Découverte de Règles Associatives Hiérarchiques entre Termes

S. Bsiri* — **H. M. Zargayouna**** — **C. C. Latiri** — **S. Benyahia**

* *Laboratoire L.R.I*
Université de Paris-Sud, ORSAY
bsiri@firtech.lri.fr
** *Laboratoire Lamsade*
Université Paris Dauphine
zargaha3@etud.dauphine.fr
Unité de recherche en Programmation, Algorithmique et Heuristiques (URPAH)
Faculté des sciences de Tunis
Département des Sciences Informatiques
Campus Universitaire, Tunis, Tunisie
chiraz.latiri@gnet.tn
sadok.benyahia@fst.rnu.tn

RÉSUMÉ. *Dans cet article, nous proposons une nouvelle approche pour la génération de règles associatives hiérarchiques entre termes, en supposant l'existence d'une taxonomie associée au corpus de textes. L'idée est de générer, en plus des règles associatives non redondantes, d'autres règles génériques, spécifiques et/ou équivalentes, par rapport à un terme donné, en exploitant les relations sémantiques existantes entre les termes dans une taxonomie. Nous discutons également de leur intérêt pour la rechreche d'information (RI).*

ABSTRACT. *In this paper, we propose a new approach to generate hierarchical association rules between terms, while considering a taxonomy of terms associated to the corpus. The attractive idea is to discover other rules from a set of non-redundant association rules, which can be generic rules, specific rules and/or equivalent rules, using the semantic relations between terms existing in the taxonomy. We will also discuss of their interest in Information Retrieval (IR).*

MOTS-CLÉS : *règle associative non redondante, règle hiérarchique, taxonomie, expansion de requêtes.*

KEYWORDS: *Information Retrieval, non redundant association rule, hierarchical association rule, query expansion.*

1. Introduction

Dans le croisement des deux disciplines, à savoir le textmining et la recherche d'information (RI), nous avons proposé dans [LAT 03b, LAT 03a] une nouvelle approche pour la génération de règles associatives non redondantes à partir de texte. Cette approche s'intègre dans la famille d'algorithmes de génération de règles associatives, basée sur la fermeture de la connexion de Galois et considérant l'analyse formelle de concepts comme fondement mathématique de base [GAN 99]. Nous avons également montré dans [LAT 03a] l'intérêt de l'utilisation des règles associatives entre termes dans l'expansion de requêtes en RI, en terme d'augmentation des taux de rappel et de précision d'un système de recherche d'informations (SRI) optimisé.

Dans cet article, nous proposons une nouvelle approche pour la génération de règles associatives entre termes, en supposant l'existence d'une taxonomie associée au corpus de textes. L'idée est de générer, en plus des règles associatives non redondantes, d'autres règles *génériques*, *spécifiques* et/ou *équivalentes,* par rapport à un terme donné, en exploitant les relations sémantiques existantes entre les termes dans une taxonomie.

Dans le cadre de notre étude, nous considérons un SRI qui prend en charge un nombre important de collections documentaires, auxquelles est associée une taxonomie propre à leur domaine. Cette taxonomie est souvent construite manuellement par des experts et des linguistes du domaine en question et elle n'est pas forcément associée à une collection particulière du domaine. Elle explicite formellement des associations sémantiques entre les termes, jugées valides par les linguistes (e.g. relation de spécialisation/généralisation).

De ce fait, nous allons exploiter les règles associatives non redondantes dérivées par l'algorithme *Gen-RA-RE* [LAT 03b], pour générer en exploitant la taxonomie, un ensemble de règles, dites *hiérarchiques*. Le but est de vérifier la validité sémantique d'un ensemble de règles associatives non redondantes, qui sont statistiquement valides (i.e. par rapport à *minsupp* et à *minconf*) et de montrer également leur intérêt en RI.

L'article est organisé comme suit : dans la deuxième section, nous discutons du problème de redondance dans les règles associatives générées. La troisième section introduit le formalisme de base pour la dérivation de règles associatives hiérarchiques. Dans la quatrième section, nous présentons l'algorithme *Gen-RH* dédié pour la génération de telles règles. La sixième section présente l'intérêt des règles associatives hiérarchiques pour la RI. Une conclusion et quelques perspectives de l'approche sont données à la fin.

2. Règles associatives non redondantes

Habituellement, en datamining le nombre d'itemsets fréquents extraits et leur taille moyenne sont élevés dans la plupart des jeux de données transactionnelles. Le nombre de règles associatives générées varie en général de plusieurs dizaines de milliers à plu-

sieurs millions [STU 01, ZAK 00]. Cependant, le problème de la pertinence et de l'utilité des règles extraites demeure un problème majeur pour l'extraction de règles associatives [STU 01]. Il est lié à la forte proportion de règles associatives redondantes, c'est-à-dire de règles véhiculant la même information. Ce constat est notamment vrai quand nous considérons un contexte d'extraction textuel où les corpus textuels explorés sont beaucoup plus denses et nettement moins éparses que les bases de données transactionnelles. D'une manière générale, une règle associative R_2 est dite redondante par rapport à la règle R_1 si la génération de R_1 implique nécessairement celle de R_2.

Néanmoins, le problème de règles associatives redondantes a encouragé le développement d'outils de classification des règles selon leurs propriétés, de sélection de sous-ensembles de règles selon des critères définis par l'utilisateur et de visualisation de ces règles sous une forme intelligible. Dans la littérature, cette sélection s'est faite selon deux approches [STU 01] :

1) *Sélection avec perte d'information* : Elle repose sur des patrons définis par l'utilisateur [LIU 99], sur des opérateurs booléens ou encore sur du SQL-Like [SRI 97, NG 98]. Le nombre de règles associatives peut être aussi réduit en les élaguant avec une information additionnelle telle qu'une taxonomie [SRI 95, HAN 95, SIN 99, LAT 01] ou une métrique additionnelle [HUE 01] (e.g. corrélation de Pearson ou le test de χ^2). L'algorithme CIARD [LAT 01], par exemple, établit un élagage sémantique qui se base sur le lien qui figure entre deux termes dans une taxonomie associée au domaine.

2) *Sélection sans perte d'information* : Elle repose sur l'extraction d'un sous-ensemble générique de toutes les règles associatives, appelé *base*, à partir duquel le reste des règles va être dérivé. Les travaux de Bastide et al. [BAS 00] utilisent les résultats de Duquenne et Guigues [DUQ 86] et de Luxemburger [LUX 91] pour présenter deux bases, afin de générer des règles avec un nombre minimal d'attributs dans la prémisse et un nombre maximal d'attributs dans la conclusion. Dans le cadre de la même approche, Zaki et al. [ZAK 00] ont proposé un système axiomatique pour la génération de la totalité des règles à partir d'une base minimale de règles associatives.

En effet, les règles associatives minimales, telles qu'elles sont définies dans la littérature [ZAK 00, BAS 00] ne peuvent pas véhiculer fidèlement les connaissances implicites, cachées dans un corpus textuel, vu qu'elles réduisent la quantité d'information convoyée par l'ensemble de règles associatives découvert initialement. Face à ce problème, dans [LAT 03b], nous avons discuté du problème des règles associatives redondantes générées par la majorité des algorithmes utilisés an datamining [AGR 94, BRI 97, GAR 98, PAS 98, BEN 02] et leur inadéquation dans le contexte textuel. Nous avons ainsi proposé un algorithme *Gen-RA-RE* pour la génération de règles associatives non redondantes [LAT 03b, LAT 03a], en se situant par rapport à la deuxième approche, à savoir *la sélection de sous-ensembles de règles sans perte d'information*. L'algorithme proposé se base sur un treillis de générateurs minimaux et sur la fermeture de la connexion de Galois [GAN 99].

Nous avons distingué deux types de règles associatives : les règles associatives *exactes*, dont la confiance est égale à 1, et les règles associatives *approximatives*, dont la confiance est inférieure à 1 [ZAK 00]. Nous différencions les règles associatives exactes et approximatives car elles possèdent des propriétés, permettant d'identifier les règles redondantes ainsi que celles qui sont moins significatives, i.e. dont la confiance est faible, parmi toutes les règles associatives valides [LAT 03b]. L'algorithme *Gen-RA-RE* [LAT 03b] dérive les règles associatives *approximatives* et *exactes* à partir d'un treillis de générateurs. Ces règles sont non redondantes de fait que nous intégrons dans notre approche le traitement les deux types de redondance déjà définis à savoir la redondance simple et la redondance stricte [AGG 98]. L'algorithme *Gen-RA-RE* est détaillé dans [LAT 03b]. Nous présentons un exemple de règles associatives non redondantes dérivées par cet algorithme (voir exemple 1).

R	A	C	D	T	W
1	×	×		×	×
2		×	×		×
3	×	×		×	×
4	×	×	×		×
5	×	×	×	×	×
6		×	×	×	

$Générateur$	$Termset\ fréquent$	$Support$
C	C	6
W	CW	5
D	CD	4
T	CT	4
A	ACW	4
AT/TW	$ACTW$	3
DW	CDW	3
DT	CDT	2
AD	$ACDW$	2

Tableau 1. (a) *Le contexte d'extraction ;* (b) *L'ensemble de concepts réduits fréquents* (CRF) *avec leurs générateurs et leurs supports respectifs*

Exemple 1 *Considérons le contexte d'extraction définie par le tableau 1* (a) *et le treillis de générateurs G (voir la figure 1). Toutes les règles associatives approximatives non redondantes qu'il est possible de dériver sont illustrées par la figure 1. Dans cet exemple, les deux règles exactes qu'il est possible de dériver, à savoir la règle* $DW \Longrightarrow C$ *et la règle* $AT \Longrightarrow CW$ *ne sont pas générées car elles sont simplement redondantes respectivement par rapport à la règle approximative* $D \Longrightarrow CW$ *et à la règle* $A \Longrightarrow CTW$.

3. Formalisme de base pour la génération de règles associatives hiérarchiques

Considérons un ensemble de n termes $T = \{t_1, t_2, ..., t_n\}$ et un corpus textuel de m documents $D = \{d_1, d_2, ..., d_m\}$. Chaque document d_i inclus dans D contient un sous-ensemble de termes, inclus dans T appelé *termset* [1].

Soit $T_k \subseteq T$, un k-termset contenant k termes. Le *support* de T_k est le pourcentage de documents dans D, contenant tous les termes inclus dans T_k. Le termset T_k est dit

1. Terminologie que nous proposons par analogie au terme *itemset* utilisé en datamining.

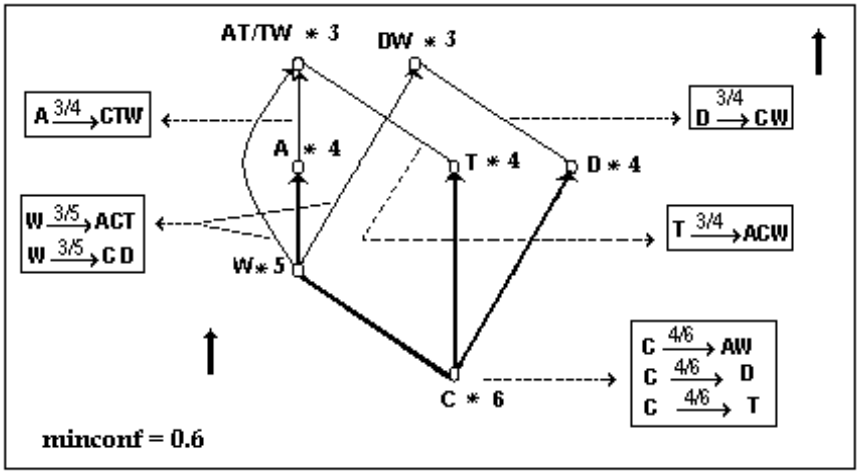

Figure 1. *Règles associatives approximatives non redondantes relatives au contexte d'extraction décrit par le tableau* 3.5.

fréquent, si son support est supérieur ou égal à seuil de support minimal exigé appelé *minsupp*.

D'une manière générale, une *règle associative* entre termes est une implication de la forme : $R : T_i \Longrightarrow T_j$, tels que T_i, $T_j \subseteq T$ sont des termsets et $T_i \cap T_j = \emptyset$. La validité de cette règle associative est appréciée selon les deux mesures utilisées en datamining, à savoir le *support* et la *confiance*.

Le *support* de la règle associative $R : T_i \Longrightarrow T_j$ exprime la fréquence avec laquelle deux termsets T_i et T_j co-occurrent ensemble. Il est mesuré par la cardinalité de l'ensemble des documents du corpus dans lesquels, les deux termsets T_i et T_j apparaissent ensemble, divisée par la cardinalité du corpus D, soit :

$$support(R) = \frac{\|D_{\{T_i \cup T_j\}}\|}{\|D\|}$$

sachant que $D_{\{T_i \cup T_j\}}$ représente l'ensemble de documents du corpus contenant en même temps les deux termsets T_i et T_j.

La *confiance* d'une règle associative $R : T_i \Longrightarrow T_j$, exprime la probabilité conditionnelle pour qu'un document contienne le termset T_j, sachant qu'il contient le termset T_i. Cette mesure indique le pourcentage de documents dans le corpus qui vérifient la conclusion d'une règle associative parmi ceux qui vérifient sa prémisse. La confiance d'une règle associative $R : T_i \Longrightarrow T_j$ se calcule comme suit :

$$confiance(R) = \frac{support(T_i \cup T_j)}{support(T_i)}$$

Nous remarquons que le support de la règle associative est divisé par le support de T_i afin d'exprimer le fait que plus ce dernier est élevé, plus nous nous attendons à ce qu'il entraîne d'autres règles associatives. Dans le cas extrême où le support de T_i est égal à 100%, c'est-à-dire tous les documents du corpus contiennent le termset T_i, la règle associative $T_i \Longrightarrow T_j$ est triviale et non significative. Ainsi, plus le termset T_i est fréquent dans la collection et plus cette règle est non intéressante. Pour filtrer les règles non valides, un seuil minimal de confiance est fixé.

Intuitivement, étant donné un corpus de textes, la règle associative $R : T_i \Longrightarrow T_j$ signifie que les documents qui contiennent le termset T_i contiennent aussi le termset T_j. Toutes les règles associatives qui vérifient les seuils prédéfinis de support *minsupp* et de confiance *minconf*, forment un ensemble de règles associatives dites *valides*. Cet ensemble représente une *base de connaissance* exprimée sous forme de termes et de liens entre les termes, formellement illustrés par les règles associatives.

Dans cet article, nous proposons un algorithme pour dériver les règles associatives hiérarchiques à partir de l'ensemble de règles associatives non redondantes minimales [LAT 03b] et d'une taxonomie. L'approche proposée tient compte d'une part des relations statistiquement valides trouvées entre les termes et qui sont formellement représentées par les règles associatives non redondantes, et d'autre part des relations sémantiques existantes dans la taxonomie.

Dans le cadre de notre étude, les règles associatives hiérarchiques sont de trois types, à savoir les règles *génériques*, les règles *spécifiques* et les règles *équivalentes*. L'intérêt de telles règles est de vérifier si les relations entre les termes qui sont sémantiquement validées par les experts et les linguistes du domaine, sont statistiquement valides selon les deux indicateurs de pertinence, à savoir la confiance et le support.

Avant de présenter l'algorithme pour la génération de règles associatives hiérarchiques, nous énonçons ainsi les définitions suivantes :

Définition 1 *Une taxonomie est une hiérarchie de termes d'une collection documentaire dans un domaine donné. Chaque noeud dans la taxonomie correspond à un terme. Elle est représentée par une structure de graphe qui illustre les relations sémantiques existantes entre les termes ainsi que le poids de ces relations, exprimé généralement par un degré d'association [SIN 99].*

Définition 2 *Une règle associative spécifique de la forme $X_S \longrightarrow Y$ est une implication entre deux termsets X_S et Y, dérivée à partir de la règle non redondante $X \longrightarrow Y$ et tel que le termset X_S est formé respectivement par tous les fils directs des termes de X, existants dans la taxonomie.*

Définition 3 *Une règle associative générique de la forme $X_G \longrightarrow Y$ est une implication entre deux termsets X_G et Y, dérivée à partir de la règle non redondante*

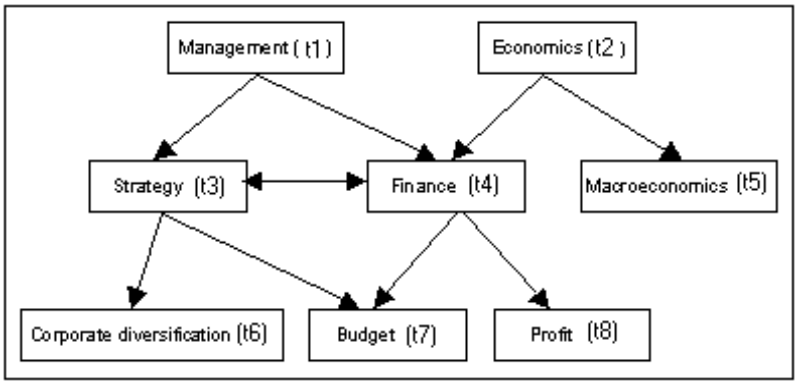

Figure 2. *Exemple de taxonomie*

$X \longrightarrow Y$ et tel que le termset X_G est formé respectivement par tous les pères directs des termes de X, existants dans la taxonomie.

Définition 4 *Une règle associative équivalente de la forme $X_E \longrightarrow Y$ est une implication entre deux termsets X_E et Y, dérivée à partir de la règle non redondante $X \longrightarrow Y$ et tel que le termset X_E est formé respectivement par tous les voisins directs des termes de X, existants dans la taxonomie.*

Exemple 2 *Un exemple de taxonomie est illustré par la figure 2. Les arcs unidirectionnels pointent du terme générique vers le terme spécifique, tandis que les arcs bidirectionnels expriment la relation entre deux noeuds voisins.*

Nous décrivons dans la section qui suit l'algorithme *Gen-RH* pour la génération de règles associatives hiérarchiques.

4. Algorithme Gen-RH

Nous supposons que nous disposons de l'ensemble de règles associatives non redondantes, dérivées à partir d'un treillis de générateurs minimaux. La génération de ces règles est détaillée dans [LAT 03b]. L'algorithme *Gen-RH* prend comme entrée l'ensemble de règles associatives non redondantes ainsi qu'une taxonomie associée au corpus et retourne comme résultat l'ensemble de règles associatives hiérarchiques. L'algorithme commence par parcourir les règles non redondantes une par une. Pour chaque terme de la prémisse, il applique la fonction *Chercher-taxo*, qui localise le terme dans la taxonomie pour trouver ses pères, ses fils et ses voisins éventuels. Une substitution du terme en cours du traitement par ses pères dans la taxonomie retourne un ensemble de règles *génériques*. Sa substitution par ses fils retourne un ensemble de règles *spécifiques* et sa substitution par ses voisins retourne un ensemble de règles

RnR :	Ensemble de **R**ègles associatives **n**on **R**edondantes.
RH :	Ensemble de **R**ègles associatives **H**iérarchiques relatives aux *RnR*.
list-père	Liste des pères d'un terme donné dans la taxonomie.
list-fils	Liste des fils d'un terme donné dans la taxonomie.
list-voisin	Liste des voisins d'un terme donné dans la taxonomie.
RG	Ensemble de règles associatives génériques.
RS	Ensemble de règles associatives spécifiques.
RE	Ensemble de règles associatives équivalentes.

Tableau 2. *Notations utilisées par l'algorithme* Gen-RH

équivalentes. Seules les règles qui satisfont les seuils *minsupp* et *minconf* sont retenues. En effet, toute la difficulté de l'approche proposée dans cet article réside dans la vérification de la validité statistique de la règle hiérarchique générée, notamment pour le calcul du support et de la confiance de la règle hiérarchique. Pour ce, l'algorithme fait appel à la fonction *Valid-règle*, que nous détaillons ultérieurement.

Nous signalons que pour chaque règle associative non redondante, il existe un ensemble de règles génériques, un ensemble de règles spécifiques et un ensemble de règles équivalentes, par rapport à la taxonomie considérée.

Le pseudo-code de l'algorithme *Gen-RH* est donné au niveau de l'algorithme 1. Les notations utilisées par cet algorithme sont illustrées par le tableau 2.

4.1. *La fonction Valid-règle*

Dans chaque itération sur la liste des pères, la liste des fils, ou la liste des voisins, la fonction *Valid-règle* permet de calculer la confiance de la nouvelle règle hiérarchique générée. Elle substitue au fur et à mesure un terme de la prémisse de la règle associative non redondante par ses pères, ses fils ou ses voisins existants dans la taxonomie. Elle calcule le support de la nouvelle règle en utilisant la fonction *Get-support* que nous détaillons dans la section qui suit. Elle commence par calculer le support du termset qui représente la prémisse de la nouvelle règle. S'il ne satisfait pas le seuil *minsupp* alors la règle n'est pas statistiquement valide. Dans le cas contraire, i.e. le support vérifie le seuil *minsupp*, le support de la règle est calculé (i.e. le support du termset formé par l'union de la prémisse et de la conclusion). Si le support de la règle ne satisfait pas le seuil *minsupp* alors la règle hiérarchique n'est pas générée et il est inutile de calculer sa confiance. Par contre, si le support de la règle satisfait le seuil *minsupp*, alors la confiance de la règle hiérarchique est calculée. Le pseudo-code de la fonction *Valid-règle* est donné au niveau de l'algorithme 2.

Algorithme *Gen-RH*
Données : *RnR*, *Taxo* : une taxonomie relative au domaine ;
Rèsultat : *RH* : ensemble de règles associatives hiérarchiques ;
Début
Pour ($i = 1; RnR \neq \emptyset; i++$) **Faire**
$RnR_i = X \longrightarrow Y$ /* X est termset tel que X = {$t_1 t_2 .. t_n$} */
Pour chaque $t_x \in X$ **Faire**
$t \longleftarrow$ *Chercher-taxo*$(t_x, Taxo)$;
Si *t.list-père* $\neq \emptyset$ **alors** /* Générer les règles génériques associées à $t_x \in$ RnR$_i$*/
Pour tout $t_j \in$ *t.list-père* **Faire**
$Confiance(r) \longleftarrow$ *Valid-règle*(X, t_x, t_j);
Si $Confiance(r) \geq$ *minconf* **alors**
$RnR_i.RG \longleftarrow RnR_i.RG \cup \{r : X\prime \longrightarrow Y$, *Support*$(r), Confiance(r)\}$
Fin Pour
Si *t.list-fils* $\neq \emptyset$ **alors** /* Générer les règles spécifiques associée à t_x de RnR$_i$*/
Pour tout $t_j \in t.list\text{-}fils$ **Faire**
$Confiance(r) \longleftarrow$ *Valid-règle*(X, t_x, P_j);
Si $Confiance(r) \geq$ *minconf* **alors**
$RnR_i.RS \longleftarrow RnR_i.RS \cup \{r : X\prime \longrightarrow Y$,*Support*$(r), Confiance(r)\}$
Fin Pour
Si *t.list-voisin* $\neq \emptyset$ **Alors** /* générer les règles équivalentes associée à t_x de RnR$_i$*/
Pour tout $t_j \in t.list\text{-}voisin$ **Faire**
$confiance(r) \longleftarrow$ *Valid-règle*(X, t_x, P_j);
Si $confiance(r) \geq$ *minconf* **alors**
$RnR_i.RE \longleftarrow RnR_i.RE \cup \{r : X\prime \longrightarrow Y$,*Support*$(r), Confiance(r)\}$
Fin Pour
Fin Pour
Fin Tant que
Fin

Algorithme 1

Fonction *Valid-règle*
Entrées : X : prémisse de la règle non redondantes RnR_i;
t_x un terme $\in X$; t_j : père, fils ou voisin éventuel de $t_x \in X$,
Resultat : Confiance de la règle hiérarchique
Début
$X\prime \longleftarrow (X \cup \{t_j\}) - t_x$
$r \longleftarrow (X\prime \longrightarrow Y)$
$Support(X\prime) \longleftarrow$ *Get-support*$(X\prime, t_j)$
Si $Support(X\prime) \geq$ *minsupp* **alors**
$Support(X\prime \cup Y) \longleftarrow$ *Get-support*$(X\prime \cup Y, X')$
Si $Support(X\prime \cup Y) \geq$ *minsupp* **alors**
$Confiance(r) \longleftarrow \dfrac{Support(X\prime \cup Y)}{Support(X')}$
retourner $confiance(r)$;
Fin si
Sinon retourner (0);
Fin si
Fin.

Algorithme 2

4.2. *La fonction Get-support*

Pour calculer le support de la nouvelle règle hiérarchique, la fonction *Get-support* commence par calculer le support du termset correspondant à sa prémisse. Pour ce faire, un parcours sur la liste des termsets fréquents, dérivés par l'algorithme *Gen-RA-RE* [LAT 03b], est effectué afin de chercher *le premier plus petit termset fréquent le contenant.* S'il n'existe aucun termset fréquent qui contient ce termset alors il en découle qu'il n'est pas fréquent et il est inutile de calculer la confiance de la nouvelle règle hiérarchique. Elle est considérée ainsi comme statistiquement invalide bien qu'elle l'est sémantiquement. Dans le cas contraire où il existe un termset fréquent contenant le termset en question, son support est stocké, et la fonction *Get-support* continue la recherche du termset fréquent contenant l'union de la prémisse et de la conclusion de la règle hiérarchique. Deux cas sont possibles :

1) Si aucun termset fréquent ne contient l'union de la prémisse et de la conclusion de la règle hiérarchique alors il est inutile de calculer sa confiance ;

2) Si le termset fréquent existe alors la confiance de la nouvelle règle est calculée. La règle sera retenue si sa confiance est supérieure ou égale à *minconf.*

Etant donné que nous nous basons sur le treillis de générateurs minimaux, à partir duquel sont dérivées les règles associatives non redondantes [LAT 03b], nous proposons de minimiser le parcours dans ce treillis afin de trouver le plus petit termset fréquent, contenant un termset $\{t_1, t_2..t_n\}$. En supposant que le nouveau terme dans la règle hiérarchique est t_1, la fonction *Get-support* ne va pas chercher le plus petit termset fréquent le contenant dans tous les noeuds du treillis. Elle commence d'abord par chercher le termset fréquent contenant t_1, et par la suite, la recherche du plus petit termset se fera parmi les pères du noeud ainsi localisé, puisque un termset fréquent est formé par les termes de tous ses noeuds fils. Toutefois, étant donné que la liste des pères d'un noeud donné n'est pas ordonnée selon leur supports respectifs, il s'avère très coûteux de comparer les supports de tous les termsets fréquents relatifs aux noeuds pères et contenant le termset de départ. Nous proposons ainsi une approche par niveaux, i.e. parcourir la liste des pères par génération (les pères puis les pères des pères, etc.). Ainsi, la comparaison des supports se fait entre les noeuds d'une même génération seulement. Afin de pouvoir respecter cet ordre de génération, nous proposons d'utiliser une file de noeuds destinée à contenir la liste des pères, en respectant l'ordre des générations. Nous procédons comme suit : au niveau du noeud de départ, nous enfilons tous ses pères ainsi qu'un indicateur de fin de la première génération, appelé *FING* (fin d'une génération). A chaque noeud rencontré dont le termset correspondant ne contient pas le termset recherché, nous enfilons tous les pères jusqu'à rencontrer l'indicateur *FING*. Dans ce cas nous enfilons un autre indicateur *FING* : indicateur de fin de la deuxième génération. Lorsque nous rencontrons un *FING*, nous vérifions si un ou plusieurs noeuds contenant le termset recherché ont été rencontrés. Si c'est le cas, celui ayant le support est le plus grand sera retenu.

Le pseudo-code de la fonction *Get-support* est illustré par l'algorithme 3. L'exemple 3 explique le fonctionnement de la fonction *Get-support* et de la file des noeuds pères

par rapport au contexte d'extraction donné par le tableau 1 et le treillis de générateurs correspondant (voir figure 1).

Fonction *Get-support*
Entrées : X' : prémisse de la règle hiérarchique ;
le treme t_j de X' traité
Résultat : le support de la règle hiérarchique.
Début
$Nd \longleftarrow$premier noeud du treillis ;
Tant que $t_j \not\subseteq fermeture(Nd)$ **Faire**
$Nd \longleftarrow Noeud\text{-}suivant$;
Fin tant que
Si $Nd \neq \emptyset$ **alors**
enfiler tous les pères de Nd dans f; /* f est une file d'attente*/
enfiler *FING* dans f;
Fin si
$Nd \longleftarrow défiler(f)$; $liste\text{-}trouvés \longleftarrow \emptyset$;
Tant que $(Nd \neq FING)$ ou $(liste\text{-}trouvés = \emptyset)$ **Faire**
Si $t_i \not\subseteq Nd$ **alors**
enfiler tous les pères de Nd dans f;
Sinon
ajouter Nd dans $liste\text{-}trouvés$;
Fin si
Tant que (f est non vide)
$Nd \longleftarrow défiler(f)$
Si $(Nd = FING)$ **alors** enfiler $FING$ dans f;
Fin tant que
Fin tant que
Si $liste\text{-}trouvés \neq \emptyset$ **alors**
retourner le support maximal $\in liste\text{-}trouvés$;
Sinon retourner 0 **;**
Fin si
Fin

Algorithme 3

Exemple 3 *Nous avons repris le treillis de générateurs (voir figure 1), à partir duquel nous pouvons dériver les termsets fréquents et les règles associatives non redondantes relatives au contexte d'extraction illustré par le tableau 1. Afin de mieux comprendre le principe de la fonction Get-support, nous avons numéroté chaque noeud au niveau du treillis de générateurs (voir figure 3). Considérons le noeud C, à partir duquel nous allons calculer le support du termset ACT. Le plus petit termset fréquent le contenant, i.e. celui ayant le support le plus grand, est recherché par niveau dans le treillis, comme suit : la fonction Get-support enfile tous les noeuds pères directs de C, à savoir* $\{W, T, D\}$. *Pour mentionner la fin d'un niveau, elle enfile également à la suite de D, la chaîne FING qui indique la fin d'une génération de noeuds pères. Elle com-*

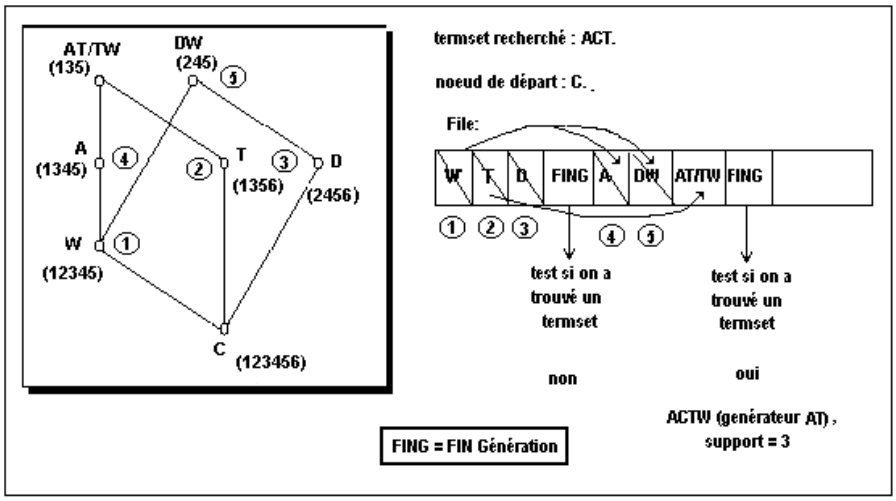

Figure 3. *Exemple d'exécution de la fonction* Get-support

mence ensuite par traiter la tête de la file, i.e. W, en vérifiant si sa fermeture[2] *contient le termset recherché. Si ce n'est pas le cas alors la fonction Get-support enfile successivement les noeuds pères directs de W jusqu'à arriver à la chaîne* FING, *qui sera enfilée en queue de file et indique que le deuxième niveau est terminé. Lorsque la fonction Get-support arrive au niveau du termset ACTW dans la file, ceci signifie qu'un termset fréquent contenant ACT est trouvé. La fonction stocke ainsi son support et continue jusqu'à ce qu'elle finisse tout le niveau, puisque il est possible de trouver un autre termset fréquent contenant ACT et ayant un support plus grand. Dans l'exemple considéré, la fonction Get-support trouve juste après ACTW la chaîne* FING, *ceci veut dire que le minsupp=3 est atteint et aucun autre termset fréquent n'a ètè retenu. Nous pouvons ainsi conclure que* $support(ACT) = support(ACTW)$.

5. Intérêt des règles associatives hiérarchiques pour la RI

Une large panoplie d'algorithmes et d'approches existent dans la littérature pour l'extraction de règles associatives entre termes à partir des données textuelles [FEL 95, FEL 96, RAJ 97, FEL 97]. Tout l'intérêt de ces règles associatives se trouve dans leur utilisation dans des domaines connexes au textmining. En ce qui nous concerne, nous orientons l'étude de l'intérêt et de l'exploitation des règles associatives entre termes

2. Dans le cadre de notre approche proposée dans [LAT 03b], la fermeture de la connexion de Galois appliquée à un générateur minimal du treillis, permet de donner le termset fréquent correspondant à ce générateur, sachant qu'ils ont le même support.

dans le contexte de la recherche d'information (RI) et plus précisément, l'expansion de requêtes.

L'expansion de requêtes est vue comme un traitement pour *élargir* le champ de recherche pour une requête. Une requête étendue va englober plus de termes reliés. Cependant, l'expansion de requêtes est basée sur la co-occurrence des termes dans un corpus. Différentes méthodes sont proposées dans la littérature [SPA 91, QUI 93, ADR 99, MAN 00]. De par ces méthodes, nous considérons que les règles associatives entre termes présentent un intérêt certain pour la RI, en les exploitant pour l'expansion des requêtes.

Les résultats présentés dans [LAT 03a] confirment l'intérêt des règles associatives pour la recherche d'information. Nous avons montré que l'utilisation des règles associatives dans le contexte de l'expansion automatique et interactive de requêtes, permet d'augmenter les performances d'un système de recherche d'information expérimental en terme de rappel et de précision [LAT 03a]. Nous pouvons ainsi dire que les règles associatives entre termes permettent d'illustrer efficacement des liens entre les termes, reflétant le contexte d'utilisation d'un terme dans un corpus ainsi que le contenu thématique des documents du corpus.

Cependant, nous pensons que les règles associatives hiérarchiques, telles qu'elles sont définies dans cet article, peuvent être d'un grand intérêt pour un SRI. Ainsi, les règles associatives spécifiques permettent de diminuer le problème de bruit lors de l'interrogation dans le sens où quand un grand nombre de documents répondent à une requête, le fait de l'étendre avec les termes provenant de ces règles, permet de spécifier et cibler explicitement le besoin de l'utilisateur. D'un autre coté, les règles associatives génériques permettent de réduire le problème de silence, puisque dans le cas où aucun document ne répond à la requête initiale, le fait de l'étendre par des termes plus généraux et qui existent forcément dans certains documents, permet d'élargir le champ de recherche.

6. Conclusion

Dans cet article, nous avons proposé une nouvelle approche pour la génération de règles associatives entre termes, en supposant l'existence d'une taxonomie associée au corpus de textes. L'idée est de générer, en plus des règles associatives non redondantes [LAT 03b], d'autres règles *génériques*, *spécifiques* et/ou *équivalentes,* par rapport à un terme donné, en exploitant les relations sémantiques existantes entre les termes dans une taxonomie. Toute la difficulté de l'approche proposée réside dans le calcul du support de la règle hiérarchique. Il serait intéressant de prouver expérimentalement l'apport de règles associatives hiérarchiques dans une application d'expansion de requêtes. Ceci, nécessite une collection test, à laquelle est associée une taxonomie du domaine, afin de pouvoir générer les règles associatives hiérarchiques.

7. Bibliographie

[ADR 99] ADRIANI M., RIJSBERGEN C. J. V., « Term Similarity-Based Query Expansion for Cross-Language Information Retrieval », *Proceedings of the International Conference ECDL'99*, 1999.

[AGG 98] AGGRAWAL C., YU P., « Online Generation of Association Rules », *Proceedings of the 14th International Conference on Data Engineering, IEEE edition*, 1998, p. 402–411.

[AGR 94] AGRAWAL R., SKIRANT R., « Fast algorithms for Mining Association Rules », *Proceedings of the 20th International Conference on Very Large Databases*, June 1994, p. 478–499.

[BAS 00] BASTIDE Y., PASQUIER N., TAOUIL R., LAKHAL L., STUMME G., « Mining minimal non-redundant association rules using frequent closed itemsets », *Proceedings of the International Conference DOOD'2000, LNCS, Springer-verlag*, july 2000, p. 972–986.

[BEN 02] BENYAHIA S., SLIMANI Y., « Discovering Association Rules using the Formal Concept Analysis », *Extraction des Connaissances et Apprentissage (Numéro spécial des Journé es Francophones d'Extraction et Gestion des Connaissances(EGC'2002)), Montpellier, France*, vol. 1, n° 4, 2002, p. 233–244.

[BRI 97] BRIN S., MOTWANI R., ULLMAN J., « Dynamic Itemset Counting and Implication Rules », *Proceedings ACM SIGMOD, International conference on Management of Data ,Tucson, Arizona, USA*, 1997, p. 255-264.

[DUQ 86] DUQUENNE J., GUIGUES J., « Famille minimale d'implications informatives résultant d'un tableau de données binaires », *Mathématiques et Sciences Humaines*, vol. 95, n° 24, 1986, p. 5–18.

[FEL 95] FELDMAN R., DAGAN I., « Knowledge Discovery in Textual Databases », *Proceedings of the first International Conference on Knowledge Discovery in Databases KDD'95, Montreal*, August 1995, p. 112–117.

[FEL 96] FELDMAN R., HIRSH H., « Mining Associations in Text in the Presence of Background Knowledge », *Proceedings of the Second International Conference on Knowledge Discovery in Databases (KDD96), Portland*, August 1996, p. 343–346.

[FEL 97] FELDMAN R., KLOSGEN W., ZILBERSTIEN A., « Document Explorer : Discovering Knowledge in Document Collections », *Proceedings of ISMIS97, Lecture Notes in AI, Springer Verlag, NC, USA*, October 1997.

[GAN 99] GANTER B., WILLE R., *Formal Concept Analysis*, Springer-Verlag, Heidelberg, 1999.

[GAR 98] GARDARIN G., PUCHERAL P., WU F., « Bitmap based algorithms for mining association rules », *Proceedings of 14th International Conference Bases de Données Avancées, Hammamet, Tunisia*, 26–30 October 1998, p. 157–175.

[HAN 95] HAN J., FU Y., « Discovery of multiple-level association rules from large databases », *Proceedings of the VLDB Conference*, 1995, p. 420–431.

[HUE 01] HUELLERMEIR E., « Implication-based fuzzy association rules », *Proceedings of the PKDD'2001 : Principles of Data Mining and Knowledge Discovery, Springer-verlag, Freiburg, Germany*, september 2001, p. 241–252.

[LAT 01] LATIRI C. C., BENYAHIA S., « Textmining : Discovering explicit formal concepts from unstructutred data », *Proceedings of the XIXéme Congrés Informatique des Organisations et Systèmes d'Information et de Décision, INFORSID'01, Suisse*, Mai 2001, p. 27–39.

[LAT 03a] LATIRI C. C., BENYAHIA S., MINEAU G., « Conceptual Non-Redundant Association Rules Discovery : Application to Query Expansion », *Proceedings of the First International Conference on Formal Concept Analysis : The State of the Art (ICFCA03), Damstadt, Allemagne*, vol. 3, February-March 2003.

[LAT 03b] LATIRI C. C., BENYAHIA S., MINEAU G., JAOUA A., « Découverte des règles associatives non redondantes : Application aux corpus textuels », *Proceedings Des Journées francophones EGC'2003, Lyon. Publié dans la revue d'Intelligence Artificielle, Vol 17, N°1, 2003, Janvier2003, p. 131 − −144.*

[LIU 99] LIU B., HSU W., ANG K. W., CHE S., *« Visually aided exploration of interesting association rules », Proceedings of the 3rd international Conference on research and development in Knowledge Discovery and Data mining (PAKDD'99), LNCS, Springer-verlag, vol. 1574, April 1999, p. 380–389.*

[LUX 91] LUXEMBURGER M., *« Implication partielles dans un contexte », Mathématiques et Sciences Humaines, vol. 29, n° 113, 1991, p. 35–55.*

[MAN 00] MANDALA R., TOKUNAGA T., TANAKA H., *« Query expansion using heterogeneous thesauri », Information Processing and Management, , n° 36, 2000, p. 361–378.*

[NG 98] NG R. T., LAKSHMANAN V. S., HAN J., PANG A., *« Exploratory mining and pruning optimizations of constrained association rules », Proceedings of the SIGMOD Conference, 1998, p. 13–24.*

[PAS 98] PASQUIER N., BASTIDE Y., TOUIL R., LAKHAL L., *« Pruning closed itemset lattices for association rules », Proceedings of 14th International Conference Bases de Données Avancées, Hammamet, Tunisia, 26–30 October 1998, p. 177–196.*

[QUI 93] QUI Y., FREI H. P., *« Concept Based Query Expansion », Proceedings of the 16th International Conference on Research and Development on Information Retrieval, ACM-SIGIR, 1993.*

[RAJ 97] RAJMAN M., BESANÇON R., *« Data mining and Reverse Engineering, Chapter 3 : Text Mining : Natural language techniques and Text Mining Application », p. 51–64, 1997.*

[SIN 99] SINGH L., CHEN B., HAIGHT R., SCHEUERMANN P., *« An Algorithm for Constrained Association Rule : Mining In Semi-Structured Data », Proceedings of the third Pacific-Asia Conference, PAKDD'99, Beijing, China, 1999.*

[SPA 91] SPARK-JONES K., *« Notes and references on early classification work », SIGIR Forum, vol. 1, n° 25, 1991, p. 10–17.*

[SRI 95] SRIKANT R., AGRAWAL R., *« Mining Generalised Associations Rules », Proceedings of the 21th International Conference on Very Large Databases, Zurich, Switzerland, 1995, p. 407–419.*

[SRI 97] SRIKANT R., VU Q., AGRAWAL R., *« Mining Association Rules with Item Constraints », Proceedings of the 3rd International Conference on Knowledge Discovery in Databases and Data Mining, Newport Beach, California, August 1997.*

[STU 01] STUMME G., TAOUIL R., BASTIDE Y., PASQUIER N., LAKHAL L., *« Intelligent structuring and reducing of association rules with formal concept analysis », Proceedings of KI'2001 conference, LNAI 2174, Springer-verlag, september 2001, p. 335–350.*

[ZAK 00] ZAKI M., *« Generating Non-Redundant Association Rules », Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, USA, August 2000, p. 34–43.*