# L'élaboration d'un réseau sémantique par le raffinement du Markov Clustering-A partir des données lexicales du roman de Saint-Exupéry, « *Le petit prince* »

Hiroyuki Akama[1], Maki Miyake[2], Jaeyoung Jung[1]

[1]Tokyo Institute of Technology, 2-12-1, O-okayama, Meguro-ku, 152-8552, Tokyo, Japon

[2]Université d'Osaka, Machikane-machi, Toyonaka-shi, 560-0043, Osaka, Japon

## Abstract

The aim of this paper is to develop a new graph clustering algorithm called Branching Markov Clustering (BMCL). This new algorithm addresses the cluster size unbalance issue that is evidenced when ordinary MCL approaches are applied to documents or corpora. Furthermore, we propose a new windowing method called Incrementally Advancing Window (IAW) that generates co-occurring word pairs that can be used as inputs to the Incremental Routing Algorithm of BMCL. Finally, the effectiveness of these techniques is tested by creating the semantic network corresponding to the story map for the very famous novel of Saint-Exupéry, "*Le petit prince*".

## Résumé

Le but de cet article est de développer un nouvel algorithme de réseau clustering nommé « Branching Markov Clustering » (BMCL). Ce nouvel algorithme aborde la question du déséquilibre de la taille des clusters mis en évidence quand les méthodes MCL ordinaires sont appliquées à des documents ou corpus. De plus, nous proposons une nouvelle méthode de fenêtrage appelée « Incrementally Advancing Window » (IAW) qui génère des couples de mots (paires de mots) qui peuvent êtres utilisés comme entrées à l'algorithme de routing incrémentiel du BMCL. Finalement, l'efficacité de cette technique est testée par la création d'un réseau sémantique correspondant au « story map » du fameux roman de Saint-Exupéry, « *Le petit prince* ».

**Mots-clés :** Markov Clustering, branching Markov Clustering, réseau sémantique.

## 1. Introduction

L'élément le plus important lors de l'élaboration d'un réseau sémantique d'un document est la sélection de « paires de mots » pertinentes, montrant une relation lexicale de type : adjacence, association ou co-occurrence. De manière à obtenir une grande justesse de sélection, on extrait seulement des cas de modification, de dépendance ou d'apposition via des logiciels d'analyse morphologique et des bases de données, cependant les informations obtenues sont souvent limitées. À l'inverse de ces approches synthétiques, les approches paradigmatiques, par le biais des « méthodes de fenêtrage », se basent seulement sur la co-présence de mots. (Burges, 1998 ; Lemaire et al., 2005).

Selon cette méthode (cf. Information Mapping Project Standford University) les mots les plus répétés d'un document sont sélectionnés à l'avance, définissant les mots-clés, ainsi qu'un cadre mobile appelé « fenêtre » d'une taille fixe limitant le nombre de mots co-occurrents (Schutze, 1997 ; Takayama et al., 1998). Cette fenêtre est supposée parcourir le document cible et s'arrêter à chaque fois qu'elle rencontre un des mots-clés pour enregistrer alors les informations sur les mots voisins apparaissant dans la fenêtre. Cependant, si l'on veut tirer

parti des avantages du réseau sémantique établi pour le document, pour le caractériser pleinement, les mots-clés doivent êtres déterminés post-hoc basé sur l'ensemble des données de co-occurrence lexicale.

Mais même si on arrive à obtenir les données d'association exhaustive, on ne peut éviter de remarquer que le réseau sémantique, lui-même, a ses propres limitations. Bien que l'on puisse visualiser l'ensemble du réseau sur un écran, tout ce que l'on peut voir sont les vertices et les liens amassés tels des nuages au loin. Ceci rend la méthode de réseau clustering telle MCL utile car elle permet de réduire le réseau sémantique en considérant un cluster de mots similaires à un « concept ».

L'algorithme de cluster Markov a été proposé par Van Dongen (2000). Il consiste en l'alternance d'un mouvement en deux étapes -*expansion* et *inflation*- afin d'atteindre la convergence d'une matrice stochastique par laquelle un réseau entier est subdivisé en « clusters durs » sans aucun chevauchement. Ordinairement, le sous-réseau de chaque cluster de Markov est de type « étoile » dont le centre est le noeud de plus haut degré et les autres noeuds ne sont reliés qu'à celui-ci. L'ordre du réseau sémantique est correctement établi par le MCL, mais les clusters de grande taille que l'on peut appeler « coeur cluster » demeurent un problème.

La distribution des mots dans un document suit la loi de Zipf et génère « un petit monde sans échelle » (Steyvers et al., 2005) dans lequel la distribution de la taille des clusters est tellement déséquilibrée qu'apparaît un cluster de Markov de taille inaccoutumée sans aucune particularité précise (« coeur cluster »).

Pour combler l'inconvénient mentionné dans l'établissement correct d'un réseau sémantique d'un document, nous proposons une nouvelle méthode de fenêtrage combinée à un nouvel algorithme de réseau clustering qui affine le MCL d'une manière différente du Recurrent MCL (RMCL) décrit par Jung et al (2006). Le Branching Markov Clustering (BMCL) pour les données de co-occurrence lexicale est obtenu en utilisant notre procédure « Incrementally Advancing Window » (IAW, « fenêtre incrémentielle avancée »).

## 2. Fenêtre Incrémentielle Avancée (Incrementally Advancing Window)

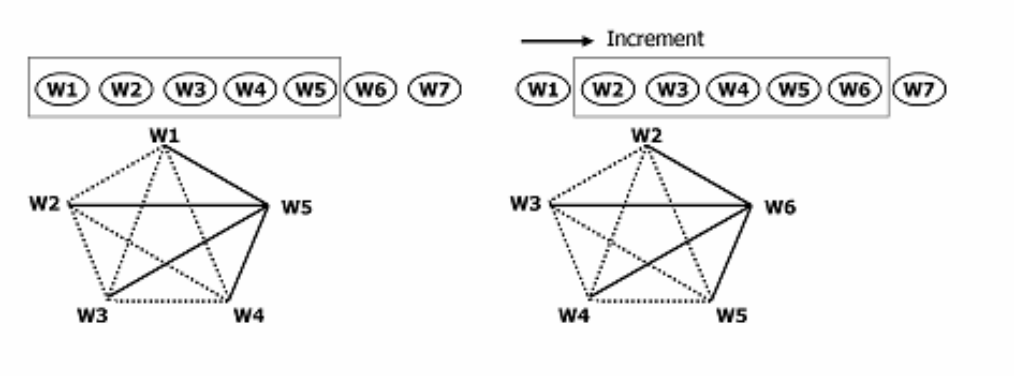


*Figure I. Mécanisme du IAW*

La méthode de fenêtrage a pour but d'obtenir exhaustivement des données de co-occurrence lexicale pour un document. Une solution pour de meilleures performances pour cette méthode est de créer une « fenêtre incrémentielle avancée » qui n'utilise aucun des termes clés centralisés. À la place, tous les mots, exceptés les « mots-bruits » ou les « mots fonctions », sont considérés comme des clés. Ce type de fenêtre ne se déplace pas à la recherche de mots préalablement sélectionnés, mais procède étape par étape à travers tout le document pour

collecter toutes les paires de mots apparaissant au moins une fois dans la fenêtre. L'état de la fenêtre est néanmoins initialisé lorsque qu'elle trouve un des « mots sections » pre-specifiés tel « chapitre ».

Pour éviter tout « double comptage » dans cette méthode de fenêtrage, on a défini la taille de la fenêtre n par la valeur du rayon (son diamètre est donc 2n+1). Dans la fenêtre de type [w(i-n), w(i-n-1),...,w(i),...,w(i+n-1), w(i+n)] où la position centrale temporaire est marquée par i, seulement les paires co-occurrentes incluant le mot le plus à droite w(i+n) doivent être prises en compte. Les exemples de paires de mots relevés ici (w(i-n) w(i+n)), (w(i-n-1)w(i+n)),...,(w(i+n-1)w(i+n)) ne seront pas comptés dans les positions suivantes de la fenêtre (cf. Figure I). L'ensemble des exemples types de paires obtenues ici, ainsi que leur fréquence rend possible la génération d'un réseau sémantique auquel nous pouvons appliquer l'algorithme de réseau de cluster Markov (MCL). Cela nous permet ainsi de collecter une série similaire ou étroitement liée de mots dans un cluster.

## 3. Branching Markov Clustering

Cependant, comme nous l'avons mentionné avant, le MCL n'est pas aussi efficace lorsqu'il est appliqué à des corpus de langage en raison de la présence anormale de cluster de grande taille. Ceci rend presque impossible de spécifier son contenu. Notre algorithme, le « Branching Markov Clustering » est une des manières de remédier au déséquilibre de la taille des clusters. Il consiste à diviser chacun des coeurs clusters de Markov en différentes branches et de redéfinir les relations adjacentes entre ces petits sous clusters.

Autrement dit, le BMCL est une manière de construire les relations adjacentes « à l'intérieur » des clusters MCL, contrairement au MCL récurrent (RMCL) qui opère entre chacun d'entre eux (Jung, 2006). Selon Jung et al (2007), il y a un type d'algorithme BMCL basé sur ce qu'ils nomment « réseau d'adjacence latente ». Ce nouveau concept signifie qu'à l'intérieur du coeur cluster (démesurément grand) une connexion virtuelle entre les vertices est possible par l'utilisation de détours via quelques vertices extérieures. Ainsi, par l'application du MCL à la matrice d'adjacence latente du coeur cluster, nous pouvons le subdiviser en une série de sous réseaux pour résoudre le problème de déséquilibre de taille de cluster. Néanmoins, il est difficile d'utiliser ce type de BMCL à un cluster de structure cohérente, dans lequel les vertices sont trop fortement connectés pour accepter le nombre approprié de liens latents. Ceci est vrai pour les gros clusters de Markov générés par les données de co-occurrence lexicale. Ces données sont obtenues par la fenêtre incrémentielle avancée (IAW) qui opère sur un texte ou un document. C'est pour cela que nous devons proposer un autre type de BMCL qui peut être appliqué à ce genre de réseau sémantique dense.

Le second type de BMCL s'opère ici par i) rapportant les séries de mots apparaissant répétitivement dans le MCL et générées par les changements de paramètres telle la taille de la fenêtre ou le seuil de fréquence des paires de mots et ii) classifiant les mots et trouvant automatiquement les configurations les plus communes qui seront considérées comme leurs racines dans un diagramme de classification taxonomique.

Nous appelons cette procédure, algorithme de routine incrémentielle « incremental Routing Algorithm ». Le renouvellement des valeurs des paramètres dans les calculs du MCL est exécuté de manière systématique pour collecter des variantes de Markov clusters. Ces variantes doivent êtres triés plus tard afin de trouver les séries de base desquelles les autres séries sont apparentées par l'ajout d'annexe. Ces « séries ancestrales » persistantes peuvent

être prises comme entités de clusters représentatifs, nous permettant à la fois de diviser et de restructurer tous les « clusters durs » qui seraient autrement finals pour l'algorithme MCL.

# 4. MCL des données IAW

## *4.1. Paramètres de réglage*

Maintenant regardons en détail le fonctionnement du MCL appliqué aux données IAW. Nous avons appliqué notre méthode au roman, « *Le petit prince* » de Saint-Exupéry (version originale française). Le choix de ce livre pour le texte cible, n'est pas seulement dû en raison qu'il est un des livres les plus traduits dans le monde entier mais aussi parce qu'il est tombé dans le domaine public à cause de la fin de la durée des droits de traduction au Japon. L'échantillon est composé de 1 312 mots porteurs de sens résultant d'une « stop liste ». Le MCL doit être lancé environ 50 fois, en changeant la taille de la fenêtre de 1 à 10 et le seuil pour la fréquence de paires de mots (thêta ; θ) de 1 à 5. (par exemple, si la valeur de thêta est 3, les paires de mots apparaissant moins de deux fois dans la fenêtre sont ignorées). La figure II montre comment le nombre de clusters MCL varie selon ces 2 paramètres.

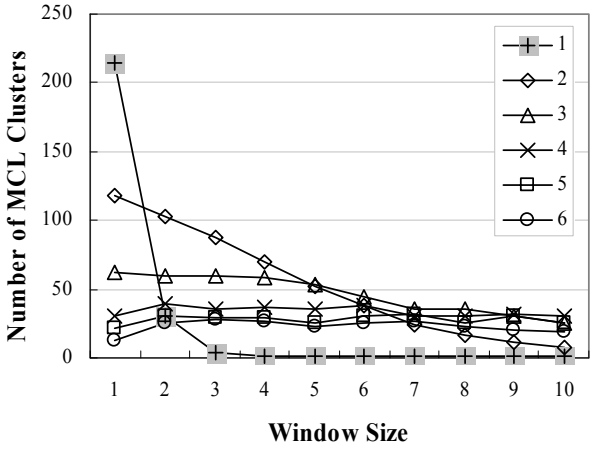


*Figure II. Taille de la fenêtre et seuil de fréquence pour les paires de mots*

Une caractéristique de cette courbe, quand la valeur de thêta est faible (de 1 à 3) est sa lente diminution avec l'augmentation de la taille de la fenêtre. Une autre caractéristique est la relative platitude de la courbe pour les valeurs de thêta supérieures à 4.

Il est en fait préférable de régler le seuil de fréquence pour les paires de mots, le plus faible possible afin de garder assez d'information. On peut noter ici, au sujet du problème de cluster de Markov volumineux que, tandis que la taille de la fenêtre augmente, le nombre de genres de mots augmente également (tous les mots sont conservés durant cet accroissement), mais nous pouvons remarquer qu'en dépit de changement complexe, d'union ou de séparation de clusters membres, qu'un seul cluster tend à absorber la partie augmentée. Particulièrement, quand la taille de la fenêtre est supérieure à 3 et thêta est égale à 1 (sans restriction), il se produit que le réseau entier ne peut plus être divisé. C'est pour cela que nous devons évaluer chacun des 60 (10x6) résultats de réseau clustering afin de sélectionner le plus approprié en tant que cible du BMCL (qui sera expliqué par la suite).

### *4.2. Modularité Q et Mesure F*

Il est bien connu qu'il existe un coefficient appelé modularité Q qui nous permet de mesurer la précision du résultat de clustering. Ce coefficient Q correspond à la différence de la distribution des liens entre un réseau d'éléments de sens et un réseau aléatoire sous les mêmes conditions de vertices. Il peut être défini (selon Newman et al.) comme $Q=\sum(e_{ii}-a_i^2)$, où $i$ est le nombre de cluster $c_i$, $e_{ii}$ est la proportion de liens internes dans le réseau entier et $a_i$ est la proportion attendue de liens de $c_i$ qui est calculée en tant que nombre total de degrés dans $c_i$ divisé par le total de tous les degrés du réseau entier (2 x nombre total de liens).

La figure III met en lumière toutes les valeurs de Q pour les 60 résultats de « MCL clusterings ». Si l'on se base sur la signification du résultat du coefficient de Modularité Q, plus la taille de la fenêtre diminue et le seuil de fréquence de paires de mots augmente et plus le résultat de clustering est supposé devenir précis.

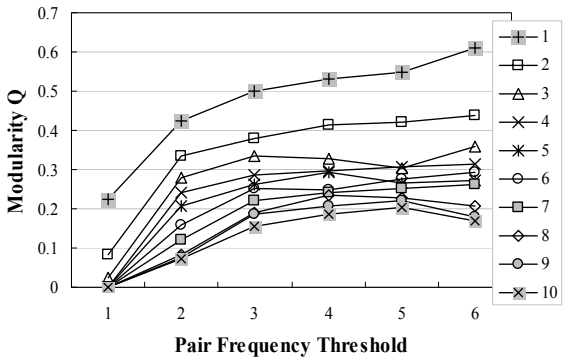


*Figure III. Modularité Q des résultats de clustering*

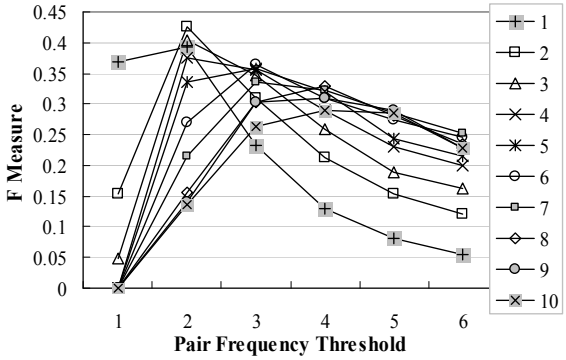


*Figure IV. Mesure F des résultats de clustering*

Cependant, lorsque le coefficient de Modularité Q enregistre la valeur maximale de 0.6089 dans les conditions d'une fenêtre de taille 1 et de seuil 6, le nombre de mots pris dans le résultat de clustering est seulement de 38, ce qui représente seulement 2,9 % du total de mots. Le taux de précision P par la Modularité Q et le taux de rappel R ont toujours un statut de compromis, il est donc préférable de calculer la mesure $F = PR\{(1-\alpha)P + \alpha R\}^{-1}$ pour optimiser la sélection des résultats les plus appropriés. La figure IV montre que le pic de la mesure F glisse de la gauche vers la droite et diminue graduellement. Ainsi nous devons adopter ici les paramètres de réglages : thêta= 2 (taille de fenêtre= 1,2,3,4), 3 (taille de fenêtre= 5,6,7) et 4 (taille de fenêtre= 8,9,10).

## 5. Algorithme du BMCL

### *5.1. Phase de détection-exploration (Detective-exploratory phase)*

Comme mentionnée ci-dessus, la figure II montre que les courbes sont similaires pour des valeurs de thêta supérieures à 3, ce qui veut dire que le nombre de clusters MCL devient plus ou moins identique au fur et à mesure de l'agrandissement de la fenêtre. Le nombre de clusters est alors approximativement constant, la correspondance entre clusters de taille de fenêtre différente peut être facilement trouvée en comparant le contenu des clusters entre eux. On peut reconnaître que chaque cluster montre une cohérence (une semi-identité) impliquant des séries de mots constants. Nous introduisons ici par exemple, l'évolution du cluster semi identique qui continue de représenter *le conflit entre enfant et adulte ayant des opinions opposées au sujet de l'astronomie*. Ces clusters sont sélectionnés en considérant les valeurs

maximales de la mesure F. Les nombres en tête expriment la taille de la fenêtre et le seuil de fréquence.

5-3 :{« astéroïde b 612 », « astronome », « démonstration »}

6-3 :{« enfant », « troisième », « grande personne », « numéro », « excuse », « tout le monde », « astéroïde b 612 », « astronome », « démonstration », « turc », « autrefois », « dédier », « froid », « demander »}

7-3 :{« enfant », « troisième », « grande personne », « numéro », « autrefois », « chiffre », « excuse », « astéroïde b 612 », « démonstration », « turc », « dédier », « froid », « européen », « habit », « demander »}

8-4 :{« enfant », « troisième », « grande personne », « excuse », « faim », « astéroïde b 612 », « astronome », « démonstration », « turc », « dédier », « froid »}

9-4 :{« enfant », « troisième », « grande personne », « chiffre », « excuse », « faim », « tout le monde », « astronome », « astéroïde b 612 », « démonstration », « turc », « dédier », « froid », « demander »}

Nous pouvons voir ici les séries de mots similaires, apparaissant répétitivement dans les multiples réseaux sémantiques et qui représentent des thèmes identiques occasionnels.

### *5.2. Phase d'optimisation-inferentielle (Optimizing-inferential phase)*

Nous avons premièrement comparé les résultats du calcul du MCL, en changeant la taille de la fenêtre, les séries de mots se chevauchant sont premièrement classifiées selon leur longueur. Parmi les séries de différentes longueurs, des relations d'inclusion ou de bifurcation peuvent être facilement trouvées. Ainsi, ce type de similarité nous permettrait d'appliquer la méthode de classification aux séries, qui peuvent être agencées sous la forme d'un arbre phylétique (Figure V) ou diagramme de Venn.

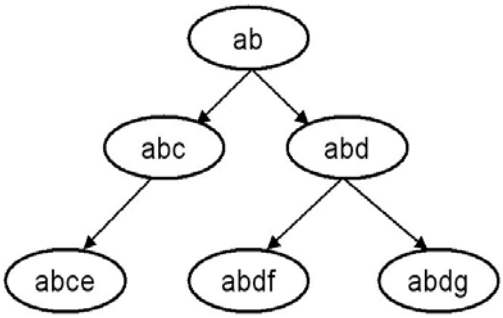


*Figure V. Arbre Phylétique*

Les séries de mots les plus courtes se chevauchant et auxquelles se rajoute d'autres mots pour créer des séries plus longues peuvent être définies comme les racines de l'arbre phylétique. Mais la longueur la plus courte (seuil) des séries ancestrales peut être changée, dans des cas ordinaires, de 2 à 4 comme point de départ de calcul. Dans cet algorithme de branchement, inspiré par la taxonomie biologique, l'origine généalogique est articulée par l'extension incrémentielle des séries (addition de mots), ceci étant un effet indirect du recalibrage de la fenêtre.

Comme la série ancestrale maintient un caractère homologue à travers plusieurs générations, l'association de mots à l'intérieur de ces séries reste stable et forte. Conséquemment, quand

les séries ancestrales sont entièrement extraites de chaque cluster de Markov, les liens connectant les mots nœud sont renouvelés pour y créer des sous-réseaux plus complexes qui sont fondés sur elles. La formule ci-après, montre les étapes de cet algorithme, qui est au coeur de notre « Branching Markov Cluster Algorithm » (BMCL).

Notation :

# signifie commentaire.

$x_m$ : Un mot

$x$ : Une série de mots

$PL(x)$ : Longueur de $x$

$(if\ x = x_1 x_2 ... x_m ... x_n, PL(x) = n)$

$WP$ : Ensemble de série de mots

Pattern($WP, l$): Sous-ensemble de $WP$ dont la longueur des membres est $l$

$(\forall x \in Pattern(WP, l) \subset WP, PL(x) = l)$

$l\min$ : Longueur minimum des séries de mots (ordinairement =2, 3 ou 4)

$MCL(i)$ : Ensemble de clusters de Markov dur généré avec la taille de fenêtre (rayon) $i$

(réglons $Max(i)$ à 10 ; Seuil de fréquence pour les paires de mots=constant=4 ou 5)

FindAncestors : Fonction pour énumérer toutes les séries ancestrales

ApplyAncestors : Fonction pour diviser chaque cluster *MCL*(i) par les séries ancestrales trouvées à l'intérieur.

RemakeAdjacency : Fonction pour calculer la matrice d'adjacence à l'intérieur de chaque cluster $MCL(i)$

$OL$ : Tous les recoupements de séries de mots générés comme suivant :

$For(i \le i, j \le Max(i, j))\{$

$OL = \bigcup_{i \ne j} (MCL(i) \bigcap MCL(j));$

$\}$

#La partie centrale du programme BMCL :

```
Foreach (l min){
   fa = FindAncestors(OL , l min);
   Foreach(MCL(i)){
     aa=ApplyAncestors(fa, MCL(i)) ;
     Foreach (cluster ⊂ MCL(i)){
          RemakeAdjacency(cluster,aa) ;
     }
   }
}
```

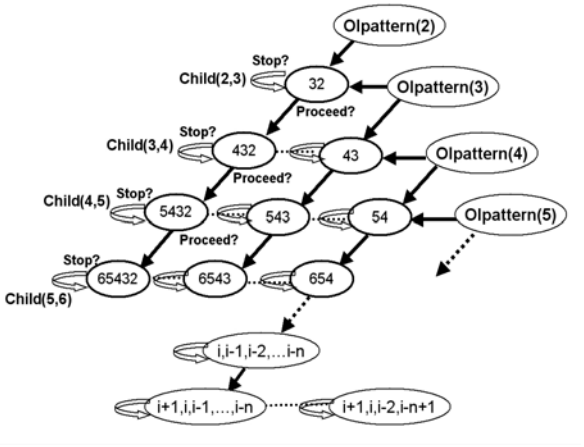


*Figure VI. Mécanisme du Programme FindAncestors*

```
FindAncestors( OL , l min ) {
olpattern(l) = Pattern(OL,l);
Foreach( l min ) {
For( l min ≤ sg ≤ Max(PL(x ∈ OL)) ) {
For( sg ≤ gen ) {
Child(sg,sg) = olpattern(sg +1);
Child (sg, gen +1) = ⋃(Child (sg, gen) ⋂ olpattern (gen));
If Child(sg, gen +1) = ∅, Break;
gen + + ;
}
sg + + ;
}

# sg = génération de départ (starting generation) ; gen = génération en cours
(on-going generation).
}
}
```

## 6. BMCL appliqué au « Petit Prince »

Le texte entier « *le petit prince* » étant pris comme exemple, considérons le résultat de calcul incrémentiel appliqué à un des cluster de Markov précédemment généré sous les conditions de taille de fenêtrage et de seuil de fréquence de mots. On peut considérer les séries ancestrales comme représentant les relations d'idées que l'auteur développe dans sa sphère d'imagination. Ces séries servent de clés pour élucider la signification idiosyncrasique des mots qu'il utilise dans son thème préféré. Ainsi, il y a un exemple notoire de mot – le verbe « *apprivoiser* » – pour lequel le terme de traduction proposé varie largement selon les traducteurs japonais (« *domesticat*e », « *accustom* », « *get acquainted* » etc). Les séries ancestrales qui incluent ce verbe nous donnent de précieuses informations qui peuvent nous aider à comprendre ce phénomène d'abondance : Ils sont composés par les mots étroitement reliés à lui, mais plus de la moitié d'entre eux sont des verbes : « chasser », « créer », « élever », « gagner », « intriguer », « jouer », « pleurer », « proposer », « revoir », « signifier », « souhaiter », « soupirer », « revenir. » Il n'est pas exagéré de dire que ces verbes pourraient contenir la plus large signification que détient ce mot énigmatique.

En ce qui concerne la sous-division du coeur cluster, nous pouvons remarquer qu'après l'application du BMCL, quelques-unes de ces séries apparaissent dans le cluster de Markov. Par exemple, le coeur cluster, sous les conditions 7-3 contient 436 mots au total, mais il est

sous-divisé par le BMCL en 119 sous-clusters qui tolèrent des chevauchements. Il est de raison de remarquer que de ces 119 sous-réseaux, on peut trouver 3 séries ancestrales de type sous-clusters « mous » incluant le verbe « apprivoiser ».

{« *apprivoiser* », « créer[1] », « élever[23] », « fusil », « gagner[4] », « gênant[2] », « intérêt[5] », « intriguer », « lien[1] », « monotone[6] », « parfait », « poliment », « pommier », « poule[36] », « proche », « proposer[7] », « renard[78] », « revoir », « signifier[5] », « souhaiter », « soupirer », « vacance »}

{« *apprivoiser* », « chasser », « créer[1] », « gagner[4] », « garçon », « jouer », « lien[1] », « pleurer[4] », « pommier », « proposer[7] », « renard[78] », « semblable[8] », « souhaiter »}

{« *apprivoiser* », « revenir »}

On peut noter que les séries les plus longues incluent celles qui sont bien plus courtes et plus particulièrement ces séries de départs les plus courtes sont formées par des paires de deux mots, ceci nous permet de comprendre plus facilement ce que ces « clusters mous » représentent en tant que composants thématiques. Les nombres ajoutés ici en tant que superscripts désignent chaque série de départ correspondant à chaque élément réduit de l'histoire.

Plus simplement, ce type de réagencement de clusters produit automatiquement une « charte » qui présente graphiquement une série de thèmes sur lesquels le texte est organisé (cf. Figure VII). Le « story map » créé par le BMCL génèrera une trame à l'intérieur de chaque thème principal et produira une meilleure représentation de l'univers du roman. De plus, si nous combinons BMCL et RMCL, ce qui est une manière de générer des liens entre les clusters de Markov, il est possible de connecter en 3D tous les niveaux de réseaux de clusters, BMCL, MCL et RMCL.

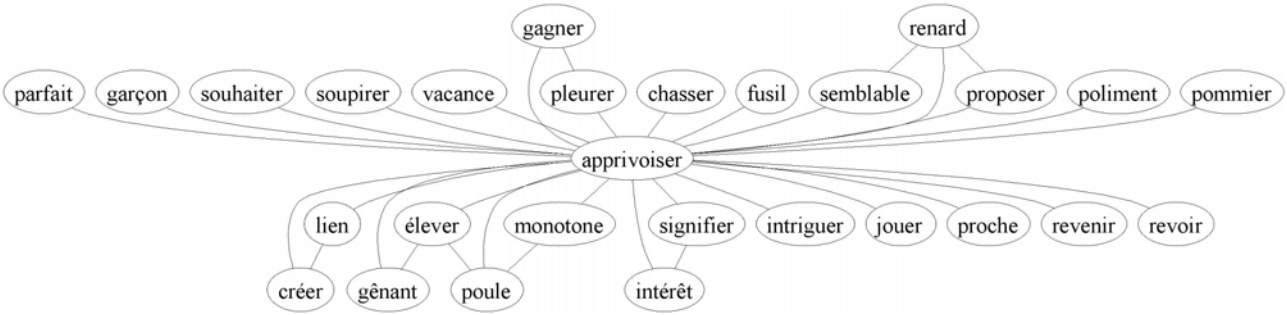


*Figure VII. Réseau Sémantique du verbe « apprivoiser »*

## 7. Conclusion

Dans ce projet, nous avons essayé de montrer que le BMCL est un outil efficace pour raffiner les résultats du MCL en collectant les entités de ces groupes « clustérisés », pris en tant que séries ancestrales pour classification. Une fois appliqué aux clusters de Markov, le BMCL divise le coeur cluster en plusieurs composants de réseaux en rétablissant dans celui-ci des liens précédents entre noeuds. Ainsi son algorithme de routine incrémentielle nous permet de résoudre le problème de cluster de grande taille et procure une précision de classification suffisante.

De plus, les résultats du BMCL appliqués à un document de données dépassent le niveau d'un simple thesaurus. Ces classes sont connectées ensemble selon leurs liens mutuels. Si la fenêtre incrémentielle avancée (IAW) est appliquée à des romans tel « *Le petit prince* », le réseau produit par le BMCL peut contribuer à l'amélioration de la compréhension de l'histoire. C'est pour cela que la mise en oeuvre de la méthode BMCL-IAW dans la « Topic Map » (ISO/IEC 13 250) en tant que « story map » pourra amener à une génération semi-automatique d'ontologies. D'autres recherches approfondies seront réalisées sur ce sujet.

## Remerciements

La réalisation de cet article a été rendue possible en grande partie grâce aux donations du 21st Century Center of Excellence Program « Framework for Systematization and Application of Large-scale Knowledge Resources ». Nous voudrions remercier ici la générosité de ce centre et également, adresser nos remerciements sincères à M. Arata Miura, ancien membre du laboratoire du Professeur Motoshi Saeki, Tokyo Tech, qui nous a aidé à programmer notre méthode de fenêtrage.

## Références

Akama H., Jung J. and Miyake M. (2007). Graph-based Linguistic Analysis on the Ideological Similarity between the Mesmerism and the Modern Stoicism. *SIG Technical Report*, Vol. 2007, No.49, 49-56.

Akama H. (2001). Computational Analysis of Similarity between the Modern Stoicism and the Mesmerism, Based on the Vector Space Model. *SIG Technical Report*, Vol.2001, No.51, 1-8.

Burgess et al. (1998). Explorations in context space: words sentences and discourse. *Discourse Process*, 25 : 211-257.

Dorow B. et al. (2005). Using Curvature and Markov Clustering in Graphs for Lexical Acquisition and Word Sense Discrimination. *MEANING-2005, 2nd Workshop organized by the MEANING Project,* February, 3rd-4th.

Enright A. J. et al. (2002). An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res.* 2002 Apr 1, 30(7): 1575-84.

Gfeller D. et al. (2005). Synonym Dictionary Improvement through Markov Clustering and Clustering Stability. *International Symposium on Applied Stochastic Models and Data Analysis (ASMDA 2005)*, 106-113.

Jung J., Miyake M. and Akama H. (2006). Markov Cluster Shortest Path Founded upon the Alibi-breaking Algorithm. *CICLing-2006, LNCS 3878,* Springer Verlag Berlin Heidelberg, 55-58.

Lemaire B. and Denhière G. (2004). Incremental Construction of an associated Network from a Corpus. In *Proceedings 26th Annual Meeting of the Cognitive Science Society*, 825-830.

Miyake M. (2007). A Network Structure of the Synoptic Gospels Employing Clustering Coefficients. *Digital Humanities 2007*, 137-139.

Miyake M. (2006). Synoptic Gospels Networked by Recurrent Markov Clustering, *Digital Humanities 2006*, 329-331.

Okamoto J. and Ishizaki S. (2001). Associative Concept Dictionary and its Comparison Electronic Concept Dictionaries. http://afnlp.org/pacling2001/pdf/okamoto.pdf.

Saint-Exupéry A. de. (1971). *Le Petit Prince*. Harcourt, Brace & World, Inc.

Schutze H. and Pederson J. O. (1997). A coocurrence-based thesaurus and two applications to information retrieval. *Information Processing & management*, 33(3): 307-318.

Schutze H. (1997). Ambiguity Resolution in Language Learning. *CSLI Publications, CSLI Lecture Notes* number 71.

Steyvers M., Tenenbaum J. (2005). The Large Scale Structure of Semantic Networks: Statistical Analyses and a Model of Semantic Growth, *Cognitive Science*, 29(1): 41-78.

Stijn van Dongen. (2000). *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht, May 2000.

Takayama Y. et al. (1999). Information Retrieval Based on Domain-Specific Word Associations. In *Proceedings of PACLING '99,* Waterloo, Ontario, Canada, June.