nous voulons en premier lieu classer. Les méthodes sans recouvrement, sans « flou », sont aptes à accompagner ce désir d'ordre.

Il existe un grand nombre d'approches permettant de créer des communautés sans recouvrement. Ces différentes approches pour développer des communautés distinctes doivent être aussi considérées comme des bases pour développer de nouvelles méthodes permettant, elles, de créer des communautés avec recouvrement. Les communautés sans recouvrement peuvent aussi être utilisées en tant qu'éléments dans des processus plus complexes. Par exemple, en ajoutant à des communautés disjointes des espaces qui, potentiellement, sont partagées entre plusieurs communautés.

Ces méthodes n'étant pas au centre de notre travail, nous ne présentons ici que succinctement les méthodes les plus utilisées ou présentant une base potentielle pour autoriser le recouvrement. Les algorithmes des méthodes sont regroupés en trois grands types :

- Les algorithmes séparatistes ;
- Les algorithmes de scission ;
- Les algorithmes de recherche de zones de forte modularité.

Les deux éléments de validation de ces communautés sont les suivants :

- La cohérence interne sur le critère de regroupement doit être la plus élevée possible ;
- La cohérence externe sur le critère de regroupement doit être la plus faible possible.

Autrement dit, les communautés doivent apparaître comme les plus homogènes possibles et posséder une grande distance entre elles.

2.2.1 Les algorithmes séparatistes

Le partitionnement de graphes

Le partitionnement de graphes est sans doute le moyen le plus ancien d'effectuer un découpage dans un graphe. Le fait que l'on prédétermine avant tout le nombre d'ensembles à créer peut faire douter qu'il s'agisse vraiment de regroupement. En effet, la première règle (nombre de communautés dans le graphe) n'est pas une règle liée aux caractéristiques des communautés à créer mais aux caractéristiques du graphe lui-même. Le plus célèbre des algorithmes de partitionnement est le « min cut » de Kernighan et Lin **[Kernigha&al-1970] (cf. figure 2.1)**.

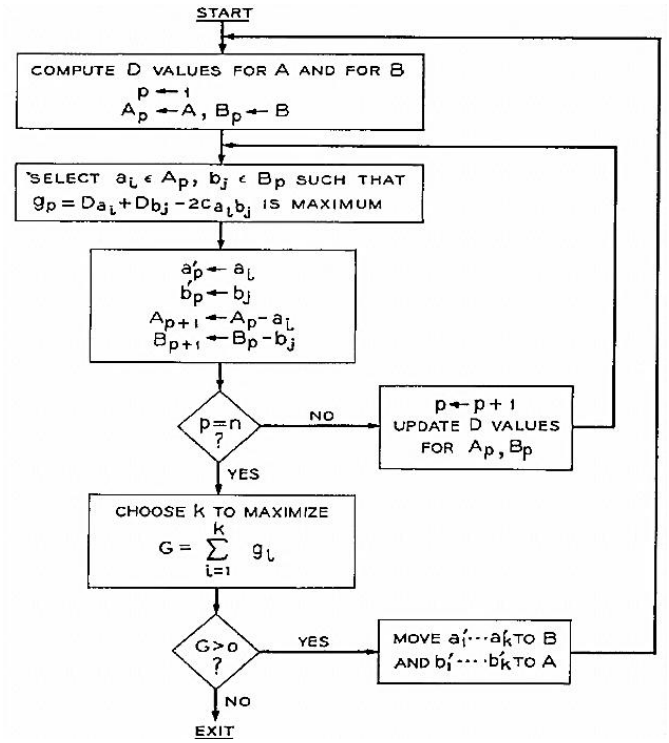


Figure 2.1 : Algorithme récursif tel que présenté en 1970 par Kernighan et Lin pour le partitionnement de graphe [Kernigha&al-1970].

Parfois aussi appelé algorithme de « migration de groupe », il calcule itérativement, pour chaque nœud et chaque déplacement entre communautés, le coût du déplacement. Ce coût est généralement le rapport entre le nombre de liaisons inter-communautés créées et le nombre de liaisons intra-communautés créées. Puis, il identifie le nœud qui produit la plus grande baisse ou la plus petite augmentation dans ce coût. Ensuite, il fait l'échange et répète

le processus en utilisant la nouvelle partition comme partition initiale jusqu'à ne plus trouver de partition de coût inférieur.

La figure 2.1 décrit cet algorithme où *G(V,E)* est un graphe constitué de l'ensemble *V* des nœuds du graphe et de *E* l'ensemble des paires d'éléments de *V* représentant les liaisons. L'algorithme démarre à partir de *X* communautés de taille équivalente qui scindent *G* en *X* partitions. Ces communautés sont générées aléatoirement et précédemment à la partie de l'algorithme présenté.

L'algorithme va optimiser deux communautés *A* et *B* parmi *X* entre elles. Dans une première phase, l'algorithme copie les deux communautés *A* et *B* dans deux communautés de travail A_p et B_p.

Nous définirons un nœud « *a* » présent dans la communauté A_p, I_a comme la somme des liens entre le nœud *a* et les autres nœuds de communautés A_p, E_a comme la somme des liens entre le nœud *a* et les autres nœuds de communautés B_p.

$D_a = E_a - I_a$ ou D_a est la différence pour le nœud « *a* » entre les liens externes et internes

Supposons maintenant un nœud « *b* » appartenant à la communauté B_p, en cas de permutation *p* des nœuds *a* et *b* le gain *G* de la permutation *p* sera de $G_p = D_a - D_b - 2C_{a,b}$, où $C_{a,b}$ est le coût possible de la relation entre les nœuds *a* et *b*.

Dans une première boucle l'algorithme va tester l'ensemble des permutations possibles en recherchant la permutation donnant le plus grand gain G_p. Cette permutation est effectuée et ainsi de suite tant qu'il existe des permutations ayant un gain positif.

Enfin, dans une dernière phase, si le coût de l'ensemble des liaisons dans les nouvelles communautés A_p et B_p est supérieur à celui des communautés *A* et *B*, celles-ci remplacent les communautés de *A* et *B*. Il est ensuite possible de relancer l'algorithme en comparant deux autres communautés de X, jusqu'à ce qu'il n'y ait plus d'amélioration possible.

Le partitionnement de données

Contrairement au partitionnement de graphe, le partitionnement de données est essentiellement utilisé sur des graphes pondérés. La pondération étant alors utilisée comme une valeur distanciatrice entre les sommets. Les algorithmes de partitionnement de données ou « data clustering » vont ensuite regrouper les nœuds en fonction de leur proximité.

Une des méthodes consiste à utiliser ces « distances » comme un système de localisation dans un espace Euclidien de *n* dimensions relativement les unes par rapport aux autres et de regrouper ensuite les sommets par zones pour créer des communautés de « voisinage » **[Jain&al-1999]**.

Communautés créées à partir d'une vision hiérarchique du graphe

Créer des communautés à partir de la vision hiérarchique d'un graphe consiste à transformer le graphe en un dendrogramme (arbre représenté par une succession de fourches). Pour cela, on considère au départ chaque sommet de l'arbre comme une communauté. On

lance ensuite un algorithme qui va rechercher pour chaque communauté, celle qui lui sera la plus proche. Ces deux communautés seront alors intégrées pour en créer une nouvelle. On traite ainsi toutes les communautés de même rang puis on recommence jusqu'à n'avoir qu'une communauté.

On relie ensuite les communautés de départ entre elles. La règle de rapprochement est variable. Elle peut être basée sur la présence de liens communs entre les sommets, ou sur le nombre de liens entre tous les sommets d'une communauté vers une autre. On peut aussi utiliser la notion de centralité (centroïd) et définir une valeur des liens en fonction de la distance au centre, comme dans la méthode de Ward **[Ward-1963]**. Dans le cas d'un graphe pondéré, on peut, bien sûr, faire intervenir les valeurs de pondération pour choisir les communautés à rapprocher.

Il suffira ensuite de choisir un niveau de séparation pour obtenir plus ou moins de communautés. Ce nombre pouvant alors aller de une (obtenue par l'agglomération de toutes les communautés de départ), au nombre de sommets de l'arbre constituant les communautés de départ.

Dans les communautés créées à partir d'une vision hiérarchique du graphe, celles fondées sur la marche aléatoire sont particulièrement pertinentes et vont constituer la base de nombreuses autres méthodes. Ces méthodes considèrent qu'un promeneur va potentiellement de nœuds en nœuds de façon aléatoire. En fonction de sa position il a donc toujours la possibilité d'emprunter une liaison pour se rendre sur un des nœuds voisins, il peut bien sûr faire aussi marche arrière. La méthode va donc permettre de définir des promenades plus ou moins probables et donc des ensembles de nœuds plus probables que d'autres. De ces ensembles vont naître les communautés **[Pons&al-2005] [Pons-2007]**. Deux nœuds sont alors dans la même communauté si, depuis leurs positions, un promeneur a une probabilité maximale de faire une promenade identique ou des promenades très proches. Le niveau de découpage sera ensuite choisi comme étant celui procurant la plus grande qualité. Le coefficient de qualité ou l'élément de vérification étant souvent basé sur les valeurs de modularité.

La méthode est remarquable à plus d'un titre : elle intègre une vision «agglomérative» nouvelle à la vision séparatiste de départ. En effet, la position de départ est locale et les « promenades » construisent la communauté en ajoutant les nœuds explorés. La démarche reste cependant séparatiste de par le fait que le nombre des communautés et donc des partitions est préalablement fixé.

2.2.2 Les algorithmes de scission

Les algorithmes de scission ont pour but de découper le graphe en deux, puis chaque nouvelle partie encore en deux jusqu'à ce qu'un nombre « satisfaisant » de communautés ait été créé. Pour cela, l'algorithme va retirer les liaisons les plus faibles ou jugées comme telles pour séparer l'objet de départ en composantes connexes distinctes ; chaque composante connexe étant alors une communauté et un nouveau point de départ.

Les différents algorithmes se différencient sur la manière de détecter les liaisons à supprimer pour créer la scission.

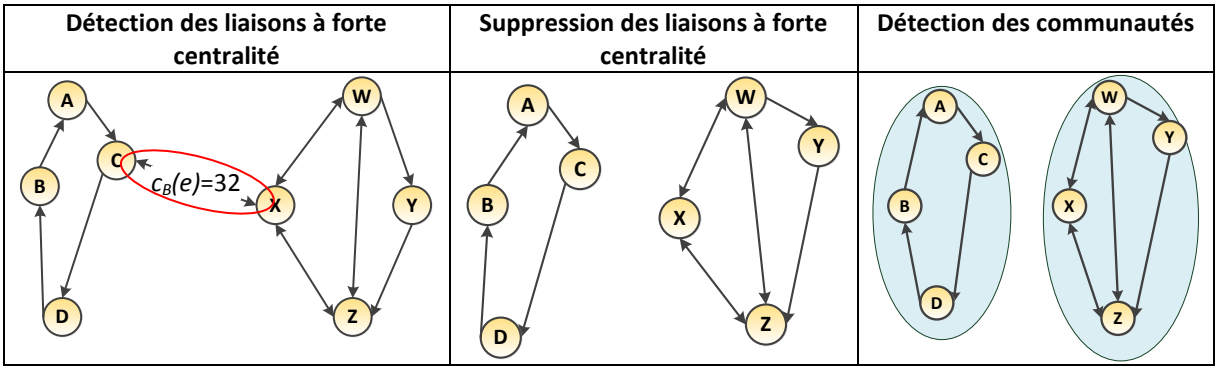


Figure 2.2 : Illustration de la méthode de M. E. J. Newman and M. Girvan pour créer des communautés basées sur des composantes connexes créées par le retrait de liaisons à forte centralité.

La méthode de Girvan et Newman illustrée figure 2.2, se fonde sur la recherche de liaisons où la centralité est la plus élevée (Liaisons où passe un maximum de chemins « le plus court » pour aller d'un nœud à un autre) **[Newman-2004-3]**.

2.2.3 Les algorithmes de recherche de zones de forte modularité

Les algorithmes de recherche de zones de forte modularité s'appuient sur la notion de modularité introduite par Newman en 2004 **[Newman-2004-2].** Ils cherchent, en se basant sur la définition d'une communauté en tant qu'élément dense du réseau, à déterminer des zones prédisposées à devenir des communautés. Pour un découpage en communautés données par un algorithme, la modularité est la différence entre la part d'arêtes intra-communautaires et la même valeur pour une répartition aléatoire des arêtes pour le graphe complet étudié. Avec une variation comprise entre -1 et 1, plus la modularité est élevée plus les communautés sont de qualité.

Finalement assez proche de la définition de la communauté « forte » donnée par Fortunato *« Une communauté forte est telle que le degré de chaque nœud interne est supérieur à son degré externe »* **[Fortunato-2010]**, les algorithmes utilisant la recherche d'une forte modularité pour définir des communautés sont très nombreux. Nous pouvons donner, à titre d'exemple, le travail de Fabrice Rossi et Nathalie Villa-Vialaneix sur l'application de ce principe sur les réseaux topologiques **[Rossia&al-2010]**. On peut aussi noter que cette démarche, est devenue une démarche de référence. En effet, on la retrouve dans bien des logiciels permettant de travailler sur les graphes. Le logiciel **R** (http://www.r-project.org/) présente dans sa librairie « **Igraph** » (http://igraph.sourceforge.net) une fonction « modularity » permettant de retourner la capacité d'un sous-graphe à devenir une communauté de « qualité ».

En 2008, Newman a proposé une version de son algorithme capable de gérer les graphes pondérés **[Newman&al-2008]**.

2.3 Les différentes méthodes de recherche de communautés avec recouvrement

On peut considérer que les premières études théoriques et fondatrices de ces méthodes ont été faites en 1965 par Lotfi Zadeh **[Zadeh-1965]**. Ce mathématicien a posé les bases d'un système de classification où les objets peuvent appartenir à plusieurs ensembles qu'il nomme « ensembles flous » ou « fuzzy sets ». Chaque élément possède un tableau d'appartenance où le degré d'appartenance à chaque ensemble flou est indiqué par une valeur entre 0 et 1. D'après ce scientifique : « *Un contrôleur électromécanique doté d'un raisonnement humain serait plus performant qu'un contrôleur classique* ».

Bien que ces travaux ne portent pas explicitement sur des graphes et l'appartenance de nœuds à des communautés, ils abordent la problématique de la complexité d'une multi-appartenance pondérée d'objets et marquent aussi la volonté de rechercher une modélisation du monde réel.

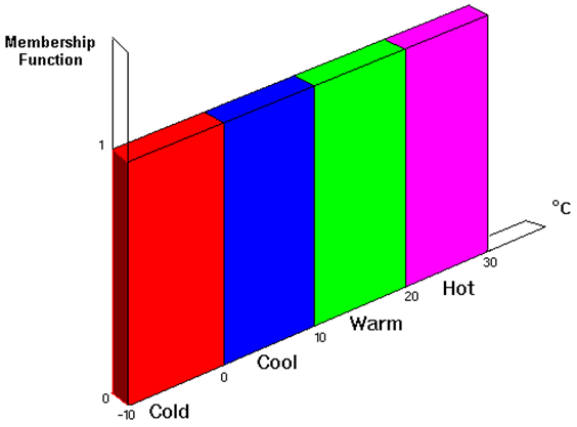
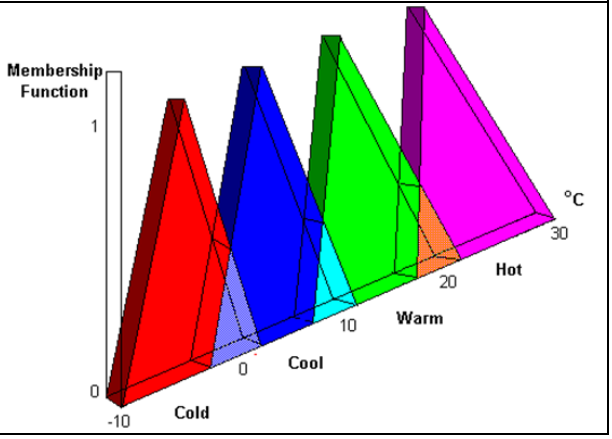
Ensembles bivalents pour caractériser la température d'une pièce.	**« Fuzzy Sets » ou ensembles flous qui représentent la température d'une pièce.**
	

Figure 2.3 : comparaison d'ensembles bivalents et de « Fuzzy Sets » (ensembles flous) source : http://www3.imperial.ac.uk/computing/.

Ces deux préoccupations sont identiques à celles qui nous animent, créant ainsi, semble-t-il, un lien entre ces travaux et nos recherches. La capacité de certains objets à appartenir à plusieurs communautés distinctes est indéniable. La complexité qu'apporte cette caractéristique dans le repérage de communautés au sein d'un graphe est importante. Les algorithmes qui le permettent seront plus complexes. Les techniques de validation devront aussi tenir compte de cette caractéristique.

Les méthodes sont regroupées en trois types principaux :

- Les méthodes de recherche de formes ;
- Les méthodes en plusieurs phases ;
- Les méthodes issues des méthodes sans recouvrement, modifiées pour permettre le recouvrement.

2.3.1 Méthodes de recherche de formes : la percolation de cliques

Dès les années 2000, J. Scott **[Scott-2000]**, identifie dans les réseaux sociaux des formes particulières au sein des graphes. Ces formes sont des sortes de motifs récurrents et connus. Le plus célèbre est sans aucun doute la **clique**. Il va au tout départ travailler sur ces ensembles. Mais il sent très vite que le système de clique s'il est référent quant à sa cohérence, n'est à la fois pas suffisamment défini et trop rigide. Pas suffisamment défini car, dans un réseau dont le degré moyen serait très élevé, cette forme n'est pas forcément suffisamment explicite. Il utilise alors la notion de « composante connexe » pour rechercher des ruptures dans un graphe de grande taille. Puis, pour pallier la rigidité des cliques il propose plusieurs adaptations avec les K-cliques et les n-cliques (qui définissent un chemin maximal entre deux nœuds de n liaisons) et enfin les k-plex dans lesquels les nœuds sont tous connectés deux à deux à l'exception de K nœuds. Nul doute que ces recherches sont les fondamentaux de la méthode la plus connue dans le recherche de communautés avec recouvrement qui est celle définie par G. Palla et al **[Palla&al-2005]**. Elle est aussi rencontrée ou nommée sous le nom de C-finder, du nom du logiciel libre l'implantant http://cfinder.org/.

Palla définit la méthode « C-finder » comme basée sur la localisation de toutes les cliques (sous-graphes complets maximaux) du réseau puis sur l'identification des communautés en effectuant une analyse en composante standard de la matrice de recouvrement entre chaque clique.

Pour donner plus de détails, nous pourrions ajouter que la méthode consiste à rechercher des cliques de k sommets nommées K-cliques, où k est un nombre entier supérieur à 3. La communauté est ensuite définie comme l'ensemble des K-cliques qui sont connectées de telle manière qu'un minimum de k-1 sommets de la clique de départ appartiennent à la clique ajoutée. Une image souvent donnée pour expliquer cette méthode est le fait que, si on créait un masque correspondant à une K-clique, toute K-clique sur laquelle on pourrait basculer le masque en gardant k-1 sommets dans le masque serait agrégée pour constituer la communauté.

La méthode décrite ici s'applique aux réseaux binaires mais elle peut être étendue à un réseau quelconque en ignorant le sens des liens et en supprimant les liens dotés d'une pondération trop faible.

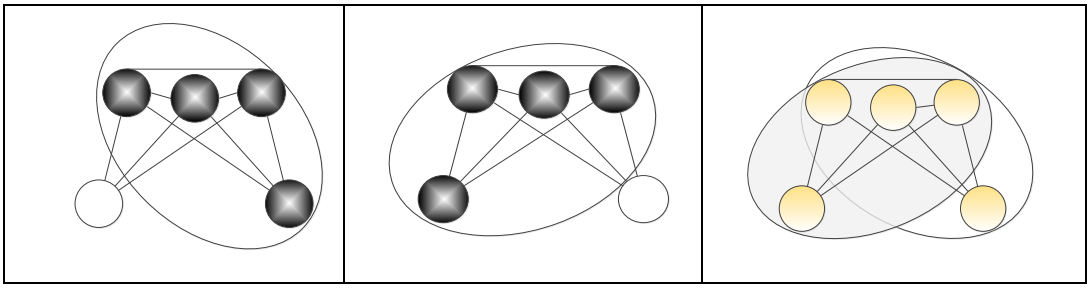


Figure 2.4 : Système de percolation de K-cliques pour k=4, par rotation de masque et communauté résultante.

Cette solution permet de créer des zones de recouvrement. En effet, un ensemble de 1 à k-2 nœuds peut être présent dans plusieurs communautés.

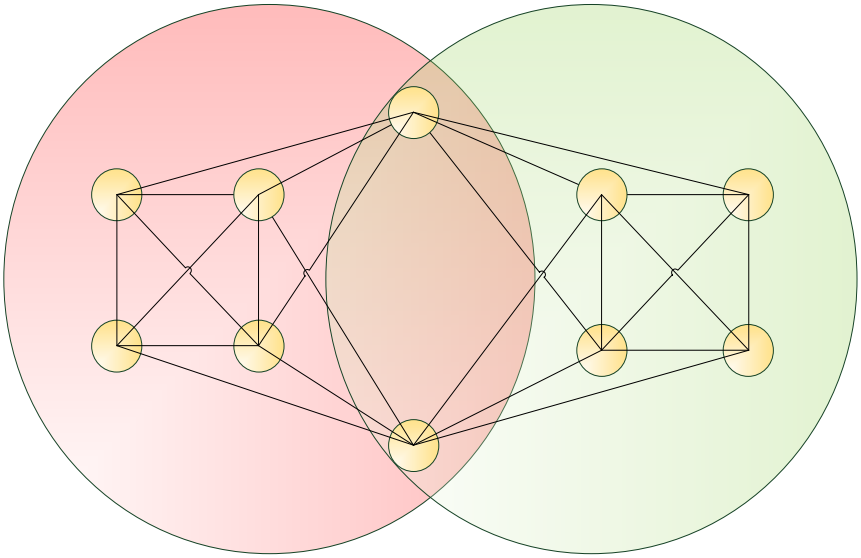


Figure 2.5 : Deux communautés de K-cliques pour k=6 partageant deux sommets.

Une autre vision de la méthode donnée par Palla lui-même est la méthode nommée « *K-clique* template rolling ».

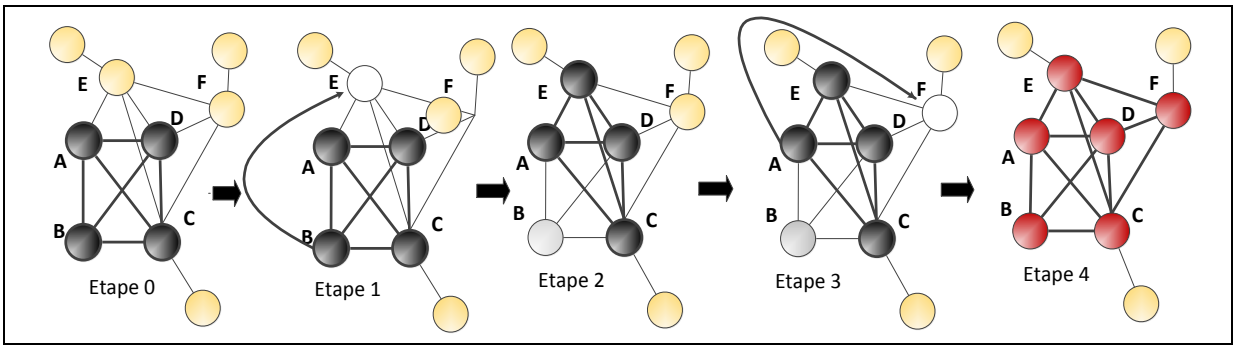


Figure 2.6 – Illustration de la méthode « *K-clique* template rolling ».

Comme on peut le voir dans la figure 2.6, la clique ABCD roule sur son axe AC en étape 1 pour former la clique ADCE en étape 2. Le point E vient ainsi rejoindre la communauté. Une autre rotation autour de l'axe EC en étape 3 permet de rajouter le point F et ainsi de créer la communauté finale ABCDEF en étape 4.

En 2007, Gregory Palla, Farkas, Pollner, Derényi et Vicsek ont fait évoluer leur méthode pour la rendre compatible avec des graphes dirigés **[Palla&al-2007]**.

Si cette solution est très performante dans le domaine du vivant, elle possède plusieurs inconvénients. Ainsi, fixer la valeur *K* du nombre de sommets à prendre en compte est critique et peut se révéler difficile. Car les graphes de terrain présentent par définition des zones de faible densité et d'autres de très forte densité. Choisir un *K* faible permettra de créer des communautés dans les zones de plus faible densité mais cela, au risque de créer des communautés immenses dans les zones de densité plus importante. Choisir une valeur de *K*

plus élevée ne nous permettra plus de créer de communautés dans les zones à faible densité. Cette méthode est très efficace dans le cas de graphe de densité relativement homogène. Cependant elle semble plus difficile à mettre en œuvre dans le cas de graphes à forte densité ou possédant une distribution très hétérogène des degrés ou encore dans l'étude de graphes pondérés.

2.3.2 Les méthodes en plusieurs phases

Les méthodes utilisant plusieurs phases sont généralement les méthodes qui vont, dans une phase de départ, rechercher un noyau ayant une très forte modularité. Ces zones vont être les parties non partagées des communautés. Ensuite, d'autres phases auront pour objectif d'enrichir ces zones noyaux de nœuds adjacents, les nœuds adjacents pouvant alors constituer des zones partagées entre plusieurs communautés.

Détection et enrichissement de noyaux

La méthode présentée par Shang, Chen et Zhou **[Shang&al-2007]** se déroule en trois phases au cours desquelles les noyaux sont détectés puis enrichis (cf. figure 2.7) :

1. la recherche des communautés « noyaux » de petite taille extrêmement connectée ;
2. la fusion de certaines communautés élémentaires en communautés plus importantes ;
3. l'expansion de chaque communauté aux nœuds périphériques.

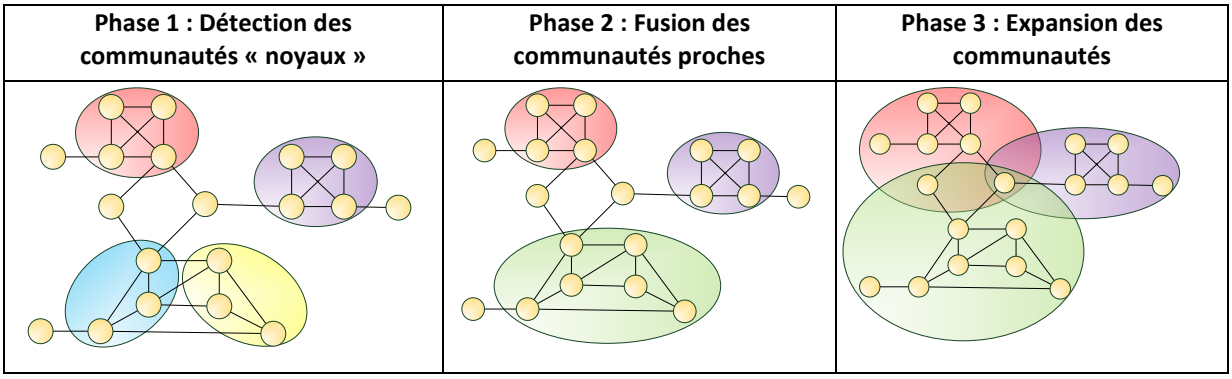


Figure 2.7 : Les trois phases de la méthode proposée par Shang et al.

La première phase consiste simplement en un repérage des ensembles de nœuds formant une k-clique telle que la clique de type (k+1)-clique contenant la clique précédente n'existe pas.

La deuxième phase va conduire à fusionner deux cliques s'il existe entre ces cliques une connectivité supérieure à une valeur prédéterminée.

La troisième phase, qui va permettre à plusieurs communautés d'être en recouvrement, visera à ajouter à une ou plusieurs communautés les nœuds qui n'appartiennent à aucune communauté. Tout nœud présentant une connectivité supérieure à une valeur prédéterminée

vers une ou des communautés sera considéré comme faisant partie de cette ou de ces communautés.

La méthode nous semble présenter beaucoup d'avantages ; en premier lieu, sa simplicité et sa capacité à évoluer, Elle est, par exemple, facilement applicable à des graphes pondérés. Les phases pourront alors évoluer pour tenir compte des spécificités de la nature d'un graphe. De plus, la recherche de noyaux est réalisable par des algorithmes différents, permettant un recouvrement entre noyaux. Les nœuds reliés par un seul lien à un nœud participant à une communauté peuvent rejoindre la dite communauté et limiter le nombre de nœuds hors communautés. La phase finale d'expansion peut, elle aussi, être adaptée selon la nature des objets.

Création de communautés puis affinage

Peu avant la publication des travaux menés par Palla, en 2009, Jeffrey Baumes, Mark Goldberg, Mukkai Krishnamoorthy, Malik Magdon-Ismail et Nathan Preston présentaient l'article *« Finding Communities by Clustering a Graph into Overlapping Subgraphs »* [**Baumes&al-2005-1]**. Cet article décrit deux nouveaux algorithmes permettant de dégager d'un graphe un ensemble de communautés.

Les auteurs font appel à la notion de « PageRank » pour mesurer l'importance d'un nœud dans un graphe. La notion de « PageRank » a été définie initialement par Page et al. en 1998 pour mesurer l'importance relative d'un site web dans Internet par rapport aux liens qu'il possède avec les autres sites **[Page&al-1998]**.

Le calcul du PageRank utilisé est basé sur la formule générale suivante :

$$PR(v) = c \sum_{u \in Bu} \frac{PR(u)}{Nu}$$

ou PR(v) est le PageRank du nœud *v*,
Bu est l'ensemble des nœuds voisin de *v*,
u est un des éléments de *Bu*, soit un voisin de *v*,
PR(u) est le PageRank du nœud *u*,
Nu est le degré de *u*,
et *c* un facteur de normalisation (choisi pour que la somme des « PageRank » soit une constante).

Il est utile de préciser que Page et al. définissent le « PageRank » sur un graphe dirigé et pondéré. Les liaisons signifiant l'existence d'un lien d'un site web vers un autre. Elles sont unilatérales et peuvent être plus ou moins nombreuses. *Nu* est alors défini comme le poids de l'ensemble des liens sortants du nœud *u* et *Bu* est alors l'ensemble des nœuds ayant un lien vers le nœud *v* **[Page&al-1998]**.

Le calcul du « PageRank » est issu d'un algorithme récursif. Cet algorithme ne possède pas toujours de point d'arrêt, puisque chaque « PageRank est dépendant du

« PageRank » de ses voisins. Pour trouver le PageRank de l'ensemble des nœuds, on initialise arbitrairement $R(u') = 1$ pour tous les nœuds u' du graphe. Puis on recalcule l'ensemble des « PageRank » du graphe, jusqu'à une convergence ou une certaine stabilité des résultats.

La méthode de création de communautés avec recouvrement proposée par Jeffrey Baumes et al. [**Baumes&al-2005-1]** s'articule autour de deux algorithmes :

- L'algorithme RaRe (Rank Removal) (cf . Figure 2.8) qui permet d'isoler un ensemble de noyaux de communautés. Pour cela, l'algorithme recherche les nœuds les plus « importants » (en utilisant, par exemple, le PageRank) et les retire du graphe de manière à ne conserver qu'un ensemble de petites composantes connexes. Ces composantes connexes donc deviennent les communautés. Les nœuds « importants » sont alors rajoutés tour à tour à ces composantes connexes pour finaliser les communautés et mettre en place les recouvrements.
- L'algorithme IS (Iterative Scan) qui permet de construire une partition du graphe localement optimale au sens de la densité. Il permet donc de raffiner les résultats obtenus avec l'algorithme RaRe de manière à obtenir une meilleure décomposition du graphe en communautés. Le principe de l'algorithme IS est simple : partant d'un ensemble de communautés racines (celles données par RaRe), un nœud quelconque du graphe est ajouté ou retiré à la communauté à chaque itération tant que la densité augmente.

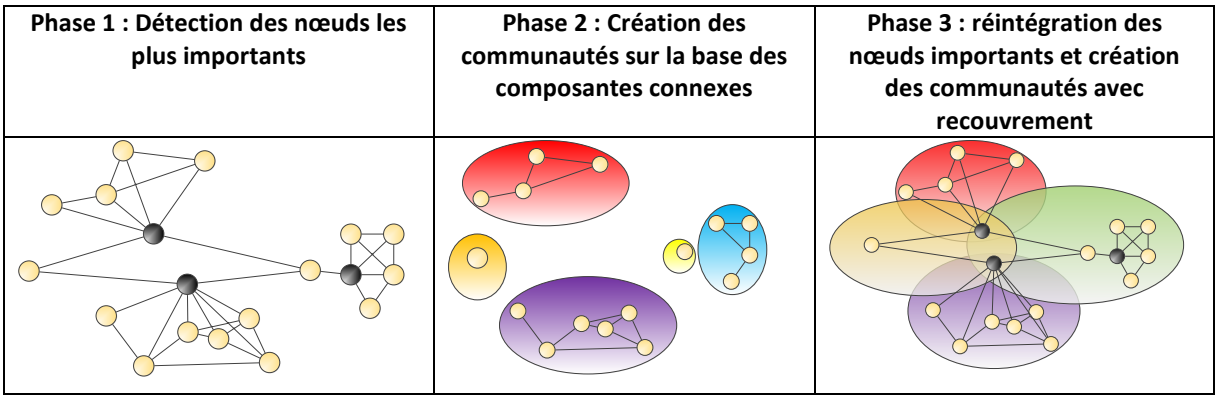


Figure 2.8 : Les trois phases de la méthode RaRe.

Même si la combinaison des algorithmes RaRe et IS permet d'obtenir des résultats corrects, leur efficacité en terme de temps de calcul n'a pas semblé satisfaisante aux auteurs. Des améliorations ont donc été proposées dans une autre publication : *« Efficient Identification of Overlapping Communities »* **[Baumes&al-2005-2]** :

- L'algorithme RaRe a été remplacé par l'algorithme LA (Link Aggregate). Partant d'un classement des nœuds selon une métrique donnée (par exemple, le « PageRank ») et d'un ensemble de communautés vide, chaque nœud est ajouté à toute communauté dont il permet d'augmenter la densité. S'il n'a été ajouté à aucune communauté existante, une nouvelle communauté est créée.

- L'algorithme IS a également été remplacé par l'algorithme IS², plus efficace car restreignant la recherche des points à ajouter, aux seuls nœuds adjacents à la communauté actuelle : sur les grands graphes, cela permet de réduire considérablement le nombre de possibilités à explorer.

La combinaison des algorithmes LA et IS² ne nécessite pas de fixer le nombre de communautés à l'avance. Le nombre de communautés proposé une fois le traitement effectué, est fonction du choix des critères d'extraction des nœuds importants.

Détection de communautés et fusion des communautés proches

En 2010, Arnau Padrol-Sureda, Guillem Perarnau-Llobet, Julian Pfeifle et Victor Munt´es-Mulero présentent l'algorithme OCA (Overlapping Community Algorithm) permettant de détecter des communautés avec recouvrement **[Padrol-Sureda&al-2010]**. Cet algorithme se veut particulièrement adapté aux très grands graphes. Il s'appuie sur l'optimisation locale d'une fonction « objectif » permettant d'évaluer la qualité d'une communauté. Constitué d'une première phase dont le but est de fractionner fortement le graphe, une seconde phase agrégera les communautés trop proches.

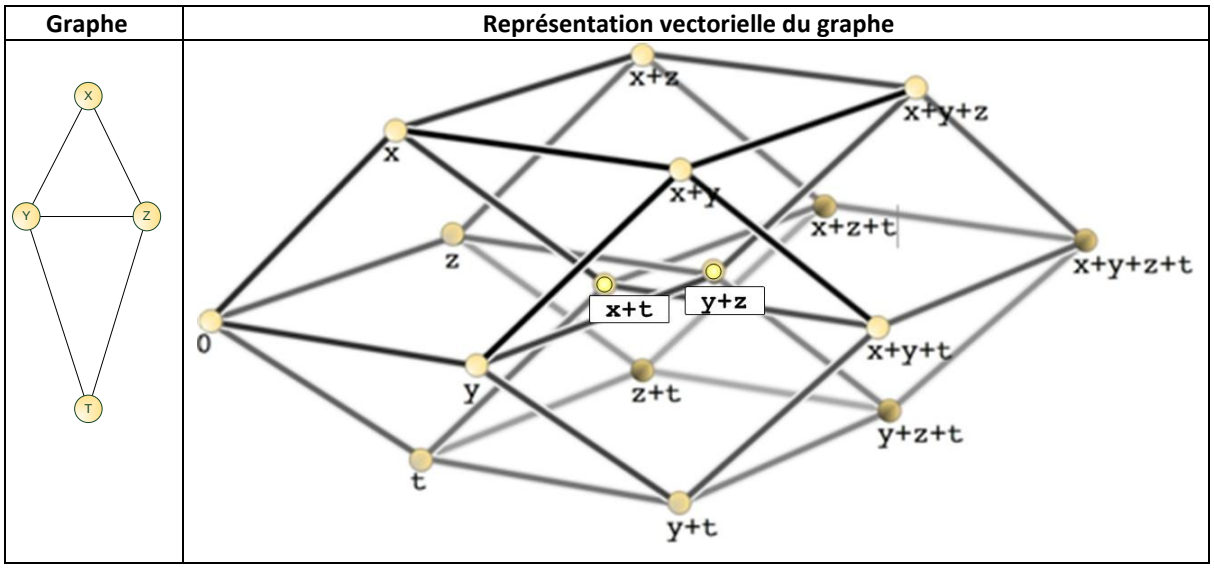


Figure 2.9 Exemple de graphe et représentation vectorielle associée.

Afin de poser les bases théoriques de leur méthode, les auteurs visualisent un graphe comme un ensemble de vecteurs dans un espace de grande dimension (cf. figure 2.9).

Chaque nœud *i* du graphe est associé à un vecteur v_i et tout sous-graphe $\{i_1,\ldots,i_n\}$ est associé à la somme des vecteurs qui le composent. La règle suivante est adoptée :

$$<v_i,v_k> = | v_i || v_k | cos(v_i,v_k) = c$$

Où $(0 < c < 1)$ s'il existe une arête reliant *i* à *k*, $c = 0$ sinon (angle droit entre v_i et v_k si i et k ne sont pas connectés).

Comme on peut le voir sur la figure 2.9, *y+z* (y et z sont connectés) est plus éloigné de 0 que *x+t* (x et t ne sont pas connectés) : la première idée est donc de choisir la norme euclidienne au carré comme fonction Θ à maximiser ($\Theta(v)$ = longueur du vecteur v au carré). Malheureusement, le maximum de cette fonction est atteint lorsque le graphe est contenu dans une seule et même communauté : en effet, comme le montre la figure 2.9, le vecteur de longueur la plus élevée est le vecteur *x+y+z+t*. Cette fonction ne peut donc pas être utilisée seule. En effet, si on la maximise toutes les composantes connexes deviendront des communautés.

De façon à éviter ce problème, les auteurs proposent de s'intéresser à l'impact du rajout ou de la suppression d'un nœud sur l'augmentation ou la diminution de Θ. Pour cela, ils introduisent la notion de laplacien dirigé $\mathcal{L}$. $\mathcal{L}$ peut s'apparenter à une dérivée lorsqu'on parle de fonction : ce laplacien permet d'évaluer dans quelles « directions » on peut espérer obtenir une augmentation de la valeur de la fonction Θ. La valeur en v de $\mathcal{L}$ pour une fonction f est définie par la formule :

$$\mathcal{L}_{\Gamma^{\uparrow},f}(v) \;=\; f(v) - \sum_{u:u\to v} \frac{f(u)}{\sqrt{\mathrm{indeg}(v)\ \mathrm{indeg}(u)}}$$

Où u représente un voisin de v appartenant à la communauté testée. La fonction indeg(x) retourne le degré entrant du nœud x.

Les variations de $\mathcal{L}$ sur l'ensemble de la communauté en ajoutant ou supprimant un nœud seront utilisées par l'algorithme OCA. L'algorithme OCA est, en fait, une première phase de l'ensemble du traitement, cette première phase recherchant les communautés indépendamment les unes des autres. Il démarre d'une graine positionnée aléatoirement dans le graphe. Le nœud permettant la plus grande augmentation de $\mathcal{L}$ est ajouté à la communauté et ce processus est répété jusqu'à ce que plus aucune amélioration ne puisse être obtenue ; ceci, autant de fois qu'il le faut pour former l'ensemble des communautés du graphe.

Tant que critère d'arrêt non satisfait **faire**
    Choisir un nœud (graine) aléatoirement dans le graphe -- (cette graine est le premier élément de la communauté k)
    Pour chaque Nœud de la composante connexe **faire**
     Si le Laplacien dirigé L augmente **alors**
      Ajouter ce nœud à la communauté k
     Fin de si
    Nœud suivant
Fin de tant que
Post-traitement des résultats : fusionner les communautés proches les unes des autres de façon à éviter des communautés trop proches et à permettre le recouvrement.

Figure 2.10 : Algorithme de la méthode d'Arnau Padrol-Sureda, Guillem Perarnau-Llobet, Julian Pfeifle et Victor Munt´es-Mulero [Padrol-Sureda&al-2010].

Le plus grand intérêt de cet algorithme est de permettre le traitement de très grands graphes en un temps relativement court : testé sur le graphe de Wikipedia (16 986 429

(16 986 429 nœuds, 176 454 501 arêtes) sur lequel un résultat a été obtenu en 3,25 heures seulement. OCA est, de plus, fortement parallélisable.

Selon les auteurs, les résultats obtenus nécessitent un post-traitement. En effet, selon la position des graines dans le graphe, des communautés extrêmement proches sont retournées. Ces communautés doivent alors être fusionnées pour arriver au partitionnement final du graphe.

Méthode spectrale et Fuzzy C-mean

Création de communautés puis fusion avec recouvrements

En 2007, dans un article nommé *« Identification of overlapping community structure in complex networks using fuzzy c-means clustering »,* Shihua Zhang, Rui-Sheng Wang et Xiang-Sun Zhang décrivent une méthode géométrique de partitionnement d'un graphe **[Zhang&al-2007]**. Cette méthode fait appel à une analyse spectrale du graphe puis à l'algorithme fuzzy c-means.

Partant d'un majorant K du nombre de communautés, l'algorithme permet d'établir un degré d'appartenance de chaque nœud à une communauté : on note u_{ik} le degré d'appartenance du nœud *i* à la communauté *k*, on aura donc $\Sigma u_{ik} = 1$ pour tout nœud *i*.

Comme beaucoup d'algorithmes de recherche de communautés, celui qui est présenté dans ce document fait appel à la notion de modularité introduite par Newman : les auteurs ont introduit une version modifiée de la modularité permettant de tenir compte des recouvrements.

Soit *A* la matrice d'adjacence du graphe et *D* la matrice diagonale (ou matrice des degrés) telle que $d_{ii} = \Sigma_j A_{ij}$.

L'algorithme se décompose en trois phases qui sont :

- Projection du graphe dans un espace euclidien

  Une méthode spectrale permettant de projeter les nœuds du graphe dans un espace euclidien de faible dimension est utilisée : cette opération nécessite le calcul des *K* vecteurs propres généralisés dominants du système $Ax = tDx$.

- Partitionnement par fuzzy c-means
  Une fois les nœuds du graphe projetés dans un espace euclidien, l'algorithme fuzzy c-means est utilisé pour former des communautés dans cet espace géométrique. Il utilise un critère de minimisation des distances intra-communautés et de maximisation des distances inter-communautés. Il calcule pour chaque nœud, un certain niveau d'appartenance à chaque classe en minimisant une fonction objective. Cet algorithme nécessite la connaissance préalable du nombre de communautés et les génère en utilisant un algorithme itératif minimisant une fonction.

Les principales étapes de l'algorithme fuzzy c-means introduit par Dunn en 1973 **[Dunn-1973]** sont :

1. la détermination arbitraire d'une matrice d'appartenance ;
2. le calcul des centroïdes des classes ;
3. le réajustement de la matrice d'appartenance suivant la position des centroïdes ;
4. le calcul du critère de minimisation et retour à l'étape 2 s'il y a non convergence de critère.

L'algorithme fuzzy c-means utilise, ici, comme fonction « objectif » la fonction définie par l'équation suivante :

$$J_m = \sum_{i=1}^{n} \sum_{j=1}^{k} u_{ij}^{m} \|x_i - c_j\|^2$$

Où u_{ij} est le degré d'appartenance de x_i dans la communauté j. c_j est le centre du $j^{ème}$ ensemble, x_i est le $i^{ème}$ point de données (dans notre cas, x_i sont les coordonnées du nœud i projeté dans l'espace euclidien). m est un paramètre de la méthode utilisé comme élément de réglage. Il permet selon les auteurs de régler le niveau de « flou », ce que nous pourrions traduire par la proportion de surfaces en recouvrement.

Afin d'identifier le nombre correct de communautés, l'opération de partitionnement est réalisée pour tous les k tels que $2 \leq k \leq K$.

- Maximisation de la modularité
  La fonction de modularité est évaluée pour chacun des partitionnements réalisés : la valeur de k maximisant la modularité et le partitionnement associés sont alors retenus.

La méthode proposée est intéressante car elle permet de visualiser le graphe dans un espace de petite dimension. De plus, seule la connaissance d'un majorant du nombre de communautés est nécessaire. En revanche, comme il est nécessaire de résoudre de nombreux problèmes aux vecteurs propres pour parvenir au résultat, l'utilisation de cet algorithme est rendue difficile sur des graphes de grande taille malgré la performance des méthodes numériques actuelles.

2.3.3 Les méthodes par déplacement d'objets

Parmi les méthodes permettant la création de communautés en recouvrement, figurent les méthodes par déplacement d'objets. L'attribution d'un nœud à une ou plusieurs communautés se fait soit en déplaçant le nœud entre les communautés soit en déplaçant des agents représentant une communauté d'un nœud à l'autre. Dans un cas comme dans l'autre**,** à chaque itération**,** on recalculera les coefficients d'appartenance entre les communautés et les nœuds. Ces méthodes sont toutes basées sur un nombre prédéterminé de communautés à

créer. Si certaines méthodes ont la possibilité d'utiliser ce nombre comme un maximum, les algorithmes par déplacement d'objets sont tous séparatistes.

Le déplacement des nœuds

En 2010, dans un article nommé *« A Game-Theoretic Framework to Identify Overlapping Communities in Social Networks »*, Wei Chen, Zhenming Liu, Xiaorui Sun et Yajun Wang proposent d'assimiler la formation des communautés à un jeu où chaque nœud serait un agent égoïste cherchant à maximiser sa propre « utilité » **[Chen&al-2010]**. Ainsi, à chaque itération, chaque nœud pourra quitter une communauté et/ou en rejoindre une autre : un changement ne sera accepté que s'il amène le nœud à maximiser son utilité.

L'utilité d'un individu est traduite par deux termes : un terme de « gain » tenant compte d'une modularité qui représente la capacité de l'individu à renforcer la communauté et un terme de « perte » qui est d'autant plus grand que l'individu fait partie d'un grand nombre de communautés.

La notion de gain s'appuie sur une version enrichie de la modularité telle qu'on peut la trouver chez Newman **[Newman-2006]**. Les auteurs parlent de « Nash equilibra **»** pour signifier que, dans le jeu de placement**,** les nœuds sont en quelque sorte coopératifs. Ils ne jouent pas les uns contre les autres mais les uns pour les autres, en équipe ou encore dans une stratégie gagnant-gagnant.

Sous certaines conditions sur les fonctions « gain » et « perte », il a été démontré que l'algorithme proposé converge vers un équilibre local donc vers une solution satisfaisante de partitionnement du graphe. La méthode présente l'avantage de n'avoir à connaître qu'un majorant du nombre de communautés.

L'accroissement des semences

L'accroissement des semences diffère fortement du déplacement de nœuds. Le mouvement est ici donné par l'accroissement de la taille de la communauté à partir de la graine.

En 2010, dans un article nommé *« Identifying Community Structures in Networks with Seed Expansion »*, Fang Wei, Weining Qian, Zhongchao Fei et Aoying Zhou décrivent une méthode basée sur un processus d'expansion à partir d'un ensemble de graines initialement réparties sur des sommets du graphe **[Vei&al-2010]**. Les auteurs définissent la méthode comme agrégative. Elle en a certains aspects. Pourtant nous la classerons ici comme une méthode séparatiste. En effet, le nombre de communautés est prédéterminé il y a donc avant tout un partage du graphe en N communautés.

L'algorithme présente le comportement suivant : à chaque itération, de nouveaux nœuds peuvent être ajoutés à chaque communauté issue d'une graine. Dans l'algorithme proposé, la probabilité de choisir un nœud libre donné est inversement proportionnelle à son degré.

Pour une communauté donnée, une fois toutes les probabilités calculées, celles-ci sont classées par ordre décroissant. Partant du nœud ayant la probabilité la plus élevée, le changement de modularité induit par l'ajout de ce nœud à la communauté est calculé : si l'ajout du nœud mène à une augmentation de la modularité, le nœud est ajouté à la communauté ; sinon, on répète le processus sur le nœud suivant.

Le principal inconvénient de cette approche est encore une fois de devoir fixer le nombre de communautés (le nombre initial de graines) à priori. De plus, le choix de la position des graines, primordial pour obtenir de bons résultats, n'est pas une tâche facile, en particulier sur des très grands graphes.

Le déplacement de particules

Le déplacement de particules est le résultat d'un paradigme où les nœuds représentent des espaces stables de résidences et les liaisons, des éléments permettant uniquement de se déplacer d'un nœud à un autre. Des particules vont se « promener » en sautant de nœuds en nœuds, chaque occupation d'un nœud par une particule lui permettant de recalculer positivement son niveau de possession.

En se fondant à la fois sur les notions de « ballade aléatoire », de niveaux d'appartenance et de déplacements choisis, Fabricio Breve, Liang Zhao et Marcos Quiles ont proposé en 2009, dans l'article « *Uncovering Overlap Community Structure in Complex Networks Using Particle Competition* », une nouvelle méthode **[Breve&al-2010]**. Cette méthode permet de détecter les recouvrements entre communautés dans un réseau complexe.

La méthodologie proposée s'appuie sur une compétition entre un ensemble de c objets mobiles, nommés particules $\{\beta_1,...,\beta_c\}$ qui évoluent dans le réseau en se déplaçant de nœud en nœud. Chaque particule représente une communauté C_i distincte. Le nombre de communautés créées est égal au nombre de particules mises en œuvre. L'appartenance d'un nœud à une communauté est alors la résultante de la possession d'un nœud par une particule. Si plusieurs particules se partagent le nœud alors nous sommes en présence de zones de recouvrement.

À l'état initial, la méthode va prédisposer les particules de façon aléatoire sur les nœuds du graphe. Chacune des particules se voit attribuer un niveau de propriété P égal sur l'ensemble des nœuds du graphe. Ce niveau de propriété est équivalent au niveau d'appartenance du nœud à une communauté.

À chaque étape de l'algorithme, les particules pourront se déplacer soit de façon aléatoire vers un nœud voisin soit de façon déterministe. Le choix du type de déplacement est fixé par un coefficient K, choisi au départ. Ensuite, de manière statistique, afin de respecter ce coefficient, la particule choisit son type de déplacement. Les particules peuvent se déplacer de deux façons dans le réseau :

- mouvement aléatoire : la particule se déplace sur un nœud adjacent avec une probabilité égale pour chacun des nœuds visitables ;

- mouvement déterministe : la particule se déplace sur un nœud adjacent avec une probabilité dépendant de son niveau de propriété sur chacun des nœuds visitables. Dans ce cas-là, la particule a tendance à visiter des nœuds où sa propriété est déjà forte par rapport aux autres particules. Le niveau de propriété exprimé par la particule est aussi le niveau d'appartenance du nœud à une communauté, puisque chaque particule représente une communauté.

Le mouvement déterministe permet aux particules de garder la main sur les nœuds qui leur appartiennent tandis que le mouvement aléatoire permet de visiter les nœuds participant au recouvrement avec une autre communauté. Le coefficient *K* sert alors à régler le niveau de « curiosité » de la particule voyageuse.

La somme de l'ensemble des propriétés des particules sur un nœud donné ne pouvant pas dépasser 100%, les particules vont alors mener une compétition pour s'octroyer la propriété des nœuds. Le niveau de propriété entre les particules et les nœuds va alors s'ajuster au fur et à mesure des itérations.

Chaque particule dispose en outre d'un potentiel indiquant sa « force ». Ce potentiel permet à la particule de pouvoir rivaliser ou non avec les autres particules : ainsi, si une particule n'est pas assez forte, elle ne pourra pas s'aventurer dans une zone appartenant à une particule plus forte qu'elle. La particule va perdre ou gagner du potentiel (ou de sa « force ») au prorata du coût de son déplacement. Si le nœud à éteindre au temps *(t +1)* est un nœud sur lequel la particule a un fort niveau de propriété sa force augmente et elle baissera dans le cas contraire.

La particule P_j possède donc à l'instant *t* un potentiel notée $P_j^{\omega}(t)$. Ce potentiel correspond à la valeur de la « force » avec laquelle la particule peut « s'affecter » un nœud. Cette valeur est bornée par ω_{min} *et* ω_{max} tel que $\omega_{min} = 0$ *et* $\omega_{max}=1$.

À chaque itération de la boucle de l'algorithme, on se place ici dans le cas où le nœud *i* a été choisi par la particule *j* au temps *(t +1)*, les propriétés et les potentiels sont mis à jour par les formules ci-dessous :

- Calcul de la propriété :

$$v_i^{\omega_k}(t+1) = \begin{cases} \max\{\omega_{\min}, v_i^{\omega_k}(t) - \frac{\Delta_v \rho_j^{\omega}(t)}{c-1}\} & \text{if } k \neq j \\ v_i^{\omega_k}(t) + \sum_{q \neq k} v_i^{\omega_q}(t) - v_i^{\omega_q}(t+1) & \text{if } k = j \end{cases}$$

Où $v_i^{\omega k}$(t) est la propriété de la particule *k* sur le nœud *i* au temps *t* et ρ_j^{ω} le potentiel de la particule *j* à un instant donné

- Calcul du potentiel :

$$\rho_j^{\omega}(t+1) = \rho_j^{\omega}(t) + \Delta_{\rho}(v_i^{\omega_j}(t+1) - \rho_j^{\omega}(t))$$

Où Δ_v et Δ_ρ sont deux paramètres permettant de contrôler la vitesse d'évolution des deux grandeurs (propriété et potentiel).

La méthode permet, après un certain nombre d'exécutions de l'algorithme, de déterminer un degré d'appartenance d'un nœud à une communauté : un nœud v_i aura un fort degré d'appartenance à la communauté C_k si le degré de propriété de la particule β_k sur le nœud $\mathbf{v_i}$ est fort. L'algorithme permettant le calcul des valeurs de propriété et de potentiel peut donc se lire ainsi :

POUR la particule *J* voulant aller en nœud *i* **faire**
POUR *K*=1 à nombre de particules **faire**
SI *k* différent de j **alors**
La valeur de propriété de la particule *K* sur le nœud *i* au temps *t+1* est égale à la valeur maximale entre la borne ω_{min} et sa valeur au temps t moins le rapport du coefficient Δ_v multiplié par la valeur du potentiel de la particule *K* sur le nœud *j* au temps *t* sur le nombre de communauté moins 1.
SI NON (k=j)
La valeur de propriété de la particule J sur le nœud i au temps *t+1* est égale à sa valeur au temps *t + différence* de propriété des autres communautés entre le temps *t* et le temps *t +1*. Ceci afin de s'assurer que la somme des coefficients de propriété de chaque nœud est conservée toujours égale à 1.
FIN DE SI
FIN DE POUR k
Le potentiel au temps *t+1* de la particule *j* sur le nœud *i* est égal au potentiel au temps *t + le coefficient* Δ_ρ multiplié par la propriété de la particule au temps *t+1* sur le nœud *j* moins le potentiel de la particule *J* au temps *t*.
FIN DE POUR voulant aller en *i*

Figure 2.11 : Algorithme de la méthode de Fabricio Breve, Liang Zhao et Marcos Quiles 2009.

Ces degrés d'appartenance permettent, en outre, de définir un indice de recouvrement o_i de telle sorte qu'un indice proche de 0 indique que le nœud appartient avec certitude à une seule communauté et un indice proche de 1 indique que le nœud appartient avec certitude à deux communautés ou plus.

Un exemple d'exécution de l'algorithme sur un réseau simple avec 5 nœuds et 2 particules est donné ci-dessous. On a choisi Δ_v=0,4 et Δ_ρ=0,9.

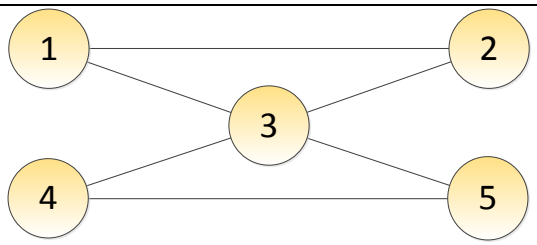
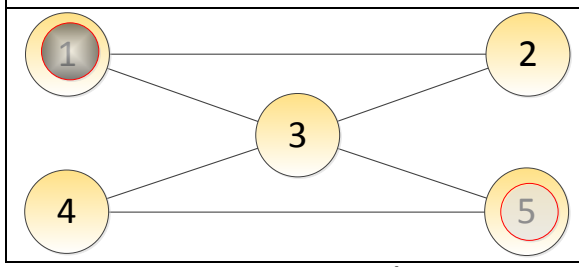
Un graphe de 5 nœuds	**2 particules (afin de créer 2 communautés) : une noire et une blanche**
	

Figure 2.12a : Les éléments en interaction pour illustrer la méthode de Fabricio Breve et al.

À chaque itération de l'algorithme, les particules recalculent les différents niveaux d'appartenance et de polarité :



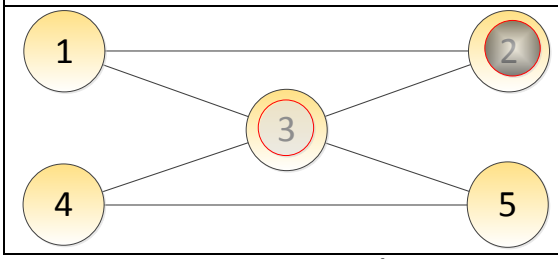
Nœuds	État 0 App.
1	0.5
2	0.5
3	0.5
4	0.5
5	0.5

Nœuds	État 0 App.
1	0.5
2	0.5
3	0.5
4	0.5
5	0.5

Particules		État 0 Pot.
	Noire	0
	Blanc.	0

Figure 2.12b : État 0 - illustration de la méthode de Fabricio Breve et al.

- À **l'état initial 0**, les deux particules sont placées aléatoirement sur le nœud 1 pour la particule noire et sur le nœud 5 pour la particule blanche. Les potentiels sont définis à ω_{min} soit à 0 et les valeurs d'appropriation de manière équitable entre les nœuds et les particules.



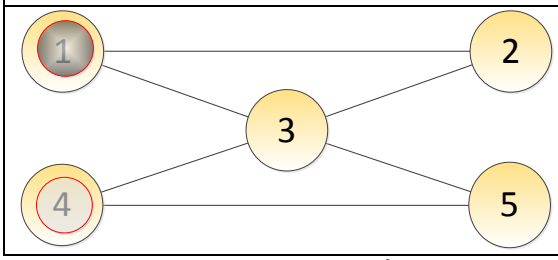
Nœuds	État 1 App.
1	0.5
2	0.5
3	0.5
4	0.5
5	0.5

Nœuds	État 1 App.
1	0.5
2	0.5
3	0.5
4	0.5
5	0.5

Particules		État 1 Pot.
	Noire	0.45
	Blanc.	0.45

Figure 2.12c : État 1- illustration de la méthode de Fabricio Breve et al.

- **À l'état 1**, les deux particules sont déplacées de manière probabiliste sur des nœuds adjacents : les valeurs d'appropriation n'évoluent pas (car les potentiels sont nuls en début d'exécution) mais les potentiels soit initialisés.

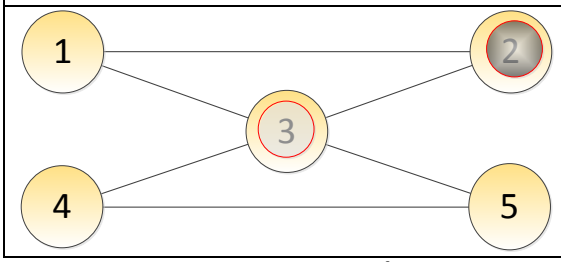


Nœuds	État 2 App.
1	0.68
2	0.5
3	0.5
4	0.32
5	0.5

Nœuds	État 2 App.
1	0.32
2	0.5
3	0.5
4	0.68
5	0.5

Particules		État 2 Pot.
	Noire	0.657
	Blanc.	0.657

Figure 2.12d : État 2 illustration de la méthode de Fabricio Breve et al.



Positionnement des particules à l'état 3

Appropriation de la particule noire

Nœuds	État 3 App.
1	0.68
2	0.763
3	0.237
4	0.32
5	0.5

Appropriation de la particule blanche

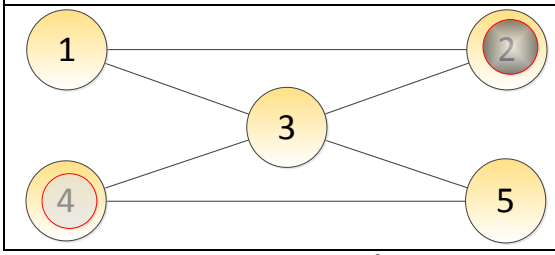
Nœuds	État 3 App.
1	0.32
2	0.237
3	0.763
4	0.68
5	0.5

Valeurs de potentiel des particules

Particules		État 3 Pot.
	Noire	0.752
	Blanc.	0.752

Figure 2.12e : État 3 illustration de la méthode de Fabricio Breve et al.

- **Aux états 2 et 3**, les particules commencent à faire évoluer leurs valeurs d'appropriation sur les nœuds de leur communauté respective : l'aléa amène la particule blanche à visiter le nœud 3 qui constitue une zone de recouvrement entre les deux communautés.



Positionnement des particules à l'état 4

Appropriation de la particule noire

Nœuds	État 4 App.
1	0.68
2	0.763
3	0.538
4	0.019
5	0.5

Appropriation de la particule blanche

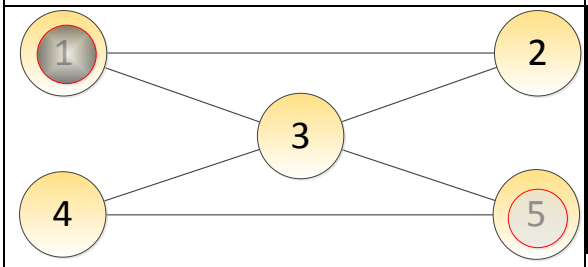
Nœuds	État 4 App.
1	0.32
2	0.237
3	0.462
4	0.981
5	0.5

Valeurs de potentiel des particules

Particules		État 4 Pot.
	Noire	0.559
	Blanc.	0.958

Figure 2.12f : État 4 - illustration de la méthode de Fabricio Breve et al.

- **À l'état 4**, la particule noire tente de s'aventurer sur le nœud 3 qui a, pour le moment, une forte valeur d'appropriation pour la particule blanche : cette valeur d'appropriation blanche empêche la particule noire de s'aventurer « physiquement » sur le nœud pour le moment, elle reste donc sur le nœud 2, son nœud de départ. La valeur d'appropriation de la particule noire sur le nœud 3 augmente au prix d'une diminution de son potentiel : cela peut être vu comme la force perdue par la particule noire dans son combat contre la particule blanche pour la possession du nœud 3.



Positionnement des particules à l'état 5

Appropriation de la particule noire

Nœuds	État 5 App.
1	0.904
2	0.763
3	0.538
4	0.019
5	0.117

Appropriation de la particule blanche

Nœuds	État 5 App.
1	0.096
2	0.237
3	0.462
4	0.981
5	0.883

Valeurs de potentiel des particules

Particules		État 5 Pot.
	Noire	0.869
	Blanc.	0.891

Figure 2.12g : État 5 - illustration de la méthode de Fabricio Breve et al.

- **À l'état 5**, la particule blanche visite pour la première fois le nœud 5. À cette étape, les forts potentiels font que les valeurs d'appropriation peuvent évoluer très vite. Les communautés commencent déjà à se dessiner, comme le montrent la forte valeur d'appropriation de la particule noire sur les nœuds 1 et 2 et celle de la particule blanche sur les nœuds 4 et 5. Enfin, la valeur d'appropriation des deux particules sur le nœud 3 (aux alentours de 0.5) signale bien que ce nœud participe à un recouvrement entre les deux communautés.

Chaque particule a à la fois exploré un territoire pour se l'approprier et défendu des positions acquises. La méthode est séduisante, elle peut effectivement apparaître comme la mise en œuvre d'un modèle intuitivement semblable avec celui du monde animal. Malgré tout il ne faut pas oublier que de nombreux paramètres comme Δ_v et Δ_ρ sont à fixer préalablement pour la faire fonctionner et que la méthode requiert également le choix du nombre de communautés à créer. De plus, chaque communauté garde le plus souvent une participation (même si elle est infime) sur chaque nœud.

2.3.4 Méthodes modifiées pour permettre le recouvrement

Avec la prise de conscience que le recouvrement est généralement nécessaire pour coller à une réalité, les auteurs de méthodes modifient des méthodes existantes qui ne permettent pas le recouvrement afin de permettre le recouvrement.

Algorithme C.O.N.G.A. modifié

Un exemple intéressant est la méthode que Steve Gregory présente, dans « *An Algorithm to Find Overlapping Community Structure in Networks* » **[Gregory-2010]**. Il s'appuie sur les travaux menés par Newman et Girvan et l'algorithme **C.O.N.G.A.** (**C**luster-**O**verlap **N**ewman **G**irvan **A**lgorithm) **[Newman&al-2004-3]**. Cet algorithme est basé sur la notion de mesure de centralité (*betweeness centrality measure*).

L'algorithme initialement proposé par Girvan et Newman consistait à retirer les arêtes dont la centralité est la plus élevée jusqu'à séparer le graphe en un certain nombre d'ensembles de nœuds disjoints. Cet algorithme n'autorisait donc pas les recouvrements entre communautés. L'idée de Steve Gregory est d'ajouter, en plus de la suppression d'une arête, la possibilité de réaliser une copie (virtuelle) d'un nœud du graphe en introduisant une arête virtuelle entre le nœud original et sa copie. De cette façon, un nœud pourra faire partie d'une ou plusieurs communautés distinctes.

L'algorithme CONGA est le suivant (où $c_B(v)$ représente la centralité du nœud v) :

TANT QUE il reste des arêtes **faire**
    Trouver l'ensemble V des nœuds v tels que $c_B(v) > \max(c_B(e))$
    SI V n'est pas vide **alors**
        Rechercher le nœud de V dont la meilleure scission a la plus grande centralité
        Réaliser une copie du nœud trouvé selon sa meilleure scission
    SINON
        Supprimer l'arête de centralité maximale
    FIN SI
    Mettre à jour les valeurs de centralité de toutes les arêtes et nœuds du graphe
FIN TANT QUE

Figure 2.13 : Algorithme CONGA pour calculer la centralité de toutes les arêtes et nœuds du graphe.

Il existe généralement plusieurs façons d'introduire l'arête virtuelle. La solution retenue par Newman et Girvan **[Newman&al-2004-3]** est celle qui permet de maximiser la centralité de l'arête virtuelle introduite (voir figure 2.14 où, partant de la configuration de départ, on retiendra la configuration A). On parle de meilleure scission d'un nœud.

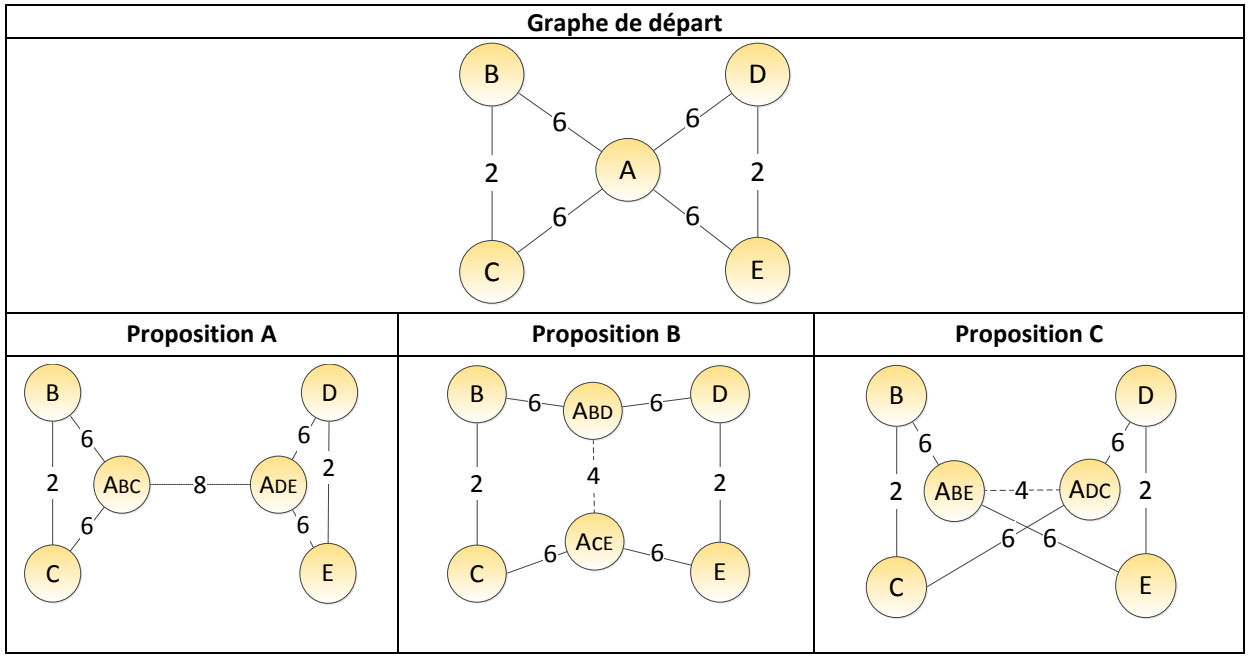


Figure 2.14 : Insertion d'une arête virtuelle pour rechercher la proposition créant la plus haute valeur de centralité.

L'algorithme a permis d'obtenir de bons résultats sur des cas tests standards. Toutefois, son temps de calcul est trop élevé pour une application sur des graphes très larges : 30411 secondes (plus de 8 heures) ont été nécessaires pour partitionner un graphe contenant seulement 3982 nœuds et 6803 arêtes (Pentium 4, 2.4 GHz).

Overlapping Stochastic Block Models modifié

Dans ces méthodes issues des méthodes de création de communautés sans recouvrement, une dernière méthode mérite notre attention. Pierre Latouche, Etienne Birmelé et Christophe Ambroise présentent, dans « *Overlapping Stochastic Block Models with*

Application to the French Political Blogosphere » **[Latouche&al-2010]**, une extension de la méthode « Stochastic Block Models » autorisant les recouvrements. La méthode proposée recherche des classes dans le graphe : plus générale que la notion de communauté, une classe peut être une communauté mais peut également décrire des types très variés de connexion entre nœuds (présence de configurations en étoiles dans le graphe par exemple).

Dans la suite, on note X la matrice d'adjacence du graphe dirigé considéré. Chaque élément de la matrice noté X_{ij} représente la présence ou l'absence d'une liaison de i à j. Partant d'un nombre de classes q fixé, l'idée est d'associer à chaque nœud un vecteur de q valeurs aléatoires (0 ou 1) Zi pour le nœud i dont chacun des q éléments suit une loi de Bernoulli.

On aura :

- $Z_i[c]$=1 si le nœud i appartient à la communauté c,
- $Z_i[c]$=0 sinon.

Plusieurs composantes du vecteur Z_i pouvant être égales à 1, cette méthode autorise bien le recouvrement entre les différentes classes.

Connaissant les vecteurs Z_i et Z_j, on peut exprimer la loi conditionnelle $X_{ij}|Z_i, Z_j$ c'est-à-dire la probabilité qu'une liaison entre i et j existe ($X_{ij}=1$) selon les valeurs prises par Z_i et Z_j c'est-à-dire selon l'appartenance des nœuds i et j à une communauté donnée.

Les auteurs proposent un algorithme d'optimisation basé sur la maximisation de la log-vraisemblance des données observées. L'objectif est donc de trouver le jeu de paramètres qui expliquent au mieux la présence ou l'absence d'arêtes (connexions) dans le réseau. Cette log-vraisemblance n'est pas calculable et les auteurs ont recours à une approximation variationnelle. L'algorithme proposé permet d'estimer à la fois les paramètres ainsi que les vecteurs Z_i, c'est à dire les classes auxquelles appartiennent les nœuds du réseau.

La méthode a pour avantage d'ouvrir de nouvelles perspectives. La prise en compte de plusieurs éléments servant à construire les communautés est sans aucun doute un axe de recherche pertinent dans les graphes de terrain. En effet, dans le monde réel, les nœuds présentent des caractéristiques supplémentaires qui ne sont pas toutes représentées par le graphe. Ces caractéristiques peuvent donner lieu à la création de classes qui seront ensuite exploitées au mieux dans la création des communautés. Cette méthode reste cependant une proposition où le nombre de communautés doit être pré-prédéterminé.

2.4 Les méthodes de validation des communautés

2.4.1 Validation qualitative

Pour valider les méthodes sans recouvrement, il peut suffire de calculer la modularité des communautés. Si celle-ci est supérieure à 0, on peut déjà décider que le « découpage » a une certaine pertinence. La plupart des méthodes sans recouvrement utilisent cette propriété pour valider qualitativement les communautés créées.

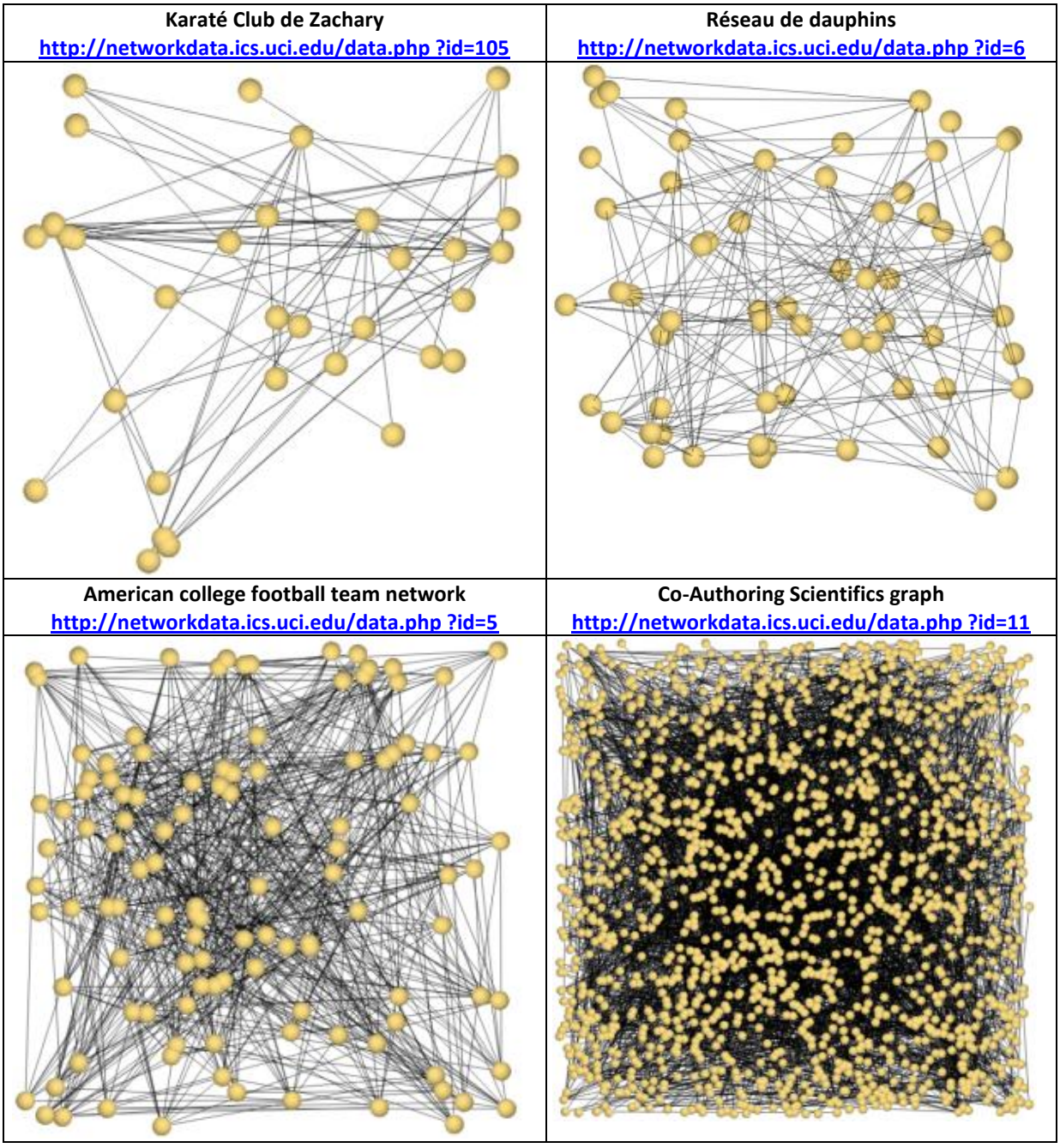


Figure 2.15 : Les graphes jouets utilisés couramment en validation des méthodes de regroupement.

La validation d'un système de création de communautés avec recouvrement est plus délicate. Les méthodes de validation de ces communautés sont dans l'ensemble des méthodes qui n'ont pour but que de chercher à démontrer la validité de l'algorithme. On choisit pour cela des graphes « jouets » de petite taille. Ces graphes que l'on va retrouver dans la majorité des expérimentations **[Chen&al-2010]**, **[Shang&al-2007]**, **[Nicosia&al-2009]**, **[Baumes&al-2005-1]**, **[Zhang&al-2007]**, **[Breve&al-2010]** ont l'avantage de présenter une solution connue.

Il y a ainsi des incontournables (cf. figure 2.15), tels le fameux « Club de Karaté de Zachary ». Dans ce club, on connaît les liens d'amitié initiaux entre les membres à partir desquels on peut construire un graphe de type réseau social. Les membres sont les nœuds et les relations d'amitié les liens. Historiquement, le club s'est ensuite scindé en deux clubs distincts après la dispute de l'entraîneur et du directeur. Chacun des deux a créé un nouveau club. Pour l'anecdote, cette scission s'est faite sans aucun recouvrement. Ceci n'empêchera pas Wei Chen, Zhenming Liu, Xiaorui Sun et Yajun Wang de nous proposer pour valider la méthode de déplacement de nœuds présentés au chapitre 2.3.3, jusqu'à six communautés en recouvrement sur ce graphe (cf. figure 2.16) **[Chen&al-2010]**. La qualité des communautés est mesurée dans ce cas sur les valeurs des modularités atteintes.

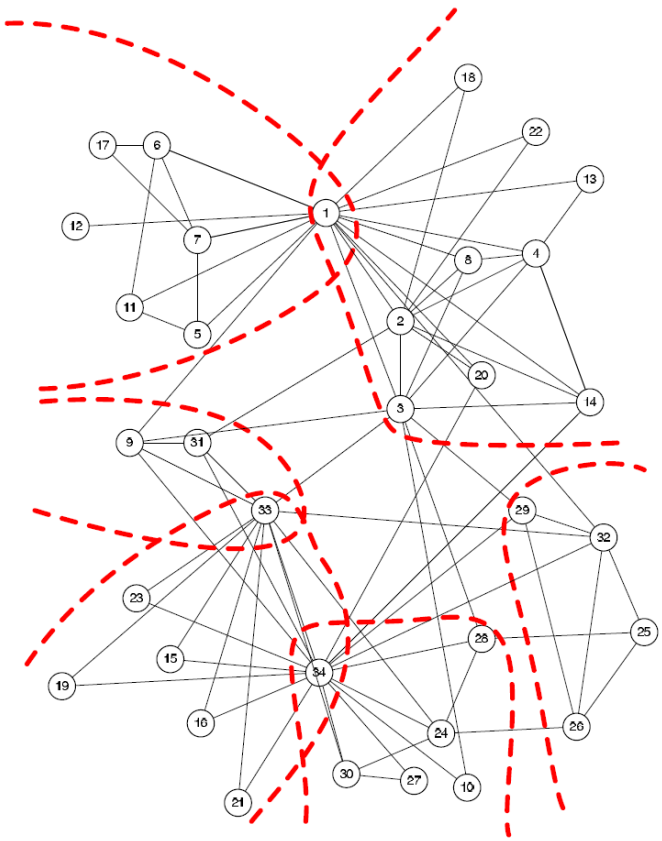


Figure 2.16 : les 6 communautés issues de l'algorithme de Wei Chen, Zhenming Liu, Xiaorui Sun et Yajun Wang [Chen&al-2010] sur le graphe représentant le Karaté Club de Zachary.

On peut d'ailleurs remarquer que la plupart des algorithmes présentés doivent être paramétrés (ne serait-ce que par le nombre de communautés à créer). Connaitre la solution permet de les choisir de façon à retomber sur le nombre de communautés souhaité.

D'autres exemples partagés par plusieurs méthodes (comme les réseaux de collaboration entre scientifiques **[Latouche&al-2010** ou la recherche des régions des équipes de football aux Etats-Unis en fonction des matchs joués) présentent aussi la même dérive. Soit le résultat recherché est connu, soit il n'est pas facilement vérifiable et donc pas vérifié. À notre connaissance, aucune expérimentation utilisant les signatures conjointes d'articles n'a cherché à croiser les résultats avec les sentiments d'appartenance à des communautés des auteurs.

De plus, ces exemples ne sont pas à proprement parler des graphes de terrain. En effet, les éléments sont en fait extraits de graphes de terrain plus complexes. Par exemple, le graphe des relations sociales des membres du Karaté Club ne représente qu'une partie microscopique d'un graphe sans aucun doute plus large des relations sociales incluant les membres du club. Les graphes de co-signatures sont aussi limités par une thématique et le plus souvent une université ou un laboratoire. Ces extractions et la petite taille permettent alors de limiter grandement les problématiques liées aux graphes de terrain.

Matthieu Latapy dans son HDR **[Latapy-2007]** déclare à ce sujet *« on se retrouve à évaluer les résultats par l'intuition qu'on a sur de tout petits exemples, sur des modèles dont on sait qu'ils capturent pauvrement la réalité et/ou sur des cas réels sur lesquels on n'a guère de point de comparaison »*.

Ces exemples jouets et de petites tailles ont peu de ressemblance avec les grands graphes de terrain. Et la principale différence est sans nul doute l'importante disparité que l'on va rencontrer dans les grands graphes de terrain sur les valeurs des degrés. Un autre aspect de la simplification de ces graphes jouets est la non-pondération des liaisons. En effet, les relations sociales n'ont pas toutes la même importance. De même, une co-signature dans un article de vulgarisation n'a pas la même valeur qu'un ensemble de co-signatures dans un domaine particulièrement pointu ou sur des articles majeurs de revue.

Ces exemples jouets ne sont pas pour autant à écarter. Les algorithmes validés à travers eux n'ont pas pour cela moins de valeur ou d'intérêt. Pratiques et de petite taille, ils permettent de comparer des résultats. Cependant, ces exemples ne peuvent être assimilés à de véritables grands graphes de terrain.

En 2009, dans l'article « *Overlapping Community Search for Social Networks* », Arnau Padrol-Sureda, Guillem Perarnau-Llobet, Victor Muntès et Julian Pfeifle **[Padrol-Sureda&al-2010]** utilisaient comme espace de test le graphe de terrain des articles de Wikipedia. Celui-ci est sans aucun doute, avec plus de 16 millions de nœuds, un grand graphe de terrain. Pourtant de notre point de vue, plusieurs aspects essentiels sont occultés. En premier lieu, le lien hypertexte qui ici est utilisé comme élément de liaison est par nature dirigé. Un article A cité par un article B ne signifie pas que l'article B citera le A. Le graphe étudié n'est pas dirigé. Il devrait aussi être pondéré, une page pouvant avoir plusieurs liens vers une autre page. Le graphe n'est pas pondéré dans l'étude. Enfin, les articles ne sont pas étudiés ici comme faisant partie d'un tout mais comme des éléments indépendants. En effet, sur chaque article Wikipédia, on retrouve des liens vers des pages repères comme l'accueil du site ou l'index ou même la possibilité de rechercher un article au hasard. On retrouve aussi un

lien vers les articles du même sujet dans les autres langues. Ces liens sont issus de la nature d'un Grand Graphe de Terrain. Les nœuds communiquent pour assurer leurs fonctions de base, mais certains nœuds ont pour fonction la communication. Dans le graphe de Wikipédia, pour pouvoir être trouvés plus facilement, beaucoup d'articles sont cités dans l'index alphabétique. Cette liste possède donc des liens vers un très grand nombre d'articles. Il existe dans l'autre sens depuis chaque page un lien vers ce glossaire. En excluant l'index alphabétique du graphe, on en change la nature. Nous reviendrons sur cet aspect des grands graphes de terrain dans la partie 3 de notre travail.

Une autre piste est souvent explorée, il s'agit d'étudier les algorithmes de création de communautés sur des graphes générés aléatoirement par des algorithmes spécifiques. Le but de ces algorithmes créateurs de graphes est de recréer un graphe aux caractéristiques identiques à celles d'un Grand Graphe de Terrain. Quelles que soient les techniques de génération **[Barabas&ali-2000] [Watts-1998**]**,** leur capacité à imiter des graphes de terrain ou des petits mondes n'est pas mesurable. De plus, les communautés ainsi obtenues ne peuvent être comparées qu'entre diverses méthodes, la nature mathématique du graphe ne pouvant être modifiée. Pour être concret, il n'est pas possible de demander à un nœud d'un graphe généré aléatoirement si effectivement il se sent bien dans cette communauté comme on pourrait le faire avec des acteurs d'un réseau social.

Ces difficultés sont particulièrement bien explicitées dans l'article publié en 2010 par Emmanuel Navarro et Rémy Cazabet : *« Détection de communautés étude comparative sur graphes réels »* **[Navarro&al-2010].** Dans cette étude, les auteurs comparent plusieurs algorithmes sur plusieurs types de graphes et notamment sur des graphes de terrain de taille « moyenne », le plus important comportant moins de 10000 nœuds. Les auteurs concluent en disant : *« ... il en ressort que le problème de détection de communautés est plus complexe sur des graphes réels que sur les graphes artificiels habituellement utilisés pour l'évaluation. L'accord entre les méthodes est faible sur les graphes réels alors qu'il est généralement important sur les graphes d'évaluation. Aussi les résultats sont bien moins robustes sur les graphes réels que sur les graphes d'évaluation. Et enfin les algorithmes ont tendance à trouver, sur les graphes réels, des « super-communautés » peu réalistes. ».*

Pour faire face à cette difficulté de validation qualitative des communautés, nous proposons une projection de ce que Matthieu Latapy suggère pour valider les caractéristiques des graphes : « la comparaison à l'aléatoire » **[Latapy-2007].** Nous proposons, la comparaison du comportement et des propriétés des communautés créées par la comparaison à des communautés créées aléatoirement. Puis, dans la mesure où cela est possible, reconduire la comparaison avec des regroupements reconnus comme des communautés valides. Il est aussi possible de comparer les communautés créées avec des ensembles de population aux caractéristiques connues. Cet ensemble de comparaisons permettra d'obtenir une estimation plus juste de la qualité du système de regroupement.

2.4.2 Évaluation de la complexité

Une autre préoccupation des utilisateurs et concepteurs de méthodes de création de communautés avec ou sans recouvrements est la complexité des algorithmes. La complexité d'un algorithme (notée α) est une indication du coût théorique « temps CPU » que l'on aura à considérer pour exécuter l'algorithme. Il est globalement calculé sur le nombre d'opérations à effectuer en fonction du nombre d'éléments. Pour exemple, le parcours d'une liste de complexité linéaire de *n* éléments sera noté $\alpha(n)$.

Nous n'avons pas, volontairement abordé dans notre travail, cet aspect et cela pour plusieurs raisons :

- l'expérience montre que, plus que la complexité des algorithmes, ce sont les technologies qui influencent les temps de traitement. Ainsi, par exemple, la mise en œuvre de bases de données permet, par l'utilisation d'index, de transformer des complexités linéaires de lecture de liste en complexités quasi constantes ;
- les algorithmes locaux qui possèdent de très hauts coefficients de complexité sont facilement parallèlisables et cet aspect n'apparait pas dans le calcul du coefficient de complexité. Le parallélisme étant aujourd'hui une technologie native sur tous les ordinateurs, il semble que le coefficient de complexité doive évoluer pour le prendre en compte ;
- les temps de traitement ne nous apparaissent pas comme une des caractéristiques à prendre en compte prioritairement tant que les communautés n'ont pas été validées qualitativement. De plus, la validation qualitative d'une méthode apporte par nature des éléments de modification. Il semble donc inapproprié de vouloir comparer la complexité relative des méthodes avant que celles-ci ne soient stabilisées ;
- les coefficients de complexité sont parfois présentés de manière avantageuse par les créateurs de méthodes sans que l'on puisse véritablement les vérifier.

Une fois que les communautés seront déterminées valides, il sera alors temps de rechercher l'algorithme le moins coûteux possible et cela en fonction des technologies présentes.

2.5 Synthèse

Cette synthèse liste en deux tableaux les méthodes présentées dans ce chapitre (cf. tableau 2.1 et tableau 2.2).

2.5.1 Caractéristiques importantes

Pour chaque méthode nous précisons si elle est applicable sur des graphes pondérés et orientés ou non et si le nombre de communautés doit être fixé à priori. Ces trois propriétés nous semblent primordiales, nous allons ici en justifier le choix par des exemples appliqués à un graphe de terrain bien connu, la ville de Kaliningrad. Nos communautés seront des arrondissements de la ville. Le but premier de ce découpage est que la circulation des personnes soit la plus simple possible au sein de chaque arrondissement. Pour des raisons de coût la mairie actuelle ne souhaite pas créer plus de deux arrondissements.

Après une étude, la mairie décide que le premier arrondissement sera constitué par la rive A et l'île B (cf. figure 2.17). Parfaitement connectés par deux ponts il semble que ces points de terre ferme puissent être rattachés. Le deuxième arrondissement est lui aussi parfaitement connecté, il semble simple de se déplacer entre la rive D à l'île C.

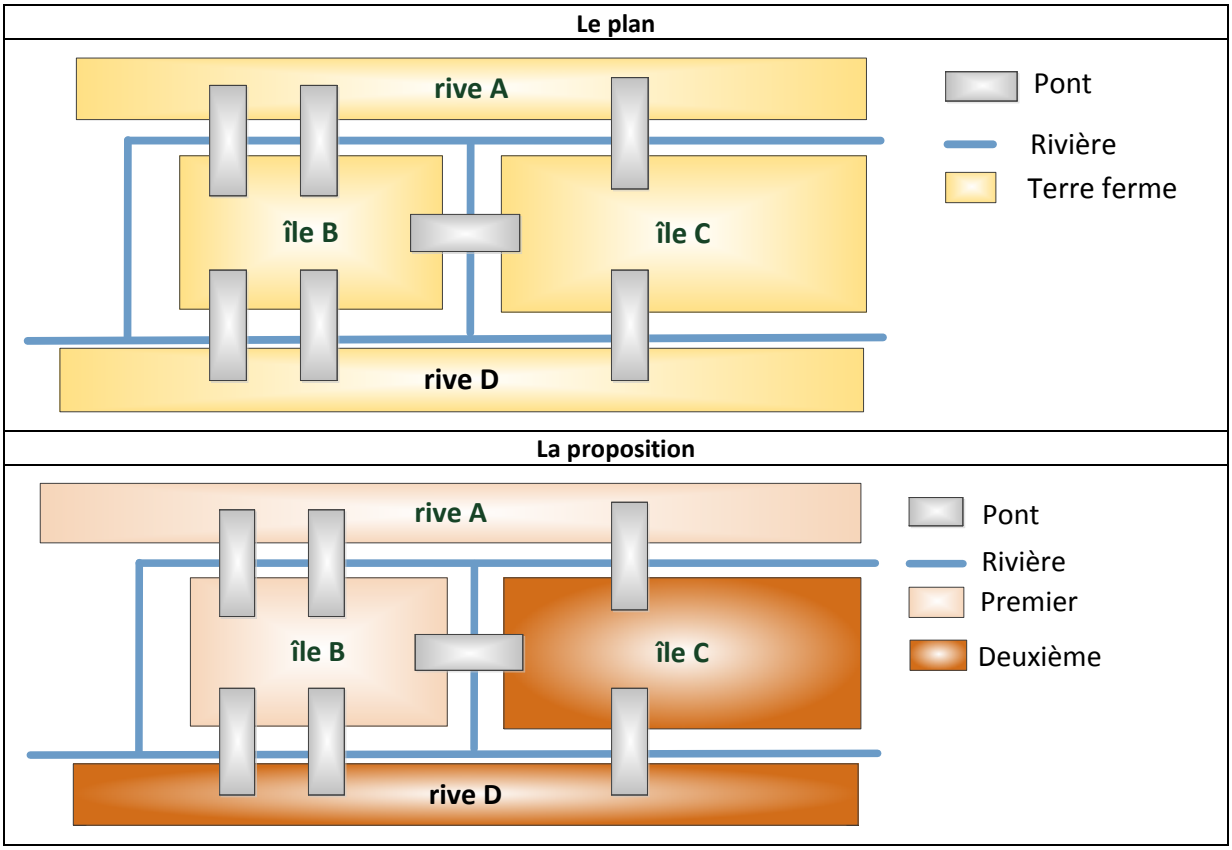


Figure 2.17 : Proposition pour créer deux arrondissements dans la ville.

Graphe Orienté

Kaliningrad a changé. Certains ponts sont maintenant à sens unique (cf. figure 2.18).

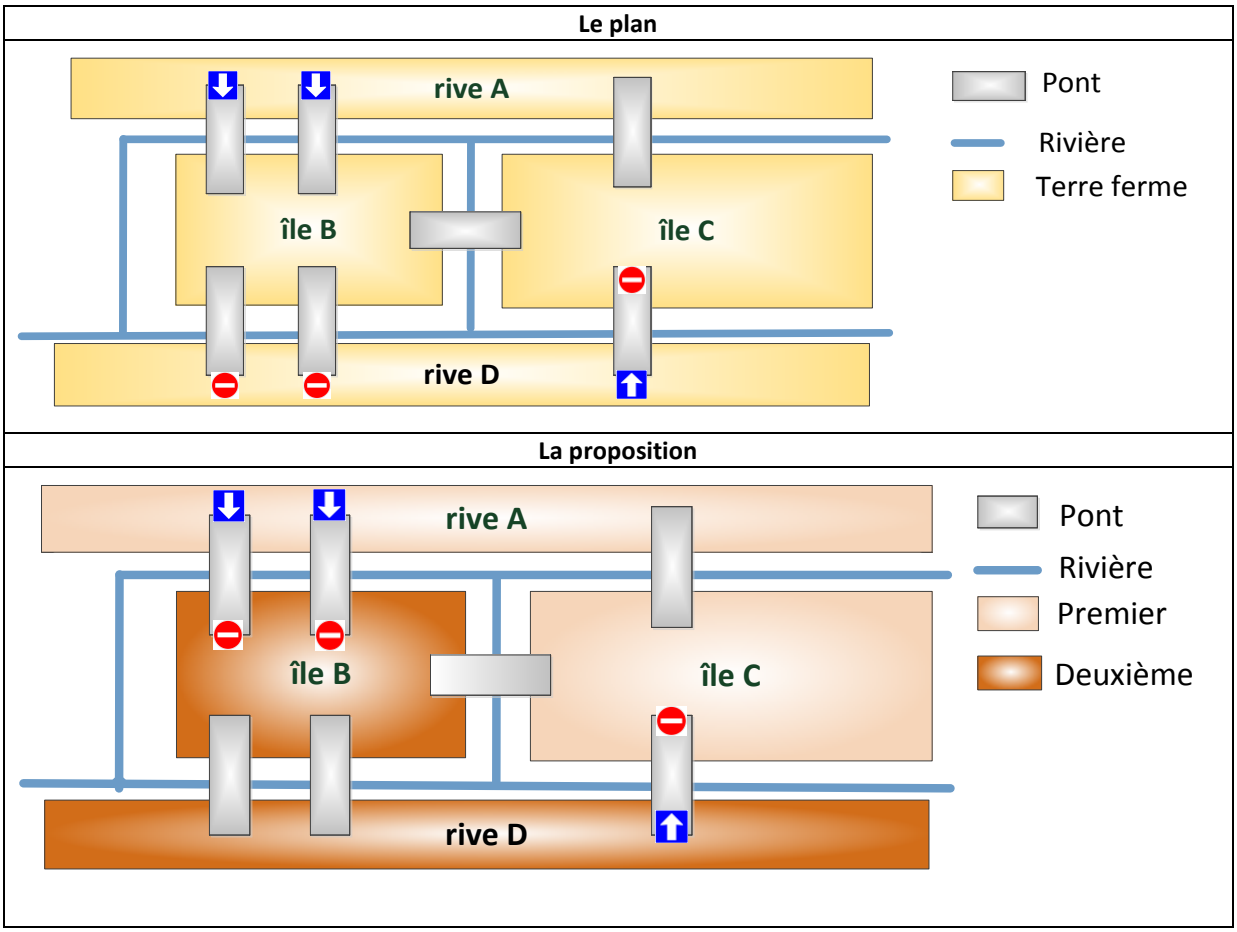


Figure 2.18 : Proposition pour créer trois arrondissements dans la ville en tenant compte des sens interdits.

Il est impossible, à cause des sens interdits, de rejoindre en voiture la rive A depuis l'île B et depuis l'île C de rejoindre directement la rive D. Utiliser un graphe orienté permettra de mettre en évidence cette caractéristique et le découpage en arrondissement proposé dans la figure 2.17 n'est plus adapté.

Comme nous le voyons ici prendre en compte l'orientation des liaisons permet d'obtenir des communautés ou arrondissements plus adaptés.

Graphe pondéré et nombre de communautés préétabli

Le problème posé est maintenant identique au problème précédent, mais certains ponts très anciens sont réservés, à un nombre de véhicules restreint (véhicules de moins de 1.5 tonnes et de moins de 1m70 de haut, ce qui correspond à environ 50% du trafic) (cf figure 2.19). De plus, ces ponts sont fermés la nuit ce qui correspond à une perte supplémentaire de trafic. Globalement ces ponts ont un trafic automobile inférieur de 60% à ceux qui ne sont pas soumis à ces restrictions. De plus, pour des raisons de sécurité certains ponts sont aussi interdits aux piétons car les trottoirs sont trop étroits.

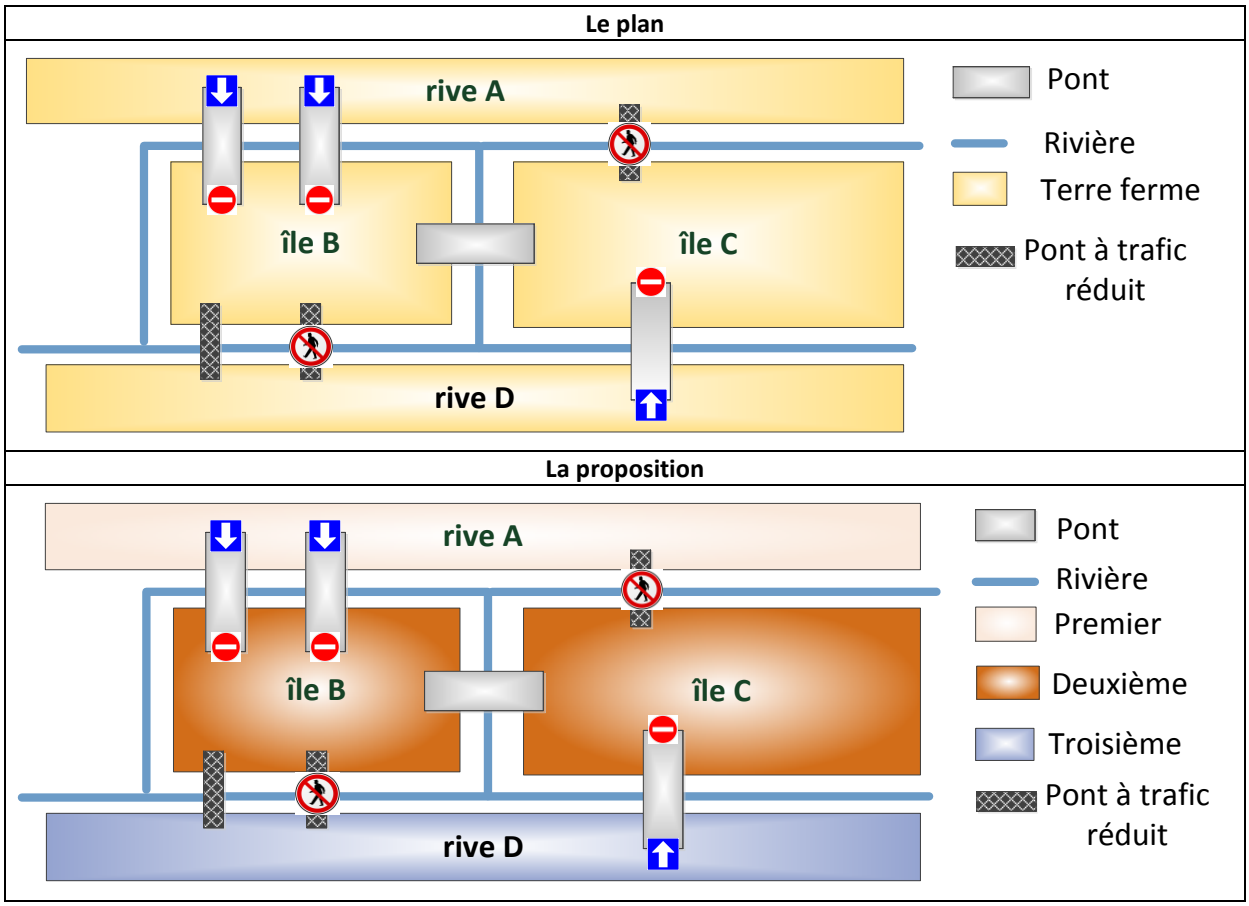


Figure 2.19 : Proposition pour créer trois arrondissements dans la ville en tenant compte des sens interdits et de la pondération des ponts par rapport au trafic automobile et piéton.

La prise en compte des différentes pondérations pour le trafic des automobiles et des piétons sur les ponts de ville, nous conduit à proposer une autre alternative. En effet, la priorité première étant de limiter le trafic inter-arrondissement, la création de deux arrondissements ne peut se faire sans une obligation de déplacement complexe. La création de trois arrondissements permet au contraire de profiter pleinement de la liaison bidirectionnelle, non limitée et ouverte aux piétons que constitue le pont entre les deux îles.

Cet exemple illustre l'importance que peuvent avoir les pondérations de liens, ainsi que la limitation de qualité qui peut être apportée par un nombre de communautés préétabli.

2.5.2 Méthodes créant des communautés sans recouvrement

Méthodes	Ref	Famille	Graphe		Nb de communautés	Résumé
Min-cut	**[Kernigha&al-1970]**	Algorithmes séparatistes	~~orienté~~	**non orienté**	prédéterminé	Basé sur le rapport du nombre de liaisons intercommunautaire sur le nombre intra-communautaire.
			~~pondéré~~	**non pondéré**		
Partitionnement par groupe de voisinage	**[Jain&al-1999]**	Algorithmes séparatistes	~~orienté~~	**non orienté**	prédéterminé	Transformation des liaisons en système de localisation dans un espace euclidien, puis découpage par zones de fort voisinage.
			pondéré	**non pondéré**		
Découpage de dendrogramme	**[Ward-1963]**	Algorithmes séparatistes	~~orienté~~	**non orienté**	Choisi en fonction du critère de qualité sélectionné (centralité)	Création d'un dendrogramme, puis création de communautés (chaque sommet de l'arbre est un départ d'une communauté). Les communautés peuvent ensuite être couplées pour en diminuer le nombre.
			~~pondéré~~	**non pondéré**		
Balade aléatoire	**[Pons&al-2005]**	Algorithmes séparatistes	**orienté**	**non orienté**	Choisi en fonction du critère de qualité sélectionné (modularité)	Algorithme qui représente le graphe comme un espace à parcourir. Plus la balade est statistiquement probable, puis l'espace est potentiellement une communauté.
			pondéré	**non pondéré**		
Heuristic procedure	**[Newman-2004-3]**	Algorithmes de scission	~~orienté~~	**non orienté**	Déterminé par le nombre de scissions.	Algorithme qui supprime des liaisons faibles de façon à transformer chaque composante connexe résultante en communauté. Les liaisons faibles sont par exemple des liaisons à forte centralité.
			pondéré	**non pondéré**		
recherche de zones de forte modularité	**[Rossia&al-2010]**	Détection de zones	~~orienté~~	**non orienté**	Déterminé par la valeur minimale de la modularité	Algorithme qui détermine les zones denses comme des communautés. Cet algorithme a la particularité de ne pas placer tous les nœuds dans une communauté.
			pondéré	**non pondéré**		

Tableau 2.1 : Méthodes créant des communautés sans recouvrement.

2.5.3 Méthodes créant des communautés avec recouvrement

Méthodes	Ref	Famille	Graphe	Nb de communautés	Résumé
C-finder	**[Palla&al-2005]**	Recherche de forme : Percolation de clique	~~orienté~~ **non orienté** ~~pondéré~~ **non pondéré**	Déterminé par le choix d'un coefficient K	Algorithme qui recherche des formes identifiables. Un nœud peut ou pas appartenir à une ou plusieurs communautés. Les nœuds ne sont pas tous rattachés à une communauté. Cet algorithme est devenu un des plus utilisés. Particulièrement efficace dans les réseaux ou le degré maximal est limité, il est plus délicat à mettre en œuvre dans de grands graphes de terrain ou le degré est très hétérogène.
Enrichissement de noyaux	**[Shang&al-2007]**	Méthode en X phases	~~orienté~~ **non orienté** ~~pondéré~~ **non pondéré**	Déterminé par le choix d'un coefficient K (phase 1) et par les règles de regroupement des noyaux (phase 3)	Algorithme qui recherche des formes identifiables identiques au C-finder, puis qui regroupe les noyaux créés et les enrichit ensuite avec les nœuds excentrés. Une grande majorité des nœuds peuvent ainsi être dans une communauté.
RaRe/IS et LA / IS^2	**[Baumes&al-2005-1]**	Méthode en X phases	~~orienté~~ **non orienté** ~~pondéré~~ **non pondéré**	Déterminé par la méthode de création des noyaux (RaRe ou LA)	Les nœuds présentant les PageRank les plus élevés sont supprimés de façon à créer des composantes connexes. Ces composantes connexes sont les communautés. Elles sont ensuite enrichies par le retour des nœuds qui sont en recouvrement.
OCA	**[Padrol-Sureda&al-2010]**	Méthode en X phases	~~orienté~~ **non orienté** ~~pondéré~~ **non pondéré**	Déterminé par le nombre de graines au départ	Cet algorithme se veut avant tout un algorithme efficace et rapide. Il nécessite une phase de post-traitement qui va regrouper les communautés trop proches.
Analyse spectrale et Fuzzy c-means	**[Zhanag&al-2007]**	Méthode en X phases	~~orienté~~ **non orienté** ~~pondéré~~ **non pondéré**	Majorant prédéterminé	Une fois les nœuds du graphe projetés dans un espace euclidien, l'algorithme fuzzy c-means est utilisé pour former des communautés dans cet espace géométrique.
Nash equilibra	**[Chen&al-2010]**	Déplacement de nœuds	~~orienté~~ **non orienté** ~~pondéré~~ **non pondéré**	Majorant prédéterminé	Un nœud se déplace d'une communauté à une autre en cherchant à maximiser son « utilité ». Un nœud utile augmente la modularité de la communauté qu'il rejoint.
Seed expansion	**[Vei&al-2010]**	Déplacement de graines	~~orienté~~ **non orienté** ~~pondéré~~ **non pondéré**	Prédéterminé	Cet algorithme calcule la probabilité qu'un nœud appartienne à une communauté de telle façon que plus son appartenance augmente la modularité plus sa probabilité d'appartenir à cette communauté est forte.
Particle Competition	**[Breve&al-2010]**	Déplacement de particules	**orienté** **non orienté** **pondéré** **non pondéré**	Prédéterminé	L'algorithme déplace des particules de façon semi aléatoire à l'intérieur du graphe. Chaque particule représente une communauté. À chaque déplacement les nœuds rejoints par la particule lui appartiennent un peu plus.
CONGA modifié	**[Gregory-2010]**	Méthode modifiée pour permettre le recouvrement	~~orienté~~ **non orienté** ~~pondéré~~ **non pondéré**	Déterminé par le nombre de scissions	Algorithme qui rajoute des liaisons virtuelles en « dédoublant » certains nœuds. Une fois la liaison portant la plus haute centralité trouvée, on peut la supprimer. Les composantes connexes créées sont alors des communautés, les nœuds dédoublés étant partagés par les communautés créées.
Stochastic Block Models modifié	**[Latouche&al-2010]**	Méthode modifiée pour permettre le recouvrement	orienté **non orienté** ~~pondéré~~ **non pondéré**	Prédéterminé	L'objectif est donc de trouver le jeu de paramètres (α,W) qui expliquent, au mieux, la présence ou l'absence d'arêtes (connexions) dans le réseau.

Tableau 2.2 : Méthodes créant des communautés avec recouvrement.

2.5.4 Conclusion

La création de communautés dans les graphes grands, petits, de terrains ou mathématiques est un axe de recherche qui est en pleine expansion. Le graphe est sans aucun doute un objet de simplification. Mais doit-il être simpliste pour autant ? Le problème initiateur des ponts de Kaliningrad n'est-il pas résolu en pondérant chaque nœud par son degré ?

Nous recherchons alors une méthode :

- permettant le recouvrement, puisque pour une même orthographe, un mot peut avoir plusieurs sens et donc appartenir à plusieurs communautés ;
- utilisant la pondération (le traitement des mots devra sans aucun doute être différent selon l'usage et une liaison représentant une co-utilisation unique pour des mots utilisés très intensément ne devra sans doute pas recevoir la même attention qu'une co-utilisation plus courante) ;
- respectant la possibilité d'orienter le graphe pour pouvoir encore affiner les proportionnalités.

Il n'est pas non plus concevable de prédire le nombre de communautés. Dans un graphe de plusieurs millions de mots, chaque communauté devra trouver et conserver ses propres caractéristiques pour conserver une qualité sémantique.

Nous n'avons finalement pas trouvé dans les méthodes proposées, celle qui nous permettrait de couvrir tous nos besoins. Nous avons donc entamé une recherche sur la création de méthodes spécifiques. C'est de ces méthodes, de leur mise en œuvre et de leur validation dont nous parlerons dans la deuxième partie de ce mémoire.

Avant de conclure ce chapitre, il convient de préciser que nous avons volontairement omis une dimension de l'étude de la création de communautés dans les graphes : le temps. En effet, un graphe est un élément vivant, nœuds et liaisons ne sont pas tous créés simultanément. Les graphes que nous étudions peuvent être comparés à des photographies présentant un instant figé.

Les travaux sur l'évolution des graphes et la dynamique de la création de communautés n'en sont qu'à leurs débuts. Nous pouvons citer ici l'article *« Detection of overlapping communities in dynamical social networks »* de Cazabet Rémy, Amblard Frédéric et Hanachi Chihab **[Cabazet&al-2010]**, dans lequel, les auteurs utilisent la date de création des liaisons comme élément référent dans la détection des communautés.

2.6 Conclusion

Depuis quelques années, la création de communautés ou d'agrégats avec recouvrements est devenue un véritable objet de recherche. C'est la conséquence de l'appropriation par des chercheurs de la théorie des graphes et de l'évolution des outils informatiques associés. Pour ces « hommes de terrain », il est souvent évident que les communautés sont sur le terrain avec « recouvrements ». Les travaux de Gergely Palla en sont une illustration dans le domaine de la biologie **[Palla&al-2005]**. Ces travaux sont le plus souvent testés sur de petits graphes de test. Il n'est, alors, pas possible d'affirmer que les algorithmes présentés pour créer des communautés restent valides sur d'autres types d'objets et sur des graphes de taille beaucoup plus importante.

La différence principale, outre la taille, entre ces graphes de test d'un domaine bien particulier et les grands graphes de terrain tels que nous les avons définis est sans aucun doute le critère de dispersion du degré des nœuds. Dans certains domaines comme ceux étudié par G. Palla (Biologie, Chimie,…) **[Palla&al-2005]**, les nœuds ont par nature un nombre limité de connexions. Il en sera de même pour les rencontres sportives entre clubs d'une région, sur une saison et les co-écritures d'articles entre chercheurs dans un espace-temps et/ou domaine de recherche circonscrit. Si l'expérimentation porte sur un Grand Graphe de Terrain, le type des objets le constituant est le plus souvent choisi pour que le graphe possède un critère de faible dispersion des degrés **[Padrol-Sureda&al-2010]**.

Ces travaux sur le « terrain », conduisent aussi les mathématiciens à considérer le domaine. Mais la recherche est ici plus théorique. Les modèles sont ensuite validés généralement soit sur des graphes jouets, soit sur des graphes créés informatiquement, dont la ressemblance avec les grands graphes de terrain n'est pas prouvée. De plus, dans une recherche théorique rien n'empêche de poser comme paramètre le nombre de communautés à créer. Ceci n'est pas toujours possible de manière précise sur le terrain. Dans ce cas la grande majorité des méthodes proposées ne sont alors pas utilisables.

Dans la « vraie vie », les réseaux sociaux ou réseaux de relations entre acteurs ne possèdent pas le plus souvent, de limite à la valeur de degré. La limite théorique pour des échanges par e-mail serait par exemple de 20% de la population mondiale (population connectée en 2010) et bien plus pour des échanges par téléphone (puisqu'Internet peut être utilisé pour les échanges téléphoniques). Il existe bien des exceptions : sur Facebook le nombre d'« amis » est limité à 5000 or Dumbar estime lui la limite des réseaux d'amis d'une personne à 148 (Nombre de Dumbar) **[Dumbar-1998]**. Cependant, remarquons que celle-ci n'est pas une limite de reconnaissance mais de confiance. Ainsi, cela pourrait être la limite de taille maximale donnée à une communauté ou le nombre maximal de communautés d' « amis » auxquelles un acteur appartient. Mais cela ne conditionne aucunement le nombre de personnes avec lesquelles il est en connexion. La limite donnée sur Facebook a pour but de protéger le système de la saturation et d'éviter de trop galvauder la notion d' « amis » en obligeant les utilisateurs à en limiter le nombre.

Dans les grands graphes de terrain, où il n'existe pas de limite au degré d'un nœud, certains nœuds portent des valeurs de degrés extrêmes. Il est, de plus, impossible ou très difficile de prédéterminer un nombre de communautés optimal à créer. C'est un réseau de ce type que nous nous proposons d'étudier dans la deuxième partie de ce travail.

Deuxième partie.

Nos propositions pour la création d'agrégats par rigidification et enrichissement

Cette seconde partie présente notre propre contribution à la recherche sur les méthodes de regroupement et de validation : la création d'agrégats par rigidification et enrichissement. Les recherches présentées dans le chapitre précédent bien que très nombreuses et en pleine évolution, n'offrent pas de solution adaptée au contexte précis des grands graphes de terrain de mots. En effet, même le procédé C-Finder **[Palla&al-2005]** qui permet d'avoir des recouvrement sans avoir à prédéterminer le nombre de regroupements n'est pas, de toute façon, adapté aux réseaux de grande taille qui présentent des zones de faible densité et d'autres de très forte densité.

Il était donc nécessaire d'imaginer des solutions spécifiques. Les méthodes que nous avons conçues - la rigidification, la Rigidification Régulée et l'enrichissement d'agrégats-sont précisément adaptées à la création d'agrégats de mots sémantiquement cohérents dans les grands graphes de terrain car elles respectent les règles suivantes :

- ne pas prédéfinir le nombre d'agrégats à créer ;
- n'étudier que des kilo-graphes ou mega-graphes (nous nous situons volontairement uniquement dans l'espace des grands graphes) ;
- n'étudier que des graphes de mots utilisés dans des requêtes d'utilisateurs. Nous n'avons pas la prétention de proposer des méthodes de regroupement efficientes, quel que soit la nature du réseau. La raison pour laquelle nous nous sommes concentrés sur un seul type de réseau (les réseaux de mots issus de requêtes) est que ce travail se positionne comme la première brique d'un système utilisant des agrégats de mots tel que présenté dans l'avant-propos ;

- cibler des agrégats ayant une forte cohérence sémantique, ce qui est le critère exclusif de qualité (celui des méthodes non spécifiques aux réseaux de mots est généralement la modularité ou d'autres caractéristiques ayant servi à créer l'agrégat).

Cette seconde partie est organisée en deux chapitres :

- Le chapitre 3 dans lequel nous présentons l'ensemble des méthodes d'agrégation que nous avons mises au point :
  - Détection de Cliques ;
  - Rigidification Simple ;
  - Rigidification Régulée ;
  - Enrichissement par Gravité.

- Le chapitre 4 où nous validons la valeur sémantique des agrégats en utilisant pour cela plusieurs procédés :
  - Un procédé que nous avons inventé :
    - Validation par comparaison de comportement de requête.
  - trois procédés que nous avons adaptés :
    - Validation par comparaison de qualité de requête ;
    - Validation par comparaison de distance entre documents retournés ;
    - Validation manuelle.

Notre travail porte donc autant sur la recherche d'une méthode d'agrégation respectant une cohérence sémantique que sur un système de mesure de cette cohérence.

Chapitre 3.

Les méthodes d'agrégations proposées

3.1 Introduction

Nous présentons dans ce chapitre quatre méthodes de création ou d'enrichissement d'agrégats, dont trois que nous avons créées. Leur avantage est qu'elles offrent toutes la possibilité de créer des agrégats avec recouvrements sans qu'il soit nécessaire de prédéfinir le nombre d'agrégats à priori. Elles sont présentées par ordre « chronologique » de conception. Chaque méthode proposée est en fait une évolution de la ou des précédentes. Pour expliquer et justifier ces modifications, il nous est apparu intéressant d'en décrire la source et les mécanismes.

Les quatre méthodes proposées sont :

- L'agrégation par détection de cliques, nommée « **Détection de Cliques** ». Cette méthode est avant tout un moyen d'évaluation de la difficulté du travail et d'apprentissage du réseau à traiter. Elle peut être considérée comme une implantation extrêmement simplifiée, de la méthode de C-Finder **[Palla&al-2005]**.
- La rigidification, méthode que nous avons créée et nommée « **Rigidification Simple** ». Cette nouvelle méthode est basée sur des règles locales. Elle a pour but dans un réseau particulièrement pollué par des liaisons de validités diverses de proposer un tri entre liaisons à écarter et liaisons à conserver. Elle est aussi une phase d'apprentissage sur le réseau, la nature et la qualité des agrégats que l'on peut espérer créer. Cette méthode issue d'une théorie mathématique de

G.C.S.P. (**G**eometric **C**onstraint **S**atisfaction **P**roblem) est à notre connaissance la première implantation de cette théorie mathématique **[Belbeze&al-2009-3]**.

- La rigidification avec régulation de taille des agrégats, méthode que nous avons créée et nommée « **Rigidification Régulée** ». Nous avons notablement amélioré la méthode précédente « **Rigidification Simple** » sur plusieurs points. Se fondant toujours sur la même théorie mathématique de GCSP, elle permet de conserver l'ensemble des nœuds dans le graphe à étudier préalablement à la création d'agrégats et d'améliorer la qualité des agrégats créés **[Belbeze&al-2009-1]**.
- L'enrichissement des agrégats, méthode que nous avons créée et nommée « **Enrichissement par Gravité** ». Cette méthode permet de rajouter aux agrégats connus des nœuds périphériques. Contrairement aux méthodes présentées dans la partie 1 comme « méthodes en plusieurs phases » **[Shang&al-2007] [Baumes&al-2005-2],** celle qui est présentée ici ne cherche pas à étendre ou créer des parties en recouvrement. Cet algorithme a pour but de rattacher de manière pondérée à un ou plusieurs agrégat(s) des nœuds isolés. De plus, l'enrichissement par gravité n'est pas une simple phase participant d'une méthode plus générale, mais une méthode à part entière. Ainsi, elle sera validée indépendamment des méthodes par regroupement **[Belbeze&al-2009-2]**.

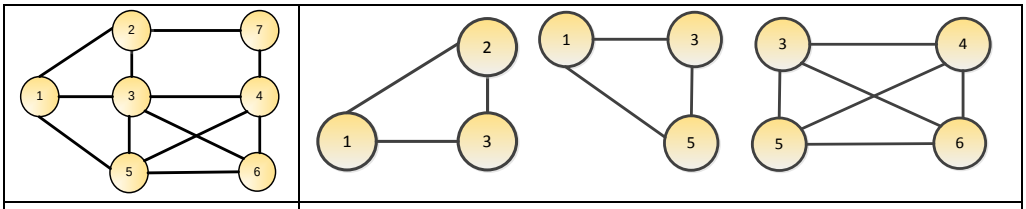
Ces méthodes ont toutes les quatre comme point commun qu'elles partent d'un nœud (ou d'un agrégat de nœuds) pour construire de manière agrégative l'ensemble recherché. En effet, elles ne travaillent pas sur l'ensemble du réseau, elles considèrent un nœud et ses voisins comme espace de première exploration. Compte tenu de notre objectif qui est de créer des agrégats de mots sémantiquement cohérents, la démarche ne nous semble pas pouvoir être séparatiste ou globale. Au contraire, l'ajout ou la suppression d'un mot pouvant faire évoluer fortement la cohérence sémantique d'un agrégat, la méthode se doit d'être une méthode d'agrégation contextuelle. Le contexte est défini par les mots déjà présents dans l'agrégat et les mots susceptibles d'être rajoutés. Les méthodes présentées ne sont donc ni séparatistes ni globales. De plus, les méthodes séparatistes sont souvent déterministes dans le nombre d'agrégats à créer. Compte tenu de la nature des réseaux et des agrégats recherchés, la prédétermination de cette valeur ne peut se faire sur de véritables fondements sémantiques.

3.2 Méthode 1 : Détection de cliques

3.2.1 La clique ou une densité maximale

Quand on cherche à former des ensembles de termes les plus cohérents possibles, le modèle de la « clique » (cf. chapitre 1) s'impose de lui-même. Il est en effet porteur de la densité maximale possible. Dans notre espace de travail, l'appartenance à une clique pour un mot-clé signifie qu'il a été employé dans au moins une recherche avec chacun des autres mots de la clique et qu'il n'existe pas d'autres mots-clés dans la population étudiée (suivant cette règle) qui ne soit pas placé dans la clique.

On peut donc penser que par le lien présent entre chacun des éléments, ce modèle garantira une forte cohérence sémantique au sein des agrégats.



a : Graphe *G*	b : Les cliques du graphe *G* : {1, 2, 3}, {1, 3, 5}, {3, 4, 5, 6}

Figure 3.1 : Un graphe et ses cliques.

A la lecture du graphe *G* représenté dans la figure 3.1a on constate que l'élément *diade {1, 2}* est un couple de mots-clés utilisés conjointement (au moins une fois) dans une recherche par un internaute. Il en est de même pour la *diade {1, 3} et la diade {2, 3}*. La *triade {1, 2, 3}* (cf. figure 3.1b) représente donc une clique de trois mots-clés, telle que chaque mot-clé a été utilisé au moins dans une requête (mais pas nécessairement la même) avec chacun des deux autres. La *clique {3, 4, 5, 6}* signifie de même que chaque mot-clé a été utilisé au moins dans une requête (mais pas nécessairement la même) avec chacun des trois autres.

3.2.2 Mécanisme de regroupement des mots-clés en cliques

Par définition, dans une population, une triade ne peut appartenir qu'à une seule clique et une clique complète contient tous les éléments de la population concernée.

Cette caractéristique nous permet, une fois toutes les cliques issues d'un mot-clé de départ *(X)* et de ses *diades (X-Y)* créées, de ne plus avoir à le considérer comme pouvant appartenir à de nouvelles cliques.

Pour chaque mot–clé X classé par MCID* Ascendant **faire**
    Récupérer les mots-clés Y en diade avec X s'il existe Z en triade avec X et Y si Y.MCID est supérieur à X.MCID
    Pour chaque diade X – Y **faire**
        Initialisation clique-en-cours
        Ajouter-a-clique (clique-en-cours, X, Y)
        Appel **Recherche-de-clique** (clique-en-cours)
    Fin Pour chaque diade X – Y
Fin Pour de chaque mot-clé X

Sub Recherche-de-clique (clique-en-cours)
    Ordonner la clique-en-cours par MCID croissant
    Si il existe des mots en corrélation avec tous les mots contenus dans clique-en-cours ordonnés par valeur de MCID et que la valeur de MCID est supérieure à celle du dernier mot **alors**
        Appel **Recherche-de-clique** (clique-en-cours)
    Sinon
        Sauvegarde de la clique
    Fin de SI
Fin de Recherche-de-clique

Algorithme 3.1 : Création de cliques (*« X.MCID » détermine l'identifiant associé au mot-clé X).

En ordonnant les mots-clés par leur identifiant (MCID) nous pouvons ainsi, à l'aide d'une fonction récursive, limiter l'exploration de la population aux mots-clés possédant un identifiant (MCID) supérieur à celui du mot-clé en test (cf. Algorithme 1). Afin d'apprécier l'avantage de cet algorithme, nous proposons d'en mesurer le gain par rapport à une exploration classique.

Considérons un ensemble *E* de mots-clés (dans un fichier de traçage) contenant 10 éléments : (*card(E)* = 10). Si nous dénombrons les tests nécessaires à la validation de cliques en testant toutes les combinaisons et en considérant les suites *(a,b,c)* et *(c,a,b)* comme différentes, nous déterminons alors que le nombre de suites à 3 éléments dans *E* est égal à :

$$A_{10}^{3} = 10 \times 9 \times 8 = 720$$

Dans l'algorithme présenté nous avons décidé de classer les éléments (les mots-clés) selon leur identifiant **MCID**, et de ne considérer comme candidat à la création d'une triade que des mots-clés ayant un identifiant supérieur à celui du mot-clé de départ. Nous établissons ensuite une relation d'ordre, permettant de ne plus revenir sur les ensembles ayant déjà été considérés. Nous pouvons donc utiliser ici des sous-ensembles en lieu et place des suites proposées par un algorithme basique effectuant un balayage total de la matrice. Dans notre exemple, le nombre de sous-ensembles à trois éléments est égal à :

$$C_{10}^{3} = \frac{A_{10}^{3}}{3!} = 720/6 = 120$$

Cela signifie que nous n'aurions que 120 combinaisons à tester au lieu de 720 pour des cliques de 3 éléments.

Dans un ensemble *X* de *n* éléments ($n \succ 10$) dans lequel on formera des cliques ayant jusqu'à 10 éléments, le nombre de recherches sera divisé alors par 10 ! = 3 628 800.

Nous avons recherché ici un modèle de regroupement simple qui devait nous servir de base d'étude. Intuitivement ce modèle semble être en capacité de créer des agrégats sémantiquement cohérents.

3.3 Méthode 2 : Rigidification Simple

La recherche d'une méthode d'agrégation souple paramétrable et tenant compte de la proportionnalité des cooccurrences par rapport à l'usage de mots nous a emmenés à nous intéresser aux travaux de la communauté mathématique sur les ensembles d'objets rigides.

3.3.1 Définition des problèmes de satisfaction de contraintes géométriques G.C.S.P (Geometric Constraint Satisfaction Problem)

Hoffman définit **[Hoffman&al-2005]** le problème de contraintes géométriques au moyen d'un tuple (E, O, X, C) où

- E est l'espace géométrique constituant un cadre de référence dans lequel le problème est défini,
- O est l'ensemble des spécifications géométriques des objets constituant le problème,
- X est un ensemble, éventuellement vide, de variables qui représentent des caractéristiques géométriques: distances, angles et ainsi de suite.
- C est l'ensemble des contraintes. Les contraintes peuvent être géométriques ou équationnelles.

Les contraintes géométriques sont les relations entre les éléments géométriques choisis parmi un ensemble prédéterminé, par exemple, la distance, l'angle, la tangence, etc.

Dans notre étude, E représente le graphe étudié ; O est constitué par la définition des liaisons ; X est l'ensemble des figures que nous considérons comme figures de référence soit : diades et triades ; C est l'ensemble des contraintes équationnelles qui vont nous permettre de conserver ou supprimer les liaisons et les contraintes géométriques qui vont nous permettre de constituer les agrégats.

Résoudre un problème de satisfaction de contraintes géométriques consiste à utiliser une méthode de résolution pour le système G.C.S.P défini.

3.3.2 Présentation de HLS

Constituée d'un ensemble de phases paramétrables, la méthode de rigidification d'Hoffmann, Lomonosov et Sitharam, nommée HLS [Hoffman&al-1997] est très souple. Ce paramétrage permet de supprimer ou au contraire de conserver des liens entre des mots-clés selon des critères variables. Dans la méthode de rigidification simple, le choix du maintien de la liaison se fait en fonction de pondérations (nombre de co-utilisations) de cette liaison et relativement au poids (nombre d'utilisations) de chaque mot relié. Cela permet de choisir une condition d'agrégation plus efficace que celle utilisée pour le regroupement des mots-clés en cliques.

Cette méthode, proposée initialement en 1997 par Hoffman, Sitharam et Lemonosov est une méthode de décomposition structurelle ascendante. Elle recherche des ensembles d'objets rigides. Ces agrégats sont ensuite assemblés récursivement. Il s'agit d'une des méthodes de rigidification récursives de G.C.S.P. (Geometric Constraint Satisfaction Problem).

Hoffman et son équipe ont toujours proposé des descriptions de leurs méthodes en termes de graphes et de transformation de graphes. Nous présenterons ici un court résumé de cette méthode qui, par une succession de phases, va permettre la création d'un agrégat à partir d'un noyau de départ par ajouts successifs de nœuds.

3.3.3 Les étapes de la méthode HLS

La méthode HLS comprend 2 étapes :

L'analyse

Une première phase d'analyse consiste à regrouper les agrégats tant que des regroupements sont possibles. La phase d'analyse se compose de trois parties : fusion, extension et condensation

- **la fusion** recherche les agrégats de taille minimale ;
- **l'extension** consiste à inclure un objet voisin dans l'agrégat courant et ce, tant qu'il existe un objet voisin à insérer (l'opération d'extension utilise un opérateur d'extension qui va étendre l'agrégat courant « *A* » aux objets voisins ; l'opérateur d'extension est défini comme une condition et doit être adapté en fonction des éléments à agréger) ;
- **la condensation** place les objets regroupés dans l'agrégat en cours de constitution et met à jour le plan d'assemblage, qui est présenté dans la section suivante.

L'assemblage

La phase d'assemblage exécute un plan d'assemblage où chaque agrégat est considéré comme un objet de départ. Nous n'utilisons pas cette phase dans nos travaux afin de mieux mesurer la qualité de la phase d'extension. Les travaux de Jermann C. présentent plusieurs mises en œuvre possibles **[Jermann&al-2004] [Jermann-2002]** de la phase d'assemblage. L'Enrichissement par Gravité présenté en paragraphe 3.20, est une instanciation possible de cette phase.

3.3.4 Implantation et adaptation de la méthode HLS

Définissons ***S*** un G.C.S.P. tel que ***S=(MC,R)*** où ***MC*** est l'ensemble nœuds et ***R*** l'ensemble des relations et où $\boldsymbol{A=(MC_A, R_A)}$. ***A*** est un agrégat dans le G.C.S.P., le voisinage de ***A*** est le sous ensemble ***MC'*** des objets géométriques de ***S*** qui sont liés par des relations à des objets de *A* par l'opérateur d'extension **O.**

Dans notre instanciation, nous définissons l'agrégat minimum comme une clique. La phase de **fusion** recherche donc ces objets.

La phase d'**extension** est déterminée par la connaissance du voisinage de l'agrégat de départ et par la capacité à étendre cet agrégat. L'opération d'extension utilise un **opérateur**

d'extension *O* obéissant à la règle suivante : le graphe de l'agrégat doit toujours rester **bi-connexe** pendant les opérations d'extension. La figure 3.2 donne un exemple du déroulement de la phase d'extension.

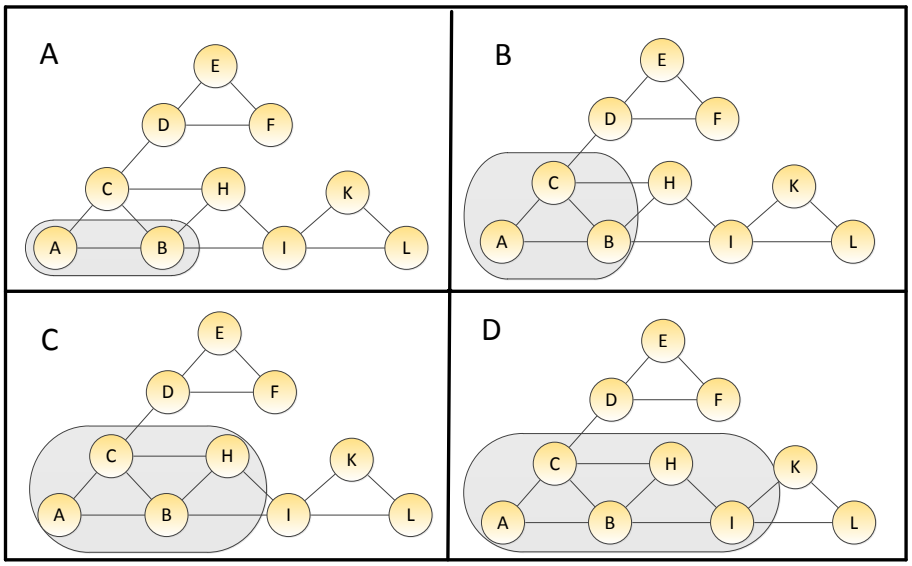


Figure 3.2. Illustration du déroulement de l'algorithme Fusion/Extension dans notre implantation de H.L.S.

D'autres critères interviennent dans l'opération d'extension, tels que le poids des mots et des relations.

Nous nommons *poids*, le nombre de recherches liées à un objet. Cet objet est soit un mot-clé soit une relation ***R*** inter mots-clés. Le poids d'un mot-clé est le nombre de requêtes incluant ce mot-clé. Le poids $\boldsymbol{PR_{AB}}$ d'une relation $\boldsymbol{R_{AB}}$ entre un mot-clé **A** et un mot-clé **B** est égal au nombre de requêtes incluant conjointement les deux mots-clés **A** et **B**.

- **Le poids d'un mot-clé :** *Nb* étant le nombre total de requêtes, $MC_{T,Q}$ étant l'élément valant 1 si le mot-clé *T* est présent dans la requête *Q* et 0 sinon. On définira le poids d'un mot-clé A noté P_A comme suit :

$$P_A = \sum_{Q=1}^{Nb} M_{A,Q}$$

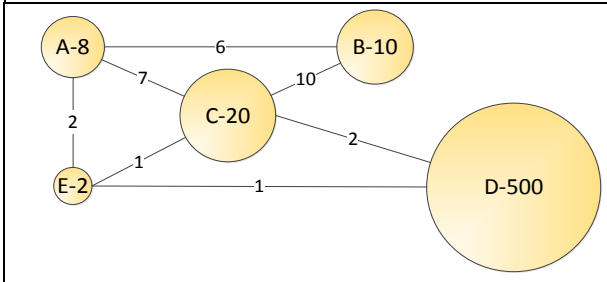
- **Le poids d'une relation :** Soient les deux mots-clés **A** et **B**, la relation $\boldsymbol{R_{AB}}$ telle que **A** $\boldsymbol{R_{AB}}$ **B**, ***Nb*** le nombre total de requêtes, ***Ri*** étant l'élément de valeur « vrai » ou « faux » si les mots-clés sont conjointement présents dans la requête (vrai valant 1, faux valant 0). On définira le poids d'une relation $\boldsymbol{R_{AB}}$ noté $\boldsymbol{PR_{AB}}$ comme suit :

$$PR_{AB} = \sum_{I=1}^{Nb} R_I$$

Remarque : Le poids total d'un mot-clé n'est pas nécessairement la somme des poids de ses relations. En effet, une même recherche peut inclure plusieurs mots-clés et donc

compter pour 1 dans le poids du mot-clé qui est en relation avec « n » mots-clés (cf. figure 3.3).

Graphe non orienté



Matrice symétrique des poids du graphe

Mot	Poids mot	A	B	C	D	E
A	8	-	6	7	0	2
B	10	6	-	10	0	0
C	20	7	10	-	2	1
D	500	0	0	2	-	1
E	2	2	0	1	1	-

Figure 3.3 : Graphe et Matrice non orientés.

Utilisation du poids pour l'orientation du graphe et l'amélioration de l'opérateur d'extension

Nous proposons de compléter l'opérateur d'extension par une prise en compte de la notion de poids relatif. Il semble évident que le poids de la relation est à comparer aux poids des mots-clés en relation. Une relation d'un poids de « 1 » entre un mot-clé A pesant « 1000 » et un mot-clé B pesant « 2 » ne représente pas du tout la même importance relative. Ainsi la relation pèse 10^{-3} du poids du mot-clé A et .5 du poids du mot-clé B. Afin de prendre en compte ce poids relatif, nous orientons et pondérons le graphe de la matrice présenté en figure 3.3. Nous utilisons pour ceci la valeur du poids du mot-clé de départ sur le poids de la relation du mot-clé de départ avec le mot-clé cible. On note ce rapport ***CFL*** ou **Coefficient de Fiabilité de Lien**.

Ainsi pour un mot-clé A en relation avec un mot-clé B noté **A R_{AB} B**, P_A le poids du mot-clé A, PR_{AB} le poids de la relation R_{AB}. On définit le Coefficient de Fiabilité de Lien du mot-clé A vers le mot-clé B noté $CFL_{A=>B}$ comme suit :

$$CFL_{A=>B} = P_A / PR_{AB}$$

La figure 3.4 présente le résultat de cette opération sur le graphe proposé en figure 3.3

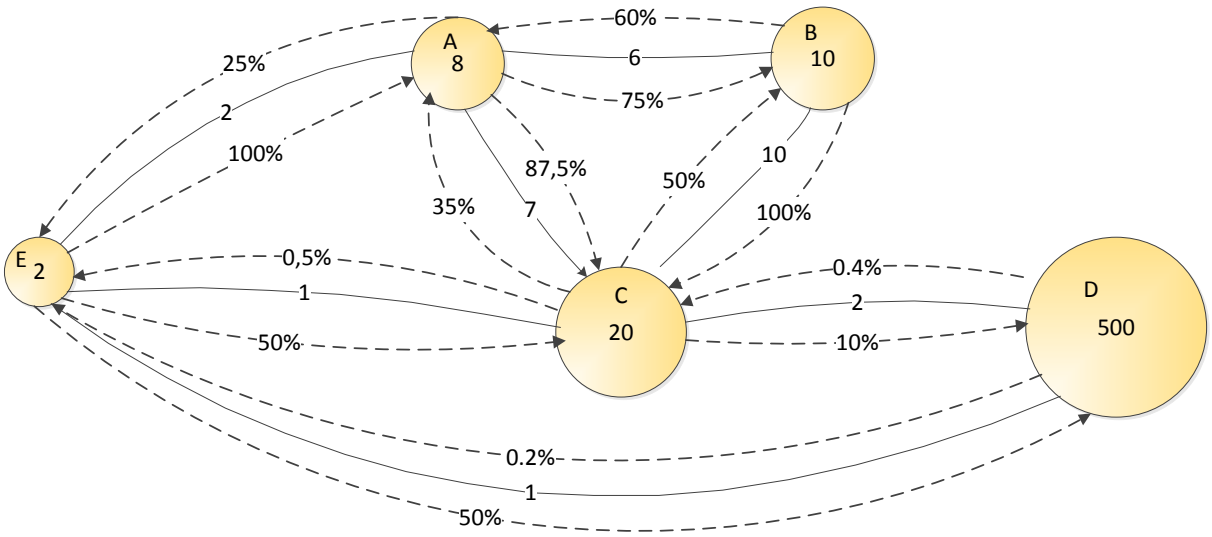


Figure 3.4 : Graphe orienté pondéré du CFL de la matrice présentée en figure 3.3. (CFL est ici présenté en pourcentage pour en faciliter la lecture).

Matrice symétrique - graphe non dirigé								**Matrice asymétrique - graphe dirigé - *CFL* (%)**						
Mot	Poids	A	B	C	D	E		Mot	Relation	A	B	C	D	E
A	8	-	6	7	0	2		A	->	-	75	87.5	0	25
B	10	6	-	10	0	0		B	->	60	-	100	0	0
C	20	7	10	-	2	1		C	->	35	50	-	10	5
D	500	0	0	2	-	1		D	->	0	0	0.4	-	0.25
E	2	2	0	1	1	-		E	->	100	0	50	50	-

Tableau 3.1. Matrice asymétrique d'un graphe orienté pondéré – CFL.

Définition d'un opérateur d'extension

L'utilisation de cet algorithme avec un opérateur d'extension qui ne tient pas compte de la valeur relative des liaisons a pour conséquence la création d'un agrégat massif de plusieurs milliers de mots-clés. Il paraît donc indispensable de définir des seuils de validité. Pour ne pas maintenir des liens présentant un *CFL* trop faible, nous ne prenons en compte que les relations présentant un *CFL* supérieur à une valeur nommée **Valeur Minimale** de *CFL* ou ***Val-Min-CFL***. De même, pour les mots de faible poids en relation avec des mots de poids fort, nous maintenons quel que soit le CFL de sens inverse toutes relations ayant un *CFL* supérieur à la valeur d'activation prédéterminée ou ***Val-Activ-CFL***. Dans cette méthode les valeurs *Val-Min-CFL* et *Val-Activ-CFL* sont définies arbitrairement après un ensemble d'essais ayant pour but de détecter un ordre de grandeur permettant à l'opérateur de fonctionner.

Dans l'exemple ci-dessus (cf. Figure 3.4), l'opérateur défini est appliqué à la phase d'extension.

L'opérateur d'extension définitif sera donc basé sur les deux règles suivantes :

- le graphe doit rester bi-connexe ;
- un *CFL* inférieur à *Val-Min-CFL* supprimera la relation sauf si le *CFL* de sens inverse est supérieur à *Val-Activ-CL*.

Dans l'exemple de la figure 3.5 nous représentons sur le graphe déjà présenté en figure 3.4 le déroulement de l'algorithme de la phase d'extension. La valeur de *Val-Min-CFL* est arbitrairement fixée à 5 et celle de *Val-Activ-CLF* arbitrairement à 20. La liaison C-D n'est pas maintenue car le $CFL_{D=>C}$ est inférieur au *Val-Min-CFL* fixé et le *CFLc=>d* est inférieur au *Val-Activ-CFL* fixé. L'élément « D » ne peut donc rejoindre l'agrégat car le graphe résultant ne serait alors plus bi-connexe.

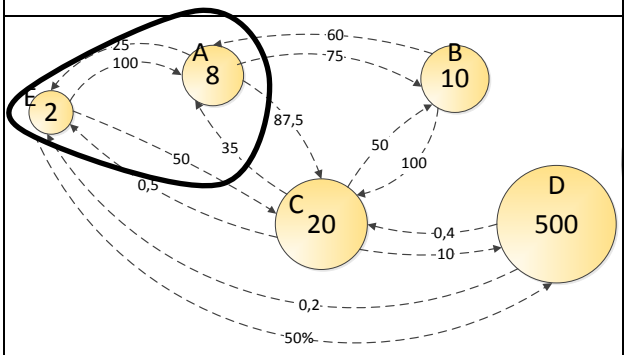
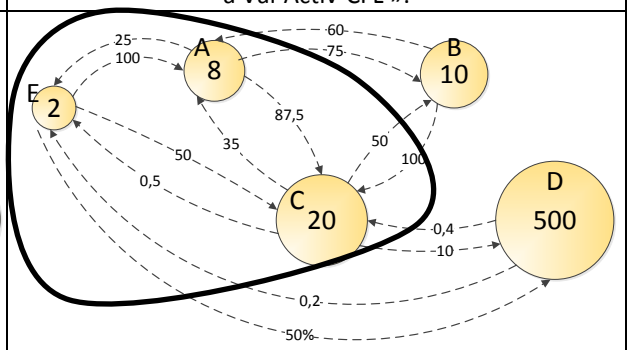
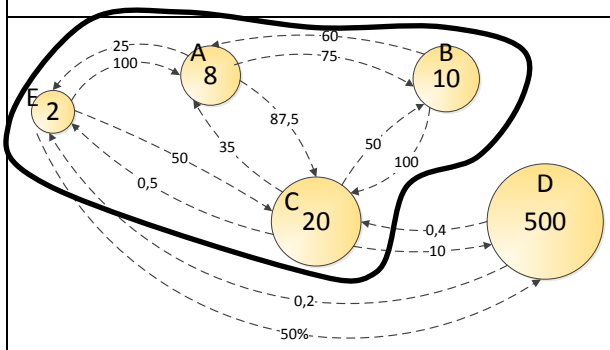
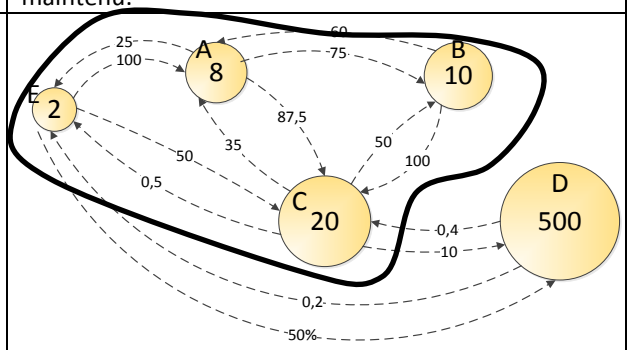
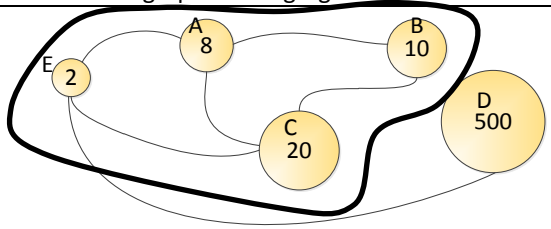
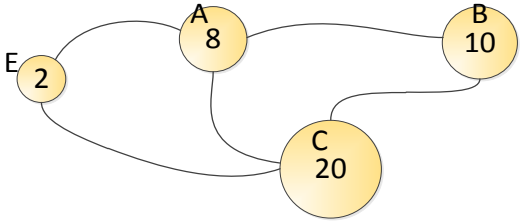
<table>
<tr><td>Étape I : Validation du lien A-E. Le lien appartient à une triade.</td><td>Étape II : Extension vers le nœud C. Validation des liens A-C et E-C : « Bien que le $CFL_{C=>E}$ soit inférieur au Val-Min-CFL le lien est maintenu $CFL_{E=>C}$ est supérieur à Val-Activ-CFL ».</td></tr>
<tr><td> 
</td><td> 
</td></tr>
<tr><td></td><td></td></tr>
<tr><td>Étape III : Extension vers le nœud B. Validation des liens A-B et C-B.</td><td>Étape IV : Tentative d'extension vers le nœud B, CFL_{D=>C} est inférieur à Val-Min-CFL et CFL_{C=>D} est inférieur à Val-Activ-CFL. Le lien C-D ne peut être maintenu.</td></tr>
<tr><td> 
</td><td> 
</td></tr>
<tr><td></td><td></td></tr>
<tr><td>Étape V : Le nœud D est définitivement exclut de l'agrégat son intégration ne permet pas de maintenir un graphe de l'agrégat bi-connexe.</td><td>L'agrégat définitif est constitué des nœuds E, A, B et C.</td></tr>
<tr><td> 
</td><td> 
</td></tr>
</table>

Figure 3.5 : Illustration du déroulement de l'algorithme Fusion/Extension en utilisant l'opérateur d'extension définitif.

Mécanisme de regroupement des mots-clés en agrégats (application de la méthode HLS)

Si un mot-clé peut appartenir à plusieurs agrégats, une paire de mots-clés constituant une diade ne peut appartenir au plus qu'à un agrégat. En effet, s'il existe un troisième mot-clé formant avec les deux premiers une triade, cette triade ne sera présente que dans un et un seul agrégat. S'il n'existe pas de triade incluant la diade alors la diade n'est dans aucun agrégat. C'est sur cette règle que se fonde l'algorithme de regroupement en agrégats proposé (cf. Algorithme 3.2).

```
Pour chaque mot-clé X faire [Phase de fusion]
        Extraire les mots-clés Y qui forment une triade valide selon l'opérateur d'extension avec X
        Pour chaque couple de mots-clés X-Y valides faire [Phase d'extension]
                S'il n'existe pas d'agrégat contenant le couple X-Y et que le couple n'a pas été testé
                alors
                        Créer un nouvel agrégat « X-Y » et ajout de X-Y
                        Tant que l'on ajoute des mots-clés dans l'agrégat faire
                                Pour les mots de l'agrégat faire
                                        Rechercher de nouveaux mots en triade
                                        Ajouter des mots-clés formant la triade avec les mots de
                                        l'agrégat
                                        Noter des couples trouvés comme « testés »
                                Fin de Pour
                        Fin de Tant que
                Fin de Si
        Fin de Pour [Fin de Phase d'extension]
Fin de Pour [Fin de Phase de Fusion]
```

Algorithme 3.2 : Regroupement des mots-clés en agrégats (application de la méthode HLS)

A titre d'exemple et afin d'éclairer le lecteur sur les résultats que la technique d'agrégation permet d'obtenir, nous proposons ici une représentation schématique des différents agrégats générés incluant le mot « Apple ».

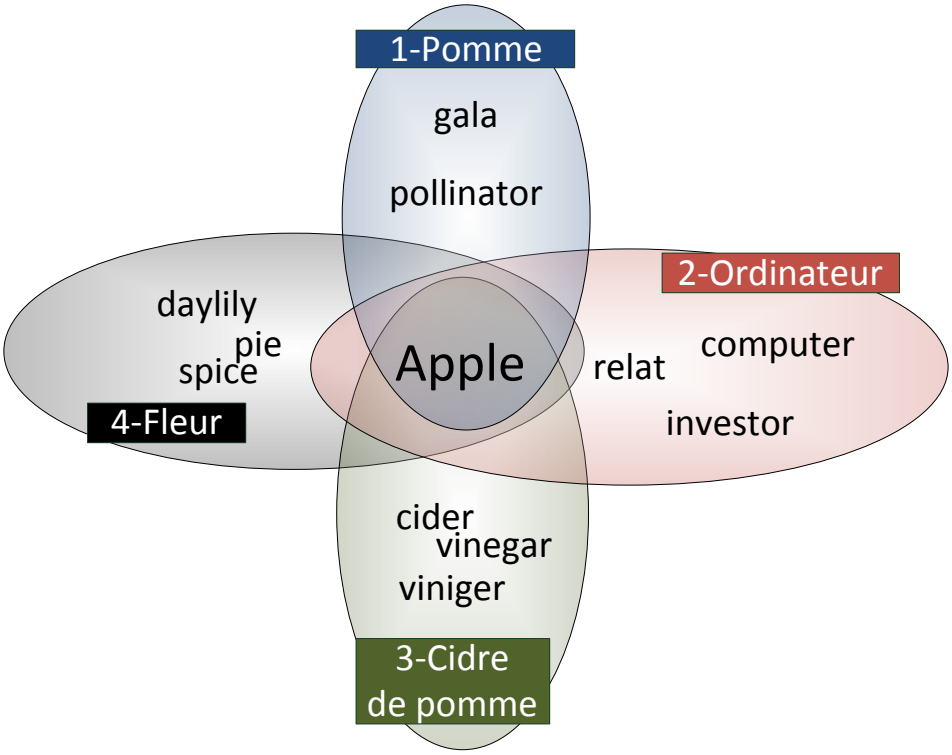


Figure 3.6 : Exemple de 4 agrégats partageant le même mot commun « Apple » résultant de notre proposition.

Comme on peut le remarquer dans la figure 3.6, les quatre agrégats sont cohérents et illustrent quatre contextes (acceptions) différents identifiés par rapport au mot-clé « Apple ». Ainsi, l'agrégat 1 fait référence au fruit (pomme) lui-même, le 2 à la marque d'ordinateur bien connue, le 3 au cidre de pomme et enfin le numéro 4 à une fleur nommée « Daylily » (un Lis) ayant pour non « Apple Pie Spice ».

Afin de valider les résultats, nous proposons plusieurs méthodes. Nous reviendrons en détail sur ces méthodes de validation dans le chapitre 4 consacré aux expérimentations et validations.

3.4 Méthode 3 : Rigidification Régulée

Cette méthode est une évolution notable de la Rigidification Simple. Elle est aussi inspirée par les travaux sur la rigidification de Hoffman et al. **[Hoffman&al-1997].** Afin d'améliorer la qualité des agrégats nous avons fait évoluer les règles gérant l'opérateur d'extension et plus précisément les règles de maintien des liaisons entre les nœuds.

3.4.1 Dans quel but une nouvelle méthode améliorée ?

Apprentissage

Les deux méthodes précédentes sont avant tout présentées comme des moyens ayant permis de construire cette troisième proposition. L'expérimentation de l'algorithme de Rigidification Simple (voir chapitre 4) et une meilleure connaissance du graphe permettent de formuler quatre observations :

1. Il existe un seuil de rupture de la validité sémantique des agrégats ;
2. les valeurs de *Val-Min-CFL* et de *Val-Activ-CFL* sont délicates à déterminer ;
3. la suppression des mots vides et/ou très courants casse des structures ;
4. les mots rares jouent un rôle disproportionné dans la construction d'agrégats.

Il est primordial de bien comprendre ces quatre points pour mesurer l'évolution entre la Rigidification Simple et la Rigidification Régulée.

Seuil de rupture de la validité sémantique

La validation de la méthode précédente nous a apporté plusieurs informations. Mais la plus importante est ce que nous nommons le « seuil de rupture de validité sémantique ». Au-delà de ce seuil, et ceci d'un point de vue statistique, nous notons un très fort affaiblissement de la cohérence sémantique telle que nous la mesurons. Ce seuil se situe, pour nos graphes de test AOL 17/04/2006 et AOL 17/03/2006 entre 30 et 40 mots. Il peut être considéré, comme l'équivalent du nombre de Dunbar dans les réseaux sociaux **[Dunbar-1992]**. Dunbar a émis l'hypothèse en 1992 que le nombre de personnes avec qui un être humain pourrait entretenir des relations durables serait limité. À partir de travaux menés sur des primates, il a estimé ce nombre autour de 150. On peut aussi faire un parallèle avec le seuil d'expansion des requêtes en science de la recherche d'informations. Harman a démontré que les requêtes devenaient moins performantes si l'on ajoutait plus de 20 à 40 mots **[Harman-1992]**. Ce chiffre a été validé plus tard par Boughamen M. et Soulé-Dupuy C. **[Boughamen-1997]**.

Les valeurs de *Val-Min-CFL* et de *Val-Activ-CFL* sont délicates à déterminer

La connaissance du seuil de rupture permet la conception d'un algorithme dans lequel il n'est pas nécessaire de fixer les valeurs de seuil *Val-Min-CFL* et de *Val-Activ-CF* de manière arbitraire et définitive**.** En effet ces seuils peuvent prendre des valeurs extrêmement différentes d'un agrégat à l'autre. L'utilisation de valeurs trop élevées fait perdre des agrégats et l'utilisation de valeurs trop faibles oblige à créer des agrégats de taille irréaliste. De plus, nous savons que statistiquement les agrégats d'une trop grande taille ont une faible cohérence sémantique. Il faut donc que l'algorithme adapte ces valeurs aux conditions locales, en limitant la taille des agrégats obtenus pour rester en deçà du seuil de rupture de la validité sémantique.

La suppression des mots vides et/ou très courants casse des structures.

Dans les grands graphes de terrain la plupart des nœuds ont par nécessité fonctionnelle le besoin d'être connectés. Cela vaut aussi bien pour la distribution du courrier dans un village (réseau postal) que pour le fonctionnement d'un ordinateur (réseau internet). Pourtant certains nœuds n'utilisent pas ces liaisons simplement pour fonctionner mais deviennent des connecteurs. Ainsi le facteur du village ou le routeur sur le réseau ont pour rôle d'effectuer les liaisons dans le réseau. Il s'agit là d'une spécialisation fonctionnelle d'un nœud du graphe qui devient un nœud connecteur ou concentrateur.

On retrouve ce phénomène dans la plupart des grands graphes de terrain, dont les objets n'ont pas par nature une limitation du nombre de connexions. C'est une des explications que nous proposons à la grande disparité du nombre de liens connectés par nœud typique des grands graphes de terrain.

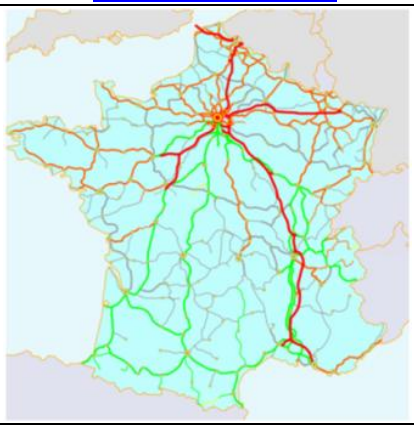
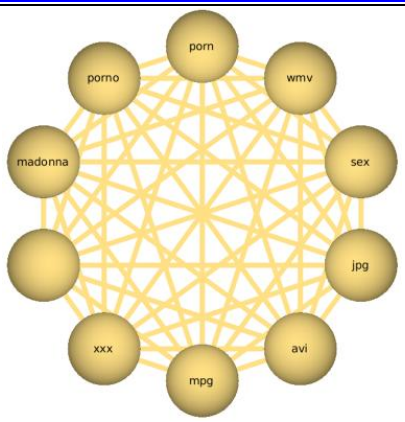
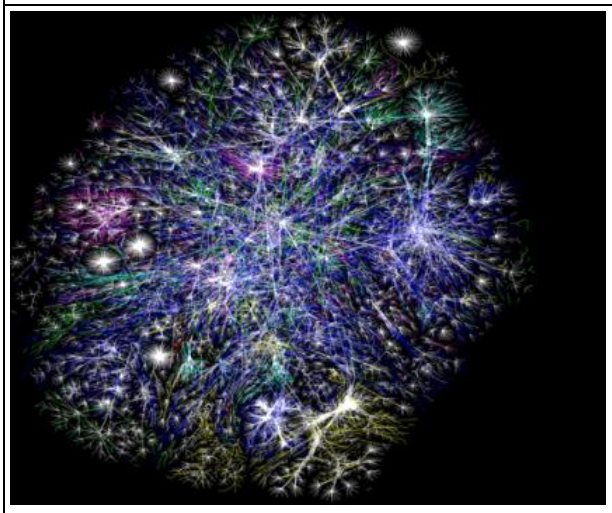
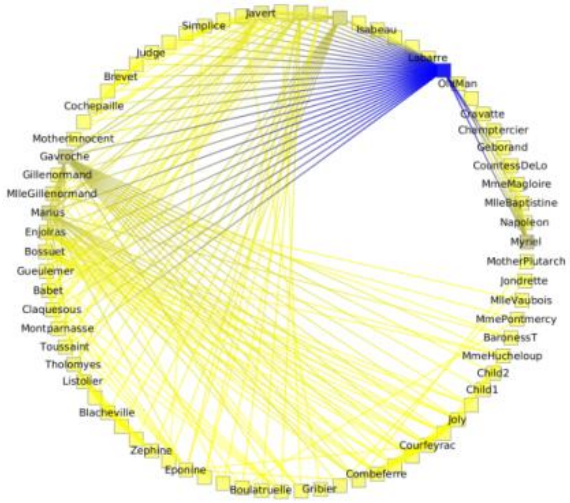
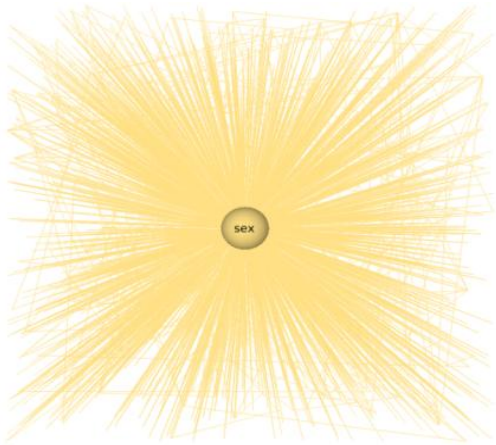
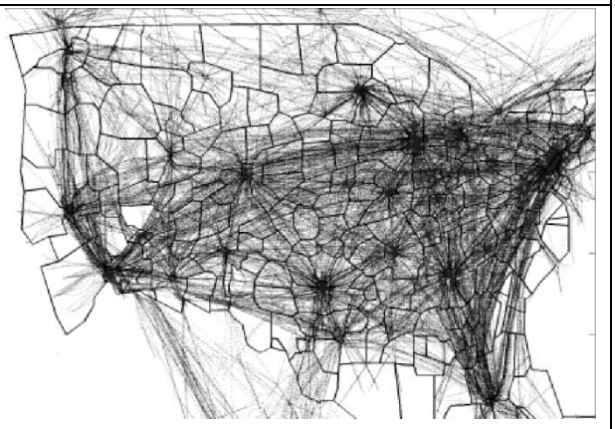
<table>
<tr><td>a - Paris hub du réseau ferré français : http://www.sncf.com</td><td>b - Les dix nœuds hubs les plus utilisés du graphe : http://antipaedo.lip6.fr/Proceedings.pdf</td></tr>
<tr><td>  </td><td> 
</td></tr>
<tr><td>c - Les hubs du trafic Internet 2006 : http://opte.org</td><td>d - Les personnages hubs du roman Les misérables de V. Hugo : http://networkdata.ics.uci.edu/data.php ?id=109</td></tr>
<tr><td>  </td><td> 
</td></tr>
<tr><td>e - Les 1000 nœuds les moins utilisés qui sont en diades avec le mot hub « sex » : http://antipaedo.lip6.fr/Proceedings.pdf</td><td>f - Les hubs du réseau de vols commerciaux USA : http://www.ce.berkeley.edu</td></tr>
<tr><td> 
</td><td>  </td></tr>
</table>

Figure 3.7 : Quelques exemples ou extraits de graphes de terrains incorporant des nœuds hubs.

Un autre exemple possible serait celui d'un réseau de transport. Une grande ville comme Paris ne possède pas seulement des routes pour y accéder, mais elle est un hub important de connexions territoriales. Cela signifie que je vais passer à Paris alors que ce n'est

pas le but final de mon voyage, mais une passerelle vers une nouvelle destination (cf. figure 3.7a). Il en est de même pour un voyage en avion venant de San-Francisco : je change à New-York pour aller en France (cf. figure 3.7b).

Comme on peut facilement le comprendre, Paris est alors victime de son succès. Les liaisons existantes attirent de nouveaux passagers, ce qui pousse à la création de nouvelles liaisons et ainsi le nœud se spécialise.

Les hôtes des réseaux informatiques chargés d'assurer ces liaisons sont nommés « switchs », « concentrateurs » ou « hubs » (cf. figure 3.7c). Nous proposons d'utiliser la même terminologie dans les graphes de terrain de type « petit monde » et de nommer « hubs » les nœuds fortement concentrateurs. La condition pour qu'un nœud hub existe dans un graphe est bien sûr que le degré de chaque nœud ne soit pas limité.

Dans les réseaux sociaux on retrouve ce même principe, prenons par exemple « la poignée de main » comme créateur de lien. Un candidat à la présidence des Etats-Unis en campagne va créer de cette manière, en quelques mois, des milliers de connexions (cf. figure 3.8c). L'embrassade (serrer dans ses bras), qui peut sembler plus intime, va aussi trouver ses « hubs ». Ainsi, nous pouvons citer le mahatma Amma qui revendique d'avoir pris 29 millions de personnes dans ses bras http://www.amma-europe.org/about-amma.html (cf. figure 3.8b). À l'échelle du quotidien, les professionnels du secteur de la santé et particulièrement en milieu rural (cf. figure 3.8a) peuvent jouer ce rôle de connecteur.

a - Médecin de campagne - Photo Doisneau 1968	b - Mahatma Amma – 2009 - Photo jean louis Audebaud	c - Obama en campagne 2010 – Photo David Mcnew
		

Figure 3.8 : Exemple de personnes « hubs » dans des réseaux sociaux.

Dunbar déclare « *En effet, la gestion de relations sociales est consommatrice de temps, ainsi le temps limite le nombre de contacts qu'une personne peut conserver et plus un réseau social est grand, moins la densité est élevée. Elle* (la densité) *argumente le coût cognitif inhérent à l'entretien de relations sociales.* » **[Dunbar-1998]**.

Toutefois dans une représentation graphique de la relation sociale, nous pouvons aussi utiliser un lien dirigé (ce type de lien prend en compte les différentes implications des acteurs des uns vers les autres). Par exemple, pour le Mahatma Amma, le lien social établi avec chacune des 29 millions de personnes embrassée a statistiquement peu d'importance. Mais cela n'est probablement pas le cas dans l'autre sens, si la personne embrassée par le Mahatma Amma n'a que très peu de contacts sociaux on imagine l'importance que ce contact peut

revêtir. On retrouve aussi l'importance de ce lien quand deux personnes ont fait ensemble la démarche de rencontrer le Mahatma. L'élimination du lien entre ces personnes et le Mahatma va alors rompre la clique.

Dans les graphes de mots il existe aussi des nœuds ayant un degré particulièrement important. Le terme le plus utilisé dans les recherches multi-mots sur Internet en avril et mars 2006, sur le moteur de recherche d'AOL est « of ». Ce mot est le mot « hub » par définition : (http://gregsadetsky.com/aol-data). Dans les deux premières tentatives pour créer des agrégats nous avions éliminé les mots de ce type. Pourtant ils peuvent participer à la construction de figures telles que des triades, qui sont la base de notre système d'agrégation.

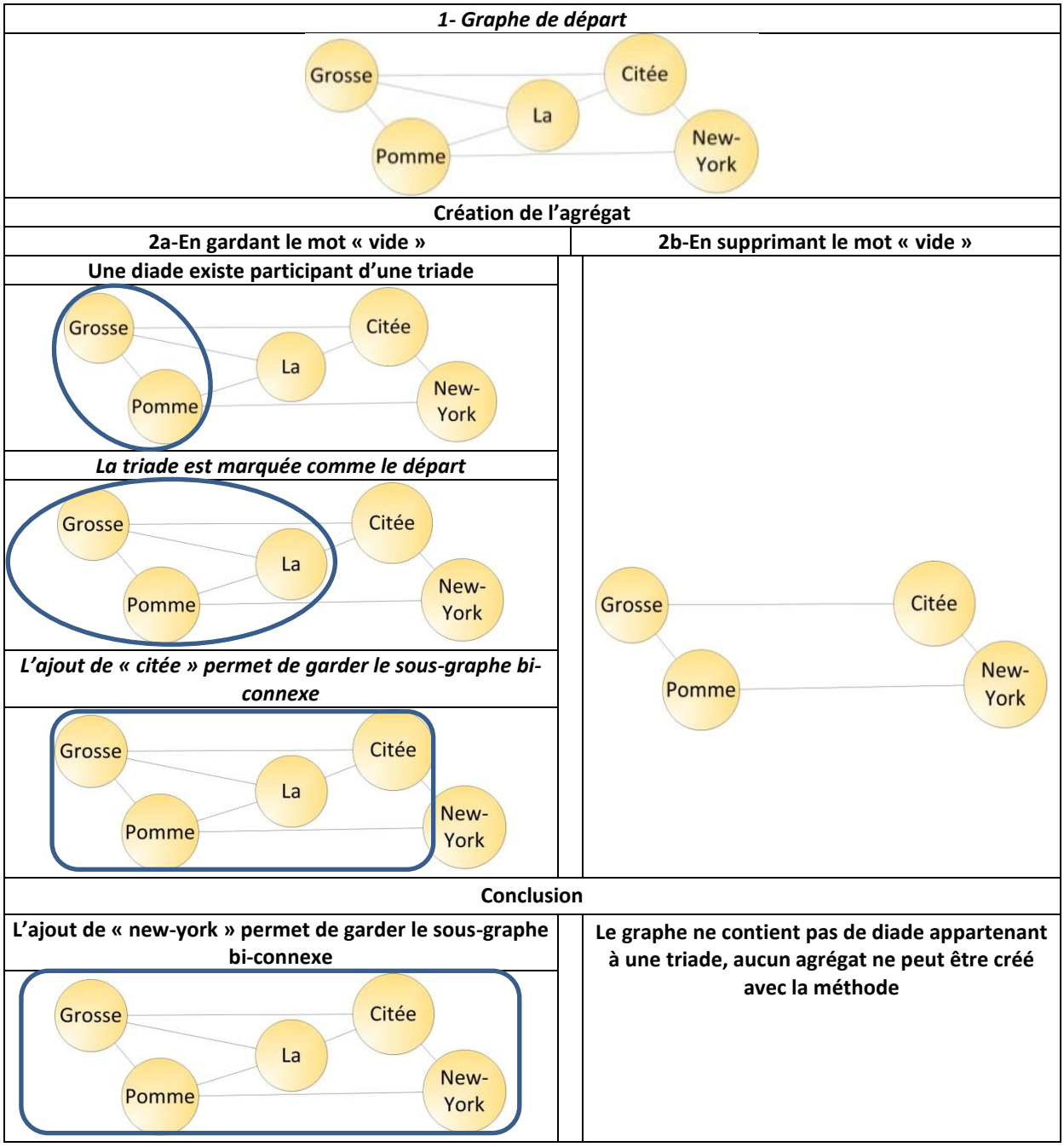


Figure 3.9 : Suppression d'un mot hub entrainant la perte de la connectivité nécessaire à la construction d'un agrégat.

Dans la figure 3.9, la suppression du mot « la » nous fait perdre la possibilité de créer un agrégat contenant les cinq mots {grosse, pomme, New-York, citée, la}.

Un rôle disproportionné pour les mots rares

En cherchant à limiter la taille des agrégats, nous notons des mots ou des nœuds qui se comportent comme des verrous de validation du lien. Ces mots sont ceux qui possèdent un très faible poids. Ces mots rares (de faible usage) font basculer systématiquement la valeur de *CFL* vers des valeurs extrêmement fortes. Ils sont alors trop fortement liés aux autres mots (et spécialement aux termes très fréquents), pour être supprimés par une augmentation de la valeur de *Val-Activ-CFL*. Ces mots de faible usage doivent pouvoir être écartés ou encore leur rôle doit pouvoir être minoré pour maintenir une taille d'agrégats en deçà de la taille correspondant au seuil de rupture de la validité sémantique.

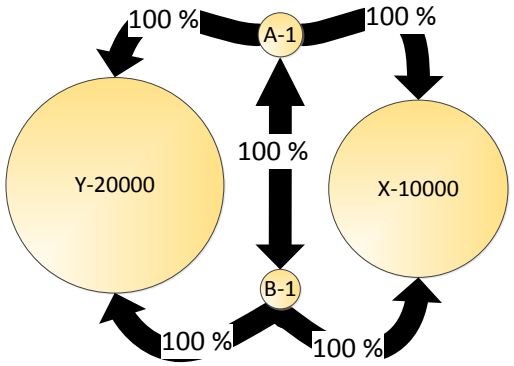


Figure 3.10 : Blocage des triades par des mots de très faibles usages.

Dans la figure 3.10, le graphe présente deux mots A et B conjointement utilisés une seule fois avec deux mots très utilisés X et Y. La requête effectuée par l'utilisateur était {A, B, X, Y}. X et Y sont des mots utilisés dans 10000 et 20000 requêtes. La valeur de CFL est pour toutes les liaisons égale à 1 (ou à 100%), ainsi la liaison est toujours valide et cela indépendamment des valeurs de *Val-Min-CFL* et de *Val-Activ-CF*.

Les nœuds peu utilisés sont alors des éléments qui verrouillent la validité des liens entre des nœuds plus utilisés, voire même des nœuds de type « hub ». Ainsi deux mots, A et B, utilisés une seule fois conduisent à la création d'un agrégat {A, B, X, Y} (quand bien même A et B n'ont été utilisés que dans une seule recherche sur une période de deux mois). Il existe un fort risque que ces mots soient, en fait mal, orthographiés. En somme, plus leur usage est marginal plus ils constituent la base inamovible d'agrégats (cf. figure 3.10). Ces mots super-agrégeants sont un problème qu'il faut traiter.

3.4.2 Présentation de l'algorithme « Rigidification Régulée »

Ce nouvel algorithme est basé sur les mêmes règles de rigidification que l'algorithme de Rigidification Simple. Mais il est écrit en intégrant la volonté de limiter la taille des agrégats. Les agrégats sont créés de façon à ce que le nombre de mots qu'ils contiennent ne

dépasse pas le seuil de rupture de la validité sémantique. La valeur des paramètres qui permettent la validation des liaisons ainsi que celles maintenant ou pas un nœud dans la structure vat être modulée de façon à limiter la taille des agrégats. Cette taille reste alors toujours sous le seuil de validité sémantique défini.

La Rigidification Régulée est toujours basée sur la méthode HLS. Son opérateur d'extension définitif respecte les règles définies précédemment :

- le graphe doit rester bi-connexe ;
- un *CFL* inférieur à *Val-Min-CFL* supprime la relation sauf si le CFL de sens inverse est supérieur à *Val-Activ-CLF*. Cette validation de la liaison est conditionnée à la taille de l'agrégat. La liaison est donc validée (ou invalidée) dans une boucle de traitement qui exécute deux actions (grâce au calcul de quatre paramètres et ce à chaque fois que l'agrégat atteint sa taille maximale). Les deux actions sont :
  - supprimer des nœuds trop utilisés (mots vides ou faibles) et très peu utilisés (mots super-agrégeant) ;
  - augmenter la valeur de *Val-Min-CFL* et de *Val-Activ-CLF* pour invalider (supprimer) les liens les plus faibles.

Garantir la validité sémantique par une Taille Maximale de l'Agrégat (*TMA*)

Grace aux résultats d'expérimentations précédentes (cf. paragraphe 4.4.2) nous savons qu'il existe statistiquement un seuil à la taille d'un agrégat pour garantir une validité sémantique. Nous nommons cette valeur ***TMA*** pour Taille Maximale de l'Agrégat.

Notre objectif est de créer des agrégats dont le nombre de mots ne dépasse pas la *TMA*, pour garantir une bonne cohérence sémantique.

Parmi les quatre paramètres utilisés pour limiter le nombre de nœuds dans un agrégat, les deux premiers permettent de supprimer les liaisons « invalides » :

- ***Val-Min-CFL*** : le lien est valide si les deux liens dirigés et pondérés le constituant sont supérieurs à cette valeur.
- ***Val-Activ-CFL*** : le lien est valide si au moins un des deux liens dirigés et pondérés le constituant sont supérieurs à cette valeur. Les deux autres valeurs sont utilisées pour écarter de l'agrégat les nœuds trop utilisés (mot vides ou faibles) et très peu utilisés (mots super-agrégeant).
- ***Poids-Min*** ou Poids minimum de validité : les nœuds ayant un poids inférieur à *Poids-Min* sont écartés.
- ***Poids-Max*** ou Poids maximum de validité : les nœuds ayant un poids supérieur à *Poids-Max* sont écartés.

À chaque fois que l'agrégat dépasse la *TMA*, une modification des quatre paramètres est effectuée afin d'écarter un certain nombre de nœuds et de liaisons.

Pour chaque diade **faire**
    Fin_de_traitement = Faux
    Tant que le nombre de nœuds de l'agrégat est inférieur au seuil garantissant la cohérence sémantique **et que** l'on peut ajouter des nœuds en gardant un graphe bi-connexe et que **PAS** Fin_de_traitement **faire**
        Recherchez de nouveaux nœuds à rajouter dans l'agrégat
    Fin de tant que
    Si le nombre de nœuds **>** 2 et nombre de nœuds **<** seuil garantissant la cohérence sémantique **alors**
        Sauvegarde agrégats
        Fin_de_traitement = Vrai
    Si Non SI nombre de nœuds **>** seuil garantissant la cohérence sémantique **alors**
        Suppression des liaisons invalides par augmentation de ***Val-Min-CFL*** et ***Val-Activ-CLF***
        Suppression des nœuds les plus utilisés Poids > ***Poids-max*** et les plus rares Poids < ***Poids-min***
        Augmentation de ***Val-Min-CFL***, ***Val-Activ-CLF***, ***Poids-max*** et diminution de ***Poids-min***
    Si non
        Agrégat marqué comme impossible
        Fin_de_traitement = Vrai
    Fin de SI
Fin de Pour

Algorithme 3.3 : Présentation simplifiée de l'algorithme de rigidification incluant un auto-paramétrage des valeurs de rigidification et une auto-suppression des nœuds et liaisons les moins signifiants.

Les deux parties de l'algorithme

Afin d'optimiser la vitesse d'exécution nous avons choisi de séparer l'algorithme en deux parties :

- une première boucle qui recherche rapidement des valeurs efficientes des quatre paramètres. Ces valeurs encadrent des valeurs qui permettrait la création de l'agrégat dans les meilleures conditions ;
- une seconde boucle qui explore finement l'espace entre les valeurs proposées par la boucle principale pour rechercher les valeurs les mieux adaptées à la création de l'agrégat dans les meilleures conditions.

Pour les deux boucles, la condition d'arrêt de la recherche d'agrégat est la même : ou la diade est incluse dans un agrégat dont la taille est inférieure au *TMA* et supérieure à 2 ou elle ne permet pas la construction d'un agrégat.

La première boucle ou « boucle principale » a pour but d'incrémenter rapidement les valeurs des quatre paramètres pour trouver les valeurs de démarrage et les valeurs finales de la seconde boucle qui est en charge d'optimiser l'agrégat.

La seconde boucle, nommée « boucle fine », cherche à construire un agrégat en affinant les valeurs proposées par la boucle principale. Elle recherche les valeurs précises des quatre paramètres permettant la construction d'un agrégat respectant les règles préétablies.

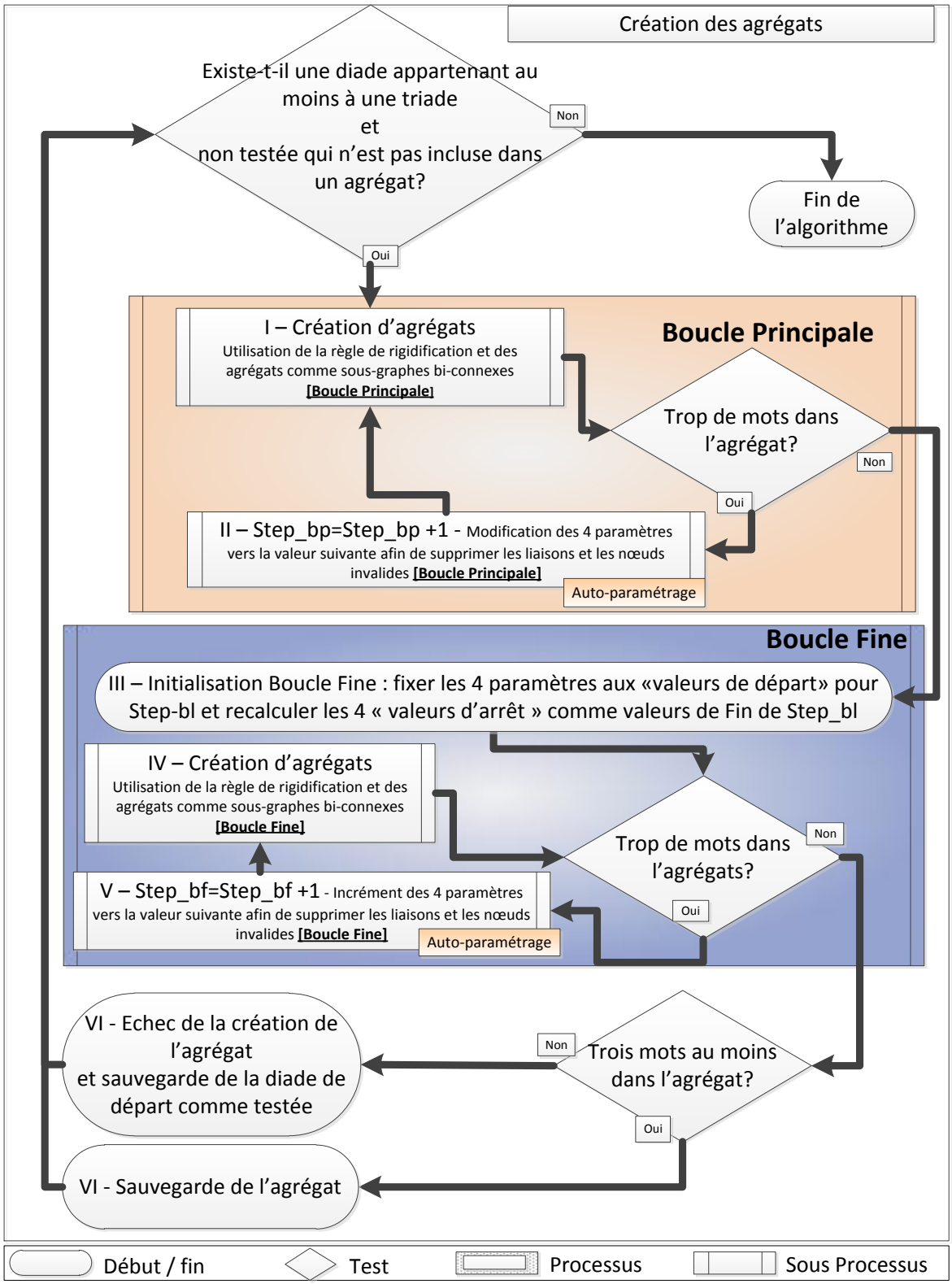


Figure 3.11 : Agrégation basée sur la limitation de la taille des agrégats et l'auto-paramétrage des valeurs de rigidification.

Les deux boucles (la boucle principale puis la boucle fine) (cf. figure 3.11) sont successivement utilisées pour la création de chacun des agrégats. Chaque boucle calcule la progression des quatre paramètres de manière dissemblable en raison des différences d'échelle des paramètres dans les deux boucles.

La boucle principale

La boucle principale a pour but de trouver les valeurs de démarrage et de fin de la boucle fine. Elle recherche grossièrement et donc rapidement des valeurs pour lesquelles l'agrégat est soit inconstructible (la diade de départ n'est plus intégrée dans une triade), soit possède moins de mots que la *TMA*.

Les valeurs finales de la boucle principale ont soit permis de créer un agrégat de taille inférieure au *TMA* soit rompu le lien de la diade de départ ou celui reliant la diade de départ et l'agrégat. La boucle principale est un test rapide permettant de trouver les valeurs des quatre paramètres pour valider ou invalider l'agrégat.

Les valeurs utilisées par la boucle principale dans l'avant dernier passage c'est-à-dire au « *step final -1* » sont toujours celles permettant la création d'un agrégat intégrant la diade de départ mais dépassant en nombre de nœuds la *TMA*. Les valeurs au « *step final* » sont celles permettant de créer un agrégat valide de taille inférieure au *TMA* ou d'invalider l'agrégat.

Dans la boucle principale les paramètres liés à l'usage des mots s'étendent sur une échelle très importante, le nombre d'utilisations (poids) d'un mot allant, par exemple, de 1 à plus de 300 000 dans une de nos expérimentations. Pour respecter une meilleure proportionnalité dans l'évolution des valeurs basées sur le poids des mots, nous utilisons une échelle logarithmique.

Dans la table 3.1 *Step_bl* représente le nombre de fois où l'agrégat a atteint la taille maximale (*TMA*) dans la boucle principale. *Avg(G.weight)* représente la moyenne du poids des mots dans l'ensemble du graphe.

Paramètres	VD : Valeurs de Démarrage	Valeurs Finales	Valeur en fonction de l'incrément Step_bp= Step_bp +1
Val-Min-CFL	VD_{VMC} = Observation	prédéterminée (10)	VD_{VMC} + (10- VD_{VMC})* (Step_bl / NbSteps)
Val-Activ-CFL	VD_{VAC} = Observation	Prédéterminée (51)	VD_{VAC} + (51- VD_{VAC})* (Step_bl / NbSteps)
Poids-Min	Poids minimum des nœuds	Moyenne des poids des mots	Avg(G.weight) ^ (Step_bl /NbSteps)
Poids-Max	Poids maximum des nœuds	Moyenne des poids des mots + 1	Max(G.weight) ^ ((NbStep - Step_bl) / NbStep) + Avg(G.weight)

Table 3.1 : Limites et calculs des valeurs des quatre paramètres dans la boucle principale.

Les valeurs de départ de *Val-Min-CFL* et *Val-Activ-CFL* sont choisies par une observation de populations spécifiques. En comparant la distribution du poids relatif des liens entre des mots typés comme monosémiques et des mots quelconques nous parvenons à déterminer des valeurs justifiées de ces deux paramètres.

Les valeurs finales de *Val-Min-CFL* et *Val-Activ-CFL* sont choisies selon les critères suivants :

- en statistique médicale par exemple, une différence supérieure à 5% entre des groupes témoins est nécessaire pour que les résultats soient considérés comme validés. Ces 5% sont appelés « signification statistique » **[Weber&al-2001]**. Il ne s'agit pas d'une barrière de validité mais davantage d'un seuil à partir

duquel les résultats sont à considérer comme de plus en plus valables. Si on double ce pourcentage, s'affiche alors un chiffre de 10% dont la valeur peut d'autant moins être ignorée. La valeur de 10% est donc choisie comme valeur de *Val-Min-CFL*, ce qui signifie qu'en aucun cas l'on ne pourra écarter une diade pour laquelle le lien représente 10% ou plus des usages pour les deux mots la constituant ;

- la valeur de 51% comme valeur finale de *Val-Activ-CF* est simplement l'expression qu'il n'est pas possible d'ignorer un lien majoritaire pour un des nœuds. Cela indique que l'on ne pourra écarter une diade si l'un des nœuds a 51% ou plus de ses usages avec l'autre nœud.

Il faut garder à l'esprit que ces valeurs sont un maximum théorique qui ne correspond absolument pas aux valeurs qui seront véritablement utilisées par l'algorithme pour construire les agrégats.

La boucle fine

La boucle fine s'exécute après la boucle principale et conserve le même algorithme que la boucle principale.

La boucle fine crée des agrégats en utilisant les valeurs finales et les valeurs précédant les valeurs finales de la boucle principale (celles utilisées dans la boucle principale au « *step final* » et au « *step final -1* »). Elles sont beaucoup plus proches entre-elles que les valeurs minimales et maximales utilisées dans la boucle principale. Du fait que, dans la boucle fine, les valeurs de départs et valeurs finales sont resserrées, l'incrémentation est différente ; un incrément de type linéaire est ainsi plus adapté qu'une variation logarithmique.

C'est entre les valeurs finales de la boucle principale (valeurs utilisées au « *step final* ») et celles à l'avant dernière exécution de la boucle principale (valeurs utilisées au « *step final -1* ») que la boucle fine cherche à optimiser et valider un agrégat de telle sorte que :

- l'agrégat possède un nombre de mots inférieur au *TMA* ;
- l'agrégat possède un maximum de mots (de façon à inclure les mots rares et les mots hubs) ;
- l'agrégat inclue la diade de départ.

Paramètres	VD : Valeurs de Démarrage	VF : Valeurs Finales	Valeur en fonction de l'incrément Step_bf= Step_bf +1
Val-Min-CFL	VD_{VMC} = *Val-Min-CFL de la boucle principale pour step_bp - 1*	VF_{VMC} = *Val-Min-CFL de la boucle principale pour step_bp*	$VD_{VMC} + VF_{VMC} * (Step\_bf / NbSteps)$
Val-Activ-CFL	VD_{VAC} = *Val-Activ-CFL de la boucle principale pour step_bp - 1*	VF_{VAC} = *Val-Activ-CFL de la boucle principale pour step_bp*	$VD_{VAC} + VF_{VAC} * (Step\_bf / NbSteps)$
Poids-Min	$VD_{Poids\text{-}Min}$ = ***Poids-Min*** *de la boucle principale pour step_bp - 1*	$VF_{Poids\text{-}Min}$ = ***Poids-Min*** *de la boucle principale pour step_bp*	$VD_{Poids\text{-}Min} + VF_{Poids\text{-}Min} * (Step\_bf / NbSteps)$
Poids-Max	$VD_{Poids\text{-}Max}$ = ***Poids-Max*** *de la boucle principale pour step_bp - 1*	$VF_{Poids\text{-}Max}$ = ***Poids-Max*** *de la boucle principale pour step_bp*	$VD_{Poids\text{-}Max} + VF_{Poids\text{-}Max} * (Step\_bf / NbSteps)$

Table 3.2 : Limites et calculs des valeurs des quatre paramètres dans la boucle fine.

Dans la table 3.2 *Step_bf* (nombre de pas dans la boucle fine) représente le nombre de fois où l'agrégat a atteint la taille maximale (TMA) dans la boucle fine.

Pour Chaque nœud X faire [***PC-1***] **faire**
Step_bp=0
Rechercher les nœuds Y tels que la diade X-Y fasse partie d'une triade valide (selon les quatre paramètres = valeurs de départ)
Pour chaque diade X-Y valide **faire [*PC-2*] faire**
Si il n'existe pas d'agrégat incluant X et Y et que le couple « X-Y » est une diade non testée **alors [*SI-1*]**
Créer un agrégat A^{X-Y} et ajouter X et Y
Fin_de_Boucle_Principale= Faux
Faire - [*Boucle Principale*]
Tant que l'on ajoute des nœuds à l'agrégat et que le nombre de nœuds est inférieur **faire**
Ajouter de nouveaux nœuds respectant les paramètres de poids et formant un sous-graphe bi-connexe
Fin de Tant que
Si le nombre de nœuds dans l'agrégat >= *TMA* **alors**
Step_bp = Step_bp + 1
Vérifier la validité de la diade [X-Y] tel que la diade X-Y fasse partie d'une triade valide (respect des quatre paramètres = valeurs calculées)
Fin de SI
Si le nombre de nœuds dans l'agrégat < *TMA* **alors**
Fin_de_Boucle_Principale = Vrai
Fin de SI
Tant que Non *Fin_de_Boucle_Principale* [*Boucle Principale*]
Calculer les valeurs de départ des 4 paramètres pour la boucle fine (step_bp-1)
Calculer les valeurs finales des 4 paramètres pour la boucle fine (step_bp)
Step_bf= 0
Suppression et recréation de l'agrégat « X-Y »
Fin_de_Boucle_fine= Faux
Faire - [*Boucle Fine*]
Tant que l'on ajoute des nœuds à l'agrégat et que le nombre de nœuds est inférieur à *TMA*
Ajouter de nouveaux nœuds respectant les paramètres de poids et formant un sous-graphe biconnexe
Fin de Tant que
Si le nombre de nœuds dans l'agrégat >= *TMA* **alors**
Step_bf = Step_bf + 1
Vérifier la validité de la diade [X-Y] tel que la diade X-Y fasse partie d'une triade valide (respect des quatre paramètres = valeurs calculés)
Fin de SI
Si le nombre de nœuds dans l'agrégat < *TMA* **alors**
Fin_de_Boucle_fine= Vrai
Fin de SI
Tant que Non Fin_de_Boucle_fine [*Boucle fine*]
Si le nombre de nœuds dans l'agrégat > 2 alors
sauvegarder l'agrégat
Fin de si
Marquer la diade comme testée
Fin de si [***SI-1***]
Fin de pour [***PC-2***]
Fin de pour [***PC-1***]

Algorithme 3.4 : Algorithme complet incluant les deux boucles de la méthode de Rigidification Régulée.

Si la boucle principale a permis de créer un agrégat, il peut sembler inutile de lancer la boucle fine. C'est le cas si l'agrégat au sortir de la boucle principale a une taille égale au

TMA. Si tel n'est pas le cas (taille inférieure), l'exécution de la boucle fine permet l'introduction de nœuds rares ou très souvent utilisés dans l'agrégat, sans dépasser la *TMA*, en respectant donc statistiquement la cohérence sémantique des agrégats.

Une limite de la méthode

Une limite de la méthode est son incapacité à ordonner les nœuds dans un agrégat ou encore à pondérer l'appartenance des nœuds à un agrégat. Les nœuds représentent l'espace sémantique d'un agrégat tout en y étant rattachés. La complexité de ce lien rend difficile toutes pondérations.

De plus, il existe plusieurs combinaisons graphiques qui ne permettent pas à un nœud de rejoindre un agrégat. Sans chercher à les citer toutes, disons simplement qu'une figure identifiable comme un sous-graphe bi-connexe ou un k-clique ne peut pas par, définition, intégrer un nœud seulement connecté à une diade (cf. image 3.6).

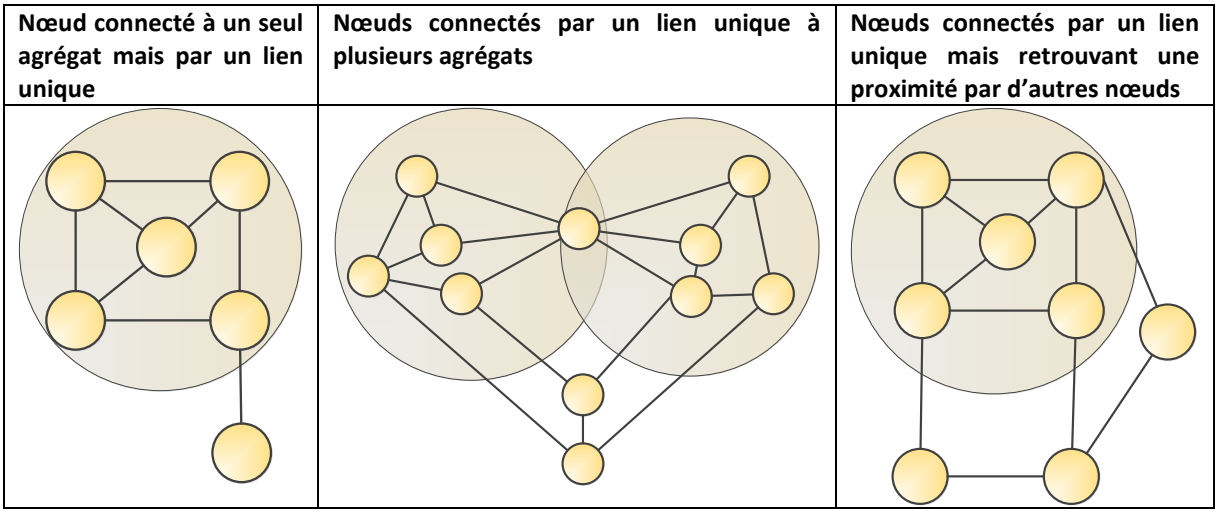


Figure 3.12 : Exemple de figures où des nœuds proches d'agrégats ne sont pas intégrés car ne répondant pas à la condition de rigidification.

Cette méthode consiste en une recherche de zone ayant une forte cohérence sémantique. Elle peut être éventuellement complétée par des post-traitements si l'on souhaite que chaque mot puisse être rattaché à un ou plusieurs agrégats. La méthode d'enrichissement d'agrégats par gravité présentée dans le paragraphe 3.5 est un exemple de méthode pouvant être utilisée dans ce but.

3.5 Méthode 4 : Méthode d'enrichissement d'agrégats par gravité

La méthode d'enrichissement d'agrégats par gravité peut être incluse en tant que phase supplémentaire d'une méthode plus générale. Elle permettrait à une méthode «mono phase » de rejoindre alors la famille des méthodes en plusieurs phases. Dans cette famille, la phase 3

de l'algorithme RaRe\IS présenté au paragraphe 2.3.2 [**Baumes&al-2005-1]** est un exemple de méthode d'enrichissement incluse dans une méthode plus générale. Elle pourrait aussi être utilisée en tant que phase d'assemblage dans les méthodes issues de HLS (cf. paragraphe 3.3.2)

Cependant, la méthode que nous avons créée est une solution indépendante de toutes méthodes et peut être implantée sur un graphe quelconque si celui-ci contient des agrégats identifiés ou connus.

3.5.1 Les objectifs d'une méthode d'enrichissement des agrégats.

L'enrichissement a pour but de créer une relation pondérée ou non-pondérée entre un nœud et un ou plusieurs agrégats.

Cependant avant de proposer une méthode d'enrichissement posons-nous la question suivante : un mot peut-il être plus ou moins proche d'un agrégat ?

Les linguistes définissent bien plusieurs zones d'influences autour d'une thématique.

Le champ lexical est défini par Marc Farayet **[Fayaret-2011]** comme *« l'ensemble des mots qui, dans un texte, se rapportent à une même réalité ou à une même idée. Ces mots ont pour point commun d'être synonymes, d'appartenir à la même famille, au même domaine ou encore à renvoyer à la même notion. »*.

Le réseau lexical est défini dans le même ouvrage comme : *« Le réseau [lexical] qui regroupe l'ensemble des mots qui désignent des idées ou des réalités qui renvoient à un même thème (=**le champ lexical) plus tous les mots** qui à cause du contexte et de certains aspects de leur signification (du fait des connotations) évoquent aussi ce thème. »*. Ainsi le réseau lexical incorpore un champ lexical en l'enrichissant.

Nous trouvons dans le schéma proposé par Marc Farayet deux espaces sémantiques où un réseau lexical incorpore un champ lexical.

Figure 3.13 : Exemple d'espaces sémantiques incorporés l'un dans l'autre « tel que » fourni par Marc Farayet [Fayaret-2011].

Le but de l'évocation des espaces lexicaux n'est pas de rechercher une identité entre eux et les agrégats. En effet, un champ ou un réseau lexical se définissent dans l'espace restreint d'un texte, ce qui n'est pas le cas des agrégats qui doivent pouvoir être créés à partir de fichiers de log de plusieurs millions de mots. Il n'en reste pas moins que Fayaret manipule des objets de même nature que ceux qui nous intéressent (les mots et les utilisations conjointes), qui une fois regroupés présentent une architecture similaire fondée sur un emboitement d'espaces sémantiques : champ lexical inclus dans un réseau lexical et agrégats « **noyaux** » inclus dans agrégats augmentés. Les agrégats noyaux se doivent de présenter une cohérence sémantique particulièrement élevée.

Il est ainsi possible d'imaginer un certain nombre de cercles périphériques. Ces zones périphériques détermineraient des zones d'influences où des nœuds satellites seraient sémantiquement liés à l'agrégat. En partant de cette hypothèse, les nœuds « en attraction » sont donc soumis à l'équivalent d'une « force de gravité » par les nœuds de l'agrégat.

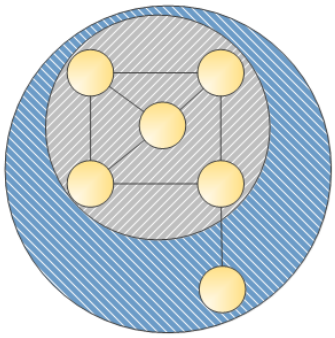
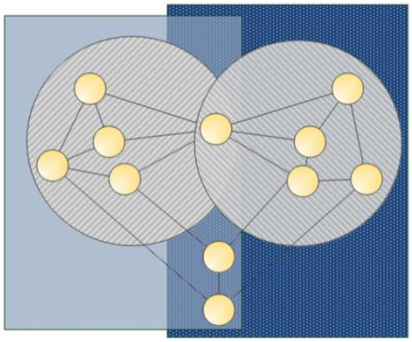
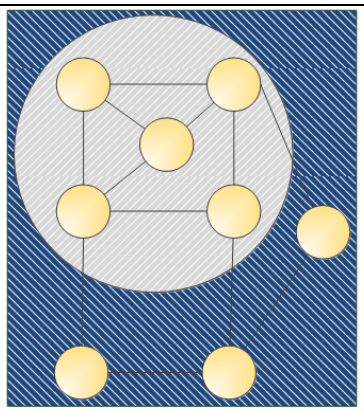
	**Création d'un agrégat augmenté par l'ajout d'un nœud à lien unique**
	**Augmentation des surfaces de recouvrement par la création d'agrégats augmentés**
	**Attachement de nœuds fortement liés à l'agrégat**

Figure 3.14 : Exemples de figures où des nœuds proches d'agrégats peuvent contribuer à la création d'un agrégat augmenté.

3.5.2 Présentation de la méthode d'Enrichissements par gravité

La méthode proposée calcule un **coefficient d'attraction** du nœud externe par l'agrégat, puis ordonne les nœuds en fonction de ce coefficient (en cas d'égalité d'attraction, le poids du nœud permettra d'arbitrer ce classement). Pour éviter des agrégats de trop grande taille, on limitera ensuite le nombre de nœuds ayant rallié l'agrégat (à quelques dizaines par exemple).

Cette technique a plusieurs avantages :

- en premier lieu elle représente un coût computationnel faible. Une fois les agrégats noyaux calculés, elle ne nécessite qu'un calcul arithmétique et un classement sur des nœuds en attraction par chacun des agrégats ;
- en second lieu, elle augmente la couverture des zones en recouvrement ;
- en dernier lieu, elle permet de réintroduire dans les agrégats des nœuds exclus par les algorithmes d'agrégation (cf. figure 3.8).

Calcul du Coefficient d'Attraction (*CA*)

Notons $\boldsymbol{CA_{X\to A}}$ le $\boldsymbol{CA}$ pour le nœud *X* et l'agrégat *A*. La valeur de $\boldsymbol{CA_{X\to A}}$ est donnée par la formule ci-dessous :

$$\boldsymbol{CA_{X\to A}} = \sum_{k=1}^{n} PLk \,/\boldsymbol{P_X} * \boldsymbol{D_{X\to A}}$$

Où *PLk* représente le poids de la liaison entre le nœud *k* interne à *A* et le nœud *X* externe à *A*. $\boldsymbol{D_{X\to A}}$ représente le degré du nœud *X* vers l'ensemble des nœuds de l'agrégat *A* et $\mathbf{P_X}$ le poids affecté au nœud *X*.

Une opération de filtrage est alors effectuée pour ne pas créer de liens trop faibles. Pour cela nous ne considérons que les *CA* au-dessus d'une certaine valeur.

Enfin, nous pouvons effectuer une classification des nœuds dans des zones d'influence en fonction de leur *CA*, ceci permettant d'avoir rapidement une représentation visuelle exploitable.

Exemple de calcul et d'ordonnancement des nœuds dans la création d'un agrégat augmenté

Dans cet exemple (cf. figure 3.15) nous calculons d'abord le coefficient d'attraction *CA* des nœuds {V, W, X, Y, Z} du graphe de la figure 3.15 pour l'agrégat A1.

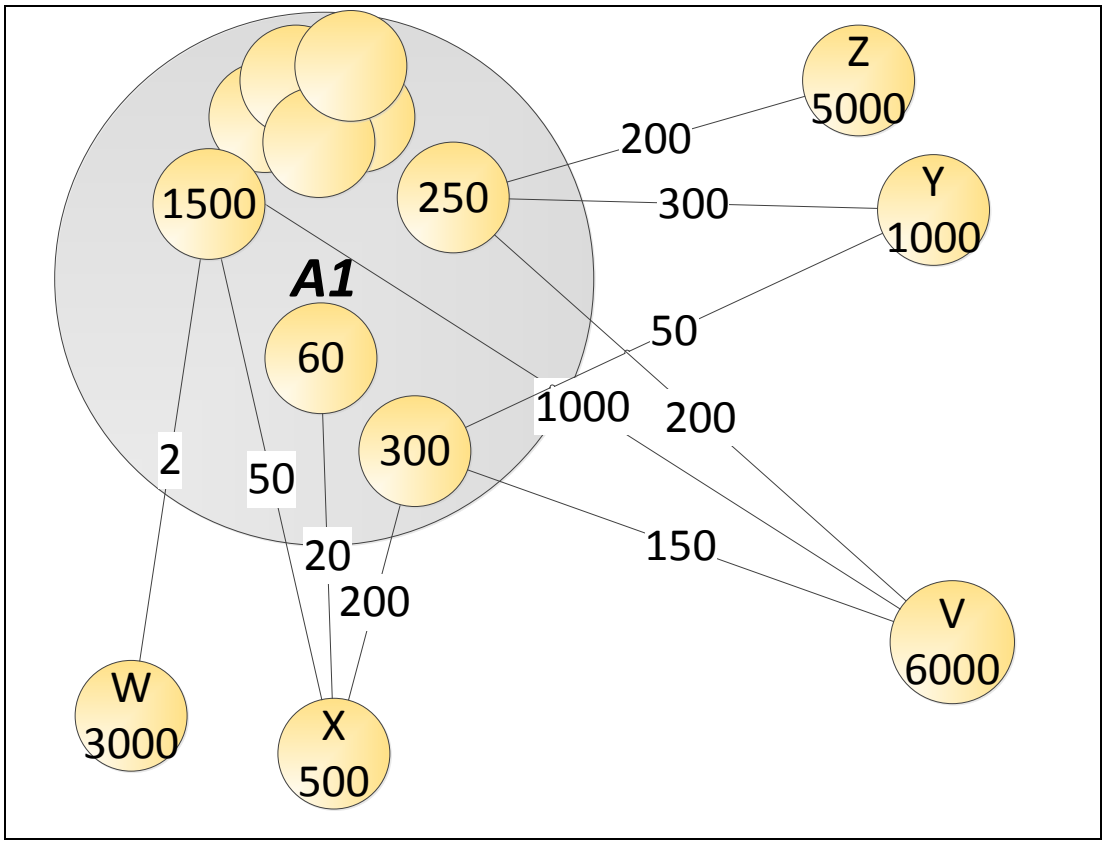


Figure 3.15 : Graphe d'un agrégat et de nœuds possibles pour la formation d'un agrégat étendu.

Le niveau de seuil de validé du *CA* est ici fixé à 10% du poids du nœud.

Nœuds	$\sum poids\ liens\ vers\ agrégat$/poids du nœud	Degré du nœud vers l'agrégat	*CA*	Le nœud est–il valide pour appartenance à l'agrégat étendu ?
V	(200 + 1000 +150) / 6000	3	0.675	Oui
W	2/3000	1	3.3 E-4	Non [Filtré]
X	(50 + 20 + 200) / 300	3	2.7	Oui
Y	(300 +50) / 1000	2	0.35	Oui
Z	200 / 5000	1	0.04	Non

Tableau 3.2 : Enrichissement de l'agrégat du graphe étudié figure 3.15.

Les nœuds sont ensuite classés par ordre décroissant de la valeur du coefficient d'attraction. En cas d'égalité si l'on veut limiter le nombre de nœuds par agrégat étendu, on utilise le poids du nœud comme élément départageant. Les mots les plus usités sont préférés aux mots rares, car, statistiquement une part importante de mots rares n'est que le fruit d'erreurs de frappe ou de fautes orthographe. L'ordre des nœuds est donc le suivant : X puis V et enfin Y. La figure 3.16 représente le coefficient d'attraction des nœuds {V, W, X, Y, Z} par rapport à l'agrégat noyau A1 du graphe étudié dans cet exemple.

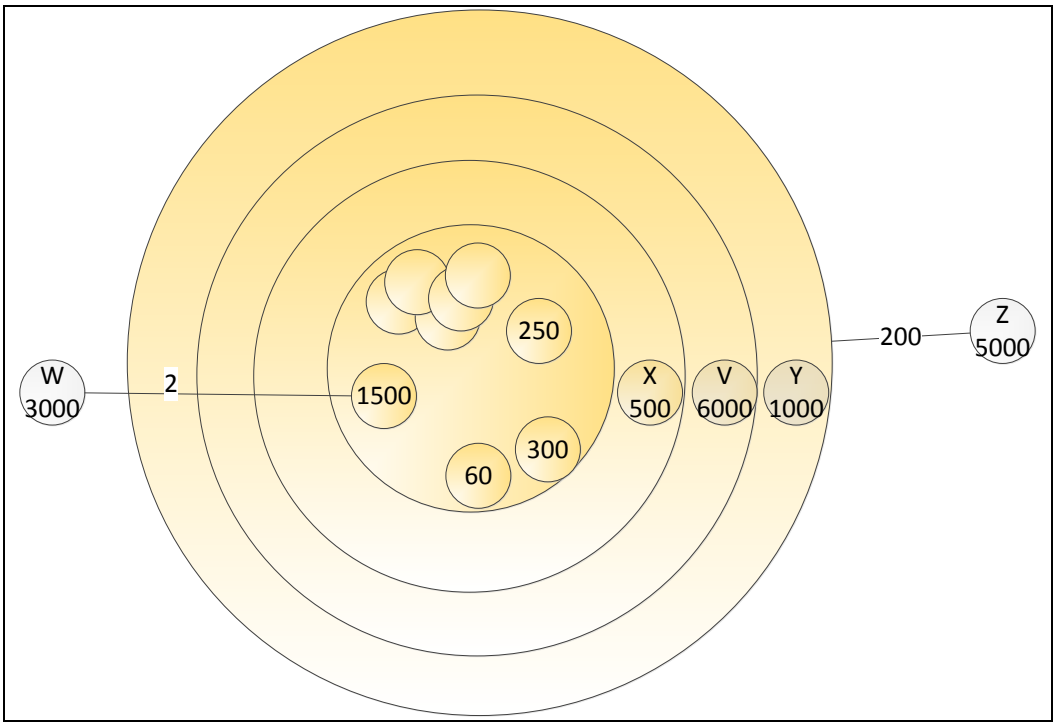


Figure 3.16 : Graphe d'un agrégat et nœuds sous attraction correspondant à l'agrégat étendu

3.6 Conclusion

Dans ce chapitre nous avons présenté trois méthodes de création d'agrégats et une méthode d'enrichissement d'agrégats. L'usage de plusieurs méthodes différentes sur un seul type de réseau (les réseaux de mots de grande taille), nous fournit des enseignements sur la nature interne de ces réseaux.

La méthode 1 ou détection de cliques

La méthode de création d'agrégats par détection de cliques a permis de valider le fait que nous ne pouvions faire l'impasse sur la pondération des liaisons. Une requête seule (sur plusieurs millions de requêtes) crée un agrégat de plusieurs mots sans que celui-ci soit valide d'un point de vue statistique.

La méthode 2 ou méthode de Rigidification Simple

La méthode de Rigidification Simple a permis de cerner que le regroupement contextuel était plus performant que la détection de cliques. Le contexte tenant compte du poids des mots et du poids relatif des liaisons. Cependant plusieurs problématiques sont alors apparues :

- les agrégats de très grande taille (plus de 100 mots) offrent peu de cohérence sémantique ;

- la nécessité de supprimer manuellement les mots vides (articles, conjonctions de coordination, …) pour ne pas créer des agrégats de taille trop importante nous fait également perdre des indications parfois précieuses ;
- certains mots faiblement utilisés deviennent de véritables verrous (ainsi par exemple, les mots mal orthographiés utilisés une seule fois, entretiennent des liens correspondant à 100% de leur usage ; ils participent donc de manière extrêmement active à la construction d'agrégats ; l'agrégat qui se veut un regroupement basé sur l'utilisation statistique des mots entre eux devient en fait un objet dépendant d'erreurs de frappe ou de problèmes d'orthographe dont la l'utilisation est très faible voire unique).

La méthode 3 ou Méthode de Rigidification Régulée

La méthode de Rigidification Régulée est une tentative de réponse à l'ensemble des problèmes rencontrés par la mise en œuvre de la méthode 2. Plus complexe, elle est une évolution de la méthode de Rigidification Simple et s'appuie, comme elle, sur les méthodes de rigidification de graphe proposées par Hoffman **[Hoffman&al-1999]** et développées par Jermann **[Jermann&al-2004]**.

Cette méthode propose de changer progressivement les règles à la fois de la rigidification et de l'appartenance au graphe en prenant en compte le contexte, afin de contenir la taille de l'agrégat. Ainsi, les conditions permettant à un nœud de rejoindre un agrégat ne sont plus statiques.

Si l'agrégat est trop peuplé, les règles vont changer : les mots de faible usage et ceux très usités vont être progressivement écartés. Les conditions de validation des liaisons sont, elles aussi, modifiées de façon à contenir la taille de chacun des agrégats.

Bien que développée et validée uniquement sur des graphes de mots, cette méthode paraît pouvoir être utilisée dans les réseaux sociaux, le nombre de Dunbar **[Dunbar-1992]** servant alors de *TMA*.

La méthode 4 ou méthode d'Enrichissements par Gravité

La méthode d'Enrichissements par Gravité, cherche à ordonner les mots dans des orbites de distance variable de l'agrégat. À titre d'exemple, la méthode de Rigidification Régulée a écarté des agrégats bon nombre de mots ayant été très peu utilisés. Il convenait donc de mettre au point une méthode capable de les réinsérer dans des agrégats enrichis. Cette technique est aussi utilisable si les agrégats sont donnés ou connus. C'est sur ce type d'exercice que nous la testons dans le chapitre suivant.

Ces méthodes peuvent encore évoluer, l'adaptation de la méthode aux réseaux de mots issus de requêtes pour la création d'agrégats sémantiquement cohérents est une démarche que nous n'avons qu'entamée. Les méthodes doivent aussi pouvoir être comparées et testées. Chaque modification doit être évaluée. L'évaluation de la cohérence sémantique d'un agrégat n'est pas simple. C'est sur cet aspect de notre travail que porte le chapitre suivant.

Méthodes	Ref	Famille	Graphe	Nb d'agrégats	Les +/-	Résumé
1. Détection de cliques		Recherche de forme : recherche de clique	~~orienté~~ **non orienté** ~~pondéré~~ **non pondéré**	Non prédéterminé = Egal au nombre de cliques	+ Faible coût computationnel - Faible cohérence sémantique - Obligation de prétraitements comme la suppression des mots vides	Chaque clique devient un agrégat
2. Rigidification Simple	**[Belbeze&al-2009-3] [Belbeze&al-2009-4]**	Méthode basée sur HLS et **GCSP**	**orienté** ~~non orienté~~ **pondéré** ~~non pondéré~~	Non prédéterminé	+ Méthode paramétrable - Valeurs des paramètres critiques et difficiles à déterminer - Obligation de prétraitements comme la suppression des mots vides et de certaines expressions	Agrégation autour d'une diade par création d'une composante bi-connexe
3. Rigidification Régulée	**[Belbeze&al-2009-1]**	Méthode basée sur HLS et **GCSP**	**orienté** ~~non orienté~~ **pondéré** ~~non pondéré~~	Non prédéterminé	+ Méthode paramétrable + Taille maximale contrôlée des agrégats + Paramètres déterminés par l'analyse du graphe et l'autorégulation ✓ Laisse des nœuds hors agrégats - Coût computationnel élevé	Agrégation autour d'une diade par création d'une composante bi-connexe avec règles de validation de la présence des nœuds et des liaisons régulées pour limiter la taille maximale des agrégats sans compromettre la création d'agrégats de petite taille
4. Enrichissement par gravité	**[Belbeze&al-2009-2]**	Méthode d'enrichissement de noyau	~~orienté~~ **non orienté** **pondéré** ~~non pondéré~~	N/A	+ Faible coût computationnel + Méthode permettant une appartenance pondérée du nœud à plusieurs agrégats • Permet de diminuer le nombre de nœuds hors agrégats	Rattachement des nœuds « hors agrégats » aux agrégats « noyaux » par l'utilisation d'un coefficient d'attraction. Ce coefficient est proportionnel au degré du nœud externe en attraction vers les nœuds interne au graphe et au nombre de nœuds internes liés au nœud externe. Il est inversement proportionnel au degré du nœud externe.

Tableau 3.3 : Synthèse des quatre méthodes proposées.

Chapitre 4.

Expérimentations, validations sémantiques et résultats de mesure

4.1 Introduction

Dans ce travail de recherche la phase d'expérimentation s'est révélée particulièrement longue en raison de la taille des graphes considérés.

Chaque méthode de regroupement proposée a été testée par des méthodes de validation sémantique différentes et sur plusieurs réseaux de mots. En effet, pour accéder à certains systèmes de validation, il a fallu accepter de ne pas toujours choisir le réseau de mots. Ainsi, pour s'insérer dans un « challenge » avec une validation manuelle, le réseau de mots dit E-donkey-5-mois a été un support imposé.

En modifiant les méthodes pour prendre en compte des réseaux sans aucune opération préalable (suppression des mots vides, mots très utilisés, …), les réseaux choisis ont aussi évolué pour aller vers des tailles plus importantes. Ceci a permis de mesurer les capacités des méthodes que nous avons mises au point, à créer des agrégats sémantiquement cohérents dans des méga-graphes.

4.2 Présentation des réseaux testés

Les algorithmes ont été testés sur six réseaux. Chacun fera l'objet d'une description plus détaillée quant à son contenu et à la façon dont il a été obtenu.

Les trois premiers réseaux sont tous issus du fichier de log d'AOL qui représente un extrait des requêtes de son moteur de recherche pour les mois d'avril et mai 2006.

On trouve parmi ceux-ci, deux réseaux filtrés (suppression de mots à faible sens) correspondant chacun à un jour de log :

1) AOL 17/04/2006 ;
2) AOL 17/03/2006.

Le troisième graphe étudié est le réseau complet et non filtré des deux mois de log d'AOL. Face au gigantisme de ce graphe nous avons limité l'étude aux agrégats contenant un des mots cibles. Ce réseau est nommé :

3) 100 mots dans AOL.

Les deux réseaux suivants sont constitués de mots tapés dans les moteurs de recherche de systèmes d'échanges « peer to peer ». On trouve :

4) E-donkey-10 semaine ;
5) E-donkey-5-mois.

Enfin le dernier réseau est issu d'un programme de campagne de validation de moteurs de recherche. Il s'agit du réseau :

6) TREC-Eval-5.

4.2.1 Les réseaux AOL

Le matériel : le « log d'AOL »

Un extrait des fichiers de log du moteur de recherche AOL.com est notre support. Cet extrait intègre trente-trois millions de requêtes effectuées du 1er mars 2006 au 30 avril 2006. Ces requêtes sont principalement rédigées en anglais. La structure du fichier intègre un identifiant, la date et l'heure de la recherche, le site éventuellement sélectionné ainsi que son rang (cf. figure 4.1).

AnonID	Query	QueryTime	temRanck	ClickURL
142	rentdirect.com	2006-03-01 07:17:12		
142	www.prescriptionfortime.com	2006-03-12 12:31:06		
142	staple.com	2006-03-17 21:19:29		
142	staple.com	2006-03-17 21:19:45		
142	www.newyorklawyersite.com	2006-03-18 08:02:58		
142	www.newyorklawyersite.com	2006-03-18 08:03:09		
142	westchester.gov	2006-03-20 03:55:57	1	http://www.westchestergov.com
142	space.comhttp	2006-03-24 20:51:24		
142	dfdf	2006-03-24 22:23:07		
142	dfdf	2006-03-24 22:23:14		
142	vaniqa.comh	2006-03-25 23:27:12		
142	www.collegeucla.edu	2006-04-03 21:12:14		

Figure 4.1 : Extrait du fichier de log AOL.com.

Ce fichier est mis à la disposition du public par la société AOL à des fins d'étude. Il est disponible sur le site http://gregsadetsky.com/aol-data.

Le réseau « AOL-17/04/2006 »

Afin de travailler sur un échantillon représentatif et néanmoins manipulable, nous avons fait le choix de limiter celui-ci à l'ensemble des requêtes d'une journée. La journée de référence prise aléatoirement est celle du 17 avril 2006.

Sur les requêtes de cette journée nous avons appliqué plusieurs règles :

1) Les mots-clés sont définis comme un ensemble de lettres sans espace. Tout espace est donc lu comme un séparateur de mots-clés.
2) Les guillemets ainsi que tous les éléments de ponctuation ont été ignorés et remplacés par des espaces.
3) Seuls les mots-clés utilisés dans une requête ayant deux mots et plus ont été conservés.
4) Seuls les mots possédant plus d'une lettre ont été conservés.
5) Certains mots non significatifs ont aussi été écartés de l'étude (cf. tableau 4.1).

.com	at be	Does	having	http if	l.	off	she	this	when	www.
al	been	dont	he	Il	la	on	so	to	where	you
all	by	el	her	in	like my	our	st	too	who	your
alt	can	elle for	here	is	ne	ours	st.	us	why	yourself
and	com	from	his	it keep	no	out	than	was	will	
are	de	had	how		of	re	th	we	with	
as	do		href				their	what	www	

Tableau 4.1 : Liste des mots exclus de l'étude en tant que mots non significatifs

- Nous avons ensuite écarté de l'étude une liste de mots considérés comme non significatifs car sur-utilisés (cf. tableau 4.2). Afin d'éviter de manipuler des mots au sens galvaudé par une trop grande utilisation, nous avons décidé de ne pas considérer les mots ayant été utilisés dans plus de 1000 recherches. Ecarter ces mots qui sont par définition les moins discriminants nous permet d'espérer éviter la construction de méga-agrégats centrés sur ces mots-clés.
  Le nombre total de recherches étudiées dans l'échantillon de la journée du 17 avril 2006 est de plus de 200 000. Ces mots sont au nombre de 14 (cf. tableau 4.2) sur 51994 mots-clés étudiés soit 0.027 % de l'échantillon.

Mots-clés	Nombre de requête	Mots-clés	Nombre de requête	Mots-clés	Nombre de requête
sale	1011	tax	1458	county	1884
york	1071	state	1532	pictures	2020
bank	1083	school	1539	new	2413
home	1139	sex	1560	free	3956
city	1273	lyrics	1561		

Tableau 4.2 Mots-clés exclus car utilisés dans plus de 1000 requêtes le 17/04/06.

Après avoir appliqué ces différents « filtres », l'objet de l'étude se présente comme un ensemble de : **51980 mots-clés utilisés dans 200646 requêtes.**

Dans ce réseau qui n'est pas un méga-graphe, l'objectif est de construire l'ensemble des agrégats possibles.

Le réseau « AOL-17/03/2006 »

Le réseau AOL-17/03/2006 est créé avec les mêmes règles que le réseau AOL-17/04/2006, la seule modification étant le filtrage sur la date des requêtes. Il contient **48568 mots-clés** et **197000 requêtes**.

Dans ce réseau l'objectif est aussi de construire l'ensemble des agrégats possibles.

Le réseau «100 mots dans AOL »

Ce réseau est constitué de l'ensemble du réseau du fichier log d'AOL des deux mois dans son entier et sans aucun filtrage. Le réseau est composé de **1 294 245 mots-clés** ou nœuds et **5 556 101 de liens**. Le nombre de requêtes considérées est de 21 059 661.

Son périmètre :

Sur ce méga-graphe, nous ne sommes pas en mesure de construire et ensuite de valider l'ensemble des agrégats possibles dans un temps raisonnable. Nous avons donc choisi 100 mots pour lesquels nous créerons tous les agrégats les incluant.

Les cent mots sélectionnés sont les dix premiers noms (propres ou communs) de dix œuvres écrites de références. Ces œuvres sont de nature différente. On peut les classer en cinq catégories :

1) Deux œuvres fondamentales de notre civilisation :
    a. la bible,
    b. la république de Platon.
2) Deux recherches scientifiques :
    a. « Le livre des révolutions » de Copernic,
    b. « De la relativité spéciale et générale » d'Albert Einstein.
3) Deux œuvres artistiques :
    a. « Your honesty » de Madona,
    b. « Roméo et Juliette » de W. Shakespeare.
4) Un site web : Linux.org
5) Trois reportages sur des conflits (cf. tableau 4.3) :
    a. Iran-Irack war,
    b. Russia et Georgia,
    c. Milosevic found dead.

Livres / sites Internet / œuvres artistiques	Mots
The Bible	book moses genesis begining god heaven earth form void darkness
Romeo and Juliet (W.shakespeare)	households dignity verona scene break mutiny civil blood hands foes
The republic (Plato)	socrates glaucon yesterday piraeus ariston prayers goddess manner festival thing
Books of revolutions (Copernic)	Holy father people revolutions spheres universe movement globe views stage
Your Honesty (Madona, 2003)	honesty choice talk love voice eyes closer baby crazy kind
Relativity special and general (Albert Einstein)	insight theory relativity readers scientific philosophical point view apparatus physics
The Iran-Iraq war : the politics of aggression (Farhang Rajaee – 1993)	Iraqui army border Kuwait August city oil persian gulf offencive
Linux.org (march 2008)	Linux Unix operating system [Torvalds]* linus assistance developers world gnu source
http://threatswatch.org/commentary/2006/04/russia-and-georgia-ready-for-w/ Russia and Georgia Ready For War : (Guest Contributor, Craig Martelle April 21, 2006)	russia georgia war middle east georgia verge wines moldovan rack
http://news.bbc.co.uk/2/hi/europe/4796470.stm Milosevic found dead in his cell (bbc news-11 March 2006)	Yugoslav President Slobodan Milosevic detention centre Hague tribunal autopsy suicide

Tableau 4.3 : La liste des 100 mots utilisés pour créer les agrégats (* le mot "Torvalds" est ignoré car il n'est pas présent dans le fichier d'AOL.).

L'idée est de partir d'un échantillon de mots issus d'espaces sémantiques différents permettant de créer des agrégats bien distincts. Toutefois, certains sujets portent sur la même thématique (sujets 1, 2 et 5) de façon à tester la capacité des méthodes d'agrégation sur des espaces sémantiques proches. Enfin, le fichier d'AOL étant essentiellement en anglais, c'est dans cette langue que les cent mots ont été choisis.

4.2.2 Les réseaux eDonkey

Les réseaux eDonkey sont des réseaux de partage de fichiers entre pairs. Conçus au départ pour permettre l'accès et le partage d'informations par tous et pour tous, ils sont souvent détournés. Ils sont utilisés pour le partage de fichiers soumis à des droits d'auteurs ou même de fichiers aux contenus illicites.

Le client le plus célèbre de ces réseaux est à cette date eMule. Dans ces réseaux « point à point », il n'est pas possible de connaître le contenu des échanges sans des accès et des équipements spécifiques. C'est en usurpant le rôle de serveur (serveurs effectuant les opérations d'inventaire et de recherche) ou de client que ces réseaux sont construits. Les deux réseaux de mots sélectionnés pour cette étude sont issus des fichiers eDonkey.

Le réseau « eDonkey-10-semaines »

La technique employée pour récupérer les requêtes utilisateurs ou les noms de fichiers échangés consiste en un rajout de serveurs « espions » dans le réseau. Les serveurs ont pour but, dans ces réseaux « point à point », de maintenir les listes des fichiers et leurs localisations, les fichiers restant physiquement sur les clients. Ainsi les serveurs espions peuvent répondre aux requêtes des utilisateurs en enregistrant celles-ci ainsi que les noms des fichiers échangés. La récupération de ce réseau est définie en détail dans l'article: « *10 weeks in the life of a eDonkey server* » **[Aidouni&al-2009]**.

Ce réseau est étudié dans le cadre de la lutte contre la pédophilie sur Internet. Plusieurs travaux incluant ce fichier ou d'autres du même type sont décrits sur le site : http://antipaedo.lip6.fr.

Le réseau est constitué par plus de 170 millions de requêtes faites par des utilisateurs recherchant des fichiers. Après avoir considéré uniquement les seules requêtes contenant plus d'un mot, il reste exactement **73 400 062 requêtes**. Le réseau comporte **2 833 164 de nœuds** et **68 millions de liaisons**. Nous n'appliquerons aucun filtre sur ce réseau.

Son périmètre :

Nous recherchons dans ce réseau les agrégats intégrant 18 mots particuliers (cf. tableau 4.4). Ces 18 mots cibles sont les « mots repères » fournis par Matthieu Latapy pour évaluer la méthode. Certains de ces mots sont des mots « bien connus » utilisés par les pédophiles. D'autres restent des mots « anonymes » que nous ne manipulons que par leur identifiant numérique. Nous ne connaissons ni leur signification ni leur orthographe.

Mots-clés	Texte	Poids		Mots-clés	Texte	Poids		Mots-clés	Texte	Poids
503664	Null	8		43970	1yo	433		28846	ptsc	3189
397675	Null	36		43170	2yo	536		26029	ygold	9183
314597	Null	65		38080	raygold	826		12603	incest	13619
39471	Null	114		166143	3yo	832		21847	pthc	45737
262249	Null	123		133912	4yo	1042				
62365	qqaazz	257		71725	inceste	1220				
112145	kidzilla	298		57572	incesti	1277				

Tableau 4.4 : Liste des mots fournis pour rechercher des agrégats les incluant.

Le réseau eDonkey-5-mois

Ce réseau est constitué de mots issus de noms de fichiers présents dans le réseau eDonkey. Un client eDonkey modifié a pendant 150 jours (environ 5 mois) interrogé des serveurs eDonkey en proposant comme requêtes des listes de mots « bien connus » comme étant utilisés par des pédophiles. Ce client a aussi demandé des fichiers à partir de mots plus génériques.

Les mots constituant les noms de fichiers représentent alors une composante connexe à intégrer au réseau. Dans ce réseau le lien entre les mots n'est donc pas : « utilisé conjointement dans une même *requête* », comme dans les autres réseaux étudiés, mais :

« présents ou dans un même nom de *fichier* ». Cette caractéristique ne change rien à la nature du réseau. C'est un réseau de mots dont les liens sont des utilisations conjointes. La pondération du mot est alors égale au nombre de fichiers dans lequel le mot apparaît. La pondération des liens est calculée à partir du nombre de fichiers où les mots sont utilisés ensemble.

Le réseau contient **2,8 millions de nœuds** distincts et **33 Millions de liens**. Il est défini plus en détail dans le document « *Automatic Identification of Paedophile Keywords* », disponible sur le site http://antipaedo.lip6.fr/T24/TR/keyword-detection.pdf **[Belbeze&al-2009-2]**.

Son périmètre :

L'enjeu du « challenge » **[Belbeze&al-2009-2]** est de trouver les 100 mots qui sont les plus pertinents comme mots utilisés en conjonction de deux listes de mots. Ces listes sont les suivantes : [*child, sex, child, porn, 1yo, 2yo, 3yo ; 4yo, 5yo, 6yo, 7yo ; 8yo,9yo,10yo,11yo, 12yo*] et [*qqaazz, aabbccddee, babyshivid, hussyfan, pthc, ptsc, r@ygold, kingpass*].

4.2.3 TREC-Eval

Le but initial de TREC-Eval est la comparaison entre différents moteurs de recherche. TREC-Eval peut comparer une liste de documents retournés par un moteur de recherche à une liste de documents théorique parfaite. Pour cela, on demande à des utilisateurs de formuler des requêtes. Les documents indexés sont ensuite classés par ordre de pertinence, manuellement par les utilisateurs. En rejouant les requêtes, les résultats retournés par les moteurs de recherche en évaluation sont ensuite comparés à la liste « idéale » proposée par les utilisateurs. Pour chaque couple {requête, document}, une note de pertinence est donnée. Ces notes sont stockées dans un fichier QREL « Query Relevance Judgments » . Chaque année une campagne de cinquante requêtes est rajoutée. La campagne s'effectue sur 300 000 articles. TREC-Eval est disponible en téléchargement sur le site : http://trec.nist.gov/

Mais, cinquante requêtes incluant juste quelques mots sur des sujets très différents sont insuffisantes pour créer des graphes suffisamment connectés et avec des pondérations signifiantes comme cela serait nécessaire pour créer des agrégats. Afin d'avoir plus de matière nous avons utilisé cinq campagnes de TREC-Eval et nous avons rajouté en tant que requête la description de la requête elle-même. Cette description est un texte court de quelques mots. La base disponible est donc au total cinq cents requêtes (250 requêtes correspondant au requêtes Trec-val et 250 requêtes correspondant à la description) et 150 000 articles.

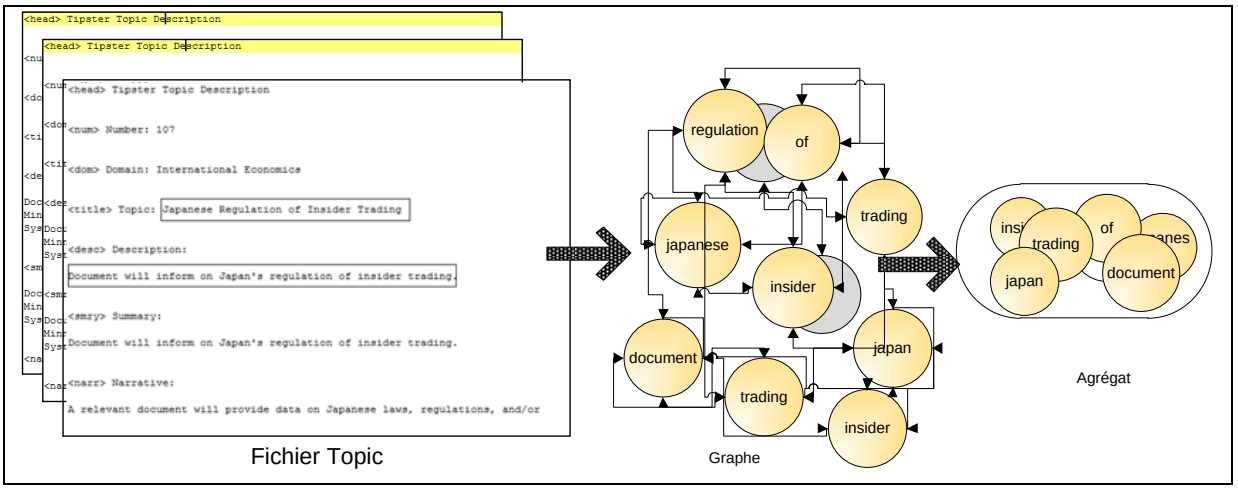


Figure 4.2 : Processus de création du réseau à partir des fichiers "Topics" de TREC-Eval.

Les requêtes et leur description sont stockées dans des fichiers « Topics » correspondant à chacune des campagnes annuelles. En excluant les requêtes d'un seul mot, il reste 199 requêtes et 250 descriptions pour un total de 944 mots-clés. L'ensemble du réseau est utilisé, aucun filtre n'est appliqué.

4.3 Les méthodes de validation sémantique

Afin, d'obtenir une estimation de la cohérence sémantique des agrégats, nous avons mis au point trois méthodes.

La première méthode effectue une validation par comparaison des comportements de populations de requêtes identifiées (aléatoires, écrites par un utilisateur, etc.). Différents comportements statistiques sont étudiés en fonction du nombre de sites retournés par un moteur de recherche. Nommée **MCCVS** pour Méthode Comparative de Coefficient de Validation Sémantique (cette méthode est de notre invention).

La deuxième méthode est une mesure de la capacité à améliorer la qualité des résultats retournés par un moteur de recherche en enrichissant la requête des mots d'agrégats associés aux mots de la recherche de base. Nous utilisons pour cela l'outil **TREC-Eval**.

La troisième méthode est très proche de MCCVS dans sa construction. Simplement, au lieu de comparer le nombre de sites retournés nous comparons la distance entre les documents proposés par le moteur de recherche.

4.3.1 Méthode MCCVS ou « Méthode Comparative de Coefficient de Validation Sémantique »

MCCVS permet la mesure de la qualité sémantique par comparaison du nombre de sites retournés par un moteur de recherche en utilisant comme requête des populations (par population nous entendons un ensemble nœuds partageant une caractéristique) de mots

identifiées **[Belbeze&al-2009-3]**. Elle a l'avantage de donner un coefficient qui est l'équivalent d'une note.

Nous considérons que si des mots sont associés dans une page web ou une requête par la volonté d'un auteur, ils sont associés « sémantiquement ».

Les postulats à la proposition d'une technique de validation sémantique par comparaison de populations de mots identifiées sont les suivants :

- Internet est majoritairement constitué de sites web et de documents sémantiquement cohérents. Nous convenons qu'il existe des exceptions telles que des dictionnaires ou des listes d'objets en vente mais les considérons comme numériquement faible et donc peu représentative. De plus, la nature comparative de la méthode permet de baisser l'influence de tels éléments.

- Les utilisateurs de moteurs de recherche sur Internet ont une conscience et une expérience suffisante pour utiliser des mots-clés ayant un lien entre eux et avec le sujet recherché.

Sur des ensembles de mots et de recherches suffisamment importants pour effectuer un traitement statistique, il devrait être possible d'observer un comportement différent, lorsque l'on compare le nombre de sites retournés par des requêtes utilisateurs à celui retourné par des requêtes combinant des mots de manière aléatoire.

Afin d'éclairer notre propos, nous soumettons en tant qu'utilisateur, trois recherches de trois mots-clés au moteur de recherche Google.com et une recherche combinant un mot-clé de chacune de ces recherches utilisateurs (cette dernière étant notre recherche aléatoire).

Comme on peut le constater dans les exemples illustrés dans le tableau 4.5 trois mots-clés pris aléatoirement dans un ensemble de requêtes donnent des résultats significativement inférieurs en nombre de sites retournés à des requêtes plus « sémantiquement cohérentes » proposées par un utilisateur. Ceci n'a bien sûr de valeur que d'un point de vue statistique ; rien n'interdisant à un monsieur « Besancenot » de placer une photo de lui-même sur Internet jouant du « B3 », modèle célèbre d'orgue de la fameuse marque Hammond, devant un plat d'épinards et de décrire celle-ci, sans que cela modifie nos observations.

ID	Requête	nb de sites retournés
1	+besancenot +état +France	850 000
2	+épinard +crème +beurre	388 000
3	+organ +hammond +B3	86 200
4	+B3 +besancenot +épinard	0

Tableau 4.5. Nombre de sites retournés par le moteur de recherche du site Google.com en fonction de la cohérence sémantique de la requête (Novembre 2010).

Recherche du coefficient « *CVSC* » ou Coefficient de Validation Sémantique Comparé.

Notre but n'est pas de fournir une méthode de validation sémantique absolue, mais d'obtenir un indice de qualité sémantique. Cet indice est défini comme un ratio et n'a donc pas d'unité. Il permet d'évaluer des méthodes de regroupements et leurs évolutions (modifications apportées). Afin de créer cette mesure, nous proposons de comparer le nombre de sites retournés par le moteur de recherche du site AOL.com à partir de requêtes basées sur des combinaisons extraites des agrégats eux-mêmes avec le nombre de sites retrouvés (par le même moteur de recherche) à partir de requêtes basées sur des combinaisons aléatoires de mots-clés (combinaisons indépendantes des agrégats construits).

Trois mots-clés représentent la taille minimale d'un agrégat (triade). Il est donc impossible de construire des recherches utilisant plus de trois mots-clés sans exclure de cette mesure les agrégats les plus petits. La validation des mots-clés par paires pourrait sans doute présenter un intérêt mais représenterait un nombre de combinaisons trop important. Nous avons donc choisi de présenter les mots-clés au moteur de recherche d'AOL.com par trio.

Toutes les combinaisons de trois mots-clés de chaque agrégat seront soumises au moteur de recherche ainsi qu'un nombre équivalent de groupes de trois mots associés aléatoirement et enfin des groupes de trois mots issus de requêtes utilisateurs. Le moteur de recherche interrogé est soit AOL.com soit Bing.com selon les expérimentations. Nous n'avons pas pu conserver le même moteur de recherche sur la totalité d'entre-elles, AOL.com ayant mis en place un système de détection de robots en janvier 2009. Le moteur de recherche utilisé est précisé dans les conditions de mesure de chaque expérimentation.

Nous comparons alors la distribution du nombre de sites retournés par chacun des groupes de trois mots, afin de comprendre si les trios de mots issus des agrégats sont plus proches du comportement des trios aléatoires ou des trios de mots issus des requêtes utilisateurs.

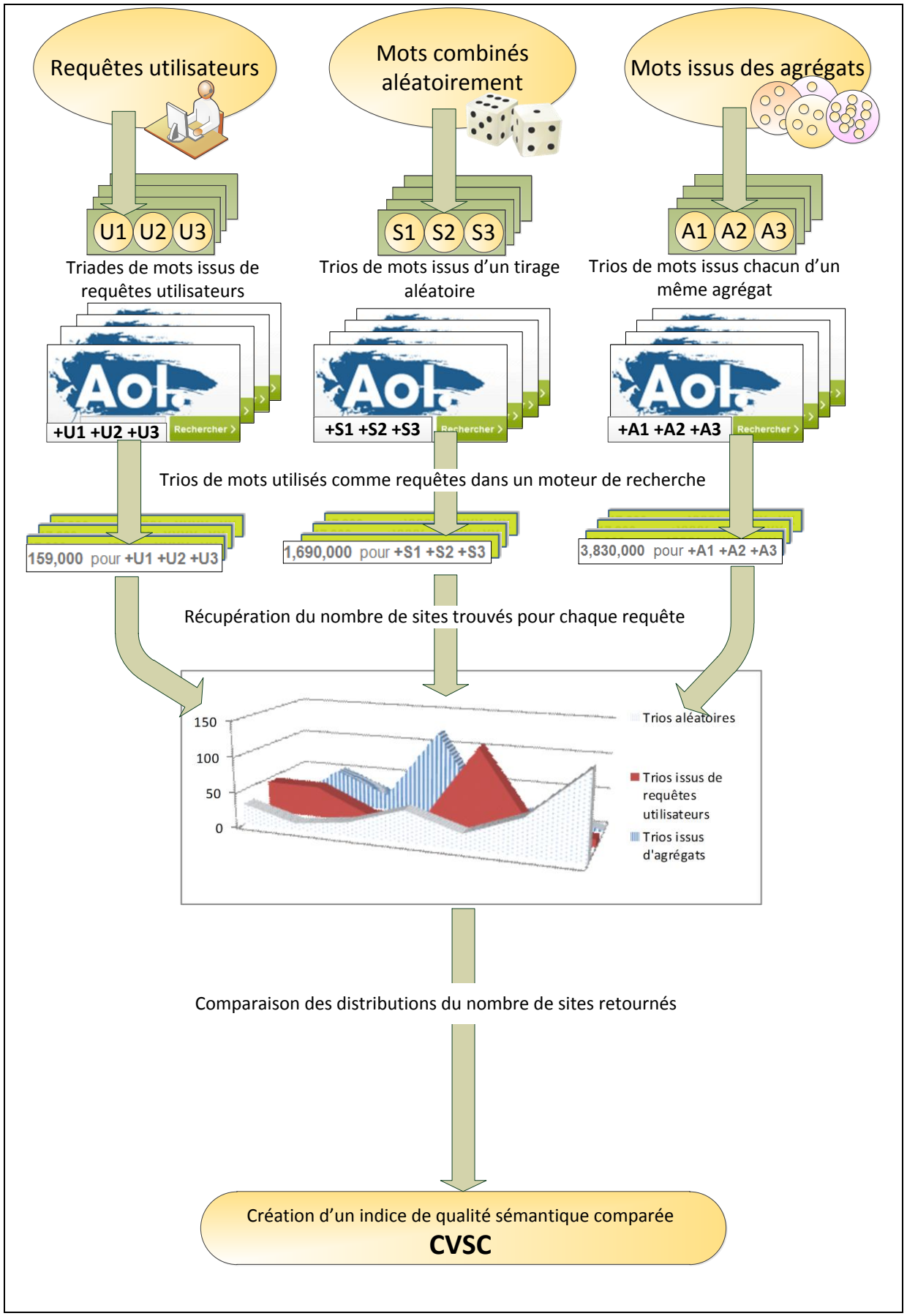


Figure 4.3 : Principe de création de l'indice sémantique.

Recherche d'un élément de comparaison

Une représentation graphique du nombre de sites retournés en fonction d'une population se heurte à quelques difficultés. L'étendue des valeurs de retour et le nombre de valeurs différentes retournées sont trop considérables pour en proposer une vision graphique. Dans notre cas, nous allons de « 0 » site retourné à plus de 99 millions de sites pour certaines requêtes.

Pour pallier ces difficultés, nous représentons les résultats selon une échelle semi-logarithmique en utilisant un regroupement des valeurs dans des classes. Un repère semi-logarithmique est un repère dans lequel l'un des axes, ici celui des ordonnées (y), est gradué selon une échelle linéaire alors que l'autre axe, ici celui des abscisses (x), est gradué selon une échelle logarithmique. L'avantage d'une représentation semi-logarithmique est son aptitude à représenter des mesures qui s'étalent sur des valeurs extrêmement larges. Des représentations semi-logarithmiques en puissance de 2 ont déjà été utilisées par Zipf dans des études sur l'occurrence des mots à l'intérieur d'un texte **[Zipf-1935]**. Ainsi, dans la figure 4.4, l'axe des abscisses est gradué en puissance de 2. En effet, pour pouvoir comparer les résultats obtenus, nous avons regroupé le nombre de sites retournés dans des classes exprimées dans un espace logarithmique. Si les échelles logarithmiques sont habituellement en puissance de 10, afin de présenter une échelle plus détaillée, nous avons choisi des classes par puissance de 2. L'axe des ordonnées représente alors le pourcentage de combinaisons trouvées par classe par rapport à l'ensemble des classes.

Recherche d'une zone identifiable comme zone de comportement différentiable

Nous comparons maintenant les deux courbes de réponses des deux espaces les plus éloignés sémantiquement selon le postulat posé au paragraphe précédent. Nous comparons la courbe issue des mots combinés aléatoirement (excluant des triades de mots utilisés dans une recherche) avec la courbe de référence issue du test de triades pour laquelle il existe au moins une recherche incluant ces trois mots-clés.

Les regroupements de trois mots combinés aléatoirement ou les regroupements de trois mots extraits d'un même agrégat sont nommés **trios** de mots. Contrairement aux regroupements de trois mots ayant été conjointement utilisés dans une même requête utilisateur que nous nommerons **triades**. Ces derniers sont d'un point de vue graphique un ensemble de trois nœuds reliés deux à deux, ce qui n'est pas le cas pour les mots combinés aléatoirement et peut ne pas l'être pour des mots issus d'agrégats.

Notre but est de comparer les distributions en fonction du nombre de sites retournés d'éléments le plus sémantiquement éloignés. Nous choisissons donc en suivant les postulats posés en début du paragraphe 4.2.1, les requêtes posées par un utilisateur et celles générées aléatoirement.

La figure 4.4 est un exemple du résultat obtenu en comparant ces deux types de requêtes. En ordonnée de la figure 4.4 les valeurs représentent le pourcentage de la classe donné en abscisse. Ainsi la classe « 0 » ou « aucun site retourné » représente 57% de

l'ensemble des tests effectués avec des trios de mots générés aléatoirement. De la même manière la classe notée 2^8 (qui correspond à un nombre de sites entre 129 et 256) correspond à 12 % de l'ensemble des tests pour les « triades de mots utilisées dans une requête utilisateur au moins ».

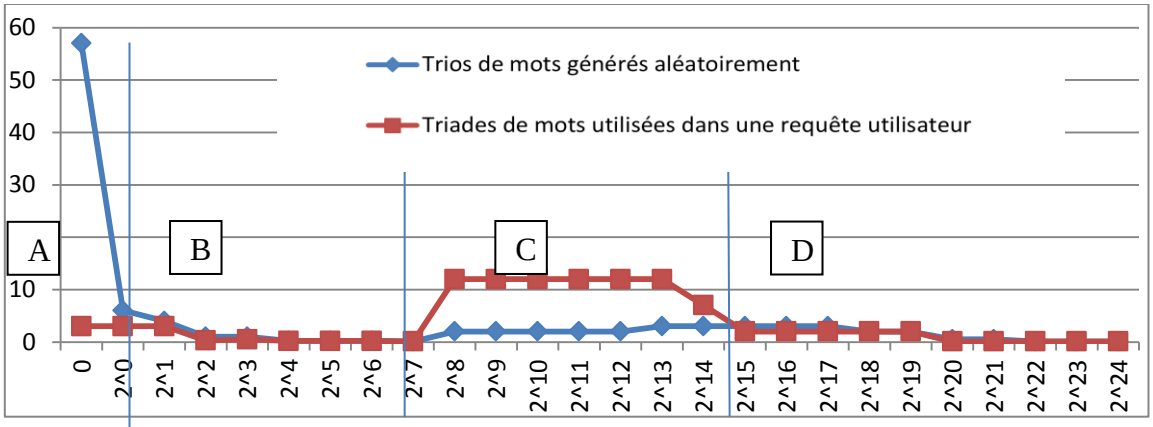


Figure 4.4. Comparaison des deux courbes susceptibles d'être les plus éloignées sémantiquement et détermination des zones à fortes divergences.

Dans cet exemple nous pouvons noter quatre zones distinctes. Les zones « B » et « D » n'offrent pas beaucoup d'intérêt, les courbes ne présentant pas de différence notable. La zone « A » est limitée à une seule valeur et ne peut donc représenter une étendue suffisante pour mener notre étude.

Nous devons repérer la zone qui exprime le plus la différence de comportement entre les deux types de requêtes. Cela nous permettra d'affiner notre analyse en nous concentrant sur un comportement véritablement différencié. Dans notre exemple de la figure 4.4, la zone « C » (cf. figure 4.5) est la zone la plus « différenciée ». Elle est, en outre, d'une plage suffisante pour nous permettre une comparaison nuancée.

Dans cet exemple, la zone « C » nous sert de zone de validation sémantique. Afin d'élaborer une comparaison rapide et arithmétique, nous allons définir un coefficient approprié.

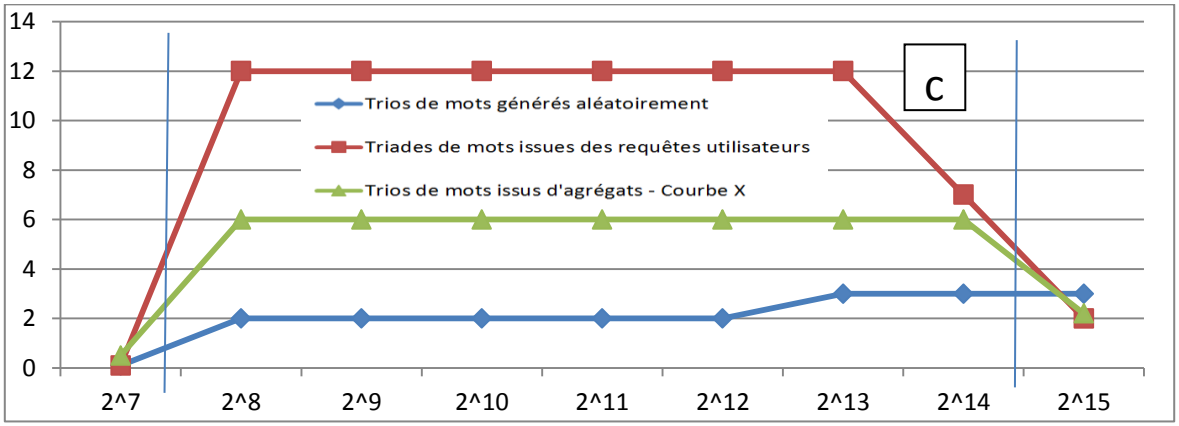
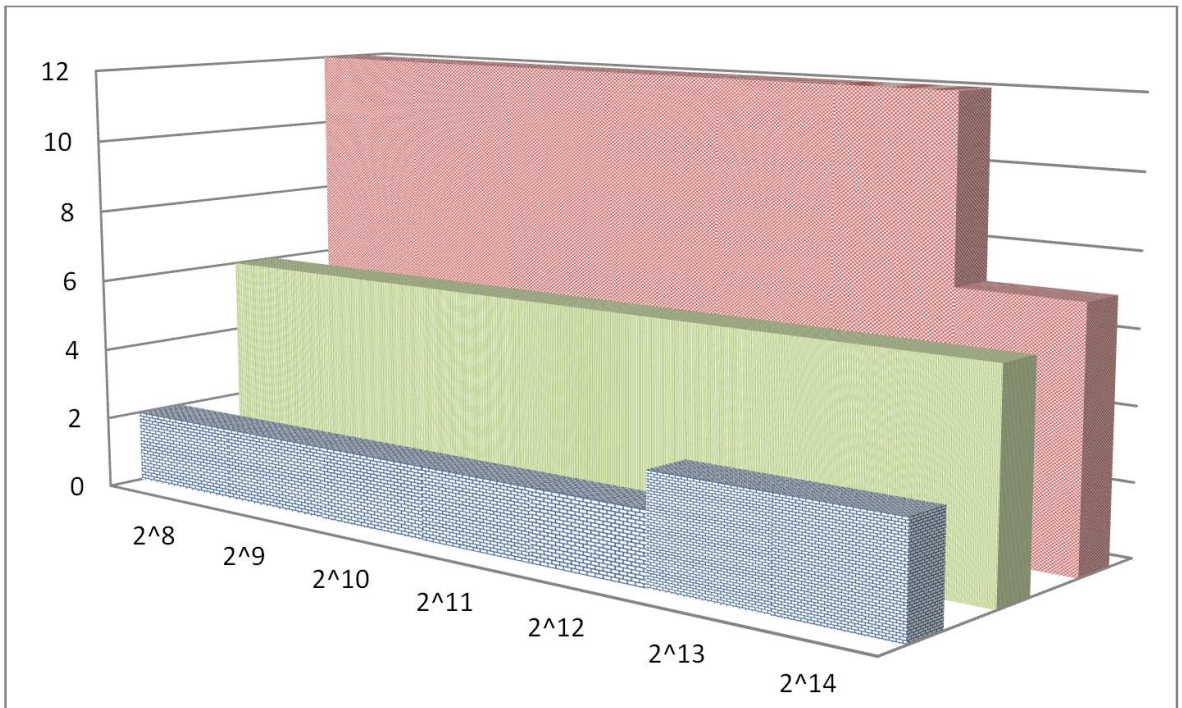


Figure 4.5 : Courbe de distribution du nombre de sites retournés en fonction de la nature de la source du trio de mots constituant la requête dans la zone « C » et incluant une courbe X.

Calcul du CVSC (Coefficient de Validation Sémantique Comparée)



	2^8	2^9	2^10	2^11	2^12	2^13	2^14
Trios de mots générés aléatoirement	2	2	2	2	2	3	3
Trios de mots issus d'agrégats - Courbe X	6	6	6	6	6	6	6
Triades de mots issues des requêtes utilisateurs	12	12	12	12	12	12	7

Figure 4.6 : Aires des distributions du nombre de sites retournés en fonction de la nature de la source du trio de mots constituant la requête dans la zone « C » et incluant une courbe X.

Nous cherchons maintenant à calculer le coefficient de validation sémantique d'une courbe *X* qui provient des mesures effectuées à partir des trios de mots issus d'agrégats (cf. figure 4.5). Nous considérons que les classes en puissance de deux forment une échelle d'indice « un » et comparons l'aire prise par les histogrammes (cf. figure 4.6). Le *CVSC*, ou Coefficient de Validation Sémantique comparée, a alors la valeur « 1 » pour l'équivalence de l'histogramme des triades (de trois mots-clés) ayant été au moins une fois utilisées dans une même recherche et 0 pour la valeur de l'histogramme des triades aléatoires.

La formule mathématique de CVSC sera donc définie pour une courbe particulière *X* comme suit :

$$CVSC_X = (A_X - A_A) / (A_R - A_A)$$

Où A_R définit l'aire de l'histogramme des triades dont tous les mots sont inclus au moins une fois tous ensemble dans une recherche. A_R est défini comme suit :

$$A_R = \sum_{i=Début-Zone-C}^{Fin-Zone-C} Yi$$

Où A_A définit la valeur de l'aire de l'histogramme des trios de mots combinés aléatoirement. A_A est défini comme suit :

$$A_A = \sum_{i=Début-Zone-C}^{Fin-Zone-C} Y'i$$

Où Ax définit la valeur de l'aire de l'histogramme des trios de mots à comparer. A_x est défini comme suit :

$$A_X = \sum_{i=Début-Zone-C}^{Fin-Zone-C} Y''i$$

Dans notre exemple (cf. figure 4.5 et figure 4.6), les valeurs seraient les suivantes :

$$A_R = \sum_{i=8}^{14} Yi = 72$$

$$A_A = \sum_{i=8}^{14} Y'i = 42$$

$$A_X = \sum_{i=8}^{14} Y''i = 17$$

$$CVSC_X = (A_X - A_A) / (A_R - A_A) = 0{,}441$$

Conclusion

La méthode d'évaluation MCCVS est basée sur le postulat que statistiquement les documents présents dans Internet et les recherches effectuées par les utilisateurs sont des éléments possédant tous deux une cohérence sémantique certaine. Le postulat inclut un autre point qui est que, statistiquement, des requêtes constituées aléatoirement présentent une cohérence sémantique plus faible que celles écrites par un utilisateur.

Si tel est bien le cas, nous devrions voir émerger sur un nombre important de requêtes une distribution du nombre de documents trouvés différente entre des requêtes sémantiquement cohérentes qui sont celles créées par des utilisateurs et des requêtes créées aléatoirement.

Cette différence de comportement, si on accepte le postulat, devient alors un élément nous permettant de valider la méthode elle-même. En effet, si la différence de comportement est suffisamment sensible, cela permet de valider que les requêtes sont bien de nature

différente. Cette nature est liée à la source des requêtes : soit il s'agit d'un esprit humain animé par une intention soit c'est un système d'association aléatoire.

En respectant des conditions de mesure identiques pour les trois sources de requêtes, en opérant sur un nombre de requêtes statistiquement significatif et en vérifiant que la différence entre les deux sources de référence (utilisateur et aléatoire) est sensible, il est alors possible de créer un coefficient pour évaluer la nature sémantique de la source des requêtes.

La méthode MCCVS que nous proposons, permet en respectant des conditions et un protocole de mesure de donner une valeur CVSC sur la cohérence sémantique des agrégats.

4.3.2 Méthode TREC-Eval : enrichissement de requêtes

Le principe de mesure

TREC-Eval est un outil basé sur un ensemble de requêtes et de documents destiné à évaluer la qualité d'un moteur de recherche en fonction de la pertinence des documents retournés. Mais, ce n'est pas là notre objectif avec TREC-Eval. En effet, nous allons utiliser un seul moteur de recherche, mais nous allons modifier les requêtes utilisateurs en les enrichissant avec des mots trouvés dans les agrégats et nous comparerons la « qualité » des requêtes originales avec celles enrichies.

Nous utilisons comme moteur de recherche « **Terrier »**. Ce moteur est disponible sur le site : http://terrier.org/

La démarche consiste à enrichir des requêtes utilisateurs avec les mots des agrégats associés aux mots déjà présents dans la requête et à comparer la pertinence des documents obtenus.

TREC-Eval permet, en effet, de comparer la qualité des résultats obtenus entre deux requêtes. Si cet outil est généralement utilisé pour comparer des moteurs de recherche sur une même requête, rien n'empêche d'utiliser un seul moteur de recherche et de comparer la « qualité » de requêtes différentes.

Pour chaque requête effectuée par Terrier nous sauvegardons les résultats dans un fichier au format attendu par TREC-Eval. Ce type de fichier est nommé fichier « Run ». Pour chaque requête TREC-Eval possède une liste de réponses « idéales ». Cette liste de réponses a été constituée manuellement et stockée dans un fichier « Qrel ». En comparant les résultats obtenus par la requête (fichier « Run ») et ceux de la réponse idéale (fichier « Qrel ») TREC-Eval note la qualité du résultat obtenu (cf. figure 4.7).

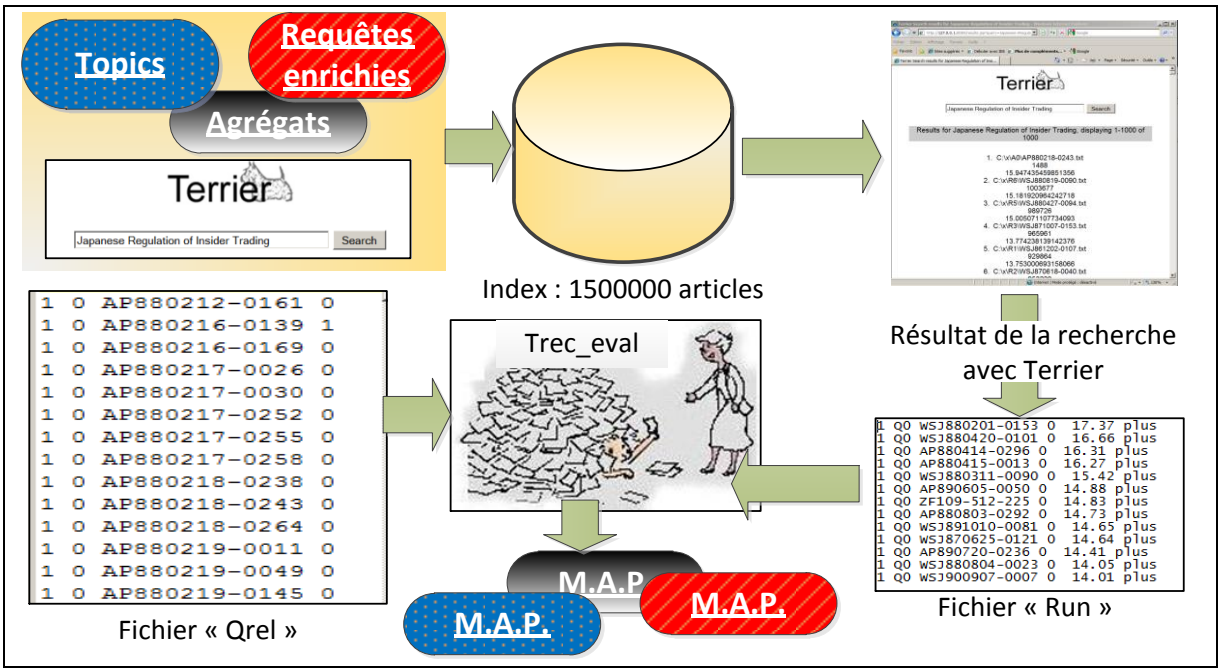


Figure 4.7 : Comparaison de résultats entre des « Topics », des agrégats et des « Topics » enrichis comme requêtes dans un même moteur de recherche « Terrier » avec TREC-Eval.

Le fichier « Run » est constitué de la liste des documents retournés par le moteur de recherche Terrier. Nous utilisons la note de pertinence ou "Rank Note" donnée par Terrier comme note dans le « Fichier « Run ».

Dans TREC-Eval une requête utilisateur est nommée « Topic ». Nous pouvons, pour une requête utilisateur de référence ou « Topic », comparer les résultats obtenus en utilisant comme requête :

- le « Topic » seul ;
- le « Topic » enrichi avec des mots issus d'agrégats ;
- l'agrégat lui-même utilisé comme requête.

Paramétrage de Terrier

Le paramétrage de Terrier dans notre cas est très limité. Il consiste à définir le nombre de documents maximum retournés pour chaque requête. Afin de pouvoir détecter des variations fines sur la qualité des requêtes, nous avons paramétré Terrier pour retourner les 1000 premiers articles. Ce paramétrage a déjà été utilisé avec succès par Chris Buckley et Ellen M. Voorhees **[Buckley&al-2004]**.

La mesure de la qualité de la requête

TREC-Eval est capable de mesurer un grand nombre de paramètres de cohésion entre la requête et les résultats retournés. Dans nos tests nous avons choisi de mesurer les valeurs de M.A.P. (Mean Average Precision) valeur de comparaison généralement la plus utilisée. La

valeur M.A.P. est définie comme la moyenne des précisions obtenues chaque fois qu'un document pertinent est retrouvé. Si d'autres mesures sont possibles il convient d'en relativiser l'importance ici. En effet, nous mesurons comparativement des populations de requêtes. Le choix du type de mesure n'a donc comme contrainte principale que celle de posséder la capacité à notifier une évolution de qualité.

La précision est définie en Recherche d'Information (R.I.) comme le nombre de documents pertinents retrouvés, rapporté au nombre total de documents retrouvés. Pour une requête *q*, *Rq* étant le nombre de documents retournés et *Pq* le nombre de documents pertinents, la précision se définit comme suit :

$$précision_q = \frac{P_q}{R_q}$$

La Mean Précision ou correspond à un calcul de la moyenne des précisons calculée pour chaque document pertinent retourné.

Supposons un Topic pour lequel il existe cinq documents pertinents appartenant à l'ensemble P_t défini comme suit :

$$P_t = \{dp_1 ; dp_2 ; dp_3 ; dp_4 ; dp_5\}$$

Supposons un ensemble de documents retourné Rq_t par une requête sur ce même Topic défini comme suit :

$$Rq_t = \{dp_1 ; dp_4 ; da ; db ; dp_5 ; dp_3 ; dc ; dd ; de ; dp_2\}$$

La précision sera pour chaque document pertinent la suivante :

dp_1	dp_4	da	db	dp_5	dp_3	dc	dd	de	dp_2
1	*1*	-	-	*0.6*	*0.833*	-	-	-	*0.5*

La Mean Precision étant la moyenne des précisions rencontrées est dans notre exemple : *MP* = (1+1+0.6+0.833+0.5) / 5= 0.7866

Pour un ensemble de requêtes, la MAP ou Mean Average Precision est la moyenne de l'ensemble des Mean Precision de l'ensemble des requêtes du jeu de test.

Conclusion

Nous savons que rajouter des mots dans une requête n'est pas toujours synonyme de meilleure performance et cela même si ces mots sont effectivement liés au contexte recherché. En effet, la recherche s'effectuant sur un grand nombre d'articles, l'ajout de mots peut en quelque sorte « flouter » la requête en ramenant des articles moins spécifiques **[Boughamem&al-1997]**.

Rajouter des mots en maintenant un même niveau de qualité de requêtes est donc déjà un défi qui nous permet de confirmer que le système d'agrégation est bien porteur d'une cohérence sémantique.

4.3.3 Méthode MCCDR ou « Méthode de Comparaison de Cohérence de Documents Retournés »

Principe et hypothèse

Cette méthode s'appuie sur la comparaison des trois populations typiques déjà rencontrées :

- trios de mots aléatoires ;
- triades présentes dans au moins une requête d'utilisateur ;
- trios de mots issus d'un même agrégat.

Le but est de mesurer la cohérence des documents retournés et de comparer ensuite la distribution de cette cohérence en fonction de la nature des requêtes. Pour cela nous mesurons la distance moyenne entre documents retournés par les trois types de requêtes.

L'hypothèse est la suivante : plus les mots d'une requête font référence à un espace sémantique précis, plus les documents retournés par un moteur de recherche dans une requête sont proches les uns des autres. Si cette hypothèse est vérifiée, nous devrions observer une différence entre les documents retournés par des requêtes construites aléatoirement et ceux obtenues avec les requêtes rejouées des utilisateurs.

Si les agrégats sont sémantiquement cohérents, on peut espérer une similarité importante et proche des « triades issues des requêtes utilisateurs » dans les documents retournés par les trios de mots issus des agrégats. La distance moyenne entre documents doit être aussi faible que possible à l'intérieur d'un ensemble de documents retournés par les requêtes effectuées à partir des mots d'un agrégat.

À contrario, la distance moyenne entre les documents retournés en utilisant deux agrégats comme sources de création de requête devrait être la plus élevée possible. C'est ce que nous tentons de mesurer en comparant deux à deux les documents récupérés par des requêtes issues de deux agrégats différents.

Requêtes et moteurs de recherche

Les requêtes sont formées de trois mots et issues d'une des trois familles (aléatoires, issues des requêtes utilisateurs et issues des agrégats). Elles sont constituées sur le même ensemble de mots.

Le moteur de recherche utilisé est bing.com. La recherche est limitée au contenu de Wikipédia par l'utilisation de la syntaxe « site : wikipedia.org ». Les articles de plus de 15000 mots et de moins de 200 mots sont écartés. Les articles de plus de 15000 mots sont en général

des sites agglomérant des listes d'articles archivés, ils sont sans valeur sémantique globale. Les articles de moins de 200 mots sont le plus souvent des articles en préparation qui ne contiennent pas vraiment d'informations ou simplement un message d'erreur (lien mort, erreur sur le site, …). Les recherches dans bing.com se font sans aucun filtre (langues, contenu pour adultes, …)

Calcul de la cohérence des documents retournés

Pour chaque requête, nous stockerons les dix premiers articles de Wikipédia valides (plus de 200 mots et moins de 15 000 mots) présentés dans la liste. L'ensemble des documents retournés par bing.com pour l'ensemble des requêtes est ensuite filtré pour enlever le code (html, xhtml, java, vbs, …) puis stocké dans une base de données.

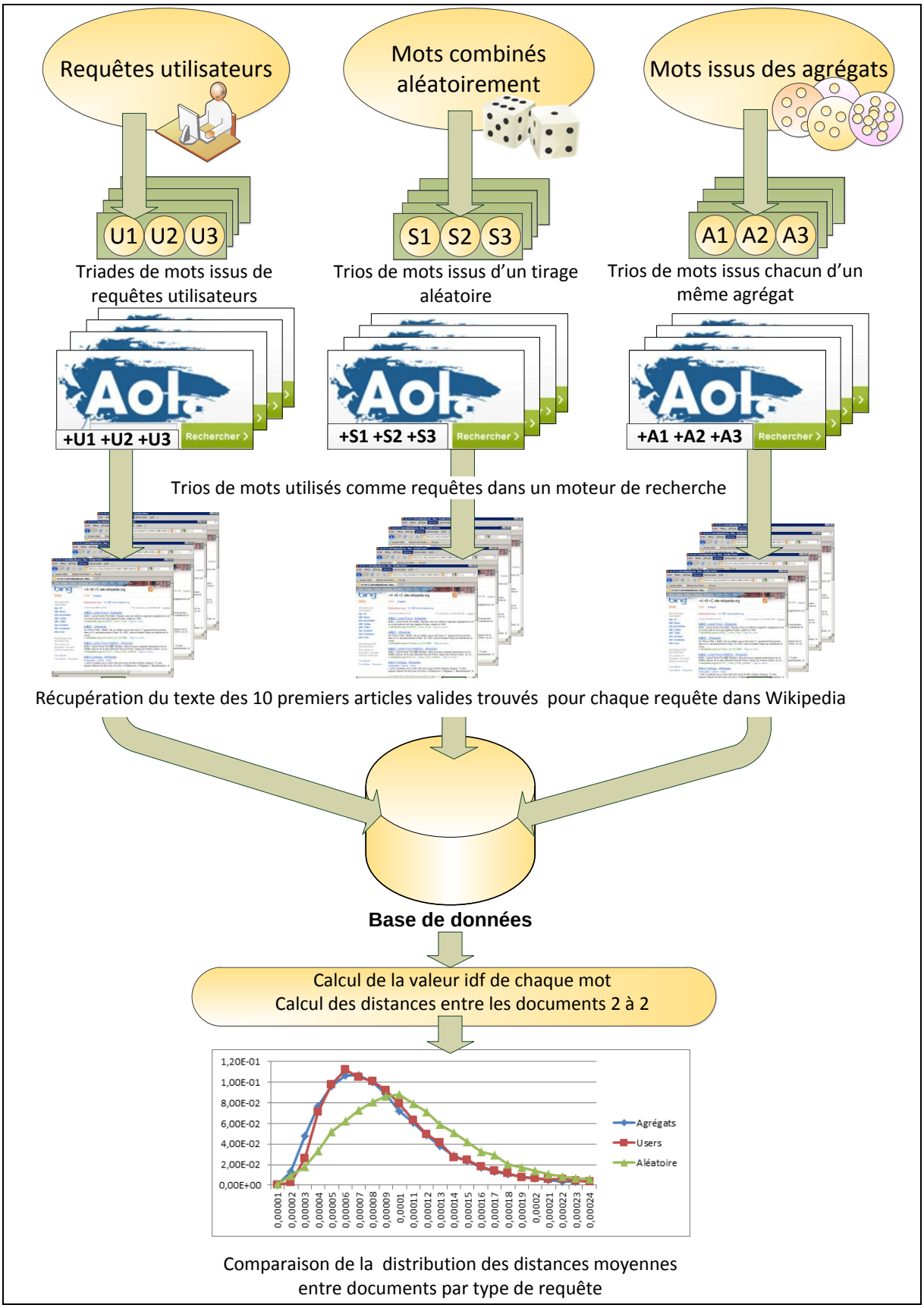


Figure 4.8 : Méthode de récupération des articles de Wikipedia et calcul de similarité entre documents retournés par type de requête.

Pondération des mots et tf.idf

Afin de comparer les articles de la façon la plus efficace possible, nous utilisons un système de pondération. Ce système connu sous le nom de « tf.idf » a pour but de valoriser les mots rares au sein du corpus et de permettre la comparaison de documents de tailles très différentes **[Salton&al-1983]**.

idf :

Chaque mot se voit affecter d'une valeur en fonction de sa présence dans un plus ou moins grand nombre de documents. La pondération est inverse au nombre de documents dans lequel le mot a été trouvé. Ainsi les mots présents dans un grand nombre de documents ne seront pas représentatifs. Au contraire, un mot trouvé uniquement dans quelques documents, présentera un poids particulièrement élevé. La fréquence inverse de présence du mot dans le corpus de documents (*inverse document frequency*) est notée *idf*. L'*idf* est aussi appelé « fonction inverse de la fréquence absolue ».

On calcule l'*idf* par la formule :

$$idf = log \frac{|D|}{|\{d_j : t_i \in d_j\}|}$$

Où $|D|$ représente le nombre de documents du corpus et $|\{d_j : t_i \in d_j\}|$ le nombre de documents dans lesquels le terme t_i apparaît. Le but est d'éliminer au plus tôt les termes de fréquence absolue élevée et de donner plus de poids à des mots discriminants. Si un mot est présent dans 25 documents sur un corpus de 1000 documents son *idf* sera de *idf* = log(1000/25) = 1.6 ; Si un autre mot est présent dans 750 documents sur un corpus de 1000 documents son *idf* sera de log(1000/750) = 0.125 et enfin si un mot est présent dans tous les documents son *idf* sera nul.

L'*idf* est donc une valeur associée au mot à l'intérieur du corpus.

tf :

Le *tf* est le poids relatif du mot dans le document. Il permet de redonner une valeur relative à la taille du document et de pouvoir, ainsi, comparer des documents de tailles différentes. Le *tf* d'un mot dans un document est calculé par la formule suivante :

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

Où $n_{i,j}$ est le nombre d'occurrences du terme dans t_i dans le document d_j. Le dénominateur est la somme de toutes les occurrences de tous les termes du document. Un terme présent 10 fois dans un document de 1500 mots aura donc un *tf* de 10/1500 soit 0.00666.

Le poids des mots à comparer est donné par :

$$tf.idf = log\frac{|D|}{|\{d_j : t_i \in d_j\}|} * \frac{n_{i,j}}{\sum_k n_{k,j}}$$

Calcul de la distance entre deux documents

Le calcul de la distance entre deux documents est ici basé sur la transformation des documents en vecteurs de n dimensions où n est le nombre de termes différents dans le document et où chaque dimension représente un terme unique. En calculant (sur l'ensemble des dimensions) le cosinus entre deux vecteurs, nous obtenons une valeur liée à la similarité entre les documents. Plus les documents sont proches et plus le cosinus est élevé.

Soient deux documents *A* et *B* tels que $A=\{a_1, a_2, \ldots a_x\}$ et $B=\{a_1, a_2, \ldots a_y\}$

n est égal au nombre de termes présents dans un document au moins.

Le calcul du cosinus ou de la similarité entre le document *A* et le document *B* sera alors :

$$S(A,B) = \cos(\theta)\frac{\sum_{i=1}^{n} a_i * b_i}{\sqrt{\sum_{i=1}^{n} (a_i)^2} * \sqrt{\sum_{i=1}^{n} (b_i)^2}}$$

Dans cet exemple nous allons mesurer la distance entre trois textes en utilisant la notion de *tf.idf.*

Texte A	**Texte B**	**Texte C**
La nuit, tous les chats sont gris, dans la nuit.	Tous les chats sont beaux dans la nuit.	Les portillons, la fenêtre sont tous en bois.

On calcule pour chaque mots sont *idf.* Les mots présents dans tous les textes présentent un *idf* nul.

mots	***idf***		**mots**	***idf***		**mots**	***idf***		**Mots**	***idf***		**mots**	***idf***		**mots**	***idf***		**mots**	***idf***		**mots**	***idf***
la	0		chats	Log(3/2)		fenêtre	Log(3)		Dans	Log(3/2)		Les	0		bois	Log(3)		gris	Log(3/2)
nuit	Log(3/2)		sont	0		en	Log(3)		Beaux	Log(3)		tous	0		portillons	Log(3)			

Sachant que le document A contient 10 mots, la valeur tf_A(nuit) =2/10. Le poids du mot « nuit » dans le premier document tf.idf_A(nuit) sera donc 2/10*log(3/2)=0.0352. La similarité entre les documents *A* et *B* est notée *S(A,B).*

$$S(A,B) = \frac{tf.idf_A(\text{nuit}) * tf.idf_B(\text{nuit}) + tf.idf_A(\text{chats}) * tf.idf_B(\text{chats})}{\sqrt{(tf.idf_A(\text{nuit}))^2+(tf.idf_A(\text{chats}))^2+(tf.idf_A(\text{gris}))^2}\sqrt{(tf.idf_B(\text{nuit}))^2+(tf.idf_B(\text{chats}))^2+(tf.idf_B(\text{beaux}))^2}}$$

$S(A,B)$ = 0,802

$$S(A,C) = \frac{0}{\sqrt{(tf.idf_A(nuit))^2+(tf.idf_A(chats))^2+(tf.idf_A(gris))^2}\sqrt{(tf.idf_C(portillons))^2+(tf.idf_C(fenêtre))^2+(tf.idf_C(en))^2+(tf.idf_C(bois))^2}}$$

$S(A,C)$ = 0

Comme on peut le voir dans cet exemple, bien que les documents A et C partagent plusieurs mots, leur coefficient de similarité est nul. La méthode peut bien sûr être améliorée par des algorithmes capables de retrouver des racines communes comme pour la langue anglaise, le célèbre algorithme de « Porter ». On peut ainsi faire des rapprochements entre des mots qui ne sont pas identiques mais qui possèdent des racines communes.

Mesure de similarité intra-requête

La mesure de similarité au sein des articles retournés par une requête est la moyenne des similarités entre chaque article. Les articles sont comparés deux à deux.

Mesure de similarité inter-requête

La mesure de similarité inter-requête des articles retournés par deux requêtes A et B est la moyenne des similarités entre chaque article retourné par la requête A avec tous ceux de la requête B. Les articles sont comparés deux à deux.

Le postulat de départ est que :

- les requêtes utilisateurs retournent des documents plus similaires que les requêtes de mots aléatoires ;
- les documents retournés par deux requêtes utilisateurs sont moins similaires que des documents retournés par deux requêtes aléatoires ;
- les documents retournés par une requête aléatoire déterminent la base de calcul qui représente le niveau 0 de similarité ;
- les documents retournés par une requête utilisateurs déterminent la valeur maximale de notre calcul qui représentent le niveau 1 de similarité.

QCSC ou le Quotient de Centralité Sémantique Comparé

La comparaison de la similarité des articles retournés par une même requête ou intra-requête entre les trois types de requêtes (aléatoires, agrégats, utilisateurs) ainsi que la comparaison de la similarité des articles retournés par deux requêtes différentes ou inter-requêtes permettra de cerner la validité sémantique des agrégats.

Calcul du CCSR_IntraQ ou Coefficient de Centralité Sémantique Comparé Intra-Requête

Nous recherchons enfin à calculer la valeur de *CCSR_IntraQ* ou le Coefficient de Centralité Sémantique Comparé Intra-Requête d'une Courbe X qui provient des mesures effectuées à partir des trios de mots issus d'agrégats (cf. figure 4.9).

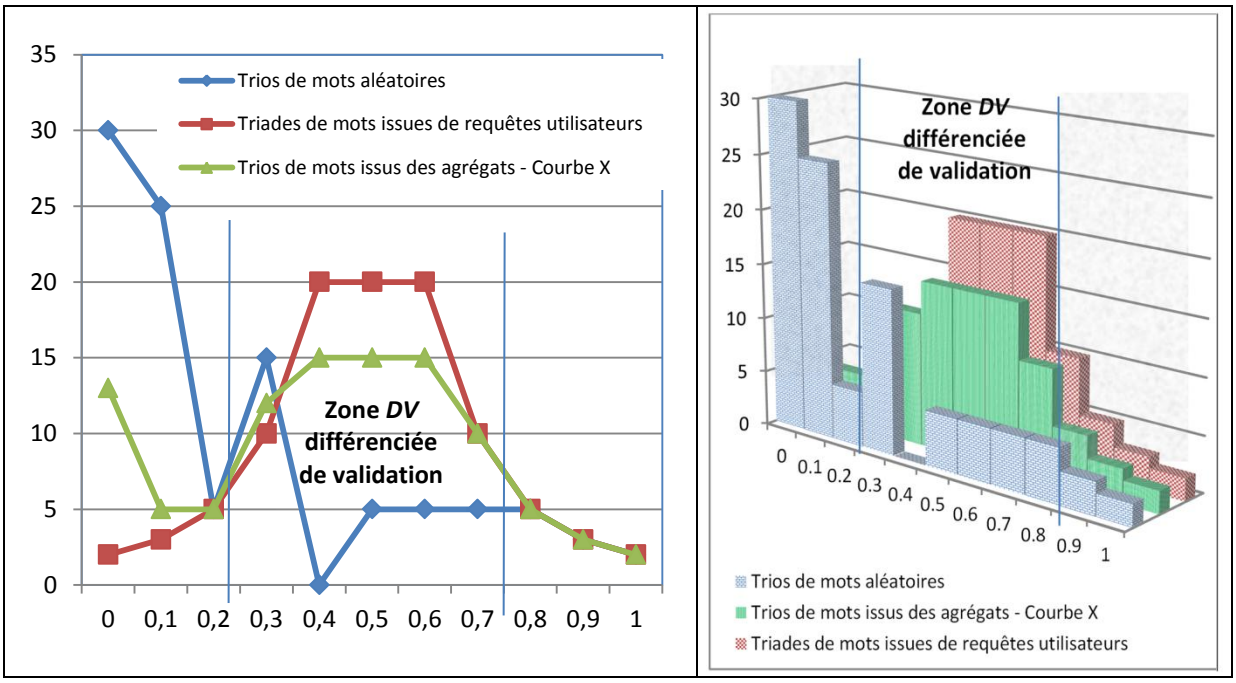


Figure 4.9 : *CCSR_IntraQ* : courbes et aires de distribution des valeurs de similarité retournées en fonction de la nature de la source du trio de mots constituant la requête et incluant en courbe X la courbe des trios de mots issus d'agrégats.

Après une détection de la zone *DV* comme étant la plus « différenciée », nous pouvons calculer *CSV-Inter-RC* sur cette zone de validation.

CCSR_IntraQ est défini comme la moyenne des aires de chaque valeur de chacune des classes incluses dans la zone étudiée :

- Chaque aire de la courbe étudiée est pondérée en fonction de la valeur des aires des deux courbes de référence sur la valeur de la classe en ordonnée de telle manière que plus la similarité de la courbe *X* se rapproche de celle des requêtes utilisateurs et s'éloigne de celle des trios aléatoires plus la valeur de *CCSR_IntraQ* augmente ;

- *CCSR_IntraQ* indique la similarité des documents retournés au sein d'une requête. Une valeur de 1 signifie une similarité comparable à celle des documents retournés par les triades issus de requêtes utilisateurs et 0 une similarité comparable aux trios de mots aléatoires.

La mise en valeur absolue des différences entre valeurs d'aires permet de ne pas avoir à tenir compte de l'ordonnancement des courbes dans le calcul relatif, le signe étant alors donné par une valeur *K* calculée pour chacune des aires selon la valeur de l'aire des requêtes issues d'agrégats par rapport aux aires définies par les courbes de références. De plus, pour chaque mesure nous maximisons la mesure à 1 et la minimisons à 0 en suivant le postulat précédemment défini qui détermine qu'il n'est pas possible d'obtenir des trios de mots plus sémantiquement cohérents que les triades utilisateurs et moins sémantiquement cohérents que les trios de mots aléatoires.

La formule mathématique de *CCSR_IntraQ* sera donc définie pour une courbe particulière *X* comme suit :

Si *(A(i) < R(i) et X(i) <A(i))* ou *(A(i) > R(i) et X(i) >A(i))* alors

K(i) = -1

Sinon

K(i) = 1

Fin de SI

$$CCSR\_IntraQ_X = \frac{\sum_{i=Début-Zone-DV}^{Fin-Zone-DV} Max(Min(|X(i)-A(i)|*K(i)/|R(i)-A(i)|,0),1)}{Nombre\_de\_classes\_ZoneDV}$$

Où *R(i)* définit l'aire de l'histogramme des triades dont tous les mots sont inclus au moins une fois tous ensemble dans une recherche pour l'ordonnée *i*.

Où *A(i)* définit l'aire de l'histogramme des trios de mots combinés aléatoirement dans une recherche pour l'ordonnée *i*.

Où *X(i)* définit l'aire de l'histogramme des trios dont les mots sont issus des agrégats dans une recherche pour l'ordonnée *i*.

CCSR_IntraQ représente le niveau de similarité entre documents pour une requête. La valeur 0 de *CCSR_IntraQ* représente alors le niveau de qualité le plus faible et 1 le plus élevé.

Calcul du CCSR_InterQ ou Coefficient de Centralité Sémantique Comparé Inter-Requêtes

Nous calculons ensuite sur la zone déclarée comme zone de validation le coefficient *CCSR_InterQ*. Ce coefficient est calculé exactement selon le même principe que *CCSR_IntraQ*, si ce n'est que les courbes de références peuvent être amenées à se croiser dans la zone *DV* ou simplement à être inversées. En effet selon l'emplacement de cette zone, les valeurs du pourcentage de comparaison de documents issus de requêtes différentes peuvent être orientées différemment (cf . figure4.10).

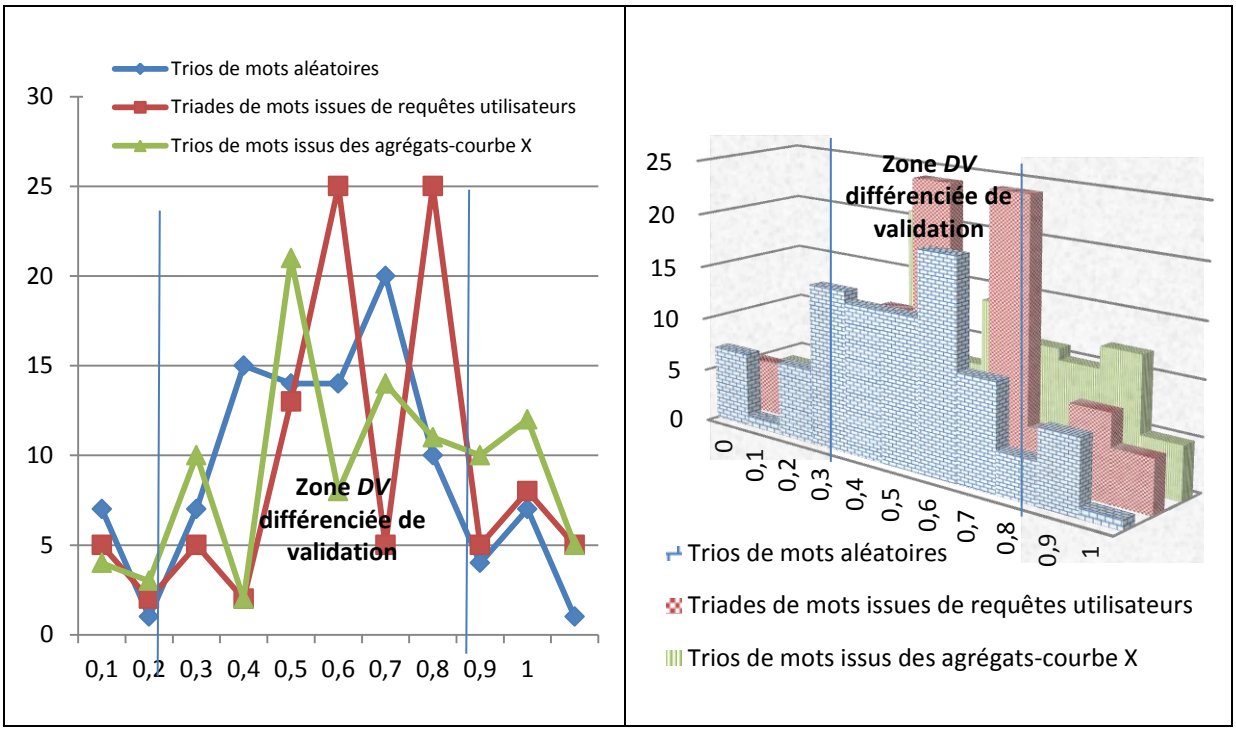


Figure 4.10 : *CCSR_InterQ :* Courbes et aires de distribution des valeurs de similarité retournées en fonction de la nature de la source du trio de mots constituant la requête.

Pour calculer la valeur de *CCSR_InterQ* ou le Coefficient de Centralité Sémantique Comparé Inter-Requêtes d'une Courbe X (qui provient des mesures effectuées à partir des trios de mots issus d'agrégats) (cf. figure 4.10), les modalités sont les suivantes :

- *CCSR_InterQ* est défini comme la moyenne des sommes des valeurs d'aires de chacune des classes incluses dans la zone étudiée.
- Chaque aire de la courbe étudiée est pondérée en fonction de la valeur des aires des deux courbes de référence sur la valeur de la classe en ordonnée de telle manière que plus la similarité de la Courbe X se rapproche de celle des requêtes utilisateurs et s'éloigne de celle des trios aléatoires, plus la valeur de *CCSR_InterQ* baisse.

Comme pour la valeur de *CCSR_IntraQ*, nous utilisons la mise en valeur absolue des différences entre valeurs d'aires de façon à ne pas avoir à gérer l'ordonnancement des courbes. Le signe est alors donné par une valeur *K* calculée pour chacune des aires selon la valeur de l'aire des requêtes issues d'agrégats par rapport aux aires définies par les courbes de référence. Nous maximisons et minimisons les valeurs de la même manière que pour *CCSR_IntraQ*.

La formule mathématique de *CCSR_InterQ* est définie pour une courbe particulière Courbe X comme suit :

Si $(A(i) < R(i)$ et $X(i) < A(i))$ ou $(A(i) > R(i)$ et $X(i) > A(i))$ alors

$K(i) = -1$

Sinon

$K(i) = 1$

Fin de Si

$$CCSR\_InterQ_X = \frac{\sum_{i=Début-Zone-DV}^{Fin-Zone-DV} Max(Min(|X(i) - A(i)| * K(i) / |R(i) - A(i)|, 0), 1)}{Nombre\_de\_classes\_ZoneDV}$$

Où *R(i)* définit l'aire de l'histogramme des triades dont tous les mots sont inclus au moins une fois tous ensemble dans une recherche pour l'ordonnée *i*.

Où *A(i)* définit l'aire de l'histogramme des trios de mots combinés aléatoirement dans une recherche pour l'ordonnée *i*.

Où *X(i)* définit l'aire de l'histogramme des trios dont les mots sont issus des agrégats dans une recherche pour l'ordonnée *i*.

CCSR_InterQ représente la similarité des documents inter-requête. La valeur 0 de *CCSR_InterQ* représente alors le niveau de qualité le plus élevé et 1 le plus faible.

QCSC ou le Quotient de Centralité Sémantique Comparé

Le Quotient de Centralité tient compte des mesures de distance faites en inter-requêtes autant qu'en intra-requêtes. Il est directement proportionnel au premier *CCSR_IntraQ* et inversement proportionnel au second *CCSR_InterQ*. Afin de tenir compte des deux coefficients et ensuite de pouvoir comparer le résultat à celui d'autres méthodes, nous définissons le *QCSC* comme la racine carrée du produit des deux coefficients. Le quotient de centralité sémantique s'exprime comme suit :

$$QCSC_X = \sqrt{CCSR\_IntraQ \times (1 - CCSR\_InterQ)}$$

4.1.1 Conclusion sur les méthodes de validation

Nous en arrivons à la conclusion qu'il existe deux démarches pour valider sémantiquement un agrégat de mots :

- La première correspond à la comparaison des distributions ou des valeurs de groupes de mots connus. Si on trouve une différence sensible entre les deux catégories de groupes de mots, il sera alors possible de situer les nôtres (ici, les agrégats) par rapport à ces groupes référents.
- La seconde consiste à confronter les agrégats de mots à une évaluation manuelle ou à un système d'évaluation manuellement étalonné par des utilisateurs. La difficulté à repérer des populations de mots véritablement représentatives et pouvant servir de populations repères, nous pousse à

rechercher une validation par des utilisateurs. L'expérimentation manuelle effectuée par un expert du domaine (si les agrégats sont créés dans un domaine précis) ou des utilisateurs est toujours tentante. En effet, c'est en fait une externalisation du travail et puisque le responsable du jugement est, soit un expert, soit un groupe d'utilisateurs ayant des exigences, nous attendons un jugement certain. Mais qu'en est-il ? Dans le paragraphe suivant, nous étudions en détail les avantages et les inconvénients de chaque méthode de validation.

Les différentes techniques de regroupement ont été utilisées en fonction de plusieurs critères et opportunités :

- la présence et la disponibilité d'experts : les validations manuelles sont liées à la disponibilité d'un expert du domaine des agrégats ;
- les résultats obtenus par les différentes méthodes : les méthodes de regroupements et de validation sont confrontées en fonction de leurs résultats respectifs. Il est inutile de tester avec plusieurs méthodes de validation sémantique une solution de regroupement qui ne fonctionne pas. De la même manière, il est peu productif de tester des agrégats créés avec plusieurs méthodes si la méthode de validation sémantique n'a pas démontré son efficacité.

Quels réseaux, testés avec quelles méthodes de validation ?

Dans le tableau 4.6 le lecteur trouve l'ensemble des méthodes d'agrégation ou d'enrichissement, les méthodes de validation et les réseaux.

Le tableau suivant permet de faire le lien entre ces diverses expérimentations.

Méthodes de Validation / Méthodes de regroupement	MCCVS	TREC-Eval	Manuelle	MCSDR
Clique	Non	Non	AOL 17/04/2006	Non
Rigidification Simple	AOL 17/04/2006 et AOL 17/03/2006	Non	Non	Non
Rigidification Régulée	100 mots dans AOL	TREC-Eval	eDonkey-10-semaine	100 mots dans AOL \Wikipédia
Enrichissement	Non	Non	eDonkey-5-mois	Non

Tableau 4.6 : les méthodes de regroupement, les méthodes de validation sémantique et les réseaux.

Ainsi, par exemple, seule la méthode de Rigidification Régulée qui apparaît comme la plus performante est validée par l'ensemble des méthodes de validation. La méthode de regroupement par cliques, qui a immédiatement montré ses limites, n'est pas suffisamment digne d'intérêt pour justifier d'autres tests.

4.4 Résultats des regroupements et validation sémantique

4.4.1 Agrégation par regroupement en cliques sur réseau AOL-17/04/2006 et validation manuelle

Matériel et conditions de test

Pour cette validation nous avons travaillé sur l'échantillon **AOL-17/04/2006**

Résultats

L'algorithme a créé **108446 cliques** de **3 à 9 mots-clés**, avec en **moyenne 3.75 mots-clés** par clique. **18600** mots-clés ne sont dans aucune clique.

En choisissant un système de regroupement favorisant fortement la cohérence du groupe, nous avons créé des groupes possédant une faible distance entre eux. Cette faible distance des agrégats a pour conséquence un grand nombre d'agrégats par rapport au nombre de mots-clés (3,2 cliques par mot-clé agrégé) et un nombre de mots-clés présents dans de très nombreuses cliques (plus de 50 mots-clés appartiennent à plus de 1000 cliques). Un système de regroupement créant plus de groupes qu'il n'existe d'objets individuels n'était pas ce que nous recherchions.

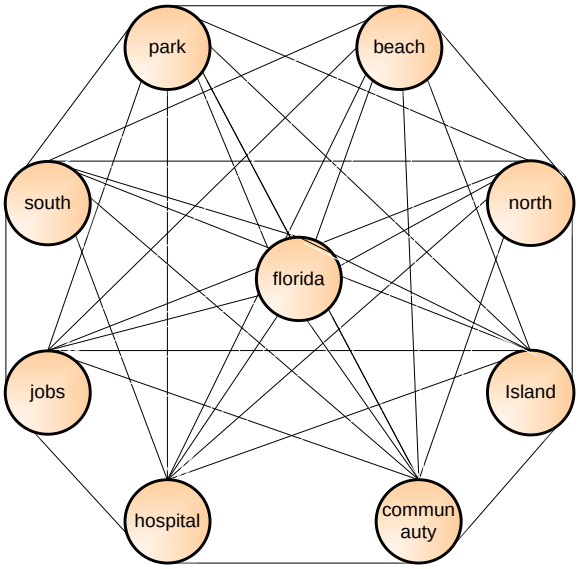


Figure 4.11. Clique à neuf éléments.

Conclusion

Les regroupements en cliques sont une étape d'étude. Les cliques relevées n'ont pas donné satisfaction en de nombreux points.

Tout d'abord, la taille « maximale » des cliques est très faible.

Ensuite, plus de 70 % des cliques ne sont en fait que le résultat d'une requête. En effet, il suffit d'une requête de neuf mots pour créer une clique d'autant de mots. Chacun des neuf mots de la requête a bien été utilisé avec les huit autres. Il n'y a pas, avec cette méthode, de pondération et donc de seuil de validation des liaisons.

Par ailleurs, une vérification manuelle rapide nous montre que les agrégats ne sont pas sémantiquement cohérents. La non prise en compte de la pondération des liens permet de créer des ensembles non significatifs. (cf. figure 4.11). Les éléments utilisés une fois conjointement créent forcément une clique. Les mots les plus utilisés servent de hubs à des cliques dans lesquelles les autres mots se sont trouvés simplement une fois « *au contact* » de tous les autres. Ceci n'est pas représentatif des « véritables » usages. Une utilisation exceptionnelle ou erronée, d'un terme provoque des liens tout aussi valides que des utilisations nombreuses.

Enfin, sur ce réseau, nous avons dû supprimer préalablement les mots vides pour éviter des agrégats encore moins cohérents. Ce type de regroupement n'est pas efficace sur des réseaux de cette nature (cf. figure 4.11). En revanche, sur d'autres réseaux, notamment sur des réseaux possédant la caractéristique d'imposer un degré limité à chaque nœud, ils peuvent être très efficaces. Le travail de Palla & all **[Palla&al-2005]** (cf. paragraphe 2.3.1) utilisant la notion d'agrégation de cliques a, sur des réseaux biologiques, donné d'excellents résultats.

4.4.2 Agrégation par la méthode de Rigidification Simple sur réseaux AOL-17/04/2006 et AOL-17/03/2006 - Validation par MCCVS

Matériel et conditions de test

Pour cette validation nous travaillons sur les réseaux : **AOL-17/04/2006** et **AOL-17/03/2006.**

Définition des paramètres de l'algorithme

Après plusieurs essais sur des échantillons, nous avons défini les valeurs des seuils : Valeur Minimale de CFL ou *Val-Min-CFL* à 5 % du poids du mot-clé et la Valeur d'Activation ou *Val-Activ-CFL* à 20 % du poids du mot-clé (cf. paragraphe 3.3).

Ces essais, effectués par approximations successives sur des échantillons du graphe, ont permis de définir des valeurs qui, tout à la fois, autorisent la création d'agrégats et limitent la taille maximale des agrégats à des valeurs qui, intuitivement, semblent correctes. Nous avons considéré que la taille maximale devait être inférieure à un millier de mots.

Ces valeurs pourront être modifiées lors de prochaines expérimentations ; ici, elles servent d'exemples et ne constituent pas le sujet de l'étude. Elles doivent cependant nous permettre de valider la méthode en créant suffisamment d'agrégats pour étudier ceux-ci, c'est ce qui a été validé par les essais préliminaires.

Nombre et nature des agrégats créés

La démarche implantée a permis de former 9 556 agrégats construits avec 38 621 mots-clés dont 24 537 mots-clés différents dans l'ensemble des agrégats (cf. figure 4.12). Le nombre moyen de mots-clés par agrégat est de 4,04. L'agrégat le plus important contient 133 mots-clés.

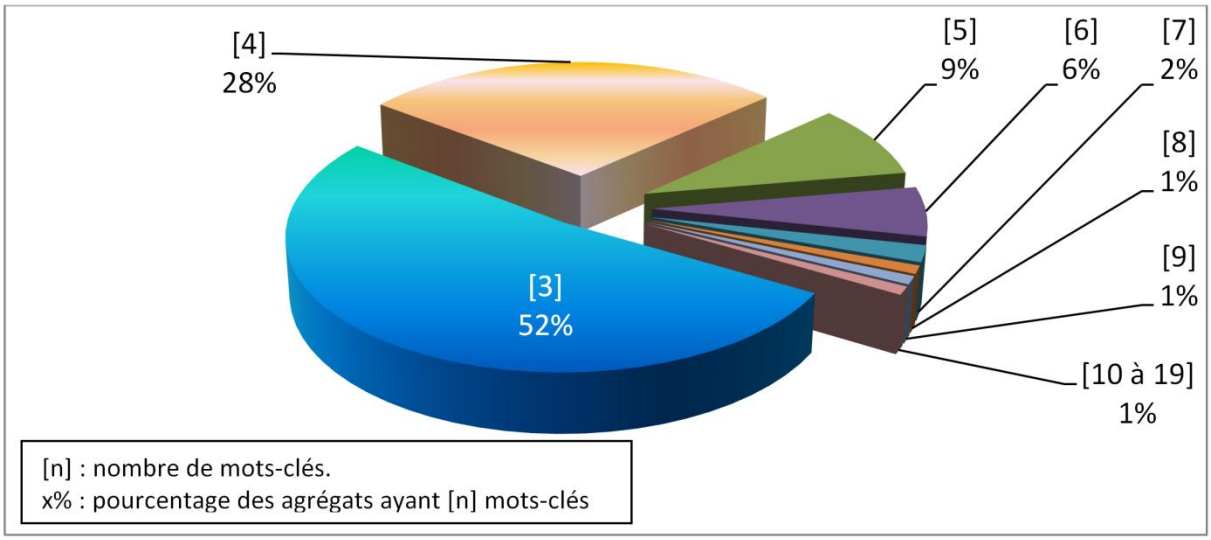


Figure 4.12 : Répartition des agrégats en fonction du nombre de mots-clés

Estimation de la qualité sémantique des agrégats

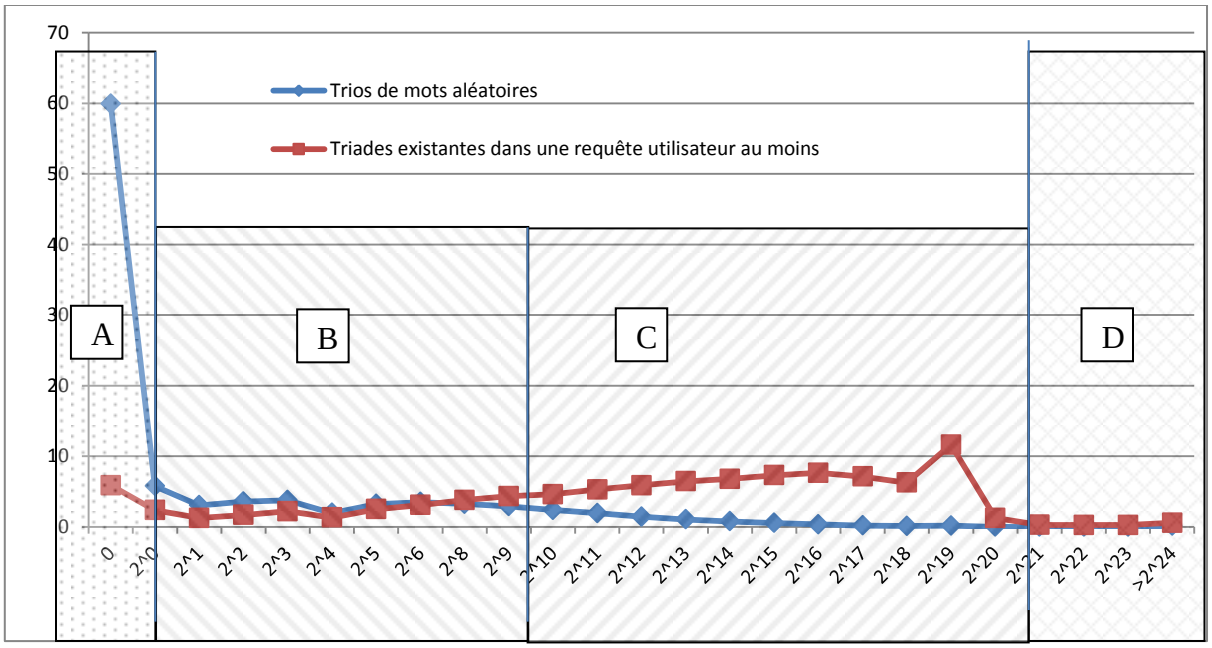


Figure 4.13 : Comparaison des réponses aux requêtes susceptibles d'être les plus éloignées sémantiquement (cf. 4.3.1) et détermination de la zone à plus forte divergence.

Nous comparons ici les deux courbes de réponses des deux espaces les plus éloignés sémantiquement selon le postulat posé en section 4.3.1. Nous comparons la courbe issue des mots combinés aléatoirement (excluant des triades de mots utilisées dans une recherche) avec la courbe de référence issue du test de triades de mots pour lesquelles il existe au moins une

recherche incluant ces trois mots-clés. Sur la figure 4.13, nous distinguons quatre zones clairement identifiables, la zone A de 0, la zone B de 2^1 à 2^9, la zone C de 2^10 à 2^20 (cf. figure 4.14) et la zone D supérieure à 2^20. Les zones « B » et « D » ne présentent pas beaucoup d'intérêt, les courbes n'ayant pas de différence notable. La zone « A » est limitée à une seule valeur et ne peut donc représenter une étendue suffisante pour mener notre étude. La zone « C » est la zone la plus singulière avec une plage suffisante pour avoir un sens. Afin de mieux percevoir l'importance de la zone « C », reprenons une lecture du graphique en omettant les zones A, B et D.

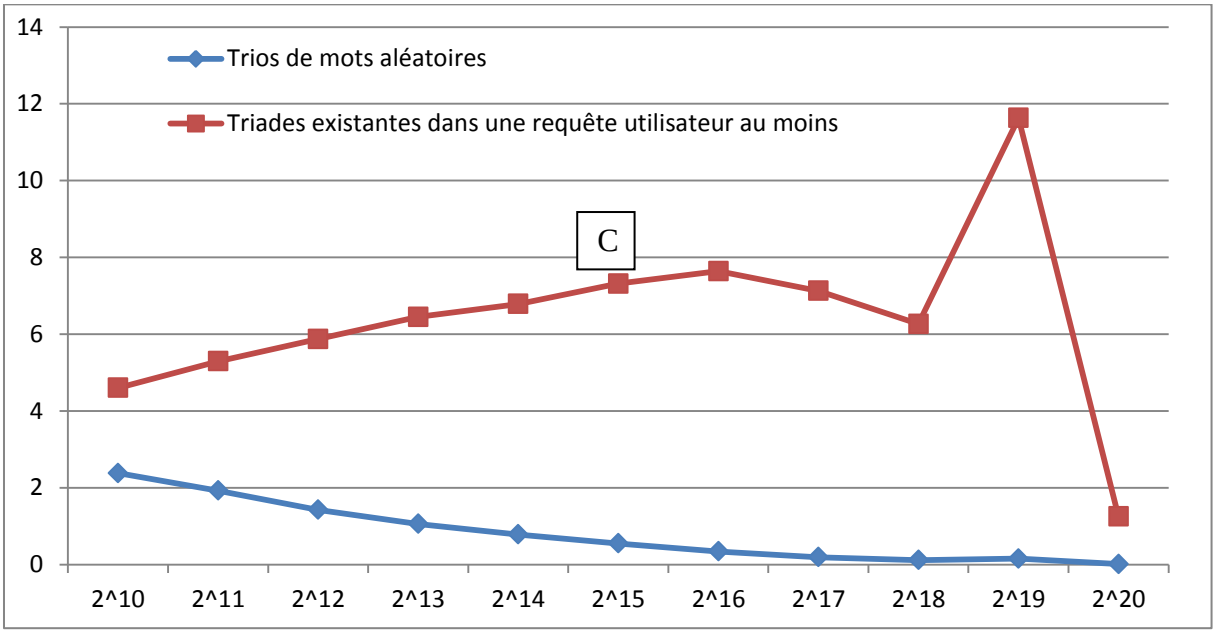


Figure 4.14 : Zoom sur la zone « C » sélectionnée comme zone d'étude.

La zone « C » nous sert de zone de validation sémantique. Afin de pouvoir élaborer une comparaison rapide et arithmétique, nous définissons un coefficient approprié.

Calcul du Coefficient de Validation Sémantique Comparée (*CVSC*)

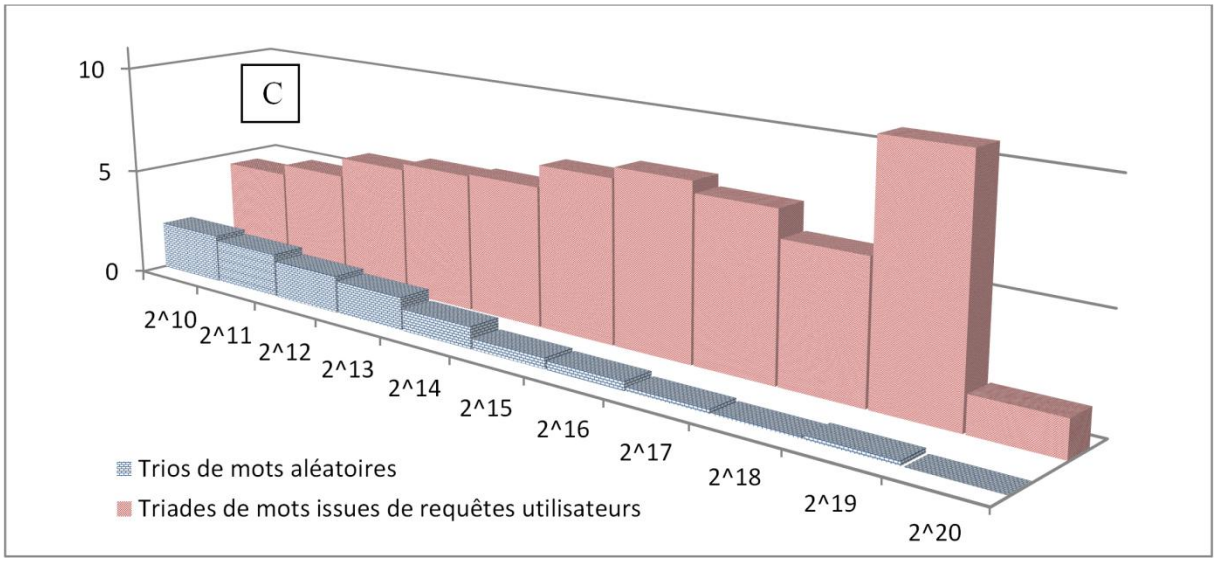


Figure 4.15 : Représentation de la Zone C en aires couvertes par les deux courbes de référence.

Nous considérons que les classes en puissance de deux forment une échelle d'indice « un » et comparons l'aire prise par les deux histogrammes. Le *CVSC*, ou Coefficient de Validation Sémantique Comparé, a alors la valeur « 1 » pour l'équivalence de l'histogramme des triades (de trois mots-clés) ayant été au moins une fois utilisées dans une même recherche et 0 pour la valeur de l'histogramme des trios aléatoires.

Où A_R définit l'aire de l'histogramme des triades dont tous les mots sont inclus au moins une fois tous ensemble dans une recherche selon la formule $CVSC_X = (A_X - A_A) / (A_R - A_A)$ (cf. paragraphe 4.31) :

$$A_R = \sum_{i=10}^{20} Yi = 70{,}24$$

Où A_A définit la valeur de l'aire de l'histogramme des triades aléatoires :

$$A_A = \sum_{i=10}^{20} Y'i = 8{,}95$$

Où Ax définit la valeur de l'aire de l'histogramme des triades à comparer :

$$A_X = \sum_{i=10}^{20} Y''i$$

Comparaison des coefficients CVSC pour des agrégats de tailles différentes

Dans un premier temps nous étudions le comportement des agrégats en fonction de leur taille. Pour plus de lisibilité nous les regroupons en cinq familles correspondant aux cinq décades : les agrégats de moins de 10 mots, ceux de moins de 20 mots et plus de 9, ceux de moins de 30 mots et plus de 19, ceux de moins de 40 mots et plus de 29 et enfin ceux de plus de 39 mots.

Le but de ce test est de détecter s'il existe une corrélation entre la taille des agrégats et la valeur du *CVSC*.

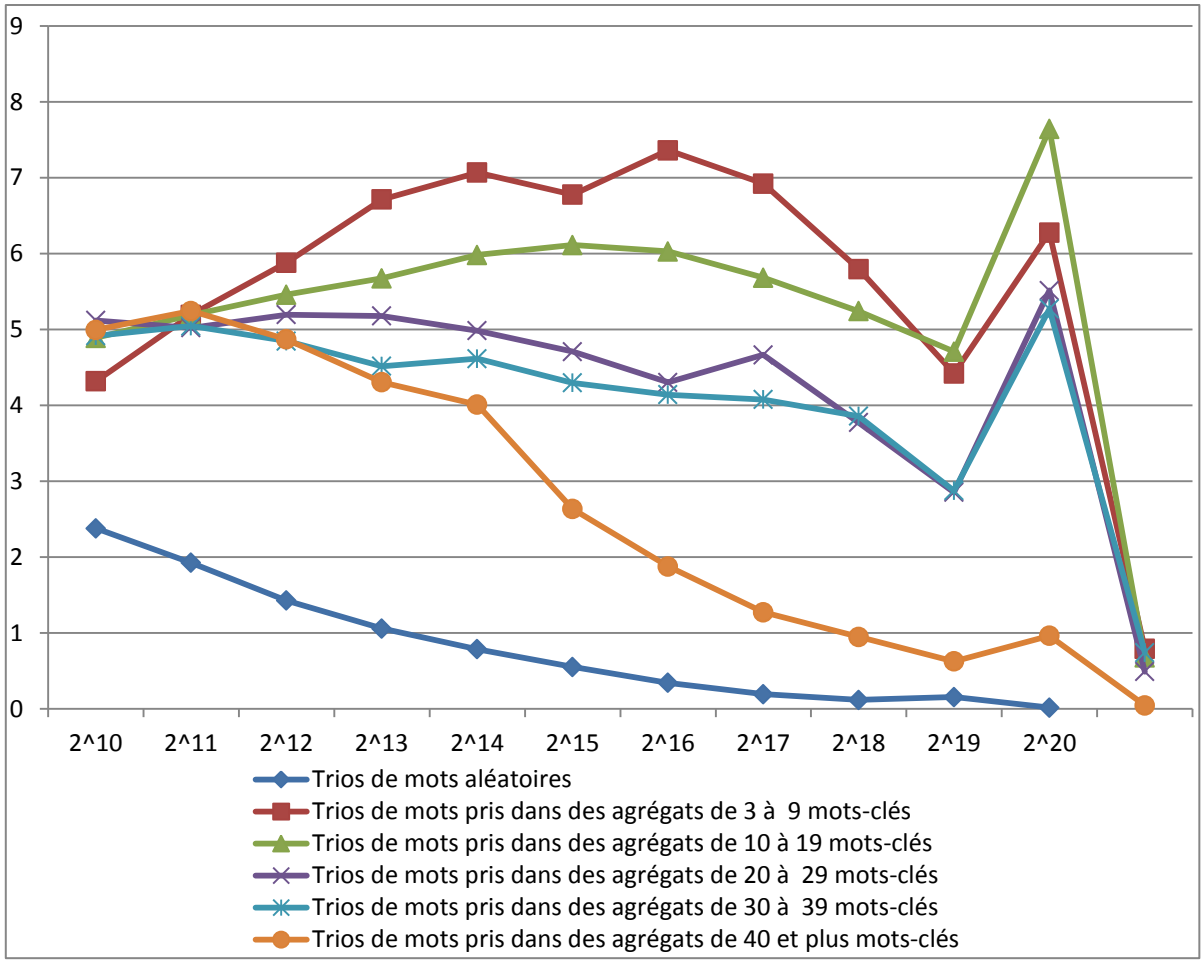


Figure 4.16 : Représentation graphique des *CVSC* en fonction de la taille des agrégats en zone « C » de validation sémantique.

Taille des agrégats en nombre de mots-clés	***CVSC***
De 3 à 9	0.89
De 10 à 19	0.80
De 20 à 29	0.61
De 30 à 39	0.57
Plus de 39	0.29

Tableau 4.7. Valeur des *CVSC* en fonction de la taille des agrégats en zone « C » de validation sémantique.

L'analyse des courbes présentées et des valeurs de *CVSC* montre une forte corrélation entre la taille des agrégats et les valeurs du coefficient. Si la taille des agrégats est inversement proportionnelle aux *CVSC* mesurés, on note un écroulement à partir de 40 mots et au-delà.

Borner la taille des agrégats est donc un moyen pour limiter le nombre des agrégats ayant une faible cohérence sémantique.

Comparaison des coefficients CVSC en excluant les recherches utilisateurs

Afin d'estimer la perte de cohérence sémantique liée à la notion d'agrégat, il est pertinent de comparer les coefficients sémantiques obtenus pour les mêmes classes d'agrégats

en excluant les triades utilisées dans une recherche au moins. Ainsi, les coefficients obtenus ne doivent leur valeur qu'à des combinaisons créées par la méthode de Rigidification Simple.

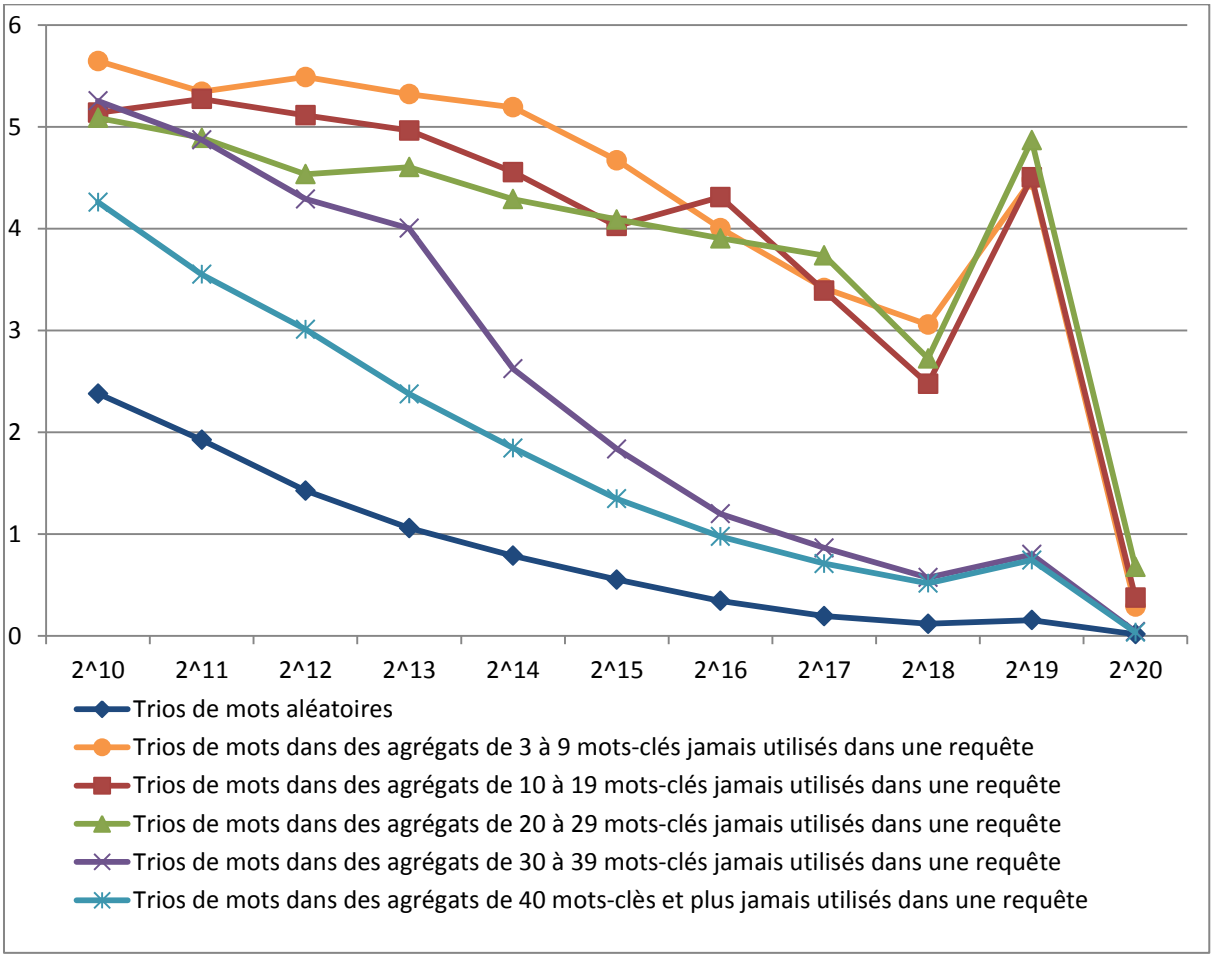


Figure 4.17 : Représentation graphique des CVSC en fonction de la taille des agrégats en zone « C » de validation sémantique en excluant les triades incluses dans une requête d'utilisateur.

L'observation des chiffres du *CVSC* (cf. tableau 4.8) des trios issus d'agrégats et n'ayant jamais été utilisés dans une recherche par un utilisateur nous conforte sur le seuil à ne pas dépasser. En effet, les agrégats de moins de 30 mots gardent un ratio supérieur à la moyenne.

Il est difficile de déterminer sans une étude détaillée au cas par cas les raisons de la baisse du coefficient. Cependant, la possibilité qu'un mot soit utilisé dans des acceptions différentes peut en être une des causes.

Taille des agrégats en nombre de mots-clés	***CVSC***	**Perte**
De 3 à 9 mots	0.62	0.27
De 10 à 19 mots	0.57	0.23
De 20 à 29 mots	0.56	0.05
De 30 à 39 mots	0.28	0.29
De 40 à 49 mots	0.17	0.12

Tableau 4.8. Valeur des *CVSC* en fonction de la taille des agrégats.

Ainsi, que ce soit de manière graphique (cf. figure 4.17) ou par le calcul du *CVSC* (cf. Tableau 4.8), on peut conclure que plus le nombre de mots-clés est important plus le *CVSC* a tendance à baisser. Cette étude révèle finalement que les agrégats d'une taille supérieure à 30 mots possèdent un *CVSC* inférieur ou égal à 0.5.

Placer une limite absolue sur une qualité aussi subjective que la cohérence sémantique d'un groupe de mots n'a bien sûr aucun sens si cela n'est pas fait de manière statistique et seulement dans le but d'étudier le comportement des agrégats.

En fixant un seuil de qualité au niveau de la valeur médiane (0.5 comme on le fait pour valider un examen), on considère que statistiquement les agrégats de plus de 30 mots-clés ne présentent pas un *CVSC* acceptable.

Mais plus que la valeur du *CVSC* elle-même, c'est la baisse brutale de cette valeur qui est intéressante. Tandis qu'entre des agrégats de 3 à 9 et 20 à 29 le coefficient baisse seulement de 9.6%, entre les agrégats de 20 à 29 et ceux de 30 à 39 le coefficient s'écroule de 50 %. La chute s'accentuant encore de 39% supplémentaire entre les agrégats de 30 à 39 et ceux de 40 à 49.

Ce test révèle donc la baisse brutale de *CVSC* pour les agrégats de taille supérieure à 30 mots.

Comparaison entre les réseaux AOL-17/04/2006 et AOL-17/03/2006

Afin de savoir si ces résultats sont liés au contexte comme, par exemple, le jour choisi dans le fichier de log, nous avons rejoué notre test sur un autre jour du fichier du log d'AOL, le 17/03/2006.

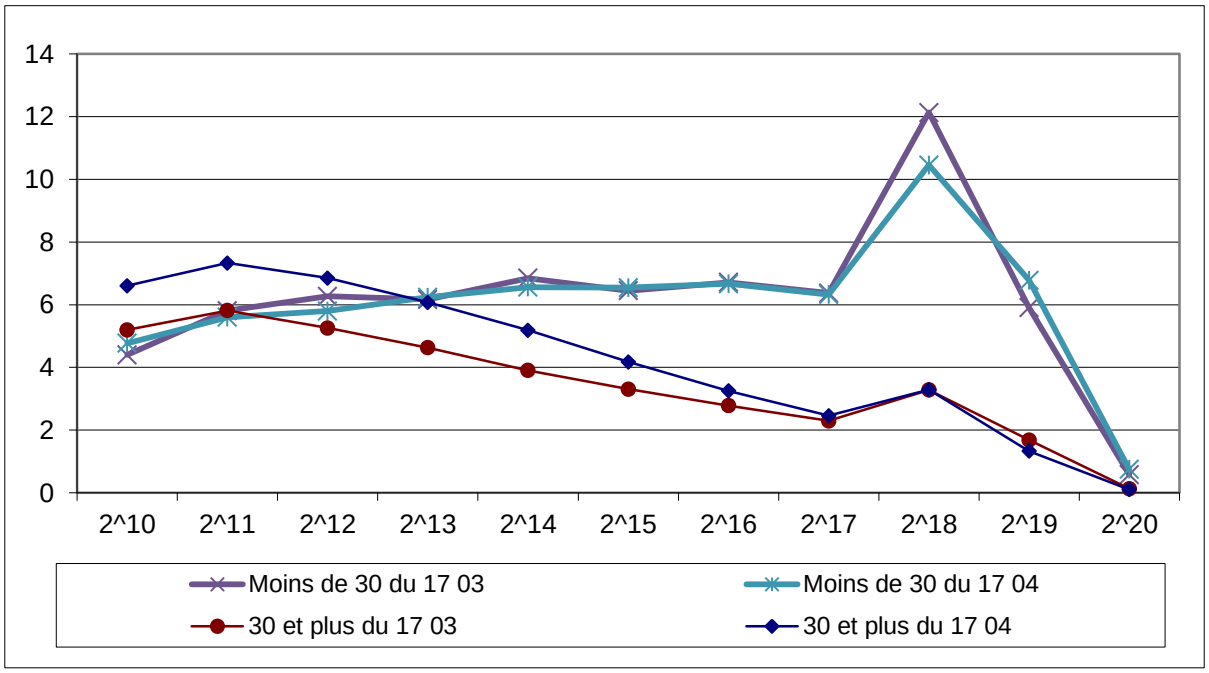


Figure 4.18 : Comparaison des courbes de *CVSC* pour les agrégats de moins et de plus de 30 mots-clés sur les deux réseaux différents.

Date	Taille de l'agrégat	*CVSC*
17/04/06	Inférieur à 30 mots	.83
17/03/06	Inférieur à 30 mots	.86
17/04/06	Supérieur à 30 mots	.45
17/03/06	Supérieur à 30 mots	.37

Tableau 4.9 : Comparatif des valeurs de *CVSC* pour les agrégats de moins et de plus de 30 mots-clés sur deux réseaux différents.

Nous avons donc traité l'échantillon du 17 mars 2006 avec une méthode HLS-CVSC identique à celle du 17 avril 2006. On observe une grande cohérence entre les courbes issues des logs du 17 avril 2006 et celles issues des logs du 17 mars 2006. Cela confirme les conclusions sur la relation entre la taille des agrégats et les valeurs de *CVSC* et le fait que cette information semble indépendante du contexte temporel.

Comparaison des CSVC entre les triades et les trios de mots au sein des agrégats

Un terme n'est pas toujours monosémique. Ainsi, les agrégats incluant des mots polysémiques, sont susceptibles de contenir des combinaisons de mots (trios) de faible coefficient sémantique, en raison de ces multiples sens.

Dans cette section, au travers de deux exemples nous illustrons la baisse du coefficient identifiée précédemment.

Le premier exemple est purement théorique (cf. figure 4.19), le deuxième est un véritable agrégat créé avec la méthode de Rigidification Simple sur le réseau AOL-17/04/2006**.**

Exemple 1

Le graphe de la figure 4.19 ci-dessous, illustre les concepts de musique et de cuisine, notamment au travers des mots « chef », « piano » et « sol ».

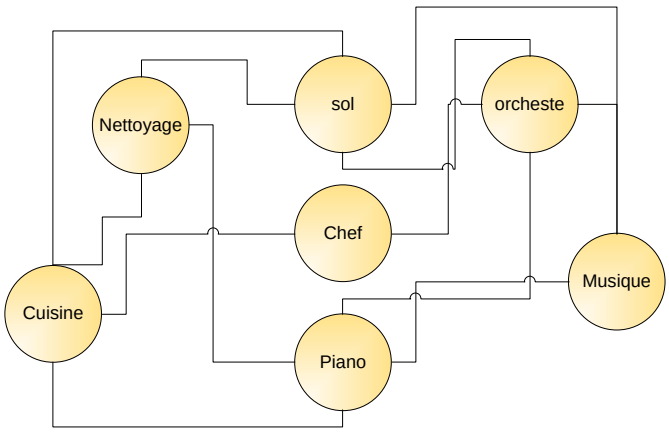


Figure 4.19 : Exemple d'agrégat intégrant des mots ayant plusieurs acceptions (musique/cuisine).

Supposons que nous ayons obtenu ce graphe à partir des requêtes utilisateurs suivantes :

- {sol, cuisine, nettoyage}
- {chef, cuisine, piano, nettoyage}
- {musique, chef, piano, orchestre}
- {sol, piano, musique}

Supposons que la méthode permette de construire un agrégat *AG* contenant tous ces mots tel que :

AG = {sol, cuisine, nettoyage, chef, piano, orchestre, musique}.

Différentes acceptations des mots « piano », « sol » et dans une moindre mesure « chef » interviennent dans cet agrégat. Lors de l'évaluation de la cohérence sémantique de cet agrégat, la combinaison systématique en trios de tous les mots-clés dans l'agrégat génère un certain nombre de trios ayant une faible cohérence sémantique. En voici trois exemples :

1) +nettoyage +musique +orchestre
2) +cuisine +chef + musique
3) +piano +nettoyage +sol

Exemple 2

Prenons un autre agrégat de mots issu du réseau **AOL-17/04/2006**, nommons cet agrégat *Agr*. Il est défini tel que :

Agr = {abiline, arunde, arundl, aubun, avalanche, b2600, car, cars, chevrolet, dealerships, electronic, fj40, fordsale, gaffn, hamptonroad, ignition, lexus, lynchb, maine, microwave, murrieta, outboard, parts, pax, selecti, ulster, uplander, used, usedfront, virgini, waterville}.

Cet agrégat a été construit notamment grâce aux requêtes utilisateurs suivantes :

la requête utilisateur « **+used +car +pax »** qui renvoie 284 000 sites : pax est une référence de pneu de marque Michelin et d'autres pièces détachées ;

la requête utilisateur « **+used +car +abiline »** qui renvoie 1 140 sites : abiline est un centre de vente et d'achat de pièces détachées ;

la requête utilisateur « **+used +car +murietta »** qui renvoie 17 100 sites : murrieta est un centre de réparation de véhicules.

Ces trois requêtes utilisateurs ont toutes des résultats situés dans la zone [C]. Cependant, le trio de mots issu de cet agrégat, utilisé comme requête dans la mesure de la cohérence sémantique, « ***+abiline +murietta +pax »*** ne retourne qu'un seul site (search.AOL.com 2010) où « Abiline » devient un prénom, « Murietta » le nom d'une ville et « pax » le mot latin signifiant « paix ».

Pour mesurer les pertes de cohérence sémantique liées à cet aspect du problème et pour mieux connaître la valeur statistique de *CSVC* sur les trios de mots issus d'agrégats par rapport aux triades issues de requête, nous avons testé séparément les trios de mots et les triades. Afin de rester sur des espaces sémantiquement valides, notre test ne comprend que les agrégats de moins de 30 mots.

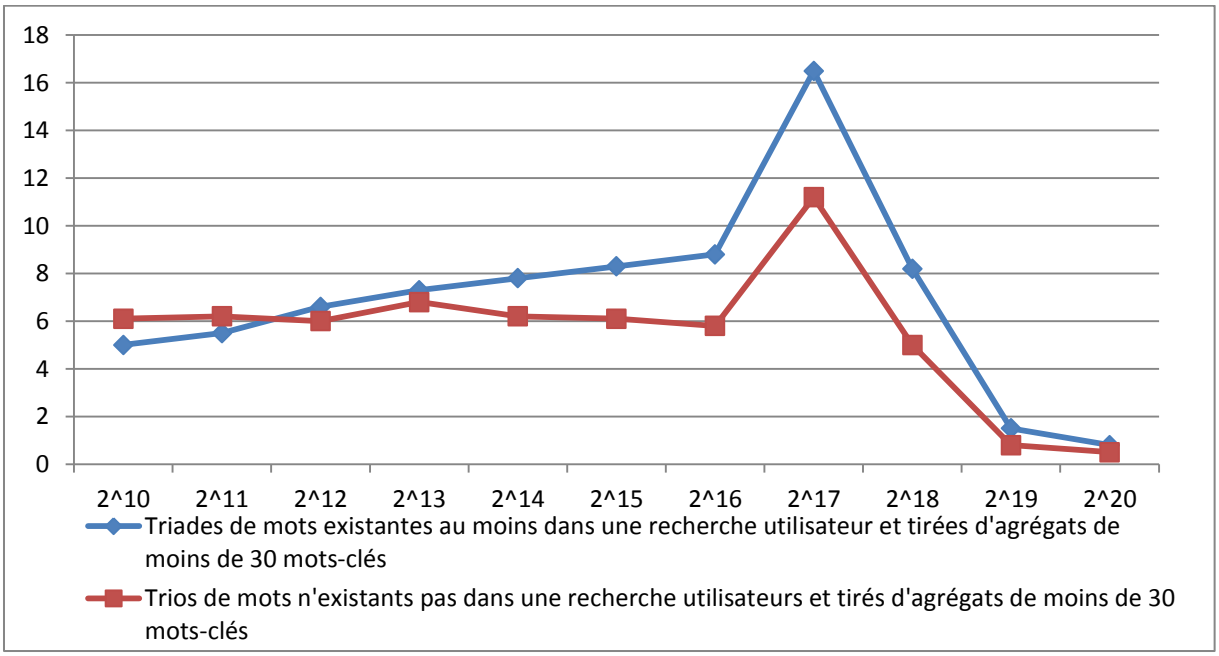


Figure 4.20 : Comparaison des valeurs des *CVSC* des triades et trios issus d'agrégats de 3 à 29 mots-clés.

Dans des agrégats de 3 à 29 mots-clés	***CVSC***
Triades utilisées au moins une fois dans une recherche	1
Trios jamais utilisés conjointement dans une recherche	.70

Tableau 4.10 : Comparaison des valeurs des *CVSC* des triades et trios issus d'agrégats de 3 à 29 mots-clés.

L'agrégat présenté dans l'exemple 2 n'a pas été soumis à ce test puisqu'il possède plus de 29 mots.

Les triades incluses dans les agrégats de moins de 30 mots obtiennent très logiquement, le score de 1. Les trios de mots (combinés depuis les agrégats de moins de 30 mots) qui n'ont jamais été utilisés dans une requête utilisateur présentent l'excellent score de 0.7.

Si l'agrégation crée bien une baisse du *CVSC*, celle-ci reste contenue au sein des agrégats de taille inférieure à 30 mots-clés.

Agrégats par la méthode de Rigidification Régulée

Matériel et conditions d'évaluation

La suppression ou la non-intégration dans les agrégats de mots au sens faible (mots de liaison, déterminants, etc.) pour en maintenir la taille est généralement préférable à la suppression des mots possédant un sens fort. Pour déterminer les valeurs de *Val-Min-CFL* et de *Val-Activ-CFL* (cf. paragraphe 3.4), nous étudions et comparons les valeurs des liaisons et plus particulièrement de *CFL* (Coefficient de Fiabilité de Lien) de deux types de mots-clés :

1) Des mots à faible sens : nous utilisons la liste de 162 mots fournis sur le site http://snowball.tartarus.org/algorithms/english/stop.txt comme liste de mots non signifiants.

2) Des mots monosémiques : nous utilisons une liste de 51 mots choisis pour leur caractère monosémique. Cette seconde liste comporte des mots aux usages très spécifiques. Fournis par des chimistes, des biologistes, des chercheurs, des médecins elle comprend des éléments du tableau de Mendeleïev, des éléments du corps humains, des codes utilisés par des pédophiles et d'autres termes très techniques (cf. tableau 4.11).

10yo	anagram	etymology	idiom	niobium	rhodium	ulnar
11yo	arginine	euphemism	indium	palindrome	scopolamine	yttrium
12yo	babyshivid	femur	innuendo	palladium	selenium	zirconium
aabbccddee	Cadmium	fibula	kingpass	pthc	sternum	
acrostic	carnitine	glutamine	lysine	ptsc	talus	
adenine	clavicle	humerus	mandible	qqaazz	technetium	
aldosterone	coccyx	hussyfan	mnemonic	r@ygold	tibia	
ambigram	collagen	hyoscyamine	molybdenum	rhetoric	tyrosine	

Tableau 4.11 : Liste de mots déterminés comme monosémiques.

Valeur de départ de Val-Min-CFL

Pour fixer la valeur de *Val-Min-CFL* nous allons comparer la nature des liens des deux populations étudiées. Plus précisément nous allons comparer la valeur la plus faible de *CFL* sur les diades où un des mots de la liaison au moins est monosémique avec la même valeur quand un des mots au moins est dans la liste des mots vides.

Le but est de prendre une valeur suffisamment basse pour conserver les mots monosémiques dans les agrégats et une valeur suffisamment haute pour exclure au plus tôt les mots vides.

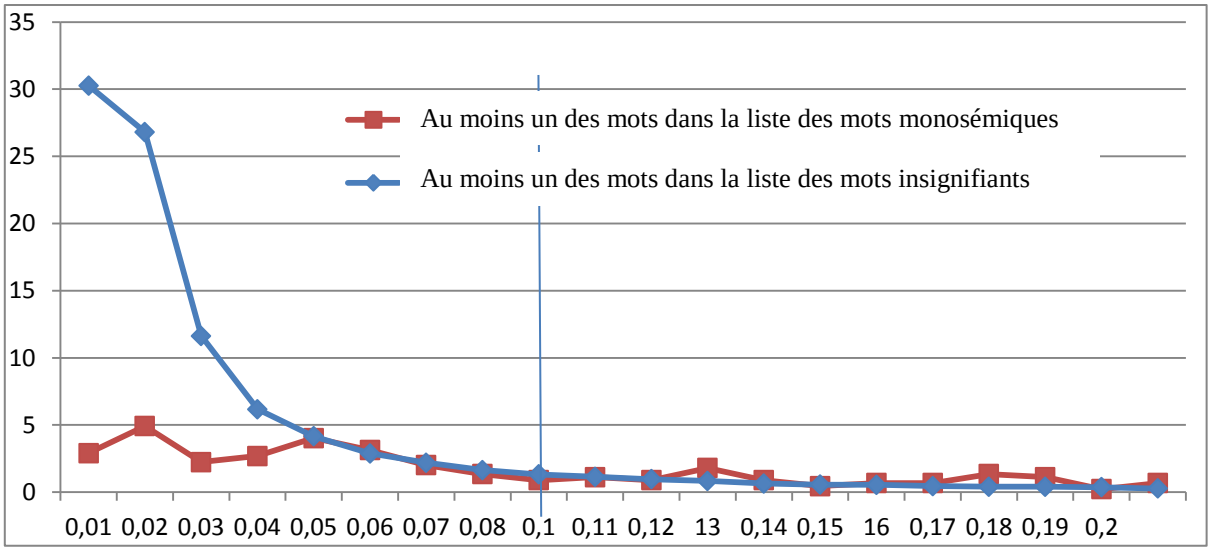


Figure 4.21 : Comparaison de la distribution de la valeur minimale de *CFL* (Coefficient de Fiabilité de Lien) dans les diades incluant un mot monosémique et celles incluant un mot vide.

La valeur la plus basse de *CFL* pour les diades incluant un mot vide au moins est dans 90% des cas inférieure à 0.1% (cf. figure 4.21). D'un autre côté, choisir cette valeur comme valeur de départ de la boucle principale pour le démarrage du paramètre *Val-Min-CFL* nous permet de conserver 75% des liens incluant un mot monosémique.

Valeur de départ de Val-Activ-CFL

Pour déterminer la valeur de départ de *Val-Activ-CFL* dans la boucle principale nous comparons la valeur maximale des deux valeurs *CFL* des diades incluant soit un mot monosémique soit un mot vide au moins.

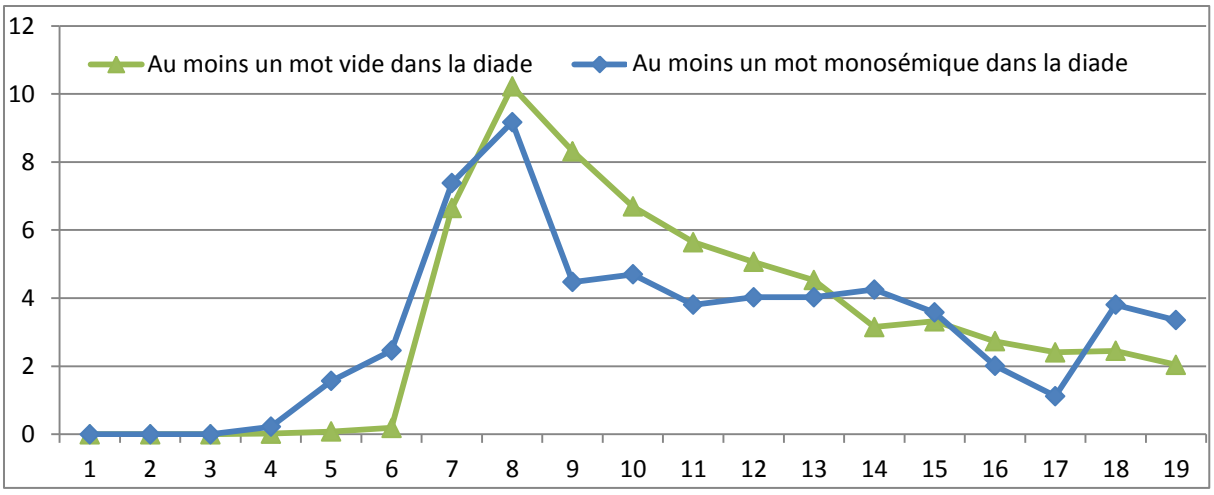


Figure 4.22 : Comparaison de la distribution de la valeur maximale de *CFL* (Coefficient de Fiabilité de Lien) dans les diades incluant un mot monosémique et celles incluant un mot vide.

S'il n'y pas de différence notable entre la distribution des diades incluant un mot monosémique et celles incluant un mot vide (cf. image 4.19), on peut affirmer qu'en-dessous de 4% le nombre de liaisons est extrêmement faible et ceci pour les deux familles de diades. Aussi nous utiliserons cette valeur pour valeur de départ de notre expérimentation.

Nombre de « pas » de la boucle principale et de la boucle fine

Nous choisissons finalement et arbitrairement 20 « pas » pour la boucle principale et 20 « pas » pour la boucle fine. Au-delà de 50 « pas » l'augmentation du nombre semble avoir un impact très minime dans la construction des agrégats. Cependant, ce nombre de 50 réclame un temps CPU trop important. Le choix du nombre de 20 nous est apparu comme un compromis raisonnable.

Moteur de recherche utilisé

Le moteur de recherche utilisé dans cette expérimentation est bing.com. Nous modifions le moteur utilisé car AOL.com détecte le fait que la tâche est robotisée et refuse de nous répondre.

4.4.3 Rigidification Régulée sur le réseau « 100 mots dans AOL » avec validation par MCCVS

Agrégats créés dans le réseau « 100 mots dans AOL »

La méthode de Rigidification Régulée nous a permis de créer, sur le réseau « 100 mots dans AOL » : 2196 agrégats de 3 à 29 éléments. Le nombre moyen de mots par agrégat est de 4.6.

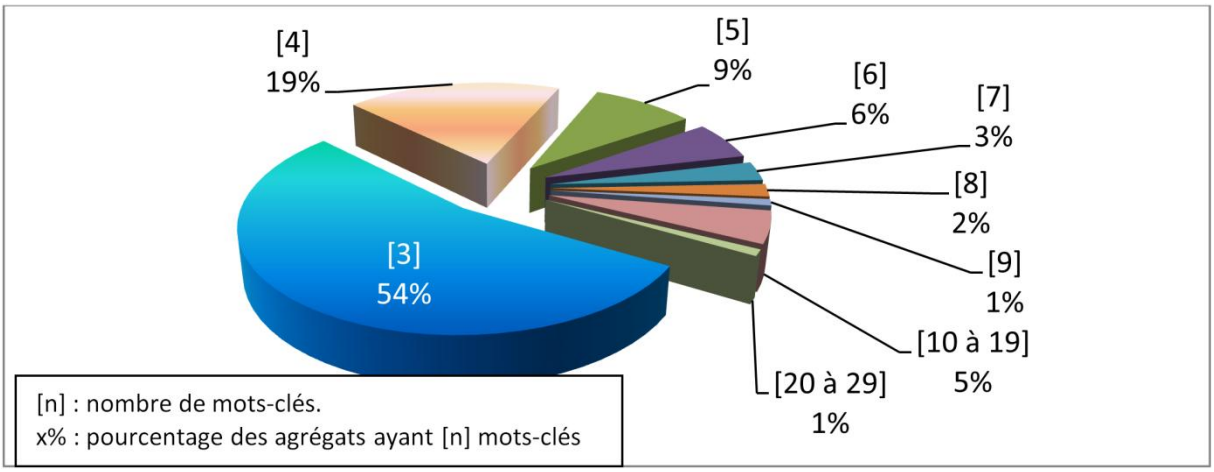


Figure 4.23 : Distribution du nombre de mots-clés par agrégat sur le réseau « 100 mots dans AOL ».

Rejet des mots vides

La méthode possède une capacité importante à rejeter les mots vides. Si dans certains cas particuliers ces mots vides sont utiles, leur éjection est souvent nécessaire pour éviter la création d'agrégats de trop grande taille. Nous ne retrouvons que 62 mots-clés de la liste http://snowball.tartarus.org/algorithms/english/stop.txt qui en contient 220 sur les 1090 mots-clés utilisés dans les agrégats.

Estimation de la qualité sémantique des agrégats par MCCVS

En suivant la méthode MCCVS, nous créons aléatoirement des trios de mots et comparons le nombre de sites retournés par un moteur de recherche avec des triades de mots ayant été utilisées conjointement dans une requête utilisateur au moins.

De la nécessité de filtrer les mots avant de les envoyer dans un moteur de recherche

Nous travaillons ici sur un réseau représentant le fichier de log d'AOL dans son ensemble. Les mots utilisés une fois ou deux peuvent être considérés, le plus souvent comme des erreurs potentielles. Pourtant, que ce soit dans le choix aléatoire d'un mot pour créer un trio aléatoire ou le test d'un agrégat, ce mot n'est pas utilisé en fonction de son usage par les utilisateurs mais par le simple fait de sa présence. Ainsi, dans la création d'un trio aléatoire, un mot utilisé une fois ou très rarement va avoir autant de chance d'être sélectionné qu'un mot utilisé des centaines de milliers de fois.

De même, les mots vides présents dans le réseau peuvent aussi se retrouver combinés. Dans ce cas, le moteur de recherche renvoie un nombre extrêmement élevé de sites trouvés. Par exemple, pour la requête « +the +and +or », big.com retourne 3 860 000 000 sites trouvés (big.com juillet 2011).

Dans le test des agrégats où l'on combine tous les mots en trios, un mot rare « pèse » aussi de manière exagérée. Supposons qu'un mot rare soit présent dans un agrégat de 10 mots, on le trouvera dans 36 des combinaisons testées ; de même, dans un agrégat de 20 mots il sera présent dans 171 combinaisons. Si ce mot est une erreur de frappe, comme c'est le cas pour la plupart de ces mots rares, il conduit le moteur de recherche à retourner très peu de sites voire aucun sur les 171 requêtes. Alors que dans le log d'AOL ce mot n'est présent que dans une seule requête sur plus de 22 millions, il intervient de manière beaucoup trop importante dans la validation des agrégats.

Donnons ici pour exemple un agrégat *Ag* créé par la méthode de Rigidification Régulée sur le réseau « 100 mots dans AOL » (les valeurs entre parenthèses sont le nombre de requêtes utilisant le terme) :

Ag = {system (15858), digestive (1378), diapra(25), demonstraion(1)}.

Les mots « diapra » et « demonstraion » n'existent pas et nous pouvons imaginer que ce sont des erreurs.

Combinaisons	Nombres de sites retournés par bing.com (décembre 2010)
+system +digestive +diapra	0
+system +digestive +demonstraion	14
+system +diapra +demonstraion	0
+digestive +diapra +demonstraion	0

Tableau 4.12 : Exemple de combinaisons de mots incluant des mots à faible usage dans des recherches.

La solution pourrait être de supprimer simplement les mots de faible utilisation et les mots vides du graphe à tester. Mais si nous voulons préserver la capacité à détecter de nouvelles communautés d'utilisateurs par l'usage de nouveaux mots et la capacité à créer des agrégats basés sur l'utilisation conjointe de mots vides, il nous faut conserver ces mots dans le graphe étudié.

Ces problèmes ont moins de conséquence dans l'étude des réseaux AOL-17/04/2006 et AOL-17/03/2006. En effet, la part de mots rarement ou très rarement utilisés (et étant des erreurs) ne peut qu'augmenter avec la taille du fichier de log. De plus, dans l'étude de ces réseaux, avec la méthode de Rigidification Simple nous filtrons préalablement les mots vides qui n'étaient donc pas présents ni dans les agrégats ni dans les requêtes de test. La méthode de Rigidification Régulée que nous évaluons ici a permis de les conserver pour les raisons évoquées plus haut.

Définition du filtre préalable avant l'évaluation sémantique

Afin de créer un ensemble valide et d'éviter des combinaisons surpondérées pour le test d'évaluation sémantique, sont exclus de l'évaluation sémantique les mots très utilisés et les mots très peu utilisés.

Les mots très utilisés sont issus de la liste déjà évoquée ; http://snowball.tartarus.org/algorithms/english/stop.txt. Bien que peu nombreux, ils représentent 10.06 % des usages (ensemble des mots multipliés par le nombre de requêtes dans lesquels le mot est présent, ces mots vides étant généralement les plus usités).

Les mots de faible utilisation sont écartés en fonction de leur valeur globale d'utilisation (ensemble des mots multipliés par le nombre de requêtes dans lesquelles le mot est présent) jusqu'à obtenir 10% des usages. Nous retirerons donc les mots qui ont été utilisés moins de 94 fois. Ainsi, nous ne conservons ni les mots définis comme vides, ni les termes de faible utilisation (présents dans moins de 94 requêtes) de façon à travailler sur des mots correspondant à 80% des usages.

Bien sûr, la démarche est la même dans les trois systèmes de génération de requêtes. Seuls les agrégats ayant au moins trois mots après filtrage sont considérés comme valides pour être évalués.

Valeur de CSVC

Avec la méthode MCSVS, il s'agit de mesurer et comparer la distribution du nombre de sites retournés sur un échantillon de 100 000 requêtes faites de trios de mots aléatoires avec 100 000 triades issues de requêtes utilisateurs. Les mots sont ici ceux définis dans le paragraphe « Définition du filtre préalable avant l'évaluation sémantique ».

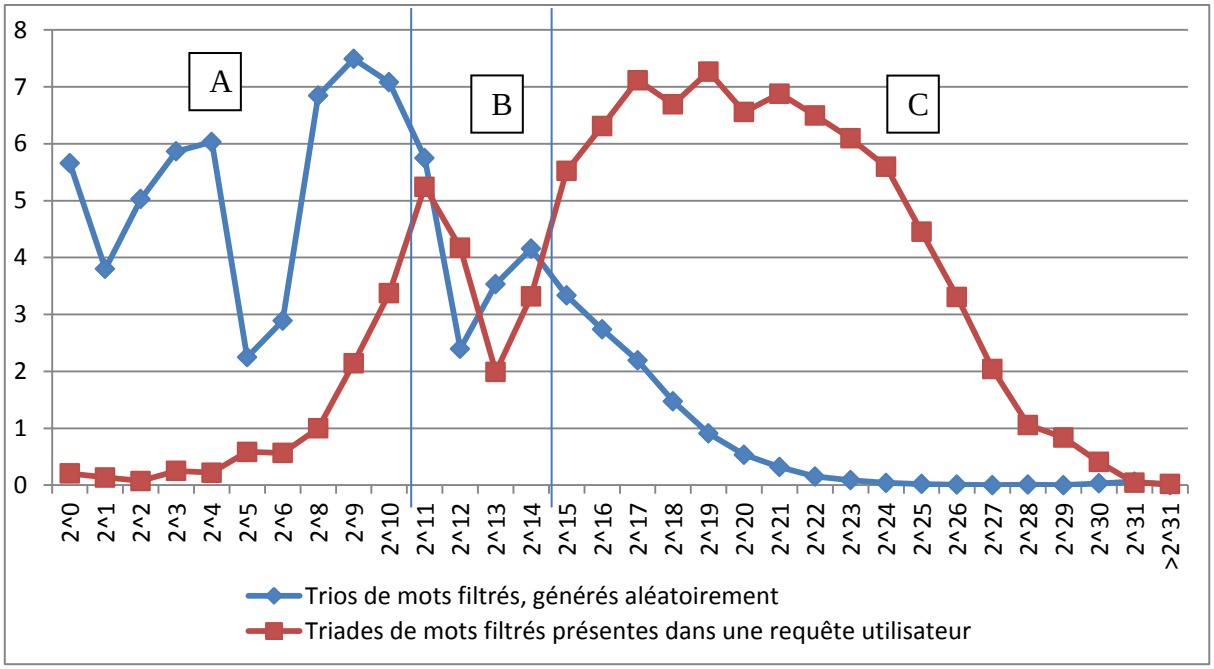


Figure 4.24 : Comparaison des réponses aux requêtes susceptibles d'être les plus éloignées sémantiquement (cf. 4.3.1) et détermination des zones à forte divergence.

L'observation des trois courbes nous permet de détecter trois zones :

- la zone « A » très accidentée ;
- la zone « B » où les deux courbes sont proches ;
- la zone « C » où les courbes sont bien différenciées et lisses. Cette dernière caractéristique confirme l'aspect non accidentel des mesures.

Nous utilisons donc la zone « C » comme zone de validation sémantique.

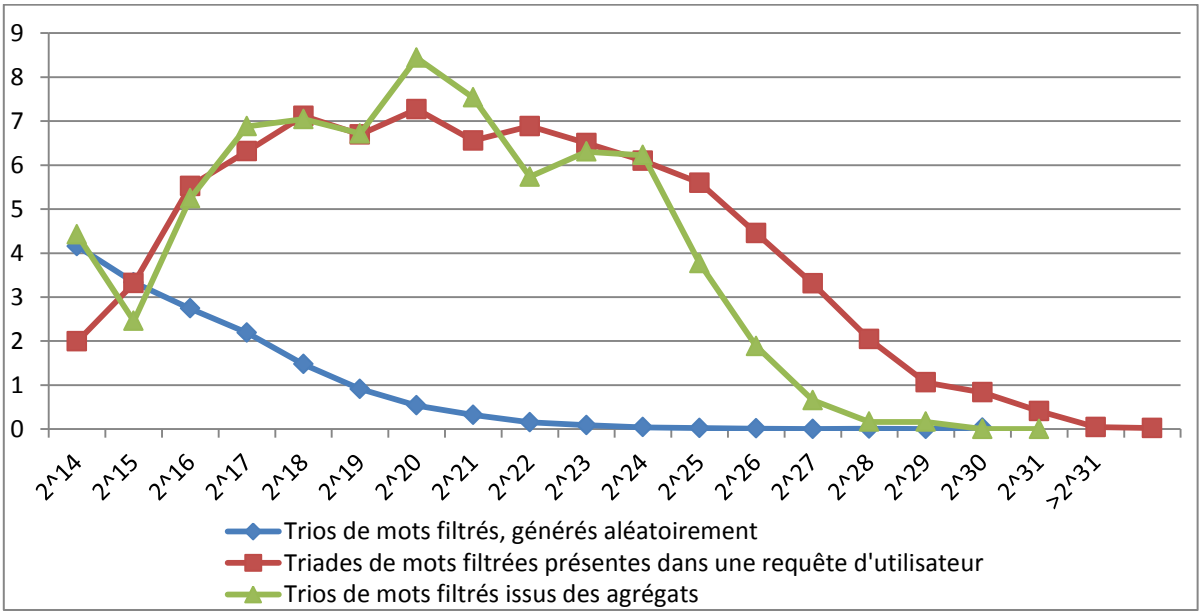


Figure 4.25 : Représentation graphique de la zone « C » de validation sémantique sur les trois courbes représentant les trois sources de requêtes.

Nous calculons le *CVSC* des requêtes construites à partir des agrégats :

$$CVSC_X = (A_X - A_A) / (A_R - A_A)$$

où A_R définit l'aire de l'histogramme des triades et où les trois mots sont conjointement présents dans une requête utilisateur au moins,

$$A_R = \sum_{i=14}^{31} Yi = 80.04$$

où A_A définit l'aire de l'histogramme des trios de mots générés aléatoirement,

$$A_A = \sum_{i=14}^{31} Y'i = 16.07$$

où A_x définit l'aire de l'histogramme des trios de mots issus d'agrégats.

$$A_X = \sum_{i=14}^{31} Y''i = 73.62$$

On a donc :

$$CVSC_X = (73.62 - 16.07) / (80.04 - 16.07) = 0.899$$

Conclusion

L'échantillon de test pour la méthode de validation MCVSV est à adapter s'il provient de très grands graphes de mots, particulièrement si ceux-ci sont « pollués » par un grand nombre d'erreurs ou de mots vides. Le choix d'écarter les mots correspondant aux 20 % les plus marginaux doit être considéré en se remémorant que MCVSV est une méthode comparative. Ainsi, si les conditions de mesure sont les mêmes pour l'ensemble des courbes repères et les éléments issus des agrégats et que les courbes de référence (aléatoires et utilisateurs) sont suffisamment différenciées, la méthode nous semble rester pertinente.

Avec une valeur de *CVSC* de .899, nous obtenons une excellente valeur du Coefficient de Validation Sémantique Comparé (en basant toujours la limite sur la valeur médiane de 0.5). La méthode d'agrégation est validée comme ayant sur des Méga-graphes de mots, la capacité à créer des agrégats qui ont statistiquement une cohérence sémantique certaine et cela depuis un réseau non préalablement filtré.

4.4.4 Rigidification Régulée sur le réseau « 100 mots dans AOL » avec validation par MCSDR.

Dans cette expérimentation nous utilisons la méthode de mesure de MCSDR ou « Méthode de Comparaison de la Similarité entre Documents Retournés » (cf. paragraphe

4.3.3) sur le réseau « 100 mots dans AOL » et les agrégats créés par la méthode de Rigidification Régulée (cf. paragraphe 4.4.3).

Filtrer les mots avant l'évaluation sémantique ?

De la même manière que dans la méthode de validation MCCVS (cf. paragraphe 4.4.3), pour le réseau « 100 mots dans AOL », nous avons choisi de supprimer les mots qui sont dans la liste des mots vides et les mots qui ont été faiblement utilisés. Le filtre est identique à celui de l'expérimentation sur ce réseau avec la méthode de validation MCCVS. Le lecteur peut se reporter au paragraphe 4.4.2 pour la description de ce filtre. Les mots conservés correspondent à 80% des usages.

La phase d'acquisition des articles de Wikipédia

Nous avons testé 6716 trios de mots pour les trois types de requêtes (aléatoires, agrégats, utilisateurs). Les dix premiers articles de Wikipédia valides (entre 200 et 15000 mots) retournés pour chaque requête ont été indexés. Le nombre de 10 représente une valeur maximale, une requête peut en retourner moins. Un total de 33845 articles a été indexé, 280530 mots différents ont été trouvés.

Résultats

Les articles retournés par le moteur de recherche pour une même requête sont comparés deux à deux. Nous observons ensuite la distribution pour la moyenne de la valeur de similarité des articles retournés par une même requête. Si un seul article est présent la valeur de similarité est considérée comme nulle.

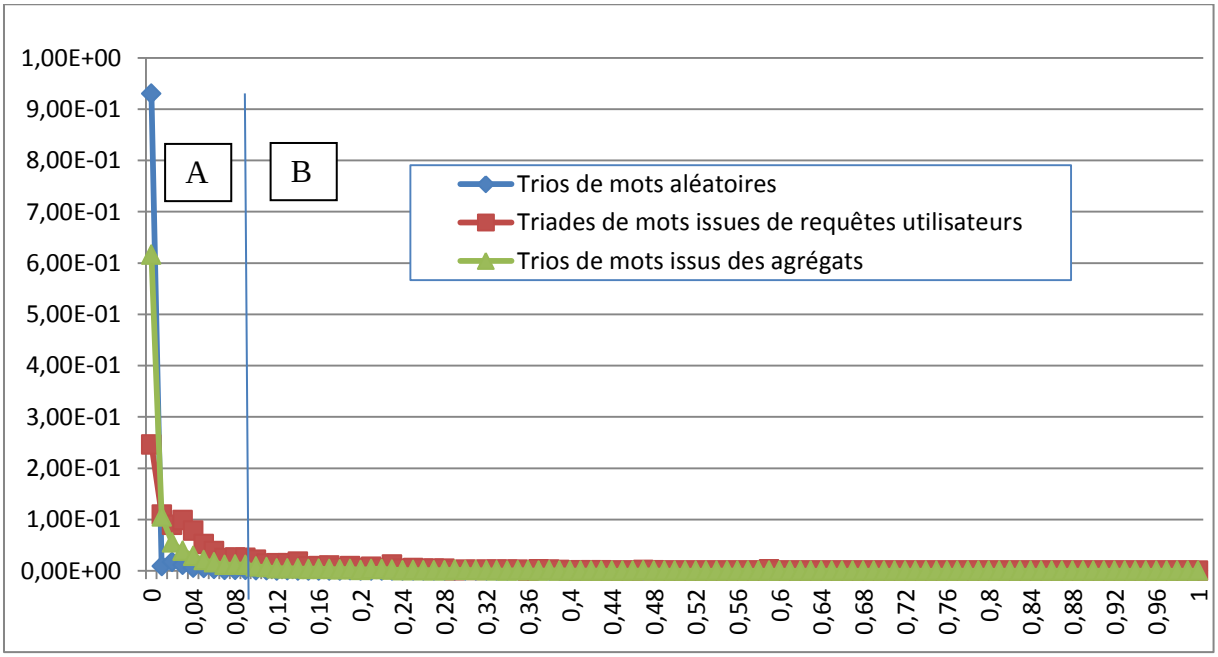


Figure 4.26 : Distribution de la moyenne des similarités entre documents retournés par les trois types de requêtes en « inter-requêtes ».

A l'analyse de la figure 4.26, on remarque deux zones :

- la zone A est la zone présentant une certaine disparité entre les courbes de référence (aléatoire et utilisateur). Cette zone est extrêmement étroite ;
- la zone B qui ne fait pas ressortir de différence notable entre les courbes de référence.

Comme on peut le constater, la différence principale entre les courbes réside dans le pourcentage de requêtes n'ayant pas retourné de site. Afin de replacer cette zone dans un espace de lecture où l'estimation des courbes est possible, nous comparons les courbes en supprimant pour chacune d'elles les requêtes ayant retourné moins de deux articles. Nous notons ensuite (toujours pour les requêtes ayant retourné au moins deux articles) la distribution des moyennes de la similarité inter-requêtes et intra-requête comme nous l'avions défini dans notre protocole de validation. La mesure des distances inter-requêtes issues des agrégats se fait entre des requêtes d'agrégats différents. Au total plus de 10 millions de comparaisons de documents ont été effectuées.

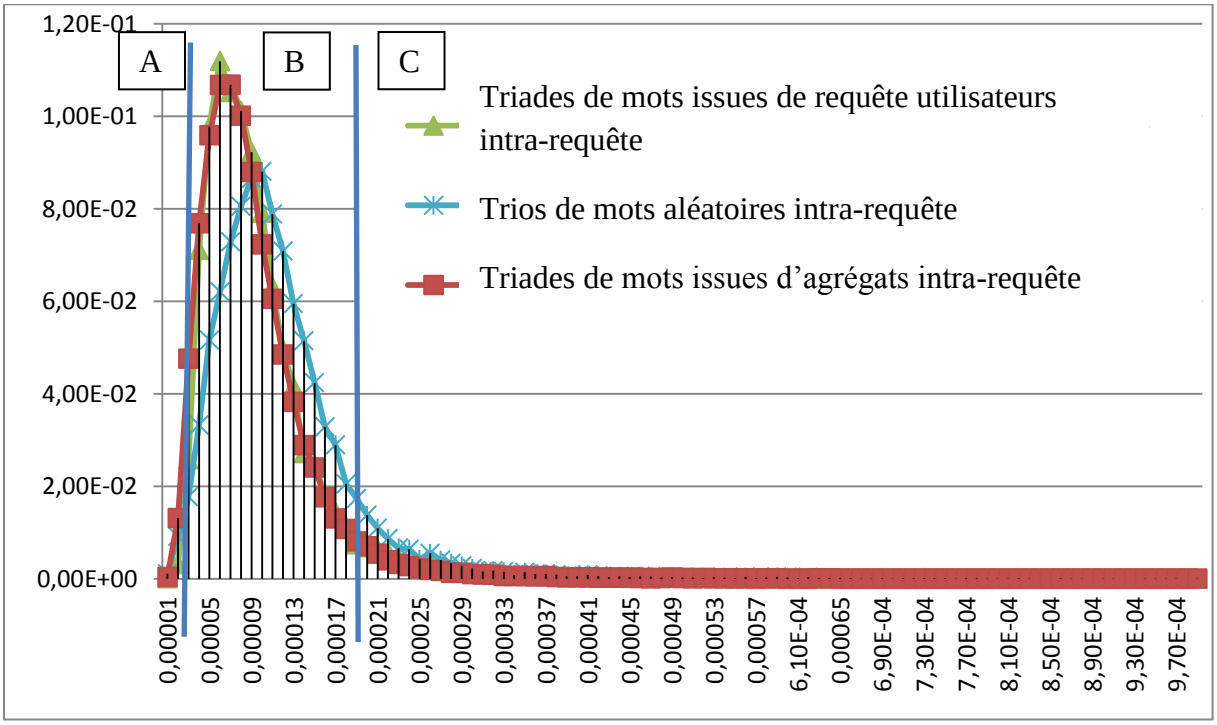
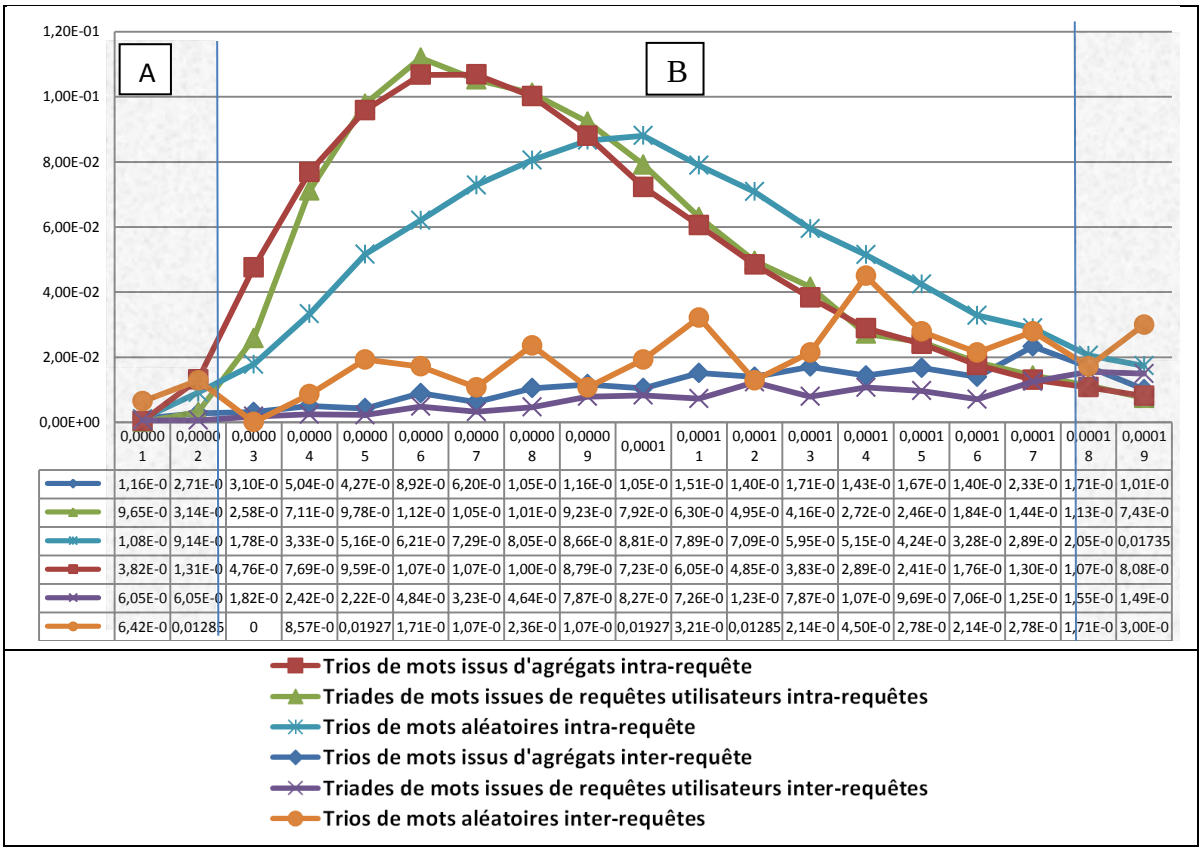


Figure 4.27 : Distribution de la moyenne des similarités entre documents retournés par les trois types de requêtes [intra-requête].

La zone B est la zone choisie comme zone « différenciatrice » sur les deux courbes de référence (cf. figure 4.27)



	0,00001	0,00002	0,00003	0,00004	0,00005	0,00006	0,00007	0,00008	0,00009	0,0001	0,00011	0,00012	0,00013	0,00014	0,00015	0,00016	0,00017	0,00018	0,00019
	1,16E-0	2,71E-0	3,10E-0	5,04E-0	4,27E-0	8,92E-0	6,20E-0	1,05E-0	1,16E-0	1,05E-0	1,51E-0	1,40E-0	1,71E-0	1,43E-0	1,67E-0	1,40E-0	2,33E-0	1,71E-0	1,01E-0
	9,65E-0	3,14E-0	2,58E-0	7,11E-0	9,78E-0	1,12E-0	1,05E-0	1,01E-0	9,23E-0	7,92E-0	6,30E-0	4,95E-0	4,16E-0	2,72E-0	2,46E-0	1,84E-0	1,44E-0	1,13E-0	7,43E-0
	1,08E-0	9,14E-0	1,78E-0	3,33E-0	5,16E-0	6,21E-0	7,29E-0	8,05E-0	8,66E-0	8,81E-0	7,89E-0	7,09E-0	5,95E-0	5,15E-0	4,24E-0	3,28E-0	2,89E-0	2,05E-0	0,01735
	3,82E-0	1,31E-0	4,76E-0	7,69E-0	9,59E-0	1,07E-0	1,07E-0	1,00E-0	8,79E-0	7,23E-0	6,05E-0	4,85E-0	3,83E-0	2,89E-0	2,41E-0	1,76E-0	1,30E-0	1,07E-0	8,08E-0
	6,05E-0	6,05E-0	1,82E-0	2,42E-0	2,22E-0	4,84E-0	3,23E-0	4,64E-0	7,87E-0	8,27E-0	7,26E-0	1,23E-0	7,87E-0	1,07E-0	9,69E-0	7,06E-0	1,25E-0	1,55E-0	1,49E-0
	6,42E-0	0,01285	0	8,57E-0	0,01927	1,71E-0	1,07E-0	2,36E-0	1,07E-0	0,01927	3,21E-0	0,01285	2,14E-0	4,50E-0	2,78E-0	2,14E-0	2,78E-0	1,71E-0	3,00E-0

Figure 4.28 : Distribution de la moyenne des similarités entre documents retournés par les trois types de requêtes inter-requêtes et intra-requête (zone B).

La valeur du *QCSC* (Quotient de Centralité Sémantique Comparé) est sur la Zone « B » telle que définie au paragraphe 4.3.3 de 0.89864.

Conclusion

Avec une valeur de *QCSC* supérieure à 0.89, la qualité des agrégats semble excellente. Cependant, la méthode utilise un moteur de recherche du marché (bing.com) dont nous ne contrôlons pas le système d'ordonnancement. Les dix premiers sites retournés le sont par des algorithmes d'ordonnancement du moteur de recherche qui prennent en compte d'autres mesures que la simple présence des mots clés.

La méthode est complexe et coûteuse sur le plan computationnel. De plus, de nombreuses difficultés techniques apparaissent. Par exemple, filtrer le code « HTML » est, quel que soit le « parser » utilisé, une opération jamais réussie à 100% sur l'ensemble des pages. De plus, nous limitons cette évaluation à une bibliothèque de documents (wikipedia.org) qui représente aussi une limite sur l'ensemble des sujets abordés.

Mais une partie de ces difficultés est compensée par la nature comparative de la méthode. Ainsi, si l'erreur est constante ou proportionnelle elle n'influe que faiblement sur la comparaison des différentes courbes.

4.4.5 Rigidification Régulée sur réseau eDonkey-10-semaine et validation manuelle

Spécificité du réseau étudié

Ce réseau inclut des recherches effectuées par des pédophiles. Il est fourni sous contrat de confidentialité par une unité de recherche du CNRS spécialisée dans la détection d'activités pédophiles sur Internet (cf. paragraphe 4.2.2.). La grande majorité des mots est anonymisée.

Paramétrage et particularité de l'algorithme

Le but de cette expérimentation est de proposer à un expert des agrégats qui sont susceptibles de contenir des mots à connotation « pédophile ». Le réseau de départ contient un certain nombre de termes pédophiles « bien connus ». L'expert espère trouver dans les agrégats, en plus du lexique « bien connu », de nouveaux mots pouvant être classés comme « utilisés par des pédophiles » ou susceptibles de l'être. Il souhaite aussi, en plus de se voir proposer des mots correspondant à de nouveaux usages, être en mesure de valider la méthode, par la présence de « mots bien connus » supplémentaires.

Le nombre maximal de mots dans un agrégat est défini à 80. Cela peut sembler important puisque dans un agrégat ayant une bonne cohérence sémantique ce nombre a été déterminé comme étant entre 30 et 40 mots (cf. paragraphe 4.4.2). La raison est que dans le cadre de cette expérimentation nous voulons proposer le plus de termes possible à notre expert. Si certains d'entre eux ne sont pas à connotation pédophile, la validation manuelle pourra le détecter. Mais en aucun cas, nous ne voudrions omettre un mot susceptible d'être classé comme « nouveau mot utilisé par les pédophiles ».

Ce nombre de 80 est une valeur maximale qui a une faible incidence sur notre résultat : Seuls 2% des agrégats dépassent la taille de 39 mots.

Valeur de départ de *Val-Min-CFL* et de *Val-Activ-CFL*

A la différence de l'expérimentation menée avec la méthode de Rigidification Régulée sur le réseau « 100 mots dans AOL », il n'est pas possible de déterminer les valeurs de départ par une observation comportementale de mots aux caractéristiques sémantiques connues (mots vides et mots monosémiques). Les valeurs de départ sont donc sélectionnées par tests successifs. Les valeurs de *Val-Min-CFL* et *Val-Activ-CFL* sont choisies extrêmement basses. Puis elles sont testées sur un échantillon du graphe et augmentées jusqu'à ce que les paramètres de *Val-Min-CFL* et *Val-Activ-CFL* permettent de créer des graphes avec un certain équilibre. Ce qui signifie que nous créons des agrégats de taille variée et que dans certains agrégats créés, nous avons la capacité d'incorporer des mots à fort et faible usage.

La valeur de départ de *Val-Min-CFL* est de 3% et la valeur de départ de *Val-Activ-CFL* est de 10%.

Résultats

Nous créons 173 agrégats répartis de la sorte et incluant 1549 mots-clés.

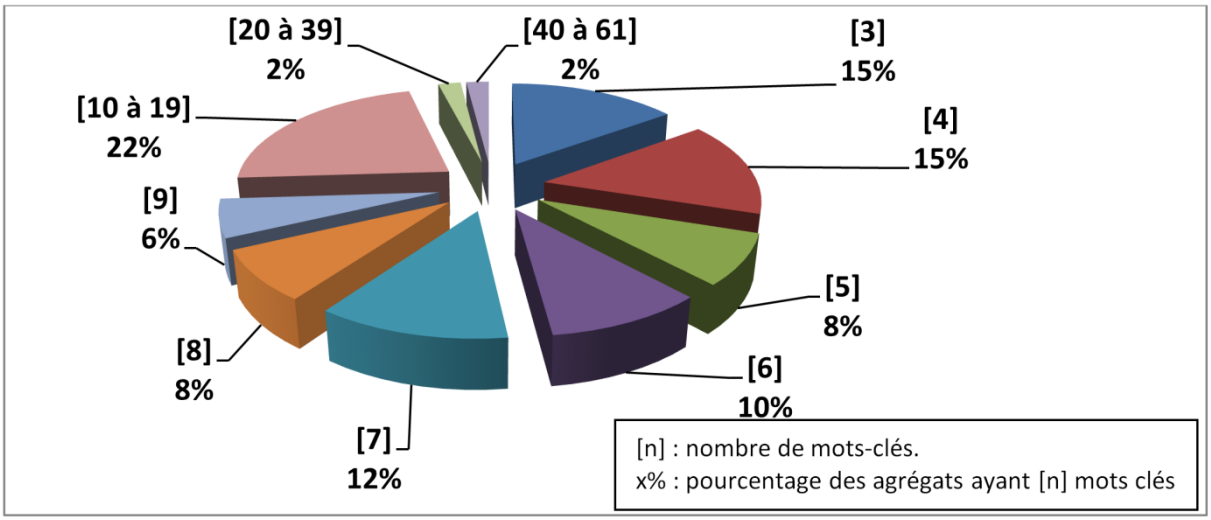


Figure 4.29 : Répartition des agrégats selon le nombre de mots-clés.

Mots-clés « bien connus »	Poids	Nombre d'agrégats	Taille Max.	Taille Moyenne	Taille Min.
pthc	45737	96	78	9	3
incest	13609	70	52	11	3
ygold	9183	19	61	15	3
ptsc	3189	14	11	6	3
incesti	1277	2	4	3.5	3
inceste	1220	3	17	12	7
4yo	1042	4	14	9	4
3yo	832	3	12	10	8

Tableau 4.13 : Agrégats incluant les huit mots « bien connus » comme étant utilisés par les pédophiles.

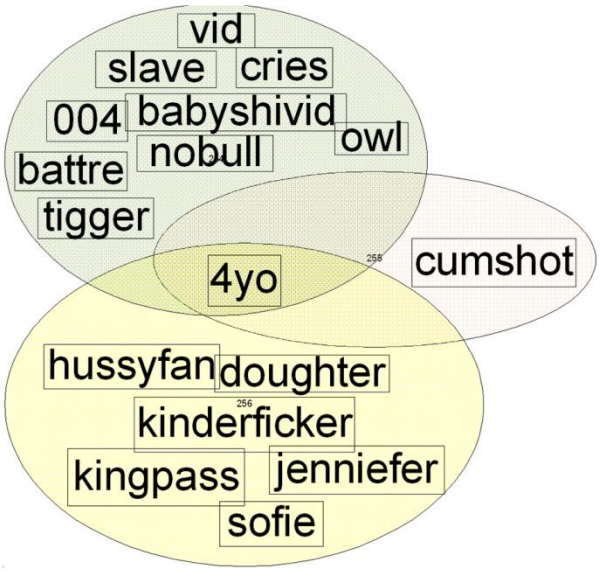


Figure 4.30 : Exemple d'agrégat autour du mot 4yo.

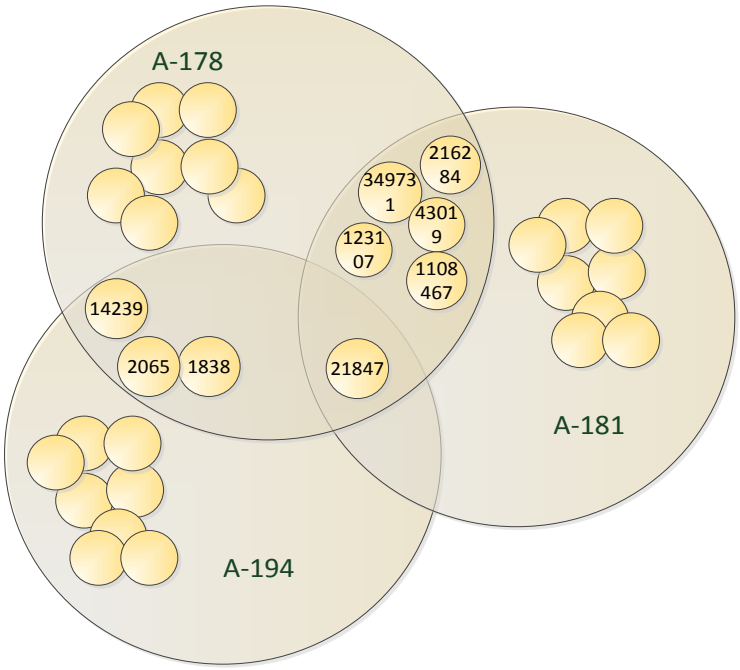


Figure 4.31 : Exemple d'agrégats avec des recouvrements importants.

Estimation de la validité sémantique des agrégats

La validation est laissée à l'entière appréciation de l'expert. Le rôle d'expert est ici joué par Matthieu Latapy. Monsieur Latapy est responsable du projet « *Measurement and Analysis of P2P Activity Against Paedophile Content* » dont on peut trouver la description sur le site http://antipaedo.lip6.fr/. Ce projet est soutenu par l'Union Européenne, l'ANR, le CNRS, l'UPMC, l'UCC, l'UL, le FDN and l'INRIA.

Cette évaluation a été faîte sans système de « note » ou de comparaison. L'expert détermine simplement que les agrégats « présentent une cohérence sémantique » ou pas et s'il découvre de nouveaux mots susceptibles d'être des mots utilisés spécifiquement dans le cadre de requêtes à caractère pédophile.

Conclusion

L'expert valide globalement que la méthode d'agrégation possède une capacité à créer des agrégats « présentant une cohérence sémantique ».

Cependant cette méthode de validation manuelle est décevante. Car la lecture des agrégats ne permet pas d'obtenir de commentaires précis. Si les commentaires de l'expert sont positifs (donc encourageants), ils ne nous guident en aucune façon pour faire évoluer les algorithmes proposés.

La récupération de nouveaux mots potentiellement utilisés par des pédophiles pose aussi la limite du travail de l'expert. Comment peut-il évaluer un système dont les résultats sont pour lui une nouvelle information dont il ne connait pas, à fortiori, la validité.

4.4.6 Méthode de Rigidification Régulée sur réseau TREC-Eval-5 et validation par méthode TREC-Eval

Paramétrage et particularité de l'algorithme

Le but, dans cette expérimentation, est de mesurer la capacité d'un agrégat à être utilisé pour compléter des requêtes utilisateurs. Une question se pose : dans les agrégats créés dans le cadre de cette expérimentation peut-on trouver des mots capables d'enrichir des requêtes ? Les mots se situant à la limite de la cohérence sémantique de l'agrégat vont fortement « bruiter » la requête et sans aucun doute faire baisser le niveau de qualité des réponses.

Pour conserver un coefficient sémantique élevé au sein des agrégats nous choisissons de limiter le nombre maximal de mots dans un agrégat à 30 mots. C'est là le premier seuil « d'écroulement » de la cohérence sémantique des agrégats (cf. paragraphe 4.4.2). Nous espérons ici réduire dans les requêtes le bruit lié à l'introduction de nouveaux mots.

Valeurs de départ de *Val-Min-CFL* et de *Val-Activ-CFL*

Le faible nombre de mots ainsi que la faible taille des échantillons ne nous permettent pas de faire une étude statistique sur le comportement des mots vides et monosémiques. Après plusieurs essais fructueux nous optons pour des valeurs de départ de 3% pour *Val-Min-CFL* et de 10% pour *Val-Activ-CFL*.

Résultats de la méthode d'agrégation de Rigidification Régulée

184 agrégats sont créés. Ils sont ensuite liés avec un ou plusieurs des 199 Topics (requêtes utilisateurs dans TREC-Eval) par la régle suivante : ¼ des mots présents dans le Topic sont présents dans l'agrégat, avec au moins un mot.

121 Topics sur 199 inclus dans l'expérimentation sont alors liés avec au moins un agrégat.

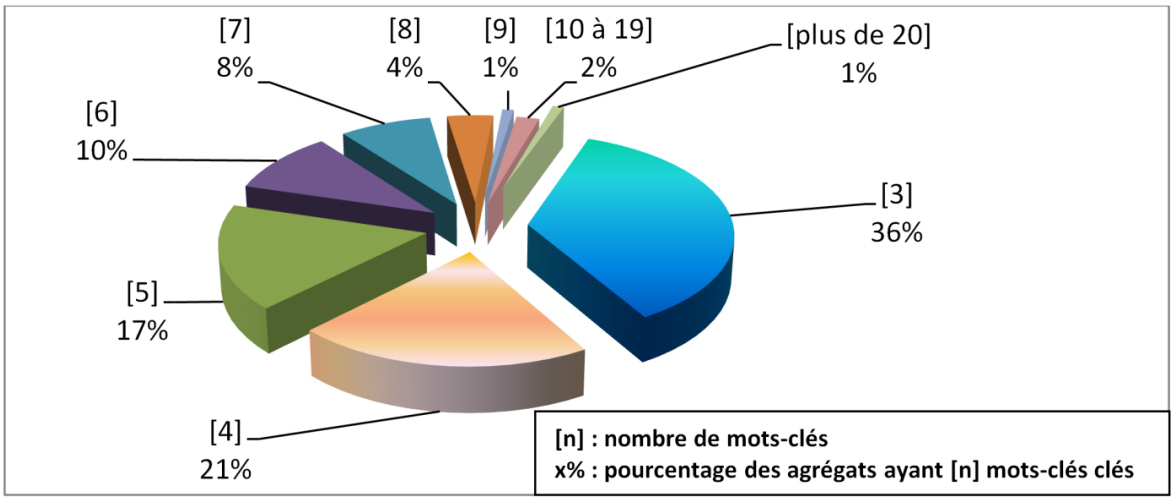


Figure 4.32 : Répartition des agrégats par nombre de mots-clés.

Estimation de la valeur sémantique des agrégats

Nous utilisons quatre types de requête :

1. Un Topic est une requête effectuée par un utilisateur.
2. Les agrégats seuls (incluant des mots du Topic) : ils sont utilisés en tant requête.
3. les Topics enrichis : par l'ajout de mots issus d'agrégats (agrégats auxquels les mots du Topic initial appartiennent).
4. Les Topics enrichis avec surpondération des mots initiaux du Topic : Les Topics sont enrichis par l'adjonction de mots issus d'agrégats (agrégats auxquels les mots du Topic initial appartiennent) mais les mots initiaux du Topic sont surpondérés dans la recherche.

Afin d'illustrer la nature des différentes requêtes manipulées, nous présentons dans le tableau 4.14 plusieurs exemples pour les quatre types de requêtes.

QUI	Topic	Agrégat	Topic enrichi	Topic enrichi avec surpondération des mots du Topic
24	New Medical Technology	and/or diseases human inherited medical potential	and/or diseases human inherited medical potential new technology	and/or diseases human inherited medical^2 potential new^2 technology^2
24	New Medical Technology	computer-aided diagnosis medical	computer-aided diagnosis medical new technologie	computer-aided diagnosis medical^2 new^2 technologie^2
24	New Medical Technology	controlling high technology transfer	controlling high technology transfer new medical	controlling high technology^2 transfer new^2 medical^2
25	Aftermath of Chernobyl	aftermath loss revenue televangelist	aftermath loss revenue televangelist of chernobyl	aftermath^2 loss revenue televangelist of^2 chernobyl^2
25	Aftermath of Chernobyl	accident chernobyl contain results	accident chernobyl contain results aftermath of	accident chernobyl^2 contain results aftermath^2 of^2

Table 4.14 :Exemple de requêtes constituées de Topics, d'agrégats, de Topics enrichis et de Topic enrichis avec surpondération.

QID	M.A.P. obtenue par le Topic	M.A.P. obtenue par l'aggregate	M.A.P. obtenue par le Topic enrichi des mots de l'agrégat	M.A.P. obtenue par le Topic enrichi des mots de l'agrégat avec surpondération des mots du Topic
24	0.0025	0.0090	0.0148	0.0159
24	0.0025	0.0010	0.0002	0.0005
24	0.0025	0.0000	0.0000	0.0008
25	0.0294	0.0000	0.0073	0.0283
25	0.0294	0.0399	0.0311	0.0304

Table 4.15 : M.A.P. pour les Topics, agrégats et Topics enrichis par l'agrégat, de Topic enrichis et de Topic enrichis avec surpondération des mots du Topic.

Sur ces 121 Topics, l'utilisation des agrégats, des Topics enrichis par les agrégats ou encore des Topics enrichis par les agrégats en augmentant le poids des mots des Topics nous a permis d'améliorer ou de maintenir la M.A.P. dans 76 cas. La M.A.P. a même été améliorée

dans 67 cas, soit dans 55% des cas. Dans le cas où plusieurs agrégats sont liés à un Topic, nous avons comparé le résultat du M.A.P. du Topic seul au meilleur des résultats.

Le seul fait que l'adjonction de mots nous permet de garder une valeur de M.A.P. équivalente est déjà un succès. En effet, cela signifie que les mots ajoutés ne viennent pas « brouiller » la requête. En ajoutant un mot même sémantiquement proche nous introduisons la possibilité de ramener des documents qui peuvent porter sur des thèmes divergents. N'oublions pas que nous ne cherchons pas ici à définir un système d'amélioration de requête, mais à déterminer si nos agrégats ont une certaine validité sémantique. Si la valeur de M.A.P. est simplement conservée cela signifie que les mots ajoutés n'ont pas « bruité » notre requête et qu'ils sont bien inclus dans un agrégat possédant une forte cohérence sémantique.

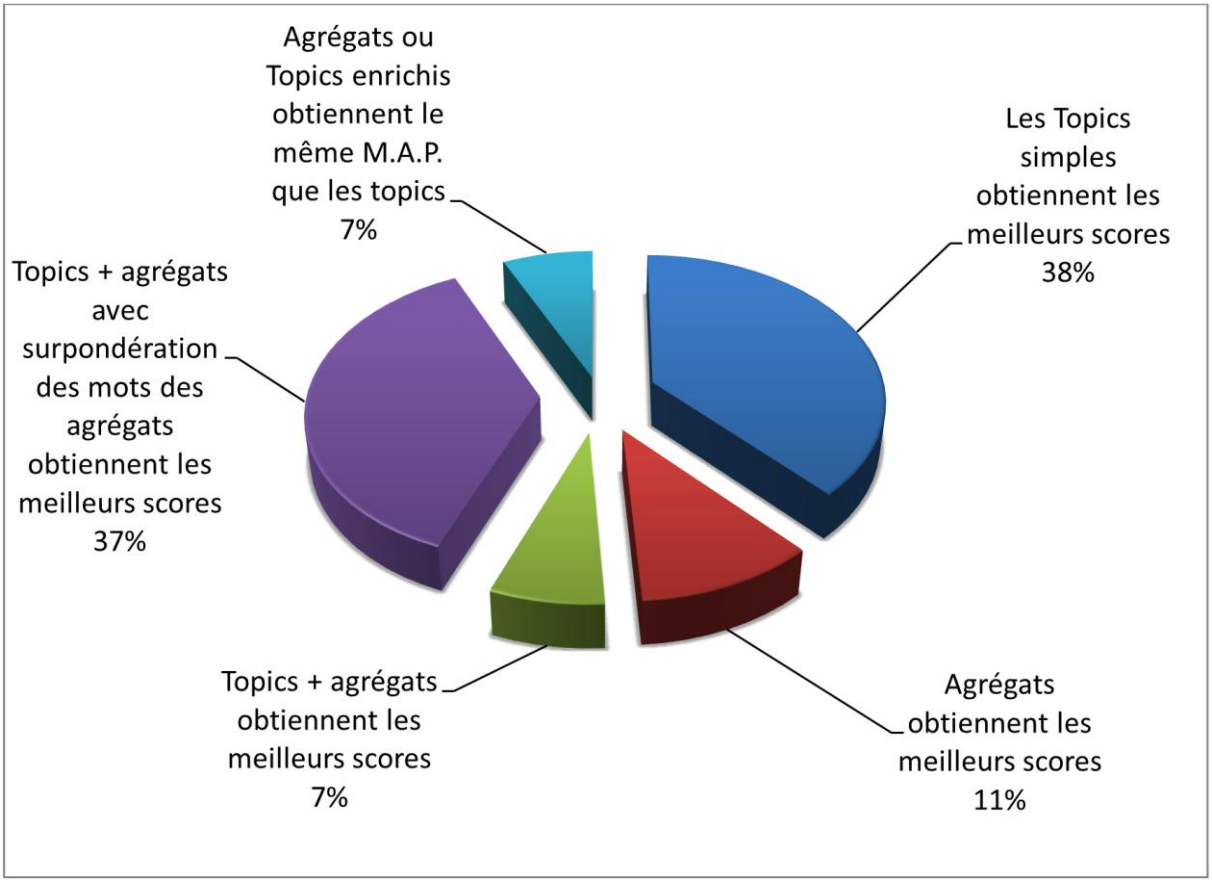


Figure 4.33 : Répartition des meilleurs résultats pour la valeur M.A.P. pour les 4 catégories de requêtes dans l'expérimentation TREC-Eval.

Les Topics simples (requêtes utilisateurs) ne sont plus efficaces que les Topics enrichis ou les agrégats utilisés comme requête ou encore les agrégats seuls que dans 38% des cas. C'est un résultat extrêmement positif.

Si nous comparons les moyennes des valeurs de M.A.P améliorées selon les 4 types de requêtes nous remarquons (pour les Topics améliorés) que :

- les agrégats seuls améliorent en moyenne la M.A.P. de 116% mais uniquement pour 21% des requêtes ;

- les Topics + agrégats améliorent encore la M.A.P. (137%) mais pour une part moins importante (seulement 12%) des requêtes ;
- Les Topics + agrégats avec surpondération des mots de l'agrégat améliorent la M.A.P. plus souvent (67%) mais moins fortement (65%) .

Contenu de la requête	Moyenne de l'amélioration	% des requêtes améliorées
Agrégat seul	116%	21%
Topic + Agrégat	137%	12%
Topic + Agrégat avec mots du Topic surpondérés	65%	67%

Table 4.16 : Répartition de l'amélioration de la valeur M.A.P. par type de requête.

Analyse des résultats

- l'agrégat seul : nous remarquons deux scénarios qui vont provoquer une nette amélioration (116% en moyenne) :
    - Dans le premier scénario, l'agrégat qui est utilisé comme élément de requête ne contient qu'un seul mot du Topic. L'agrégat est alors décalé sémantiquement du Topic. Il est cependant en réalité plus proche de la véritable thématique de la recherche et les résultats sont meilleurs.
    - Dans le deuxième cas, l'agrégat contient plusieurs mots du Topic. Dans ce cas les mots ajoutés ont permis de préciser la thématique. Nous nous rapprochons alors du type de requête Topic + agrégats, même si tous les mots de la requête ne sont pas présents.
- Topic + agrégat : les mots du Topic sont tous présents. L'agrégat intervient toujours en précisant la requête. Quand cela fonctionne les améliorations sont très importantes (137%) et on obtient des scores de M.A.P. très élevés. Cependant le risque de « bruiter » la requête en introduisant des mots fait que la réussite est plus rare.
- Topic + agrégat avec mots du Topic surpondérés : le but est de baisser le « bruit » introduit par des mots vides ou moins spécifiques tout en profitant des mots qui vont « compléter la requête ». Pour cela, les mots du Topic sont surpondérés. L'amélioration est alors plus fréquente mais moins importante, ce qui est logique, le poids de l'enrichissement est plus faible donc le risque moins important.

Conclusion

Cette étude sur les valeurs d'amélioration des requêtes ne doit pas nous détourner de notre objectif qui est de savoir si le système d'agrégation par Rigidification Régulée permet de créer des ensembles porteurs d'une thématique.

La réponse donnée par cette évaluation est bien sûr positive. L'utilisation des agrégats seuls ou conjointement avec les Topics a maintenu la précision de la requête identique dans 62% des cas.

Avec une amélioration dans plus de 55% des cas, les agrégats prouvent à la fois leur cohérence sémantique et le fait qu'ils sont même capables de servir de système d'amélioration des requêtes.

4.4.7 Méthode d'enrichissement des agrégats AGGR sur réseau « eDonkey-5 mois » et validation manuelle (challenge)

Matériel et conditions de test

Nous utilisons ici un réseau de très grande taille. Les agrégats sont fournis par un expert sous la forme de deux listes de mots. Il ne s'agit plus de construire des agrégats mais de chercher à les enrichir. La méthode ne possède pas de paramètre. Elle permet de simplement retourner une liste de mots ordonnée selon un coefficient d'attraction envers l'agrégat.

Résultats sur réseau eDonkey-5-mois – Validation manuelle

La validation est ici une validation manuelle comparée. Dans le cadre d'un « challenge » **[Belbeze&al-2009-2]**, des experts comparent plusieurs méthodes qui ont pour objectif de retourner deux listes de cent mots chacune.

En plus du réseau, l'organisateur du challenge propose deux listes de mots. Ceux-ci sont des mots « bien connus » comme étant utilisés par des pédophiles. Les experts sont des professionnels de la recherche de pédophiles sur Internet. Ils ont la charge de comparer la capacité des méthodes à retourner des mots en employant un classement en quatre types :

- type 1 : le mot est spécifiquement un mot pédophile connu. Il n'a pas d'autre utilisation. C'est généralement un code, par exemple : « pthc » ;
- type 2 : le mot est utilisé par les pédophiles, mais il peut être utilisé dans d'autres contextes, par exemple : « child » ;
- type 3 : le mot est inconnu des experts, mais il n'a pas d'autre sens connu, c'est soit un nouveau mot de « type 1 », soit une erreur ;
- type 4 : le mot n'a pas de caractéristique pédophile propre, par exemple : « jpg ».

Une comparaison détaillée des méthodes présentées est disponible dans l'article *http://antipaedo.lip6.fr/T24/TR/keyword-detection.pdf* et nous encourageons le lecteur désirant plus de détails à le consulter. La conclusion de ce comparatif nous informe de plusieurs points :

- il apparait que les méthodes fondées sur les cooccurrences entre les mots sont les plus efficaces ;
- les méthodes recherchant les mots directement reliés aux mots du registre pédophile fonctionnent de manière plus efficace que les méthodes plus complexes.

Conclusion

La méthode AGGR a donné des résultats intéressants. Elle sera même classée dans le comparatif comme une des deux meilleures méthodes capables de ramener des mots ayant du sens autour des agrégats proposés : « *They show that, even at the word level, AGGR and COOC significantly surpass other methods. They are able to construct lists of 30 keywords, half of which are classified as specific paedophile keywords by more than half our experts.* » **[Belbeze&al-2009-2]**.

Mais l'élément essentiel de ce comparatif reste la mise en évidence de deux points :

- le premier point est l'importance de la cooccurrence et de sa pondération relative à l'usage global du mot ;
- Le second est que des systèmes non optimisés et automatiques peuvent extraire des espaces sémantiques cohérents de fichier de log de moteurs de recherche.

L'importance des cooccurrences d'usage est directement donnée par la conclusion du challenge « les méthodes recherchant les mots directement reliés aux mots du registre pédophile fonctionnent de manière plus efficace que les méthodes plus complexe ». La pondération permet de faire baisser la présence de mots de « type 4 » dans les agrégats en les situant comme non spécifiques.

Enfin, la capacité de deux méthodes, dont AGGR, à présenter des résultats de qualité aux experts prouve le bien-fondé de ces travaux.

4.5 Conclusion

Au cours de ce travail, plusieurs points, concernant l'identification d'agrégats dans de grands réseaux de mots utilisés conjointement, ont pu être clarifiés :

- les méthodes d'agrégation doivent traiter la liaison en fonction de sa nature et de son importance relative à l'usage des mots (ce qui signifie que nous devons utiliser des graphes pondérés et dirigés) ;
- la reconnaissance de « figures » fortement connectées comme des cliques ne permet pas à elle seule de détecter des ensembles thématiques cohérents. Les comparaisons de méthodes menées pour la recherche de mots utilisés par les

pédophiles a permis de montrer que les méthodes utilisant la pondération relative sont les plus efficaces **[Belbeze&al-2009-2]**.

Ceci pourrait donc aussi signifier de manière plus générale que les méthodes dîtes « séparatistes » ne seraient pas la bonne voie. De plus, la plupart des méthodes séparatistes requièrent comme paramètre le nombre d'agrégats à créer, elles partent de l'ensemble du graphe pour rechercher un nombre de sous-ensembles. Ce qui est à l'opposé des méthodes d'agrégation locales basées, elles, sur une analyse contextuelle et locale.

La validation d'un agrégat de nœuds issu d'un processus de regroupement dans un graphe est d'autant plus difficile que sa définition est incomplète. Dans le cas qui nous occupe, même s'il existe une parenté entre l'agrégat et le champ lexical, nous ne sommes pas parvenus à définir l'agrégat précisément. Le champ lexical est défini pour un contexte qui est textuel, l'agrégat est défini dans un réseau. La taille moyenne des textes étudiés par les linguistes et celle des réseaux de mots que nous étudions sont suffisamment éloignées pour que la nature des travaux ne puisse être comparée.

Dans nos méthodes de validation, nous avons seulement cherché à mesurer la cohérence sémantique du regroupement. Pour cela, trois types de méthodes ont été utilisées :

- les méthodes par comparaison de la distribution de certaines mesures, pour des catégories entre des combinaisons de mots particulières et des combinaisons des mots issus d'agrégats ;
- les méthodes de validation basées sur le jugement d'un expert pour des regroupements de mots dans un domaine particulier (la pédophilie dans le cadre de ce travail).
- Des méthodes mixtes qui comparent les résultats du comportement de combinaisons de mots par rapport à une « baseline » construite manuellement.

Chaque type de méthode possède ses propres limites et ses qualités :

Les méthodes de comparaison comportementale d'un type de mot présentent l'avantage considérable de s'auto valider. En effet, la distance (ou différence) de comportement entre les ensembles aléatoires et ceux considérés comme sémantiquement valides est directement lisible comme le facteur de qualité de telles méthodes.

En revanche, ces méthodes sont lourdes à mettre en œuvre. En effet, fondées sur un comportement statistique, elles ne peuvent être considérées valides que si elles sont appliquées sur des échantillons de grande taille.

Les évaluations manuelles, si elles ne sont fondées que sur le simple avis d'un expert sont sans doute les moins informatives. Les observations de quelque ordre que ce soit sont finalement peu instructives. Comment évaluer, sans référentiel, un élément tel que la cohérence sémantique d'un agrégat ? La question n'a pas trouvé de réponse.

Il n'en reste pas moins, que sur le plan humain, la parole d'un expert validant la qualité sémantique d'un agrégat de mots créé par une méthode est incontournable. La nature particulièrement subjective de ce qu'est la cohérence sémantique ne peut se contenter de système de mesure automatique.

La comparaison de regroupements avec une base de qualité construite manuellement et étalonnée comme TREC-Eval est sans doute plus adaptée pour valider nos agrégats. Malheureusement, la taille de la base de TREC-Eval est encore trop faible pour servir d'outil de mesure absolu.

La véritable évaluation consisterait sans doute à récolter les niveaux de satisfaction des utilisateurs d'un système tel que celui décrit dans notre avant-propos. La mise au point (en vrai-grandeur) d'un système de création de lien social autour des agrégats permettrait alors de juger de leur cohérence sémantique.

Toutefois, nous devons valoriser la réussite de la démarche concernant la cohérence entre plusieurs méthodes de validation. Ainsi, les agrégats créés avec la méthode de Rigidification Régulée sur le réseau « 100 mots dans AOL » ont été testés avec pratiquement les mêmes résultats par deux méthodes comparatives : MCCVS (Méthode Comparative de Coefficient de Validation Sémantique) et MCCDR (Méthode de Comparaisons de Cohérence de Documents Retournés). La première méthode évalue le Coefficient de Cohérence Sémantique Comparé de ces agrégats à 0.899 et la seconde donne une valeur de 0.898 pour le Quotient de Centralité Sémantique Comparé. La proximité de ces résultats encourage à penser que l'usage de plusieurs méthodes de validation sémantique est souhaitable, leur résultat respectif pouvant alors se valider l'un l'autre.

Enfin, quelques mots sur les technologies utilisées pour ces expérimentations : nous avons utilisé des systèmes de bases de données pour stocker et étudier les graphes. Or, beaucoup de chercheurs « chargent » directement les graphes en mémoire dans des structures chainées représentant le graphe. Ceux-ci sont souvent persuadés que, par sa simplicité, ce système est le plus rapide. C'est sans doute le cas pour des opérations de boucles systématiques. Mais les bases de données ont de nombreux avantages :

- elles permettent de stocker infiniment plus de matière que la ram disponible sur l'ordinateur (dans le cas de Méga-Graphes, elles sont donc une aide précieuse) ;
- dans le cas d'études et de recherches de type « Brain Storming » sur le graphe le langage SQL permet interactivement d'ausculter et de retourner des informations très rapidement ;
- en changeant très peu de codes on peut travailler sur toute la base, un extrait ou un type de données particulier ;

- les nouvelles fonctions de type « select into » fournies par les éditeurs permettent d'extraire rapidement une partie du graphe choisi selon toutes les conditions possibles ;

- grâce à des index bien choisis il est possible d'accélérer l'extraction de données de telle sorte que les réponses soient immédiates alors qu'une boucle en mémoire consommera toujours un temps proportionnel au nombre d'éléments ;

- les moteurs de base de données modernes savent parfaitement paralléliser les requêtes de façon à utiliser les machines modernes multiprocesseurs (cela permet de profiter immédiatement de la puissance de calcul maximale de la machine sans avoir à écrire un code complexe parallélisable) ;

- il est possible de stocker l'avancée des travaux dans la base et de reprendre naturellement un travail en cours, ce qui permet la reprise sur incident simplement.

Il nous semble donc utile de ne pas écarter systématiquement les technologies de type « base de données » pour qui veut se confronter aux très grands graphes de terrain. Elles ont aussi, nous devons en convenir, des inconvénients. La simplicité apparente de l'usage de ces systèmes de gestion de bases de données cache des algorithmes très complexes. Souvent, pour des raisons commerciales, ces algorithmes sont peu détaillés. Il devient alors très difficile d'en prévoir les performances et plus encore dans des conditions d'usage intensif.

Conclusion générale et perspectives

L'objet de ce travail était l'étude de regroupements d'objets dans les réseaux de grandes tailles.

Dans une première partie, après une brève présentation des caractéristiques des grands graphes de terrain et des petits mondes, nous avons exposé un éventail des techniques de regroupement de nœuds existantes. Nous avons plus particulièrement insisté sur les techniques permettant des recouvrements d'agrégats de nœuds.

Dans une seconde partie correspondant à notre contribution, nous avons décrit plusieurs algorithmes permettant de créer ou d'enrichir des agrégats de mots issus de réseaux provenant de fichiers de log de moteurs de recherche (AOL.com, Edonkey). L'évolution de ces algorithmes a permis de proposer une méthode capable d'extraire des agrégats d'un méga-graphe de terrain, sans que celui-ci ait à être préalablement filtré. Par la reconnaissance et l'éviction de nœuds concentrateurs ou « hubs » et de nœuds présentant des liens faibles nous avons pu contenir la taille des agrégats de façon à en garantir la cohérence sémantique.

Certains résultats sont à considérer comme acquis. Cependant, les techniques d'agrégation présentées, comme celles de validation sémantique, sont le début d'une recherche à approfondir.

À propos des techniques d'agrégation dans les graphes de mots utilisés conjointement dans des requêtes nous pouvons conclure que :

- les algorithmes d'agrégation ne peuvent être génériques et se doivent de considérer et la nature des réseaux étudiés et la nature des regroupements à créer ;
- dans les grands graphes de terrain de mots, si l'on cherche à créer des ensembles sémantiquement cohérents, les techniques séparatistes et

déterministes ou encore ne travaillant pas en respectant la pondération des graphes sont à écarter. Les techniques agglomératives sont préférables ;

- il existe statistiquement un nombre maximal de mots à ne pas dépasser dans un ensemble si l'on veut conserver une bonne qualité sémantique sur l'ensemble des agrégats créés (dans les réseaux étudiés ce nombre est entre 30 et 40 mots) ;
- il est possible de créer des agrégats présentant statistiquement une bonne cohérence sémantique.

Les méthodes de rigidification constituent une base qui est ouverte aux modifications ; le choix de créer des agrégats comme des composantes bi-connexes n'est qu'une des possibilités ; ce choix pouvant être même modifié dynamiquement de telle manière qu'un graphe trop grand puisse se voir imposer comme règle d'être tri-connexe et ensuite quadri-connexe s'il est encore trop important et ainsi de suite. La souplesse de cette méthode et le support de différentes contraintes possibles de l'opérateur d'extension en fait une excellente base d'étude.

Les validations humaines ne présentent souvent que peu d'intérêt pour faire évoluer les méthodes. Elles demandent beaucoup d'efforts. Sans références comparatives mesurables, elles ne sont qu'un avis. Quand elles ont pour but la comparaison de différentes méthodes, elles ne parviennent le plus souvent qu'à un classement des dites méthodes. Pourtant, ce type de validation est indispensable. En effet, puisqu'il n'existe pas, à ce jour, de système étalonné de mesure de la cohérence sémantique des agrégats, seul le jugement d'un esprit humain, expert du domaine, peut fournir un avis de référence.

Il est parfois possible d'utiliser des systèmes de satisfaction étalonnés, comme nous l'avons fait avec TREC-Eval, et d'en tirer alors des conclusions plus précises.

Une mesure incontestable reste celle de la satisfaction d'usage. Pour l'obtenir, il est nécessaire de passer par une utilisation fonctionnelle des agrégats, comme la création de communautés dynamiques d'utilisateurs (cf. Avant-propos). La satisfaction d'usage des utilisateurs d'un tel système devient ainsi la mesure indiscutable de la qualité sémantique des agrégats.

La comparaison de comportements d'ensembles d'éléments déterminés reste à nos yeux la solution la plus efficace. Il convient bien sûr de posséder un échantillon suffisant et des éléments aux caractéristiques fortement identifiées. Si les résultats ne sont pas différentiables la mesure n'a alors pas lieu d'être. Dans le cas contraire, on peut estimer qu'elle possède une vraie valeur. Cette valeur doit cependant être relativisée en fonction : de la taille de l'échantillon, de la distance entre les populations identifiées et des critères ou comportements observés. Il est important ici de noter que les mesures de moyenne, d'écart type ou de variance ne sont pas toujours à considérer. L'observation de la distribution des valeurs retournées par la mesure des différentes populations permet de détecter des zones plus distantes. Ces zones sont alors un espace valide dans la comparaison de ces populations prédéterminées et des agrégats.

Ces algorithmes qui ont, dans une certaine mesure, fait la preuve de leur efficacité peuvent être validés par des utilisateurs intégrés aux communautés ; une communauté d'utilisateurs pouvant alors se définir comme un système interactif toujours en évolution. En proposant, pondérant ou écartant des mots, les adhérents d'une communauté peuvent la faire vivre et améliorer la pertinence des agrégats et donc la manière dont la communauté peut se proposer à de nouveaux membres. Il reste là tout un équilibre à inventer entre interactions humaines et agrégations algorithmiques.

En plus de la participation des utilisateurs, les pistes d'amélioration de la qualité des agrégats sont très nombreuses. Il est possible de créer des liens supplémentaires entre des mots par l'utilisation de dictionnaires, de sites encyclopédiques ou encore de dictionnaires ontologiques. Dans ce cas, le lien serait porteur, par sa pondération, de la signification de la précédence d'un mot dans la définition d'un autre. Des algorithmes comme celui de « Porter » **[Sparck&al-1997]** permettraient de créer des liens supplémentaires entre des mots ayant une racine commune. Ainsi, de nouveaux graphes seraient créés puis combinés entre eux pour trouver les agrégats. En effet, si la pondération de liaisons hyper-définies est un élément qui a permis d'améliorer la qualité des agrégats, elle est aussi responsable de certaines limitations. Une fois définie et pondérée elle ne peut représenter qu'un seul type de relation. D'un autre côté, il n'est pas question de revenir à des liaisons non spécifiées et donc non pondérées.

Au contraire, notre conviction est que la qualité des agrégats ne saura progresser qu'en affinant le plus possible les informations que sous-tendent les liaisons. Nous pensons qu'une des pistes permettant de mieux cerner ces relations est de combiner plusieurs graphes. Nous entendons par « combinaison » l'ensemble des opérations mathématiques possibles.

Pour expliciter notre propos, nous proposons en conclusion de ce travail et pour sourire, un petit jeu utilisant ce type de raisonnement :

Notre graphe représente un réseau social. C'est en fait une soirée où « il faut être ». Nous désirons identifier des communautés « d'affection réciproque » (sans plus les définir), présentes dans cette soirée. Nous nous fondons pour cela sur les éléments verbaux échangés. Le graphe est dirigé et pondéré selon le sens de « qui parle à qui » et le temps de parole est représenté par l'épaisseur du trait. Pour simplifier on suppose ce temps de parole équitablement partagé si les flèches vont dans les deux sens.

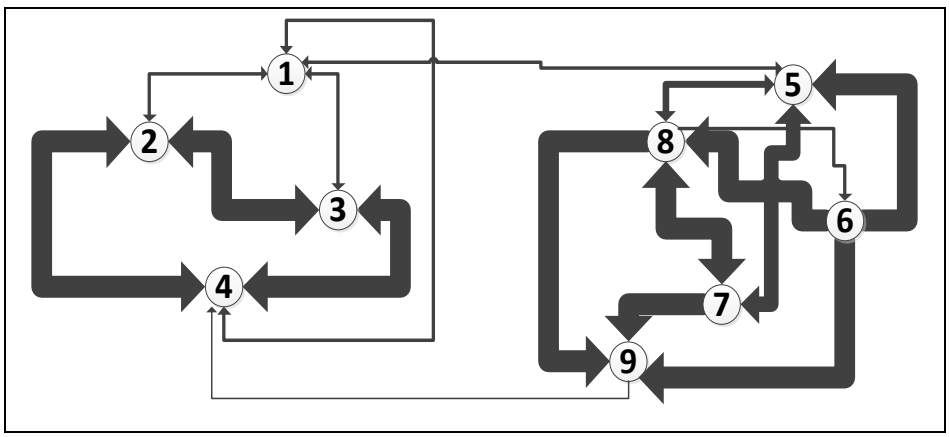


Figure C.1 : Cherchez la ou les communautés dans une soirée – « Qui parle à qui ? ».

À vous de jouer. Pouvez-vous donner un découpage de manière intuitive ?

Vous avez probablement proposé deux communautés avec {2, 3, 4} pour l'une et {5, 6, 7, 8, 9} pour l'autre. Peut-être avez-vous ajouté le « 1 » dont les échanges sont moins importants et avez-vous proposé {1, 2, 3, 4} et {5, 6, 7, 8, 9}. C'est une possibilité.

Supposons à présent que nous recevions quelques informations supplémentaires :

- 1 et 5 sont des serveurs derrière le bar.
- 2, 3 et 4 sont les parents d'une famille dont 4 est le fils.
- 7 et 8 sont les parents d'une famille dont 9 est la fille et 6 le grand-père. 8 et 6 ont beaucoup commandé à boire au serveur 5. Quant à 6, le grand-père, il est sourd. Personne ne lui parle et personne ne lui répond.
- 2, 3 sont un couple de médecins et ont échangé toute la soirée sur un cas difficile.
- Nous savons aussi que 1 et 5 (les serveurs) ont échangé sur des sujets professionnels et que les invités leur ont adressé la parole uniquement pour commander à boire.

Nous pouvons maintenant tracer un nouveau graphe. Ce graphe est celui des temps de paroles échangés dans le cadre professionnel. Nous intégrerons ici les demandes faites aux serveurs pour commander à boire comme des éléments professionnels ou fonctionnels.

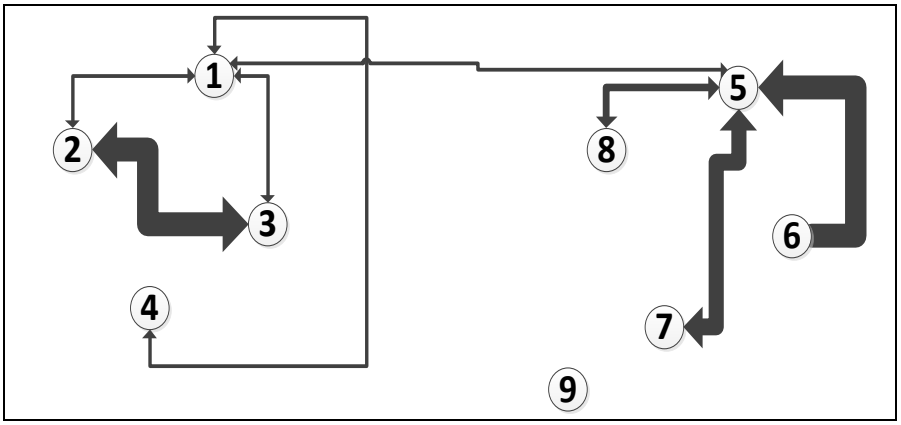


Figure C.2 : « Qui parle à qui de sujets professionnels ou fonctionnels».

Imaginons, ensuite, un graphe où le poids de la liaison serait défini comme inversement proportionnel à la différence d'âge. Les liaisons de trop faible poids (>10 ans) ne sont pas considérées. 4 et 9 ont le même âge, soit 16 ans, 3 a 37 ans, 5 a 29 ans, 7 a 40 ans, 8 a 50 ans, 2 a 55 ans, 1 a 56 ans et 6 a 85 ans (cf. figure C.3)

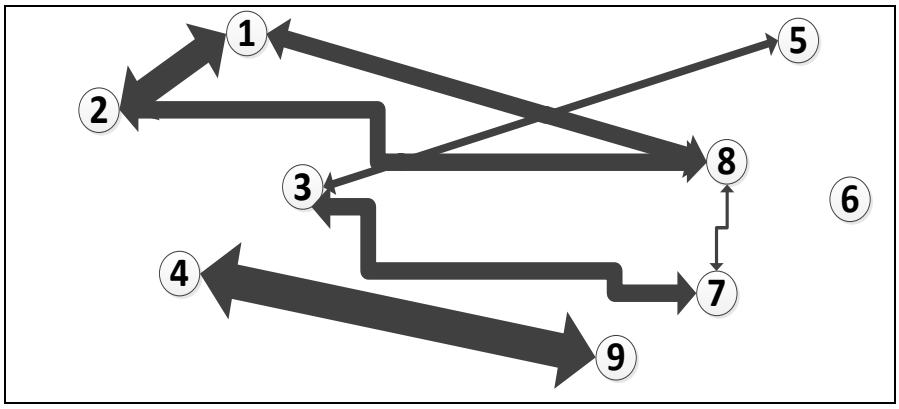


Figure C.3 : « Liaisons inversement proportionnelles à la différence d'âge ».

Dans notre quatrième graphe, les liaisons seront inversement pondérées à la distance entre les lieux d'activité professionnelle des acteurs. Pour la clarté du schéma, les distances supérieures à 100 km (jugées insignifiantes) ne sont pas représentées.

- 2 et 3 travaillent à l'hôpital de Nancy
- 8 travaille à Paris
- 7 est femme au foyer à Palaiseau
- 6 est à la retraite dans le Lot
- 1 et 5 travaillent dans un grand restaurant de Lyon
- 4 et 9 sont pensionnaires dans le même lycée international à Vérone en Italie

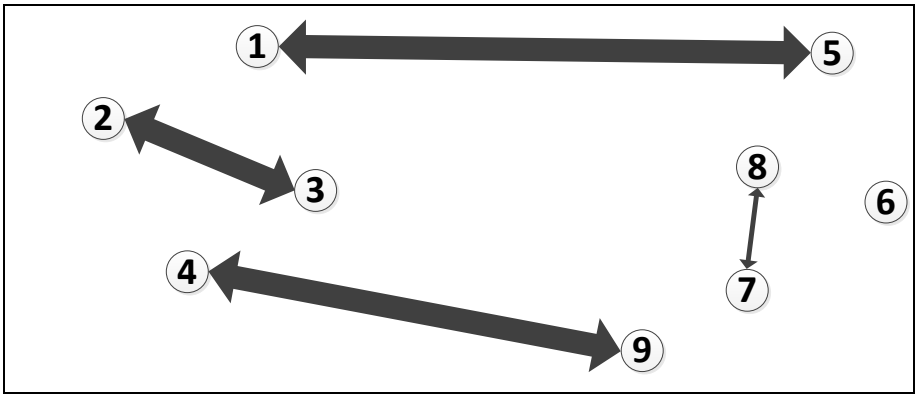


Figure C.4 : « Liaisons inversement proportionnelles à la distance des lieux de travail ».

Nous nous proposons maintenant de combiner nos graphes de telle sorte que :

- Nous supprimions les conversations professionnelles et fonctionnelles en effectuant le calcul C1-C2. Cela revient à supprimer du graphe C1 les éléments de liaisons présents dans le graphe C2 (cf. figure C5).

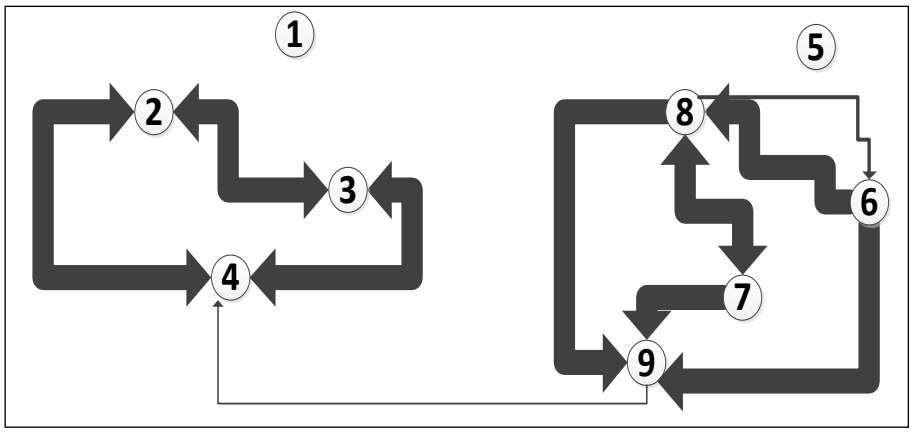


Figure C.5 : C1-C2.

- Nous pondérons les conversations du résultat de C1-C2 par le graphe C3 : soit (C1-C2)*C3 (cf. figure C.6)

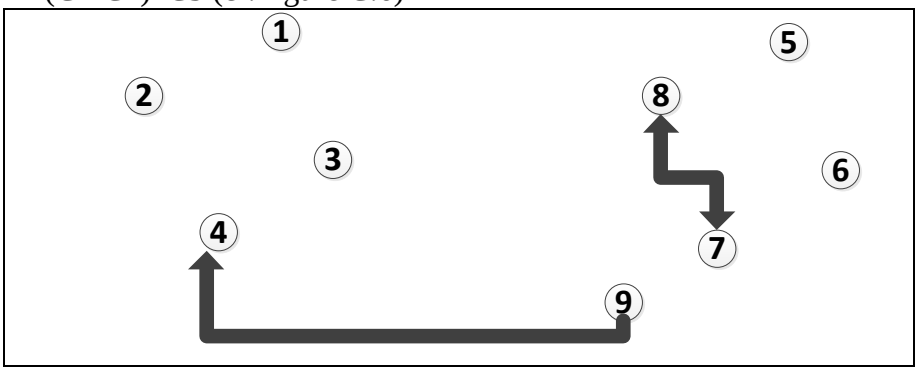


Figure C.6 : C5 *C3.

- Enfin nous pondérons de nouveau le résultat obtenu précédemment par C4, soit (C1-C2)*C3*C4 (cf. figure C.7)

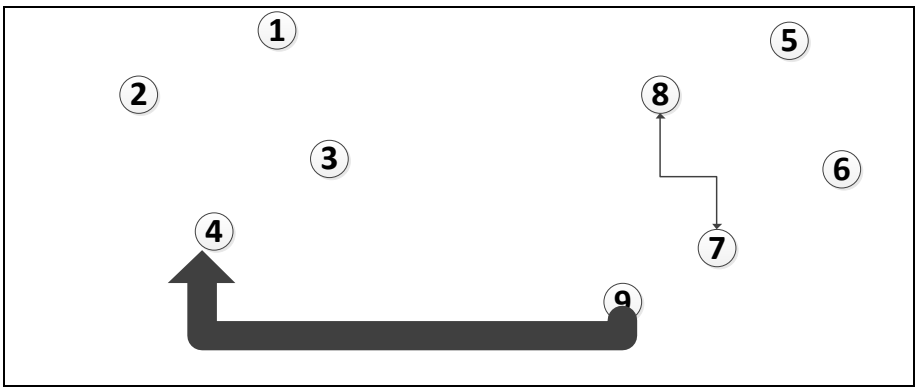


Figure C.7 : (C1-C2) *C3*C4.

Décidément, la mère de famille (6) a bien fait de trouver suspecte sa très jeune fille (9) qui, de toute la soirée, n'a parlé avec aucun des membres de sa famille et a finalement apostrophé ce jeune homme de la famille d'en face (4). Au fait, 4 se nomme Roméo et 9 se fait appeler Juliette. Alors ?

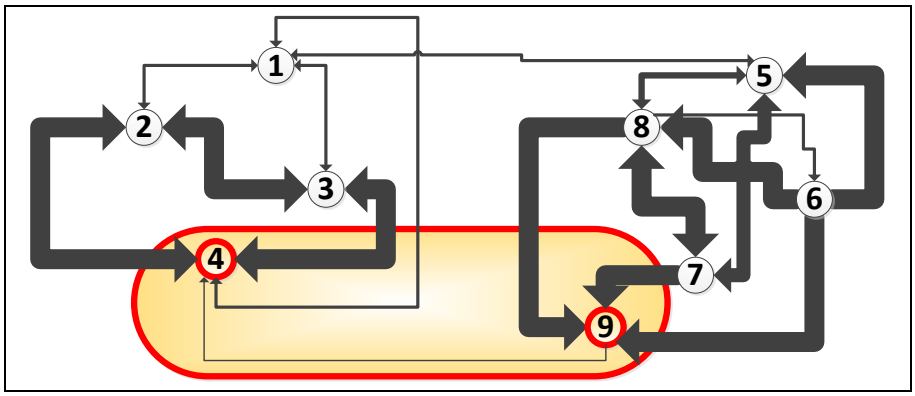


Figure C.8 : La communauté des amoureux : {4,9}

Les mots eux aussi peuvent être placés dans des graphes multiples afin de représenter différentes informations et différents types de liens.

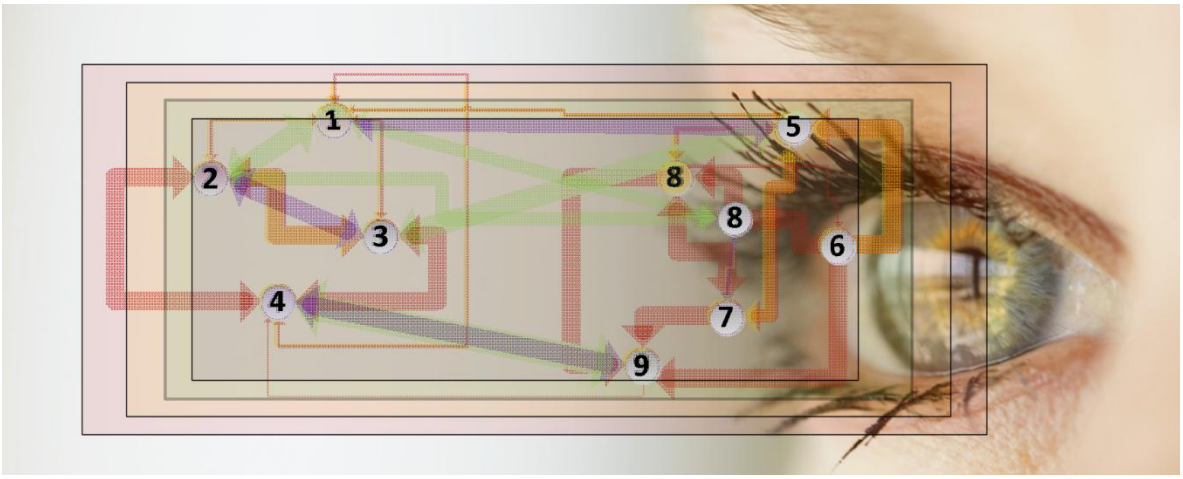


Figure C.9 : le chasseur d'agrégats combine les graphes pour mieux découvrir les agrégats

Comme un chasseur d'images qui empilerait des filtres pour trouver la bonne lumière (cf. figure C.9), le chercheur d'agrégats, lui, combinerait des graphes pour trouver le bon modèle. Les mots pourraient, par exemple, être liés par leur appartenance à une langue commune. Cette liaison serait alors pondérée par la référence à un registre de langue commun. Dans un autre graphe, les liaisons représenteraient l'existence de la paire de mots dans une même définition de dictionnaire, dans une expression, dans une ou plusieurs branches d'ontologie ou encore dans un article encyclopédique. La pondération serait alors, dans ce cas, la distance des mots entre eux et le nombre d'éléments (articles, définitions) de références partagés. On peut aussi imaginer des graphes qui figureraient la géolocalisation des utilisateurs, les liaisons étant alors pondérées par la distance moyenne entre les utilisateurs. La combinaison des différents graphes apporterait, peut-être, alors, de nouveaux éléments permettant d'améliorer la qualité sémantique des agrégats.

Et pourquoi pas des graphes qui représenteraient l'usage conjoint des mots en fonction de l'appartenance des utilisateurs à des communautés ? Le graphe serait alors pondéré positivement par le fait que certains mots de la requête seraient déjà dans un agrégat

correspondant à une communauté dont notre utilisateur ferait partie. Les communautés participeraient de ce fait à la création des agrégats et donc des communautés dynamiques.

Nous voudrions conclure ce travail sur une note plus personnelle à savoir : « comment vit-on cinq ans dans un Grand Graphe de Terrain ? »

Un Grand Graphe de Terrain est indicible. Il n'est pas résumable, fût-il construit de mots. Après plusieurs années à côtoyer le log d'AOL c'est avec la plus grande humilité que nous convenons n'en avoir qu'une infime idée.

Les ensembles constitués de millions d'objets ne peuvent être perçus que globalement, à travers des chiffres tels que les moyennes de telles ou telles valeurs ou alors « au microscope » par l'observation d'exemples concrets de quelques échantillons. La lecture des distributions des valeurs caractérisant un graphe est une vision intermédiaire. En cela, elle est pertinente mais aussi bien parcellaire.

Notre sentiment d'incompétence à percevoir la nature de ces grands graphes de terrain au bout d'un si long temps de recherche est bien réel. La frustration est d'ailleurs partagée et la recherche pour visualiser les graphes est un domaine où art, informatique et mathématiques sont fortement mis à contribution.

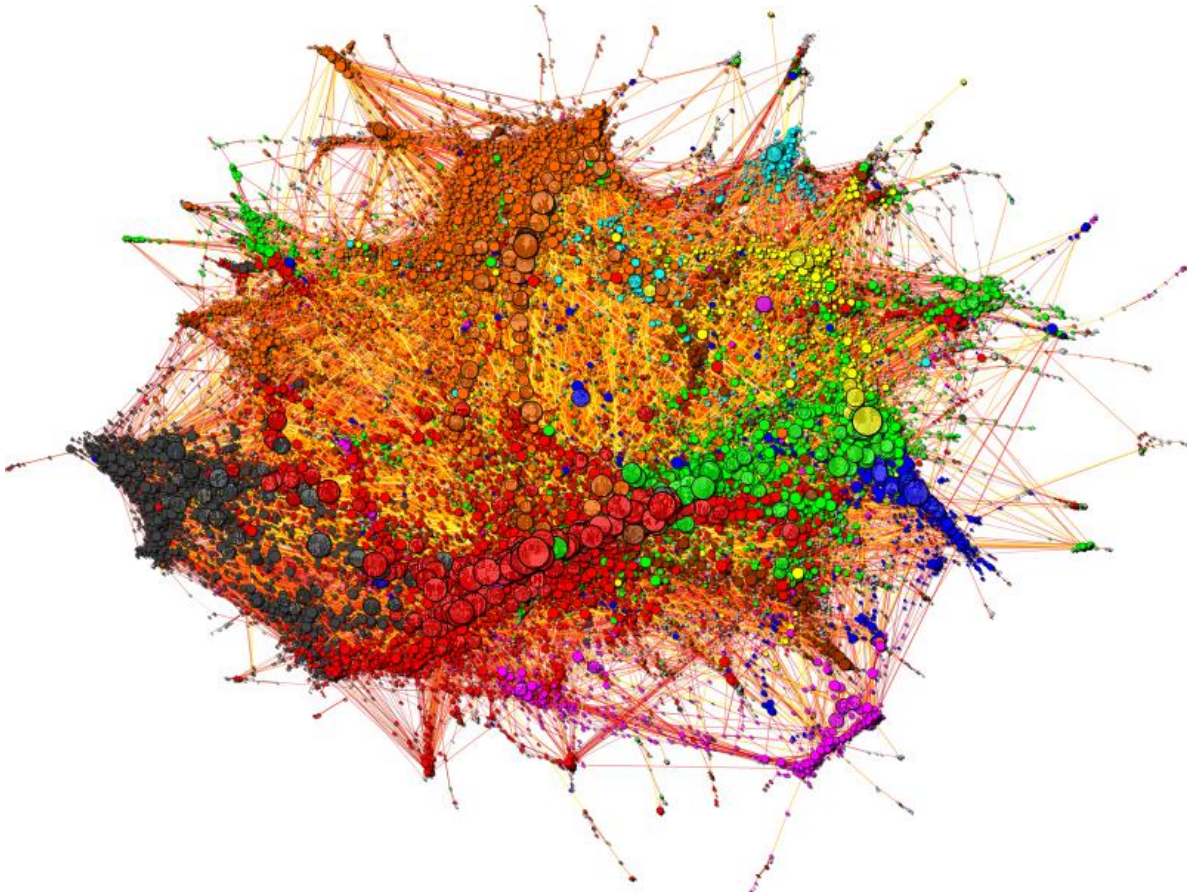


Figure C.10 : Graphe de terrain des coopérations entre artistes de la base de données last.fm.
La couleur est donnée en fonction du style de musique : rock (rouge), pop (vert) et le hip-hop (en bleu)... Auteur Tamas Nepusz, co-créateur du logiciel IGraph.

Le graphe que nous avons sans doute le plus « apprivoisé » est celui que nous avons le moins « regardé ». Expliquons-nous : en travaillant sur les données pédophiles des réseaux eDonkey-10-semaines nous n'avions pas les mots au format texte (ceci pour des raisons légales) mais seulement des identifiants numériques. Cela nous a interdit de lire le graphe comme un ensemble de mots ou de retrouver des espaces sémantiques. Cela nous a aussi empêché de traiter différemment certains mots (en pratiquant par exemple des exclusions sur des mots vides). Nous avons alors travaillé sur le graphe comme un artiste sur une matière inconnue. Nous avons cherché des points d'appui, de rupture, des nœuds de matière. Comme un sculpteur ou un potier qui sent sa terre et sait à l'avance quand elle va rompre, nous avons beaucoup appris de ce graphe, par ses réponses aux contraintes, celles que nous lui faisions subir par l'utilisation des algorithmes d'agrégation.

Bibliographie

[Aidouni&al-2008] Assia Hamzaoui, Matthieu Latapy and Clémence Magnien, *Detecting Events in the Dynamics of Ego-centered Measurements of the Internet Topology, Modeling and Optimization in Mobile,* Ad Hoc and Wireless Networks (WiOpt), 2010 Proceedings of the 8th International Symposium, pages : 491-498, 2008.

[Aidouni&al-2009] Frédéric Aidouni, Matthieu Latapy and Clémence Magnien. *Ten weeks in the life of an eDonkey server, Sixth International Workshop on Hot Topics in Peer-to-Peer Systems,* Parallel & Distributed Processing, IPDPS 2009. IEEE International Symposium on, pages 1-5, 2009.

[Albert&al-1999] R.Albert, H.Jeong, A.-L.Barabási, *Diameter of the World Wide Web*, Nature Magazine, vol.401, pages : 130-131, 1999.

[Bailey&al-1986] Bailey D. A., Cuny J. E., *Graph grammar based specification of interconnexion structure for massively parallel computation*, Lecture Notes in Computer Science 291 Grapg Grammars and their Application to Computer Science 3rd International Workshop, Warrenton, Virginia, USA, pages : 73-85, 1986.

[Barabas&ali-2000] A.-L. Barabasi, R. Albert, H. Jeong, and G. Bianconi, Power-Law *Distribution of the World Wide Web*, Science 287 2115a, 2000.

[Barabasi&al-1999] A. Barabasi et R. Albert, *Emergence of scaling in random networks*, Science 286, pages : 509-512,1999.

[Baumes&al-2005-1] Jeffrey Baumes, Mark K. Goldberg, Mukkai S. Krishnamoorthy, Malik Magdon-Ismail, Nathan Preston : *Finding communities by clustering a graph into overlapping subgraphs*, IADIS AC 2005 : Algarve, Portugal , pages : 97-104, 2005.

[Baumes&al-2005-2] Jeffrey Baumes, Mark K. Goldberg, Malik Magdon-Ismail, *Efficient identification of overlapping communities*, IEEE international conference on intelligence and security informatics, Atlanta GA, pages : 19-20, 2005.

[Belbeze&al-2007-1]* Christian Belbèze and Chantal Soulé-Dupuy, *Which contribution of the Web services in the improvement of Web searching ?* A behavioural study of the net surfers, International Conference on Enterprise Information Systems (ICEIS 2007), Funchal, Madeira, Portugal, , Vol. 4, INSTICC Press, pages : 129-137, 2007.

[Belbeze&al-2007-2]* Christian Belbèze, Max Chevalier and Chantal Soulé-Dupuy, *Web Services Based Information Access Architecture*, IADIS International Conference WWW/Internet Freiburg Allemagne, IADIS, pages : 119-127, 2008.

[Belbeze&al-2007-3]* Christian Belbèze and Chantal Soulé-Dupuy, *Apport des services Web dans l'amélioration de l'accès[1] à l'information sur le Web ?*, pages : 35-52, CORIA 2007, 2007

[Belbeze&al-2008]* Christian Belbèze and Chantal Soulé-Dupuy, *Web services based information access architecture*, IADIS International Conference WWW/Internet (ICWI 2008), Freiburg, Allemagne, pages : 119-127, 13-15 october, 2008.

[Belbeze&al-2009-1]* Christian Belbèze and Matthieu Latapy, *Detecting keywords used by paedophiles*, Poster, International Conference Advances in the Analysis of Online Paedophile Activity Paris, France, http://antipaedo.lip6.fr/Proceedings.pdf, 2-3 June, 2009.

[Belbeze&al-2009-2]* Christian Belbèze, David Chavalarias, Ludovic Denoyer, Raphaël Fournier, Jean-Loup Guillaume, Matthieu Latapy, Clemence Magnien, Guillaume Valadon, Vasja Vehovar, and Ales Ziberna, *Automatic Identification of Pedophile, Keywords keyword-detection*, Measurement and Analysis of P2P Activity Against Paedophile, Content project, 2009.
http://antipaedo.lip6.fr/T24/TR/keyword-detection.pdf

[Belbeze&al-2009-3]* Christian Belbèze, Max Chevalier, Chantal Soulé-Dupuy, *Agrégats de mots-clés validés sémantiquement Pour de nouveaux services d'accès à l'information sur Internet*, Document Numérique 1279-5127, pages : 81-105 - doi :10.3166/dn.12.1.81-105, décembre 2009.

[Belbeze&al-2009-4]* Christian Belbèze, Max Chevalier, Chantal Soulé-Dupuy, *Semantic comparaison of keywords aggregates*, IADIS International Conference Information Systems 2009 , pages 161-168, Barcelona, Spain, 2009.

[Belbeze&al-2012]* Christian Belbèze, Max Chevalier, Chantal Soulé-Dupuy, Evaluation rapide du diamètre d'un graphe, EGC, pages 17-28, Bordeaux, 2012.

[Berge-1958] Claude Berge, *Théorie des graphes et ses applications*, Collection universitaire de mathématiques, Dunod, Paris, 1958.

[Berge-1970] Claude Berge, *Graphes et hypergraphes*, Monographies Universitaires de Mathématiques, No. 37. Dunod, Paris, 1970.

[Bollobas-1998] Béla Bollobas, *Modern Graph Theory*, Graduate Text in Mathematics, Springer 1998.

* Publication effectuée par l'auteur dans le cadre de cette thèse

[Botafogo&al-1991] Rodrigo A. Botafogo and Ben Shneiderman, *Identifying Aggregates in Hypertext Structures, Proceedings of the third annual ACM conference on Hypertext*, San Antonio, TX, USA, pages 63-68, 1991.

[Boughamem&al-1997] M. Boughamem and C. Soulé-Dupuy, *Mercure at Trec 6*, In processing of the 6th International Conférence on Text retrieval, Trec6, NIST Gaithersburg (Maryland, USA), 1997.

[Breve&al-2010] Fabricio Breve, Liang Zhao and Marcos Quiles, *Uncovering Overlap Community Structure in Complex Networks Using Particle Competition*, Artificial Intelligence and Computational Intelligence, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, pages : 616-628, 2009.

[Britton&al-2008] Tom Britton, Maria Deijfen, Andreas N. Lagerås, and Mathias Lindholm - J., *Epidemics on random graphs with tunable clustering*, Appl. Probab. Volume 45, Number 3, pages : 743-756, 2008.

[Buckley&al-2004] Chris Buckley and Ellen M. Voorhees, *Retrieval evaluation with incomplete information*, In Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2004), pages 25–32, Sheffield, UK,. ACM Press, 2004.

[Cabazet&al-2010] Cazabet Rémy, Amblard Frédéric, Hanachi Chihab , *Detection of overlapping communities in dynamical social networks* , In Proceedings of the 2010 IEEE International Conference on Social Computing (Minneapolis - USA), (Best SIN Symposium Paper Award), 2010.

[Chen&al-2010] Wei Chen, Zhenming Liu, Xiaorui Sun and Yajun Wang, *A game-theoretic framework to identify overlapping communities in social networks*, Data Mining and Knowledge Discovery archive, Volume 21 Issue 2, September 2010 , Kluwer Academic Publishers Hingham, MA, USA, DOI : 10.1007/s10618-010-0186-6, pages : 224-240, 2010.

[Cucala-2009] Lionel Cucala, *Détection d'agrégats spatiaux pour données ponctuelles*, Manuscrit auteur, publié dans "41èmes Journées de Statistique, http://hal.inria.fr/docs/00/38/65/70/PDF/p16.pdf , SFdS, Bordeaux, 2009

[Dunbar-1992] Dunbar, *Neocortex size as a constraint on group size in primates*, Journal of Human Evolution 22, doi :10.1016/0047-2484(92)90081-J, 1992.

[Dunbar-1998] Dunbar, *Coevolution of neocortical size, group size and language in humans*, Behavioral and Brain Sciences 16 (4), R.I.M. (1993), pages : 681-735, 1993.

[Dunn-1973] J.C. Dunn, A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters, J. Cybernet. 3, pages 32 :57, 1973.

[Estrada&al-2010] F. J. Estrada, A. D. Jepson, and C. Chennubhotla, *Spectral Embedding and Min-Cut for Image Segmentation*, British Machine Vision Conference, London, U. K., 2004.

[Faebert-2002] Richard Faebert, *En réseau sur Internet*, page 15, Cahiers pédagogiques débattre en classe, pages 15, Février 2002.

[Faust-2010] Katherine Faust, *A puzzle concerning triads in social networks : Graph constraints and the triad census*, Social Networks, Volume 32, Issue 3, pages : 221-233, 2010.

[Fayaret-2011] Marc Farayet, *La communication – Mieux se comprendre pour mieux s'entendre*, Edition APARIS- Edifree 93200 Saint-Denis, 2011.

[Fortunato-2010] Santo Fortunato, *Community detection in graph V2*, Physics Reports 486, pages : 75-174, 2010.

[Gaignon-2006] Christophe Gaignon , *De la relation d'aide à la relation d'êtres, la réciprocité transformatrice*, Paris, L'Harmattan, page : 211, 2006.

[Gaume-2004] B. Gaume, *Balades Aléatoires dans les Petits Mondes Lexicaux*, In : I3 Information Interaction Intelligence vol.4 - n°2 CEPADUES édition (Computer sciences), 2004.

[Gregory-2010] Steve Gregory, *An Algorithm to Find Overlapping Community Structure in Networks*, PKDD 2007 Proceedings of the 11th European conference on Principles and Practice of Knowledge Discovery in Databases, Varsovie, Pologne, pages : 91 – 102, 2007.

[Hagen&al-1992] Hagen L., Kahng, A. B., *New Spectral Methods for Ratio Cut Partitioning and Clustering*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 11, No. 9, pages : 1074-1085, Sept. 1992.

[Harman-1992] Harman D., *Relevancefeeedback revisited*, In Processing of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Pages : 1-10, 1992.

[Hoffman&al-1997] C. Hoffman et R. Jaon-Arinyo, *Symbolic constraints in constructive geometric constraint solving*, Journal of Symbolic Compution, 23, pages :287-299, 1997.

[Hoffman&al-1999] C. Hoffman, A. Lomonosov et M. Sitharam, *Planning geometric constraint decomposition via optimal grap transformations*, Actes de la conference AGTIVE '99, LNCS 1779, Spinger-Verlag, pages 309-324, 1999.

[Hoffman&al-2005] C. Hoffman, R. Joan-Arinyo. *A Brief on Constraint Solving*, Computer-Aided Design and Applications, Vol. 2, No. 5, pp. 655-663, 2005.

[Jain&al-1999] A. K. Jain, M. N. Murty and P. J. Flynn, *Data clustering : a review, ACM Computing Surveys*, 31(3), pages : 264-323, 1999.

[Jermann&al-2004] Jermann C., Neveu B., et Trombettoni G., *Algorithmes pour la détection de rigidités dans les CSP géométriques*, Journal Électronique d'Intelligence Artificielle (JEDAI-JNPC'03), 2004.

[Jermann-2002] Jermann J., *Résolution de contraintes géométriques par rigidification récursive et propagation d'intervalles*, Thèse de Doctorat, Université de Nice Sophia-Antipolis, pages : 104 ; 121-160, 2002.

[Kernigha&al-1970] B. W. Kernighan and S. Lin. *An efficient heuristic procedure for partitioning graphs*, *Bell System Technical Journal*, 49(2), pages : 29-308, 1970.

[Latapy-2007] Matthieu Latapy, Habilitation à diriger les recherches, Université Pierre et Marie CURIE, 2007.

[Latouche&al-2010] Pierre Latouche, Etienne Birmelé et Christophe Ambroise, *Overlapping Stochastic Block Models with Application to the French Political Blogosphere*, Annals of Applied Statistics, 2010.

[Leskovec&al-2005] Jurij Leskovec, Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg and Christos Faloutsos Realistic, *Mathematically Tractable Graph Generation and Evolution*, Using Kronecker Multiplication, ECML/PKDD 2005, Porto, Portugal, 2005.

[Magnien-2009] Magnien C., M. Latapy, M. Habib : *Fast Computation of Empirically Tight Bounds for the Diameter of Massive Graphs*, ACM Journal of Experimental Algorithmics, 13, 10 :1.10–10 :1.9., 2009

[Mashaghi&al-2004] A. R. Mahaghi, A. Ramezapour, V. Karimipour, *Investigation of a protein complex network*, The European physical journal B, Condensed matter physics ISSN 1434-6028 , vol. 41, no1, pages : 113-121, 2004.

[Meirieu-1996] Philippe Meirieu, *Outils pour apprendre en groupe*, Sixième édition, pages 36-37, Chronique Sociale, 1996.

[MikaVrije-2005] Peter MikaVrije , *Ontologies are us : A unified model of social networks and semantics*, 4th International Semantic Web Conference, ISWC 2005, Galway, Ireland, pages : 522-536, 2005.

[Millgram&al-1969] Stanley Milgram and Travers, Jeffrey, *An Experimental Study of the Small World Problem*, Sociometry, Vol. 32, No. 4, pages : 425-443, 1969.

[Navarro&al-2010] Navarro Emmanuel, Cazabet Rémy, *Détection de communautés, étude comparative sur graphes réels*, Marami10, Toulouse, 2010.

[Newman&al-2004-1] M. E. J. Newman and M. Girvan. *Finding and evaluating community structure in networks*, Physical Review E (Statistical, Nonlinear, and Soft Matter Physics), 69(2) :026113, 2004.

[Newman&al-2004-3] M. E. J. Newman and M. Girvan, *Finding and evaluating community structure in networks*, Physical Review E, 69(2) :026113, vol. 70, 2004.

[Newman&al-2008] M. E. J. Newman and E. A. Leicht , *Community structure in directed networks*, Phys. Rev. Lett. 100, 118703, 2008.

[Newman-2004-2] M. E. J. Newman, *Fast algorithm for detecting community in networks*. Phys. Rev. E 69, 066133, 2004.

[Newman-2006] M. E. J. Newman, *Modularity and community structure in networks*, Proceedings of the National Academy of Sciences, 103(23) :8577–8582, Published online 2006 May 24. doi : 10.1073/pnas.0601602103, PMCID : PMC1482622, 2006.

[Nicosia&al-2009] V Nicosia, G Mangioni, V Carchiolo and M Malgeri2, *Extending the definition of modularity to directed graphs with overlapping communities*, Journal of Statistical Mechanics : Theory and Experiment, Vol. 2009, No. 03. (01 March 2009), P03024, 2009.

[Padrol-Sureda&al-2010] Arnau Padrol-Sureda, Guillem Perarnau-Llobet, Julian Pfeifle and Victor Muntes-Mulero, *Overlapping Community Search for social networks*, icde, 2010 IEEE 26th International Conference on Data Engineering (ICDE 2010), pages : 992-995, 2010.

[Page&al-1998] Lawrence Page, Sergey Brin, Rajeev Motwani, Terry WinogradL, *The pagerank citation ranking : Bringing order to the web*. Stanford Digital Libraries Working, Stanford Digital Libraries SIDL-WP-1999-0120, 1998.

[Palla&al-2005] Gergely Palla, Imre Derenyi, Illes Farkas, and Tamas Vicsek, *Uncovering the overlapping community structure of complex networks in nature and society*. Nature, 435, pages : 814-818, 2005.

[Palla&al-2007] G. Palla, I. J. Farkas, P. Pollner, I. Derényi and T. Vicsek, *Directed network modules,* New Journal of Physics, doi :10.1088/1367-2630/9/6/186, 9, 186, 2007.

[Pastor-Satorras&al-2001] R. Pastor-Satorras and A. Vespignani. *Epidemic spreading in scale-free networks*. Physical Review Letters 86 (14), pages : 3200–3203. arXiv :cond-mat/0010317. Bibcode 2001PhRvL..86.3200P. doi :10.1103/PhysRevLett.86.3200. PMID 11290142, 2001

[Piaget-1969] Jean Piaget, *Psychologie et pédagogie,* Paris : Denoël, 1969.

[Pons&al-2005] Pascal Pons and Matthieu Latapy, *Computing communities in large networks using random walks*, In Proceedings of the 20th International Symposium on Computer and Information Sciences (ISCIS'05), volume 3733 of Lecture Notes in Computer Science, Springer, pages : 284-293, Istanbul, Turkey, 2005.

[Pons-2007] Pascal Pons, *Détection de communautés dans les grands graphes de terrain,* **Thèse,** Université Paris 7 Denis Diderot ufr d'informatique, 2007.

[Radicchi&al-2004] Filippo Radicchi, Claudio Castellano, Federico Cecconi, Vittorio Loreto, and Domenico Parisi, *Defining and identifying communities in networks*, PNAS, 101(9) : pages : 2658–2663, 2004.

[Rao&al-2010] Francesco Rao, Sean Garrett-Roe, and Peter Hamm, Structural *Inhomogeneity of Water by Complex Network Analysis*, J. Phys. Chem. B, 114 (47), 2010.

[Rossia&al-2010] Fabrice Rossia and Nathalie Villa-Vialaneix, *Optimizing an Organized Modularity Measure for Topographic Graph Clustering : a Deterministic Annealing Approach*, Neurocomputing, Volume 73, Issues 7-9, Pages : 1142-1163, Mars 2010.

[Salton&al-1983] Gerard Salton, M.J. McGill, *Introduction to modern information retrieval*, isdn : 0070544840, McGraw-Hill, 1983.

[Schaffer-2007] Schaeffer E. S., *Graph clustering*, Computer Science Review, 1(1), pages : 27–64, 2007.

[Scott-2000] J. Scott, *Social network analysis, a handbook*. Deuxième edition, Edition Sage, 2000.

[Shang&al-2007] Shang, Chen et Zhou, *Detecting Overlapping Communities Based on Community Cores in ComplexNetworks*, CHIN. PHYS. LETT. Vol. 27, No. 5, 058901, 2010.

[Sparck&al-1997] Karen Sparck Jones and Peter Willet, *Readings in Information Retrieval*, San Francisco : Morgan Kaufmann, ISBN 1-55860-454-4, 1997.

[Vei&al-2010] Fang Wei, Weining Qian, Zhongchao Fei and Aoying Zhou, *Identifying Community Structures in Networks with Seed Expansion*, DASFAA (1) 2010, pages : 627-634, 2010.

[Villa&al-2009] Villa N. and Rossi F. , *Méthode de classification organisée pour la recherche de communautés dans les réseaux sociaux*, In Actes des 41èmes Journées de Statistique, SFdS, Journée satellite STID (conférence invitée), Bordeaux, France. Mai 2009.

[Vygotsky-1932] Lev Vygotsky : *Pensée et langage*, Gosizdat, Moscow, Chapitre 2, 1932. http://www.marxists.org/archive/vygotsky/works/words/ch02.htm

[Ward-1963] J. H. Ward. *Hierarchical grouping to optimize an objective function*, Journal of the American Statistical Association, 58(301), pages : 236-244, 1963.

[Watts-1999] Watts D., Networks, *Dynamics, and the Small-World Phenomenon*, American Journal of Sociology, Vol. 105, No. 2, September 1999.

[Watts&al-1998] Watts D.J., Strogatz S.H., *Collective dynamics of 'smallworld'networks*, Nature, 393, pages : 440-442, 1998.

[Weber&al-2001] Weber A.E., Roy E., Blais L. et coll., *HIV risk and prostitution among females in the Montréal Street Youth Cohort*, Journal canadien des maladies infectieuses, , 12 (Suppl B) **:** 67B, 2001.

[Zadeh-1965] Zadeh, L. A. *Fuzzy sets, Information and Control*, Vol. 8, pages : 338-353, 1965.

[Zhang&al-2007] S Zhang, RS Wang, XS Zhang, *Uncovering fuzzy community structure in complex networks*, Phys Rev E Stat Nonlin Soft Matter Phys. 2007 Oct ;76(4 Pt 2) :046103. Epub, 2007.

[Zhanga&al-2007] Shihua Zhanga, Rui-Sheng Wangb and Xiang-Sun Zhanga, *Identification of overlapping community structure in complex networks using fuzzy c-means clustering*, Physica A : Statistical Mechanics and its Applications, Volume 374, Issue 1, 15 January 2007, pages : 483-490, 2007.

[Zipf-1935] Zipf G. K., *The Psychobiology of Language, an Introduction to Dynamic Philology*, Boston, Houghton-Mifflin, 1935.

[Zittoum-1997] Tania Zittou : Conflit de points de vue socialement expérimenté et cognitivement résolu. Cahiers de Psychologie 33, pages : 27-30, 1997.

Index