Classification et désarticulation de graphes de termes

Anne Berry[1], Bangaly Kaba[1], Mohamed Nadif[2], Eric SanJuan[2],
Alain Sigayret[1]

[1] LIMOS – Université Blaise Pascal – France
[2] IUT de Metz, LITA – Université de Metz – France
{berry, kaba, sigayret}@isima.fr, {nadif, eric.sanjuan}@iut.univ-metz.fr

Abstract

Algorithm CPLCL (Classification by Preferential Clustered Link) for the classification graphs of syntactic variations, introduced by Ibekwe-SanJuan (1997), has the mathematical properties of single link clustering, since it is partly founded on the concept of component related to a subgraph, while avoiding the chain effect as much as possible. In spite of these properties, this algorithm of hierarchical classification does not make it possible to avoid the formation of some classes of too great dimension. However, the visualization of subgraphs corresponding to these classes with the AiSee interface highlights obvious structural graph properties, which enable their decomposition with recent formal tools. This motivates us in developing underlying graph-theoretical aspects with algorithm CPCL presented in TermWatch and in defining a decomposition algorithm for these classes.

Résumé

Ibekwe-SanJuan (1997) introduit un algorithme de classification de graphes de relations syntaxiques, qui possède les propriétés mathématiques de la classification par lien simple (CLS), tout en limitant les effets de chaînes. Malgré cela, cet algorithme de classification hiérarchique, implanté dans le système TermWatch, ne permet pas d'éviter la formation de quelques classes de trop grande dimension. Cependant, la visualisation des sous-graphes correspondant à ces classes avec l'interface AiSee met en relief d'évidentes propriétés structurelles de graphes qui permettent leur décomposition avec des outils formels récents. Le développement des aspects de la théorie des graphes sous-jacents à l'algorithme CPCL nous a alors permis d'aboutir à un algorithme de décomposition que nous introduisons ici.

Keywords: text mining, clustering, ultrametrics, minimal separators, terminology, linguistic relations, graph decomposition.

1. Introduction

La principale motivation de cette communication est d'illustrer comment la théorie des graphes peut aider à concevoir des méthodes de classification de données textuelles non fondées sur le seul paradigme de la co-occurrence. En effet, le formalisme de cette théorie permet une représentation efficace d'indices de similarité, avec d'importantes applications en sciences sociales avec la théorie des réseaux sociaux (Social Networks). Les récents outils de visualisation de graphes tels que AiSee[1], offrent de plus des interfaces conviviales et interactives pour explorer les données ainsi représentées. Enfin, et c'est en cela que consiste la principale contribution de cette communication, de récents résultats de cette théorie sur la notion de séparateur minimal permettent de désarticuler des graphes complexes sur leur seules propriétés structurelles.

[1] Cette interface implémente les algorithmes introduits par Sander (1996) et est disponible sur http://www.aisee.com.

On applique ici ces notions à une méthode présentée par Ibekwe-SanJuan et SanJuan (2002a) et qui a été aujourd'hui totalement implantée dans le système TermWatch (Ibekwe-SanJuan et SanJuan, 2002b). Il s'agit d'une méthode de classification non supervisée de termes extraits avec INTEX (Siberztein, 1993), dont le critère d'association est exclusivement fondé sur les relations de variations linguistiques.

Les résultats présentés par Ibekwe-SanJuan et SanJuan (2002a) et validés par un spécialiste en VST (Ibekwe-SanJuan et Dubois, 2002), portaient sur un corpus de textes composé de publications scientifiques en anglais sur les procédés de panification[2], collecté pour répondre à un besoin de veille scientifique et technique (VST).

Le système utilise un algorithme de classification de graphes de relations syntaxiques initialement introduit par Ibekwe-SanJuan (1997) sous le nom de CPCL (Classification by Preferential Clustered Link). Cet algorithme possède les propriétés de la classification par lien simple (CLS), tout en limitant les effets de chaînes sur ce type de données textuelles. Malgré cela, il ne permet pas d'éviter la formation de quelques classes de trop grande dimension, or la visualisation des sous-graphes correspondant à ces classes avec l'interface AiSee met en relief des propriétés structurelles de graphes qui permettent leur désarticulation. Cette observation nous amène à introduire ici un algorithme de décomposition de ces classes dont nous analysons la complexité.

Le reste de l'article est structuré de la façon suivante :

Dans la section 2, nous revenons sur l'extraction par le logiciel INTEX des unités textuelles dont il est question ici et sur leur mise en relation par TermWatch. Nous le faisons en nous basant sur la représentation informatique de ces termes qu'utilise ce système, ce qui nous permet de donner une définition simple du graphe étudié.
La section 3 revient sur l'algorithme CPCL qui est cette fois décrit en termes de réduction de graphes. Ce point de vue nous permet de situer exactement l'algorithme CPCL vis-à-vis de la CLS classique.
La section 4 représente la principale contribution de cet article. Après un exposé des méthodes récentes de désarticulation d'un graphe, nous appliquons cette approche à la décomposition automatique de la plus grosse des classes formées par l'algorithme CPCL sur le corpus de panification. Les propriétés structurelles que cette classe partage avec les autres classes de dimension similaire nous permettent de proposer un algorithme de décomposition de ces classes.

2. Définition d'un graphe de termes

Le but de cette section est de préciser exactement le type de graphe de données textuelles que nous étudions ici. Comme il s'agit d'un graphe automatiquement extrait et réduit par le système TermWatch, il nous est nécessaire de rappeler très sommairement l'architecture de ce système.

TermWatch (Ibekwe-SanJuan et SanJuan, 2002b) est donc un système de classification non supervisée de termes extraits de textes, destiné à la Veille Scientifique et Technique (VST). Appliqué à des ensembles de textes scientifiques et techniques ce système produit une carte de thématiques présentée sous forme d'un graphe. La figure 1 donne l'architecture logicielle de ce système, qui comprend aujourd'hui quatre modules.

2.1. *Extraction des sommets du graphe : les termes*

La technique d'extraction des unités textuelles repose sur des avancées récentes de la terminologie computationnelle (Jacquemin, 2001). Dans le cas du système TermWatch, il s'agit de syn-

[2] Ce corpus a été gracieusement fourni par l'Unité de Recherche et Innovation de l'INIST.

1.	**Extraction de termes d'un corpus de textes en anglais** : appel au logiciel INTEX en mode ligne de commande en utilisant les automates introduits par Ibekwe-SanJuan et SanJuan (2002a).
2.	**Construction du graphe des termes** : Mise en relation des termes par la recherche de variations syntaxiques en utilisant les ressources linguistiques d'INTEX (Dictionnaire DELAF).
3.	**Classification ascendante hiérarchique CPCL** en quatre phases : (a) Partition de l'ensemble des variations en deux classes COMP et CLAS. (b) Extraction des composantes connexes du sous-graphe des variations dans COMP. (c) Réduction du graphe en un graphe valué de l'activité des variations dans CLAS. (d) Agglomération des composantes connexes en classes.
4.	**Génération d'interfaces AiSee et HTML** : - Edition des graphes au format GDL (Graph Description Language) pour AiSee. - Génération de liens hypertextes pour la navigation dans le graphe de termes.

Figure 1. Architecture de TermWatch

tagmes nominaux de plusieurs mots susceptibles de désigner, par leurs propriétés syntaxiques et grammaticales, un objet ou une notion du domaine (donc un terme). Ils sont spécifiques au domaine considéré et ils ont fréquemment une occurrence unique dans tout le corpus de textes.

Comme indiqué précédemment, nous reprenons ici le corpus sur les procédés de panification utilisé dans Ibekwe-SanJuan et SanJuan (2002a). De ces textes ont été extraits 3.651 termes ayant au moins un modifieur, après élimination des termes candidats les plus invraisemblables par un indexeur humain de l'INIST.

Dans le système TermWatch, ces termes candidats sont actuellement codés sous forme de couples modifieurs-centre (M, c) où c représente le *centre* d'un terme et M est une suite de *modifieurs* (essentiellement des noms et des adjectifs) associée à ce centre.

La table 1 montre trois exemples de termes extraits avec leur codage dans le système.

candidat terme	M	c
wheat dough surface stickiness	wheat dough surface	stickiness
baking property of frozen wheat dough flour	frozen wheat dough flour baking	property
greater intensity of aroma	greater aroma	intensity

Table 1. Exemples de termes extraits avec leur représentation dans TermWatch

2.2. *Extraction des arêtes du graphe : les variations syntaxiques*

Dans Ibekwe-SanJuan (1997), il est montré que la notion d'association entre termes, dans un but de classification non supervisée, pouvait être entièrement fondée sur les relations de variations linguistiques que peuvent partager ces termes. Cela représente une intéressante alternative à l'utilisation de la co-occurrence comme critère d'association, tout particulièrement lorsqu'il s'agit d'extraire une information rare.

Pour cette approche, des relations de variation ont été expérimentées ; elles sont décrites en détail dans Ibekwe-SanJuan et SanJuan (2002b)[3].

[3] Cet ensemble de variations, défini sur des opérations de surface d'insertion et de substitution d'éléments, est extrêmement réduit vis-à-vis de celles plus structurelles que peut extraire un système tel que FASTR (Jacquemin, 2001). De plus, le système TermWatch se limite actuellement à la recherche des seules relations syntaxiques, il n'intègre pas la recherche de variations sémantiques telles que la synonymie. L'objectif du système n'étant cepen-

La représentation des termes, précisée dans la sous-section précédente, nous amène à présenter une définition des relations sur lesquelles nous nous baserons pour décrire complètement le graphe qui résulte de la recherche de variations entre termes, ainsi que pour en dégager les premières propriétés structurelles. On considère classiquement quatre types de relations syntaxiques entre termes. Ces relations, utilisées depuis Ibekwe-SanJuan (1997) dans un cadre de classification non supervisée, peuvent être définies comme suit : deux termes $t_1 = (M_1, c_1)$ et $t_2 = (M_2, c_2)$ peuvent être en relation par :

- **COMP** s'ils ont même centre et que M_2 peut être obtenue, à partir de M_1, par substitution d'un unique élément ou par insertion, à une position donnée de M_1, d'une suite de modifieurs.
- **SubCen2** si leurs centres sont différents mais que M_1 et M_2 sont formées d'un même unique élément.
- **SubCen3** si leurs centres sont différents mais que M_1 et M_2 sont composées d'une même suite d'au moins deux mots.
- **Exp** si leurs centres sont différents et que la concaténation de M_1 avec c_1 est une sous-chaîne de M_2 (M_2 est de la forme AM_1c_1B).

Avant de définir le graphe de termes sous-jacent à ces relations de variations, nous allons rappeler quelques notations et définitions fondamentales.

Un **graphe non-orienté** G est un couple (V, E) où V est un ensemble fini quelconque dont les éléments sont appelés **sommets** et E est un ensemble de paires de V (parties de V ayant exactement deux éléments) dont les éléments sont appelés **arêtes**. Une arête e est dite **incidente** à un sommet v si $v \in e$.

A toute relation binaire R sur un ensemble X, on peut associer le graphe non orienté $G_R = (X, E_R)$ où E_R dénote l'ensemble des paires d'éléments de X en relation par R (*i.e.* $E_R = \{\{u, v\} : (u, v) \in R\}$). Réciproquement, à tout graphe $G = (V, E)$, on peut associer la relation binaire symétrique R_G définie sur V par $R_G = \{(v, v) : v \in V\} \cup \{(u, v), (v, u) : \{u, v\} \in E\}$.

Un graphe est une **clique** quand tous les sommets sont deux à deux reliés par des arêtes (*i.e.* $\forall u, v \in V$, $\{u, v\} \in E$). $G' = (V', E')$ est un **sous-graphe** de $G = (V, E)$ si $V' \subseteq V$ et $E' \subseteq E$. Un graphe est dit **connexe** s'il est possible de passer d'un sommet quelconque à un autre par une suite d'arêtes. Un **cycle** est un 'circuit' qui permet de partir d'un point et d'y revenir, sans rencontrer d'arête qui puisse servir de raccourci (**corde**). Un **arbre** est un graphe connexe et sans cycle ; une **forêt** est formée de plusieurs composantes connexes qui sont des arbres.

Il existe des graphes, très proches des arbres, qui forment la classe des *graphes triangulés* ; ils sont définis par l'absence de *cycle* de plus de trois sommets : cela revient à dire que les seuls cycles sont formés par des 'triangles'. La figure 2 donne un graphe triangulé.

Nous pouvons maintenant revenir à notre étude des quatre relations de variations définies ci-dessus. Celles-ci induisent dès lors, sur un ensemble T de termes, un graphe non-orienté $G_0 = G_{COMP \cup CLAS}$ avec CLAS=SubCen2 $\cup$ SubCen3 $\cup$ Exp. Ce graphe est appelé **graphe de termes** de T.

On remarque notamment que :

dant pas l'extraction de ressources terminologiques mais la classification non supervisée de données textuelles, cet ensemble de variations est considéré comme suffisant pour intéresser un veilleur qui cherche à disposer de cartes thématiques (Ibekwe-SanJuan et Dubois, 2002) générées automatiquement. L'adjonction de nouvelles relations pour affiner le processus de classification est à l'étude.

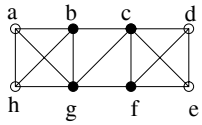

Figure 2. Un graphe triangulé : les seuls cycles sont des 'triangles'

1. Si δ est la longueur maximale d'un terme dans l'ensemble T, $|COMP \cup CLAS| < \delta.|T| << |T|^2$. La matrice d'adjacence du graphe associé à T est très creuse.

2. G_{Exp} est une forêt et $G_{Exp \cup SubCen2}$ est un graphe triangulé.

La remarque 1 justifie l'introduction en Ibekwe-SanJuan (1997) de l'algorithme CPCL spécifique à ce type de données. Nous ré-écrirons cet algorithme dans la section suivante en adoptant le point de vue de la théorie des graphes, ce qui mettra en valeur la compatibilité de cet algorithme avec la conservation de sous-graphes triangulés induits par la remarque 2 précédente. C'est en effet la persistances de ces structures triangulées dans les graphes obtenus par réductions successives, qui est la clef de l'algorithme de désarticulation des graphes volumineux qui sera présenté dans la section 4.

3. Réduction du graphe de termes

Le graphe auquel nous allons appliquer des algorithmes de désarticulation n'est pas le graphes des termes, mais un graphe réduit. Pour parfaitement le décrire il nous faut encore préciser exactement en termes de graphes l'algorithme de classification CPCL mis en œuvre par Term-Watch. On obtient ainsi un exposé simplifié de cet algorithme qui met en relief ses différences et ses points communs avec la CLS classique, ce qui nous permet d'évaluer avec exactitude sa complexité dans le pire des cas.

L'algorithme CPCL introduit en Ibekwe-SanJuan (1997) utilise, d'une part, la relation COMP pour réaliser une première réduction du graphe G_0 et d'autre part, l'union CLAS des autres relations pour dégager des classes susceptibles de représenter des thématiques du domaine au moyen d'une classification ascendante hiérarchique (CAH).

3.1. *Calcul des composantes connexes*

Nous commençons par introduire les notations nécessaires relatives aux graphes réduits.

Si θ est une relation d'équivalence sur V, on note $\theta(V)$ l'ensemble des classes d'équivalence de θ, et pour chaque v dans V, $\theta(v)$ l'élément de $\theta(V)$ contenant v. Si R est une relation binaire quelconque, on note R^* la plus petite relation d'équivalence contenant R. Un graphe $G = (V, E)$ a pour composantes connexes l'ensemble $R^*_G(V)$. Enfin, un sous-graphe $G' = (V', E')$ de G est connexe si $R^*_{G'} = V' \times V'$.

Soit $G = (V, E)$ un graphe et θ une relation d'équivalence sur G. On note $G/_\theta$ le graphe $(\theta(V), E/_\theta)$ tel que $E/_\theta = \{\{\alpha, \beta\} : (\exists u \in \alpha)(\exists v \in \beta)\ \{u, v\} \in E\}$.

Le graphe réduit a comme sommets l'ensemble des classes d'équivalence de θ et l'on trace une arête entre deux classes si et seulement si il existe au moins une arête du graphe initial qui intersecte ces deux classes.

Pour un ensemble de termes, la première étape de la réduction consiste alors à calculer le graphe : $G_0 = G_{COMP \cup CLAS}/_{COMP}$.

3.2. *Agglomération CPCL des composantes connexes*

Pour former des classes susceptibles de représenter des thématiques, on procède alors par classification hiérarchique en choisissant comme critère la somme des proportions des liens de variation de même type entre deux composantes connexes.

Plus précisément, on munit le graphe G_0 d'une valuation d_0 de ses arêtes définie pour tout $\{\alpha, \beta\}$ dans $E_{COMP \cup CLAS}/_{COMP}$ par :

$$d_0(\alpha, \beta) = \sum \left\{ \frac{|(\alpha \times \beta \cup \beta \times \alpha) \cap R|}{|R|} : R \in \{SubCen2, SubCen3, Exp\} \right\}$$

On obtient ainsi un critère de similarité qui tient compte de la nature des liens entre deux agrégats, mais la matrice de similarité induite par ce critère reste une matrice très creuse. Pour réduire G_0 en se basant sur ce critère, il est alors naturel de considérer la classification par lien simple (CLS). En effet, la CLS présente l'avantage d'induire une unique ultramétrique et s'exprime en termes de graphes réduits puisqu'elle consiste à calculer pour les différentes valeurs σ prises par d_0, les graphes G_0/R^*_σ tels que $R_\sigma = \{(\alpha, \beta) : d_0(\alpha, \beta) > \sigma\}$. Cependant, les résultats ainsi obtenus restent insatisfaisants, comme le montre le tableau 2.

La solution présentée par Ibekwe-SanJuan (1997) consiste à confondre les sommets du graphe liés par des arêtes dont la valuation est supérieure à celles des autres arêtes adjacentes. Il s'agit en fait de considérer les maximaux locaux de la fonction d_0. On calcule ainsi une suite de graphes G_i et de valuations d_i telles que pour $i \geq 0$: $G_{i+1} = G_i/_{\theta^*_i}$ avec :

$$\theta_i = \{(\alpha, \beta) : (\forall \gamma)\ (d_i(\alpha, \beta) \geq \max\{d_i(\alpha, \gamma), d_i(\gamma, \beta)\}\}$$

la valuation d_{i+1} étant définie comme dans la CLS par :

$$d_{i+1}(a, b) = \max\{d_i(\alpha, \beta) : \alpha \in a, \beta \in b\}$$

Il découle de cette caractérisation que l'ultramétrique associée à cette classification hiérarchique est unique et inférieure à l'ultramétrique de la CLS. De plus, elle peut être calculée en temps $O(|V|.\delta.|\Im(d)|)$ où $|V|$ est l'ensemble des sommets de G_0, $\delta < |V|$ est le degré maximal d'un sommet $x \in V$ et $|\Im(d)|$ est le nombre de valeurs prises par d.

3.3. *Comparaison des classifications CPCL et CLS des composantes connexes*

L'algorithme présenté dans Ibekwe-SanJuan (1997) semble éviter l'effet de chaîne dans le cas de données creuses discrètes présentant une grande amplitude de valeurs. Sur les données du corpus de panification, on obtient les résultats présentés dans la table 2. Dans cette table, i désigne l'itération, NB_CLS (resp. NB_CPCL) le nombre de classes non triviales (contenant au moins deux éléments) pour la CLS (resp. CPCL), COUV_CLS (resp. COUV_CPCL) le nombre de sommets classés et Max_CLS (resp. Max_CPCL) la taille maximale d'une classe pour la CLS (resp. CPCL).

On constate que le nombre de classes décroît avec i pour l'algorithme CPCL tandis qu'il atteint un maximum à $i = 19$ pour la CLS. C'est la classification associée à $i = 3$ qui représente le meilleur compromis entre le nombre de classes non triviales, leur taille maximale et leur couverture pour l'algorithme CPCL. C'est cette classification qui a été validée dans Ibekwe-SanJuan et Dubois (2002). Il est intéressant de noter qu'aucun niveau de la CLS n'optimise de cette manière ces trois critères.

i	NB_CLS	NB_CPCL	COUV_CLS	COUV_CPCL	Max_CLS	Max_CPCL
1	1	54	3	173	3	9
2	2	44	6	192	3	12
3	3	33	9	195	3	29
7	13	15	48	219	7	176
14	19	13	87	223	29	187
25	13	13	190	223	125	187

Table 2. Nombre, couverture et taille maximale des classes obtenues par les algorithmes CLS et CPCL lors de différentes itérations

4. Une décomposition structurelle des clusters

Quoique la méthode de classification précédente limite les effets de chaîne, certaines classes restent trop grosses pour être interprétables par l'utilisateur. Ainsi, dans le cas du corpus sur la panification, la plus grosse classe obtenue à la troisième itération avec l'algorithme CPCL contient 219 termes répartis en 29 composantes. La structure de cette classe automatiquement libellée *dough behaviour* est développée dans la figure 3 telle que AiSee permet de la visualiser avec ses liens vers les autres classes. Les éléments de cette classe *dough behaviour* sont représentés par des rectangles et sont disposés vis-à-vis des autres classes représentées par des cercles. Comme cela a été signalé par Ibekwe-SanJuan et SanJuan (2002b), on trouve dans cette classe deux grosses composantes (n° 1, 2) très proches, formées autour des termes *wheat protein* et *wheat flour dough* qui totalisent l'essentiel des liens vers l'extérieur (vers les autres classes et composantes). On remarque aussi que ce sont deux très petites composantes (n° 3, 4) formées autour des noms-centres *fibre* (*cellulose fibre, dietary fibre,...*) et *roll* (*crisp roll, hard roll, bread roll*) qui jouent un rôle de "passerelle" ou de "connecteur" dans la structure de cette classe.

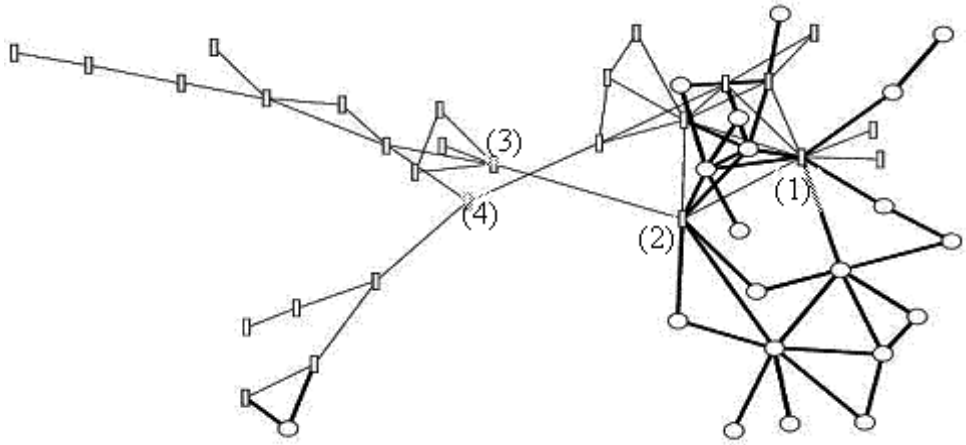

Figure 3. Structure de la classe dough behaviour

Nous allons dans cette section nous attacher à décomposer ce type de classe en nous efforçant de respecter leur cohérence structurelle.

4.1. *Les séparateurs minimaux complets*

Des travaux récents ont montré que les graphes pouvaient se décomposer de façon très cohérente en utilisant leurs articulations naturelles, que nous allons décrire, et qui sont basées exclusive-

ment sur des propriétés structurelles des graphes. Des premiers résultats très intéressants ont été obtenus avec l'examen des concepts engendrés par une base de données binaire, dans la recherche systématique de motifs pour la fouille de données (data mining) (Berry et Sigayret, 2002).

Dans les graphes simples que sont les arbres, les articulations sont les noeuds qui ne sont pas des feuilles, les feuilles étant les sommets de l'arborescence qui ont au plus une arête incidente. Le propre d'une articulation est que son retrait définit plusieurs sous-graphes connexes. De bonnes décompositions sont obtenues en recopiant une articulation dans les différents sous-graphes ainsi définis.

Lorsque le graphe considéré n'est pas un arbre, comme c'est le cas du graphe représentatif de notre corpus, on se ramène utilement à cette décomposition en cherchant des articulations qui ne sont pas formées d'un seul sommet, mais d'un groupe de sommets qui sont tous reliés entre eux. On exige toujours bien sûr, que le retrait de ce groupe de sommets définisse plusieurs sous-graphes connexes.

Pour préciser les outils que nous mettons en œuvre, sur un graphe qui n'est pas un arbre, les articulations que nous utilisons portent l'appellation de *séparateurs minimaux*. Nous pouvons en donner la définition formelle suivante : un sous-ensemble S de sommets est un *séparateur minimal* si en retirant les sommets de S ainsi que toutes les arêtes issues des sommets de S, on obtient un graphe qui n'est pas connexe (c'est-à-dire que l'on ne peut pas forcément aller d'un point à un autre en suivant un chemin dans le graphe); de plus, on demande qu'il y ait deux sommets a et b qui sont séparés par le retrait de S (a et b appartiennent à deux parties connexes ainsi définies), et tels qu'aucun sous-ensemble propre de S ne parvienne à séparer a de b par son retrait.

Un graphe, en général, peut posséder un nombre exponentiel de séparateurs minimaux, ce qui rend cet outil inimplémentable ; par contre, il est prouvé que le nombre de séparateurs minimaux complets (*i.e.* formant une clique) est faible (il y en a moins que de sommets). De plus, la décomposition décrite ci-dessus pour les arbres, et qui s'appelle en général décomposition par *séparateurs minimaux complets*, préserve les cycles (Trajan, 1985), et la décomposition du graphe ainsi que l'énumération des tous les séparateurs minimaux complets peut se faire rapidement (en temps $O(|V||E|)$, voir Berry et Bordat, 1997), à l'aide d'algorithmes tels que LEX M (Rose *et al.*, 1976) ou MCS-M (Berry *et al.*, 2002).

Un des critères qui permet de dire que les graphes triangulés sont très proches des arbres est que tous leurs séparateurs minimaux sont complets, et qu'il y en a un très petit nombre, ce qui les fait ressembler aux points d'articulation d'un arbre.

Une décomposition par séparateurs minimaux complets peut se faire soit entièrement, en recommençant sur les sous-graphes obtenus jusqu'à ce qu'il ne reste plus de séparateur complet dans aucun d'entre eux,

4.2. *Application à la classe* Dough behaviour

Examinons maintenant la classe *Dough behaviour* de notre corpus, visualisé sous forme d'un graphe. A quelques anomalies près, la structure présentée est très proche de celle d'un arbre. Il apparaît seulement quelques anomalies : d'abord quelques cycles de longueur trois (des 'triangles'), qui sont des anomalies qui ne la font pas s'éloigner de façon significative d'un arbre, comme nous l'avons expliqué ci-dessus; ensuite, il apparaît un unique cycle de longueur six, représenté en gras sur la figure 4.

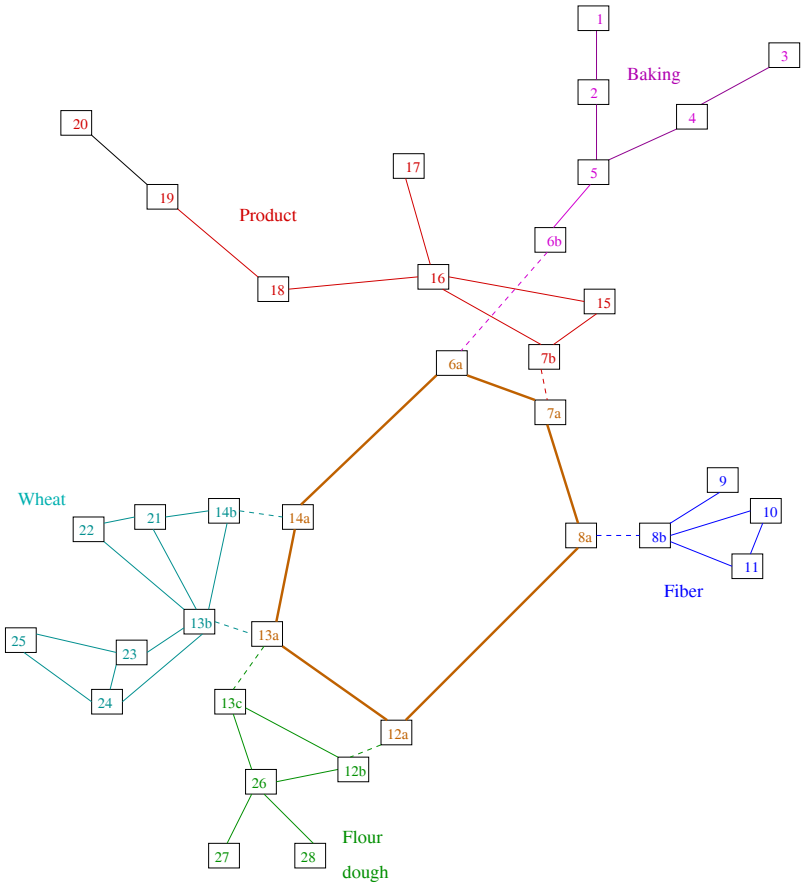

Figure 4. Une décomposition de la classe Dough behaviour *suivant des articulations équilibrantes ; les pointillés représentent les endroits où l'on a recopié des sommets d'une articulation pour décomposer*

Notre première impression est que, étant donnée la structure presque arborescente du graphe, le cycle représente réellement une anomalie; nous avons donc tenté de l'utiliser comme base pour notre décomposition. Nous avons en conséquence tout d'abord choisi des articulations qui bordaient cette anomalie, ce qui permet de l'isoler. Nous obtenons, en plus de ce cycle, deux morceaux qui sont des arbres, et trois morceaux avec quelques triangles, mais qui restent très proches d'un arbre puisqu'ils sont des graphes triangulés. Le résultat est relativement bien équilibré et les 6 morceaux obtenus pas trop gros. La figure 4 illustre ce résultat. Nous y avons renommé les sommets, pour plus de simplicité. Un premier intérêt de la notion de séparateur minimal est donc de pouvoir visualiser un sous-graphe aussi imbriqué que celui des sommets rectangulaires de la figure 3 par un ensemble d'articulations représentées en figure 4.

Cependant, pour vérifier de manière empirique si la décomposition obtenue respectait la nature syntaxique des liens représentés par ce graphe, nous avons recherché des éléments lexicaux récurrents dans les libellés des composantes appartenant à une même articulation. Ces libellés sont les termes de la composante ayant un nombre maximal de variantes (Termes avec la plus forte activité de variation). Par définition, il se peut que cet élément ne soit pas unique. Nous avons alors trouvé un élément léxical qui caractérise chacun des cinq morceaux différents du cycle. Ces éléments sont indiqués sur la figure 4. Cela signifie qu'une composante appartient à l'une de ces cinq articulations si et seulement si au moins un de ses libellés contient l'élément lexical associé à l'articulation. Il est possible de trouver d'autres décompositions de cette classe

avec cette même propriété, l'intérêt de la décomposition proposée ici est d'avoir pu être déduite des seules propriétés structurelles du graphe. D'autant plus que l'on obtient cette même propriété des éléments lexicaux pour la décomposition des quatre autres plus grosses classes qui présentent respectivement :

– Une structure d'arbre (sans cycle) ;
– Un cycle de longueur 4 plus un cycle de longueur 3 ;
– Un cycle de longueur 3 ;
– Trois cycles de longueur 3 et un cycle de longueur 4.

Il est cependant aisé de trouver des contre-exemples théoriques pour lesquels la décomposition introduite ici n'aura pas cette propriété des éléments lexicaux récurrents, d'où la nécessité de confronter cette décomposition à un grand nombre de corpus pour la valider de manière empirique. Ce travail est en cours.

4.3. *Algorithme de décomposition*

Nous proposons de façon plus générale et systématique le procédé algorithmique suivant, qui dans un premier temps décompose complètement le graphe, ce qui permet de récupérer d'une part la liste des séparateurs minimaux complets et d'autre part la liste des sous-graphes correspondants (appelés des 'atomes'); dans un deuxième temps, on examine les atomes contenant des cycles anormaux (c'est-a-dire de longueur au moins 4) et on utilise les séparateurs complets qui les bordent pour décomposer le graphe de départ. Après cette phase, si l'un des sous-graphes obtenus reste trop gros, on le redécompose en utilisant un séparateur complet qui est 'central'.

Pour décomposer, on utilise une numérotation des sommets fournie par LEX M (Rose *et al.*, 1976) ou MCS-M (Berry *et al.*, 2002), ce qui coûte $O(|V||E|)$, suivi du procédé proposé par Trajan (1985) et détaillé dans Berry et Bordat (1997), qui ne requiert pas de temps supplémentaire. Déterminer si un atome ne contient pas de cycle de longueur au moins 4 revient à vérifier si cet atome est un graphe triangulé, ce qui coûte $O(|E|)$, en utilisant un autre algorithme (Rose *et al.*, 1976). Enfin, trouver un séparateur minimal complet qui soit 'central' peut se faire en utilisant la numérotation calculée par LEX M ou MCS-M puis en prenant un numéro 'moyen'; cette approximation marche bien dans la pratique. On obtient l'algorithme suivant :

Algorithme de décomposition de graphes de termes

Donnée : Un graphe $G = (V, E)$.

Résultat : Une décomposition de G à l'aide de séparateurs minimaux complets.

begin
  Procéder à une décomposition totale en sous-graphes appelés 'atomes' ;
  $\mathcal{K}$ est l'ensemble des séparateurs minimaux complets calculé;
  pour chaque Atome A obtenu **faire**
    si A n'est pas un graphe triangulé **alors**
      Utiliser tous les séparateurs complets de $\mathcal{K}$ intersectant C pour décomposer G;
  si Un des morceaux obtenus en décomposant G est trop gros **alors**
    CHOISIR (s'il reste des séparateurs complets) un séparateur complet central pour décomposer ce morceau.
end

5. Conclusion

L'algorithme de classification CPCL avait été intuitivement conçu pour préserver les propriétés du graphe des termes. L'implantation de cet algorithme sous la forme d'une succession de

parcours des arêtes et de réductions a permis l'obtention d'un système efficace, adapté à la classification de grands graphes creux. Mais l'intérêt d'introduire la théorie des graphes dans la classification des données textuelles ne se limite pas à la formalisation des méthodes par CAH. Nous avons illustré ici sur un exemple, comment la notion de séparateur pouvait permettre de décomposer des classes trop compactes. Nous avons aussi montré comment les récents résultats de la théorie des séparateurs minimaux permettent d'implanter très efficacement ce type de décomposition. Par la suite, nous allons tenter de valider sur un grand nombre de corpus la méthode introduite ici, qui couple classification CPCL et désarticulation des classes trop compactes. Nous espérons cependant aller bien plus loin en appliquant les algorithmes de désarticulation de graphes directement au graphe de termes, ce qui permettrait de remplacer la notion de composante connexe utilisée dans l'algorithme CPCL par celle d'articulation.

Références

Berry A., Blair J.R.S. et Heggernes P. (2002). *Maximum Cardinality Search for Computing Minimal Triangulations*. L. Kucera, Lecture Notes in Computer Science, Springer Verlag.

Berry A. et Bordat J.-P. (1997). *Decomposition by clique minimal separators*. Rapport de Recherche 97213. LIRMM.

Berry A. et Sigayret A. (2002). Representing a concept lattice by a graph. In *Proceedings Discrete Maths and Data Mining Workshop, 2nd SIAM Conf. on Data Mining (SDM'02)*, Arlington (VA), submitted to Discrete Applied Mathematics.

Jacquemin C. (2001). *Spotting and discovering terms through Natural Language Processing*. MIT Press.

Ibekwe-SanJuan F. (1997). *Recherche des Tendances Thématiques dans les Publications Scientifiques. Définition d'une Méthodologie Fondée sur la Linguistique*. Thèse de doctorat, Université Stendhal, Grenoble III.

Ibekwe-SanJuan F., SanJuan É. (2002). From term variants to research topics. *International Journal on Knowledge Organization (KO), special issue on Human Language Technology*, vol. (29/3-4).

Ibekwe-SanJuan F. et Dubois C. (2002). Can Syntactic variations highlight semantic links between domain topics ?. In *Proceedings of the 6th International Conference on Terminology and Knowledge engineering (TKE 2002)*: 57-63.

Rose D. J., Tarjan R. et Lueker G. (1976). Algorithmic aspects of vertex elimination on graphs. *SIAM Journ. Comput.*, vol. (5): 146-160.

Sander G. (1996). *Visualisierungstechniken für den Compilerbau*. Dissertation, Pirrot Verlag & Druck.

SanJuan É. et Ibekwe-SanJuan F. (2002). Terminologie et classification automatique des textes. In *Actes des JADT 2002*: 677-688.

Silberztein M. (1993). *Dictionnaires électroniques et analyse automatique des textes. Le système INTEX*. Masson.

Tarjan R. (1985). Decomposition by clique separators, *Disc. Math.*, vol. (55): 221-232.