



HAL
open science

ARSA : Analyse de Réseaux Sociaux pour les Administrations

Etienne Cuvelier, Géraldine Bous, Marie-Aude Aaufaure, Gerald Kleser

► **To cite this version:**

Etienne Cuvelier, Géraldine Bous, Marie-Aude Aaufaure, Gerald Kleser. ARSA : Analyse de Réseaux Sociaux pour les Administrations. 2012. hal-00704314

HAL Id: hal-00704314

<https://hal.science/hal-00704314>

Preprint submitted on 5 Jun 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ARSA: Analyse de Réseaux Sociaux pour les Administrations

Une expérience d'intégration de réseaux sociaux internes et externes dans une administration

Étienne Cuvelier¹, Géraldine Bous², Marie-Aude Aaufaure¹,
Gerald Kleser²

1. *Équipe Business Intelligence, Laboratoire Mathématiques Appliquées aux Systèmes, École Centrale Paris, France.*

etienne.cuvelier@ecp.fr

2. *Business Intelligence Practice, SAP Research, Sophia Antipolis, France.*

geraldine.bous@sap.com

RÉSUMÉ. Dans cet article nous relatons un premier retour d'expérience sur le projet ARSA qui tend à intégrer l'utilisation de réseaux sociaux dans l'administration, avec une expérimentation en collaboration avec la ville d'Antibes. Cette intégration est déclinée en deux directions: d'une part la mise en place d'un réseau social interne qui tient compte de l'organisation hiérarchique de l'administration et d'autres types de relations; d'autre part un monitoring de l'e-réputation de la ville sur le réseau externe Twitter. Les deux axes de cette intégration sont exposés en détail dans cet article.

ABSTRACT. In this paper we give a first summary of the ARSA project which tends to promote the using of social networks in public administrations, with experimentation in collaboration with the town of Antibes. The integration of social networks in administrations is tested in two directions: on the one hand we address the design and the installation of an internal social network that takes into account the hierarchical organization of the administration, but allows other types of relationships; on the other hand we setup a monitoring tool of the e-reputation of the town of Antibes on the external network Twitter. The two axes of this integration are discussed in detail in this paper.

MOTS-CLÉS: Réseaux Sociaux, Visualisations, Twitter, e-réputation

KEYWORDS: Social Networks, Visualizations, Twitter, e-reputation.

DOI:10.3166/ISL.x.y.1-23 © 2012 Lavoisier

1. Introduction

Les outils de réseaux sociaux ont connu un développement hors norme au cours de ces dernières années, apportant de nouveaux moyens d'interactions sociales et modifiant ainsi les façons de communiquer et de partager l'information. Bien qu'ayant dans un premier temps connu une forte percée dans le cadre privé, ces outils atteignent maintenant les entités tant publiques que privées que sont les administrations et les entreprises. Les plus réactives n'ont pas tardé à utiliser ces outils pour, par exemple, communiquer avec le public ou surveiller ce qui est dit à leur propos dans ces nouveaux médias, faisant ainsi émerger le concept d'e-réputation. Mais les outils de réseaux sociaux n'ont pas seulement modifié le rapport à l'information des particuliers, ils ont aussi modifié certains comportements, comme le recours au crowdsourcing ou l'établissement (ou rétablissement) de liens sociaux qui n'existeraient pas (ou plus) sans ces nouveaux médias. C'est dans ce contexte que c'est déroulé le projet ARSA (Analyse des Réseaux Sociaux pour Administrations), dont nous fournissons un premier retour d'expérience dans cet article. Ce projet pilote de deux ans (fin prévue en décembre 2011) en collaboration avec la ville d'Antibes (France) se concentre essentiellement sur deux points:

- une adaptation de l'outil SNA (Social Network Analyzer), développé par SAP pour l'analyse des réseaux sociaux, aux besoins des administrations publiques,
- la mise au point d'un outil de monitoring et d'analyse de l'e-réputation de la ville d'Antibes sur le réseau Twitter.

Nous nous focalisons sur l'aspect *réseau social interne* comme une extension de l'intranet de la ville d'Antibes ayant pour objectif d'améliorer la collaboration transversale. Il s'agit non seulement de pouvoir visualiser et partager des données, mais également de pouvoir explorer visuellement la structure hiérarchique de l'organisation, ainsi que de modéliser et visualiser des projets collaboratifs. Selon la culture de l'entreprise et la sensibilité des données partagées, il peut être nécessaire dans ce contexte de gérer les droits d'accès en fonction, par exemple, du statut hiérarchique de la personne. D'autre part, dans un souci d'écoute de la satisfaction de ses administrés et usagers, ce projet propose à la ville d'Antibes d'écouter ce qui bruisse à son propos sur un des *réseaux sociaux externes* les plus actifs, à savoir Twitter, et d'organiser le résultat de ce monitoring de façon à attirer l'attention sur les informations les plus relayées à son propos. Un tel monitoring, en permettant de percevoir en temps réel quelles sont les opinions relatives à la ville d'Antibes, donne notamment la possibilité à cette dernière d'avoir un feedback beaucoup plus rapide sur l'action de son administration.

Cet article est structuré en deux grandes parties, la première étant consacrée au réseau social interne mis en place grâce à l'outil SNA. Outre une description de SNA, une analyse des différentes possibilités et pistes d'amélioration de l'outil est effectuée. En particulier, nous traitons l'aspect visualisation et exploration des réseaux à entités et relations multiples. La deuxième partie se concentre sur la problématique du mo-

monitoring de l'e-réputation sur les réseaux sociaux externes, avec un premier prototype pour Twitter.

2. Réseaux sociaux internes

Le phénomène de partage et d'échange collectif d'informations a commencé à éveiller l'intérêt des entreprises. Des études récentes montrent en effet que la collaboration transversale dans une entreprise permet, d'une part, de valoriser les compétences des collaborateurs et, d'autre part, de favoriser innovation et productivité (Singh, 2005 ; Nieto, Santamaría, 2007). Or, il est souvent difficile pour une entreprise de mettre en place une telle approche, non seulement à cause de la structure organisationnelle qui a tendance à favoriser les échanges hiérarchiques et intra-équipes, mais parfois également à cause de sites géographiquement distribués (Singh, 2005). L'utilisation d'outils de réseaux sociaux, permettant de créer des communautés virtuelles par centre d'intérêt, occupation ou autres, est perçue comme une solution potentielle à ce problème ou – en tout cas – comme une approche favorisant sa résolution (McAfee, 2006 ; 2009 ; Bughin, 2008). L'adoption des outils du web 2.0 au sein des entreprises, ainsi que les fonctionnalités requises par ceux-ci pour répondre aux exigences des entreprises, restent cependant peu étudiées jusqu'à présent (Richter, Riemer, 2009). Les administrations publiques de villes de moyenne et grande taille ont, très souvent, des problèmes similaires à ceux décrits ci-dessus. Elles doivent en effet concilier tâches de gestion et services rendus à la population; ces services couvrent eux-mêmes une large gamme de domaines, allant de la maintenance routière, la planification urbaine, jusqu'à l'assistance à la personne. Une administration publique – en tant qu'entreprise – regroupe un grand nombre de personnes, souvent géographiquement distribuées, avec des compétences et fonctions couvrant une large gamme de métiers. Cette structure inhomogène rend complexe la gestion et l'administration d'une ville; en particulier, il est difficile pour les agents d'utiliser de manière efficace les corps de métiers et les compétences présentes au sein des équipes. Les administrations publiques constituent donc un cas d'étude intéressant pour déterminer les fonctionnalités requises par des outils d'analyse de réseaux sociaux professionnels, en particulier celles permettant de réduire l'effet des contraintes structurelles et favorisant la collaboration transversale.

2.1. Cas d'utilisation

Dans le cadre du projet ARSA, plusieurs cas d'utilisation concrets ont été établis à la demande de l'utilisateur final, l'administration de la ville d'Antibes.

La première exigence formulée par notre partenaire fut d'avoir à disposition un moyen pour faciliter la collaboration transversale entre le personnel de la ville. En réponse à cette attente, un outil faisant fonction d'annuaire étendu du personnel, permettant d'effectuer des recherches et d'explorer visuellement la structure hiérarchique de l'organisation fut mis à disposition de l'utilisateur. Un cas d'utilisation important est la modélisation de projets collaboratifs transversaux avec la possibilité de partager des documents ou autres ressources.

Deuxièmement, la ville d'Antibes a souhaité avoir à sa disposition un outil qui permette d'analyser visuellement le réseau, tout en y représentant, de manière simultanée et claire, plusieurs types d'entités et relations. Cela permet, par exemple, de visualiser des projets collaboratifs ou des flux de données (par ex. flux de courriels) en superposition à la structure hiérarchique de la ville.

Il y a également un autre intérêt dans la modélisation d'entités et relations autres que les liens entre personnes: créer un outil d'analyse qui regroupe les fonctionnalités d'outils de réseaux sociaux *et* de Business Intelligence. Plus particulièrement, l'analyse des ressources et des coûts selon la structure hiérarchique. Considérons l'exemple du parc logistique dont dispose une ville. Les véhicules peuvent être caractérisés selon plusieurs critères, comme par exemple leur type ou leur disponibilité. Une exploration hiérarchique permet de calculer les proportions de véhicules disponibles par types, ainsi que de visualiser clairement la composition du parc et comment celle-ci se décline en fonction des divers services qui composent l'administration. De la même manière, il est possible d'envisager une analyse des coûts par entité, selon la hiérarchie.

Ces cas d'utilisation, qui diffèrent notablement de l'utilisation et l'analyse classique des réseaux sociaux, imposent le développement d'outils spécifiques permettant non seulement d'intégrer plusieurs sources de données, mais aussi de modéliser et visualiser plusieurs types d'entités et relations. Dans ce qui suit, nous présentons une brève description de SNA, un outil créé à cet effet; par la suite, nous décrivons les pistes de recherche et développement de l'outil issues des exigences de l'utilisation dans le contexte des entreprises et administrations publiques.

2.2. *Le logiciel SNA*

2.2.1. *Description de l'outil*

Le logiciel SNA¹ (Social Network Analyzer) est un logiciel d'analyse de réseaux sociaux. Plus qu'une interface pour se connecter au réseau et visiter la page d'autres membres, il s'agit d'un outil destiné à l'exploration, à la recherche et au partage d'informations au sein d'une entreprise. Il est, par ailleurs, également possible d'utiliser l'outil pour étendre le réseau à des personnes extérieures à l'entreprise, comme des fournisseurs ou des clients. SNA est composé d'une interface administrateur pour la gestion et l'importation des données, la modélisation des entités et des relations, ainsi que pour la gestion des droits d'accès. D'autre part, il se compose d'une interface utilisateur, représentée à la figure 1, qui permet de visualiser le réseau et les informations qui s'y trouvent. En outre d'être visuel, l'outil est également interactif; autrement dit, la sélection d'un nœud permet de visualiser le réseau autour de celui-ci, ce qui rend l'exploration du réseau simple et rapide. L'outil remplit la fonction d'annuaire étendu, dont les champs sont configurables, et comporte un moteur de recherche. Il offre également la possibilité de mettre des ressources (documents, liens, etc.) à disposition

1. Une démonstration de l'outil est disponible sur <http://sna-demo.ondemand.com/>

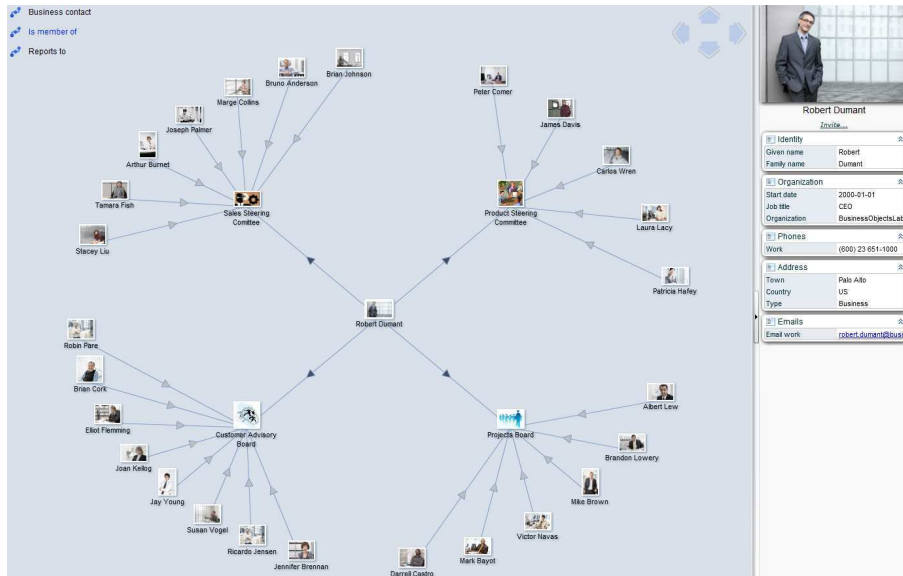


Figure 1. Interface utilisateur du logiciel SNA.

d'autres utilisateurs. Finalement, SNA dispose d'un moteur analytique permettant de calculer des statistiques sur les informations disponibles dans les profils. Par exemple, pour un champ «domaine de spécialité», l'outil permet de collecter les diverses spécialités, calcule automatiquement le pourcentage de profils qui y correspondent, pour finalement afficher le résultat dans un diagramme circulaire.

2.2.2. Extensions de l'outil

Si SNA offre des fonctionnalités qui permettent de répondre à bon nombre des attentes de la ville d'Antibes dans le cadre du projet ARSA, certaines pistes restent encore à explorer. La possibilité d'intégrer des données de types divers a un intérêt certain dans le cadre de ce projet, mais cela soulève plusieurs défis techniques.

Un premier défi concerne les droits d'accès hiérarchiques. L'intégration de données sensibles, tels des budgets ou des coûts, avec d'autres données accessibles à tous les utilisateurs, demande une gestion des droits d'accès plus élaborée. Il s'agit de définir les droits selon le type d'utilisateur et, plus précisément, selon la position hiérarchique de celui-ci dans la structure organisationnelle. En général, on peut partir du principe qu'un utilisateur peut avoir accès aux informations sensibles des personnes se trouvant directement ou indirectement sous sa responsabilité, mais pas à celles de ses supérieurs hiérarchiques ou des personnes travaillant dans d'autres services. Il existe essentiellement deux possibilités pour approcher ce problème (Sandhu, Samarati, 1994). Une solution est d'attribuer les autorisations lors de la connexion de l'utilisateur sur le réseau, c'est-à-dire, de prédéterminer une image des données auxquelles celui-ci aura accès lors de la connexion. Cette solution est possible tant que

la taille du réseau et le nombre de données est faible. À l’opposé, solution privilégiée dans le cadre d’ARSA, il est possible de restreindre l’accès aux données non-autorisées en temps réel, au cours l’exploration des données.

Un deuxième défi concerne l’analyse des données sous contraintes structurelles telle que évoquée dans les cas d’utilisation. Qu’il s’agisse d’explorer, en suivant la structure organisationnelle, le parc logistique de la ville ou la nature des dépenses, il est nécessaire de pouvoir formuler des requêtes avancées sur les données du réseau. L’idée derrière ce type d’analyse avancée se trouve à l’intersection entre le concept de réseau social, en tant que graphe possédant une structure, et celui de l’exploration de données structurées tels que les cubes de données dans le cadre de l’entreposage de données (data warehousing) (Han, Kamber, 2006). Une solution simple et novatrice à ce problème consiste à intégrer la structure hiérarchique de l’organisation, c’est-à-dire une partie du réseau, en tant que dimension à part entière dans un cube de données. Cette solution a l’avantage de pouvoir exploiter les connaissances et les outils existants dans le domaine de l’entreposage de données.

Finalement, la troisième extension de l’outil issue des cas d’utilisation du projet ARSA est l’analyse visuelle des réseaux sociaux comportant plusieurs types de relations et entités. Ce point exige le développement d’algorithmes de visualisation dédiés, que nous détaillons ci-après.

2.3. Analyse visuelle de réseaux sociaux

L’une des caractéristiques distinctives de SNA est d’offrir la possibilité d’explorer le graphe du réseau de manière visuelle. Or, l’existence de plusieurs types d’entités, ainsi que de plusieurs types de relations les reliant, pose également des défis au niveau de la visualisation et, plus spécifiquement, au niveau du traçage du graphe à l’écran. Bien que la visualisation et le «dessin de graphes» (*graph drawing*) aient déjà fait l’objet de nombreuses études (Battista *et al.*, 1999 ; Herman *et al.*, 2000), il existe toujours un nombre important de sujets de recherche ouverts dans ce domaine, notamment en ce qui concerne la visualisation et l’exploration de relations multiples (Landesberger *et al.*, 2011). La plupart des travaux récents dans ce domaine utilisent des techniques diverses pour grouper les nœuds lors de l’affichage (Heer, Boyd, 2005 ; Muelder, Ma, 2008), parfois en combinaison avec des techniques de filtrage pour simplifier le graphe au préalable (Shen *et al.*, 2006 ; Wattenberg, 2006 ; Sun *et al.*, 2009 ; Cao *et al.*, 2010). Cependant, aucune de ces approches ne s’adresse directement à la visualisation *structurée* des différentes relations, tel que nous le proposons ci-après.

2.3.1. Description du problème

L’objectif premier dans la visualisation d’une structure hiérarchique pure est d’éviter ce que l’on appelle communément «l’effet spaghetti», c’est-à-dire, de calculer la position des nœuds et des arcs (ou arêtes) sans qu’il y ait superposition, ainsi que de minimiser autant que possible le nombre d’arcs qui se coupent entre eux. La méthode de Sugiyama pour le dessin de graphes hiérarchiques (Sugiyama *et al.*, 1981) permet

de résoudre ce problème pour des arbres, ainsi que pour des graphes orientés. Il existe plusieurs implémentations heuristiques de cette méthode qui permettent d'obtenir de bons résultats avec un faible temps de calcul (Battista *et al.*, 1999; Eiglsperger *et al.*, 2005). Si le traçage de structures hiérarchiques ne pose pas de problème particulier dans le cas de graphes classiques, c'est-à-dire définis la sur base d'une seule relation, cela devient beaucoup moins vrai lorsqu'il s'agit de visualiser plusieurs relations en même temps. Un premier exemple concret de cette idée est la visualisation de processus au sein de la structure hiérarchique, comme par exemple le suivi d'échanges d'informations entre collaborateurs qui ont mené à la résolution d'un problème donné. Une analyse visuelle telle que celle-ci facilite la compréhension des étapes de résolution du problème, ainsi que le rôle des personnes et services qui sont intervenus, permettant ainsi à l'analyste de suggérer des solutions pour cibler des démarches plus rapides et efficaces. Un deuxième exemple concerne la visualisation de projets collaboratifs. Dans ce cas, il s'agit, en superposition à la structure hiérarchique, de visualiser un graphe supplémentaire composé d'un nœud «projet» relié aux collaborateurs qui y contribuent. De surcroît, la structure du projet peut elle-même être hiérarchique en fonction des rôles et responsabilités attribuées aux collaborateurs.

Dans ces deux exemples, il s'agit donc de visualiser deux graphes simultanément: la structure hiérarchique de l'organisation, ainsi qu'un graphe supplémentaire modélisant une structure définie par une autre relation. La solution la plus simple à ce problème consiste à tracer la structure organisationnelle, puis d'y superposer le graphe secondaire en traçant des arcs pour relier les sommets concernés. Cette solution fournit malheureusement des résultats visuellement médiocres, puisqu'il est impossible d'empêcher la superposition ou l'intersection d'arcs en procédant de la sorte. Une solution plus efficace consiste donc à utiliser des algorithmes de prétraitement ayant pour objectif de «fusionner» les deux graphes, pour appliquer ensuite des algorithmes tels que celui de Sugiyama. La visualisation de *graphes composés* de ce type a fait l'objet de très peu d'études jusqu'à présent (Sugiyama, Misue, 1991; Bertault, Miller, 1999; Forster, 2002; Raitner, 2004; Holten, 2006; Burch *et al.*, 2010) et ne s'adressent pas à la visualisation interactive des relations multiples. En effet, l'aspect interactif complique le problème: la mise en page graphique optimale pour un graphe composé d'une hiérarchie H et d'un graphe G_i n'est pas nécessairement la même que celle du graphe composé de H et de G_j , $i \neq j$. Or, la difficulté dans ce cas est de maintenir une cohérence entre les visualisations successives pour ne pas détruire la *représentation mentale* que l'utilisateur se construit du graphe, critère dont l'importance a été prouvée de manière expérimentale (Purchase, 2000; Purchase *et al.*, 2006; Saffrey, Purchase, 2008).

2.3.2. Heuristique pour la visualisation relations multiples linéaires dans un arbre

Une solution simple pour permettre une exploration interactive des diverses relations est de calculer préalablement la disposition visuelle d'un graphe composé incluant toutes les relations. Une sélection (ou désélection) de relations correspond dans ce cas à afficher (resp. cacher) une partie du graphe composé sans avoir à modifier le reste. Soit une hiérarchie $H(V, E)$, c'est-à-dire un arbre orienté de sommets $v \in V$ et

d'arcs $(v, w) \in E$ avec $E \subseteq V \times V$. En parallèle, soit un nombre I de graphes orientés $G_i(V_i, E_i)$, $i \in I$, avec $V_i \subseteq V$. Nous considérons le cas particulier où chaque G_i modélise un processus linéaire, c'est-à-dire une séquence de sommets $(v_{i1}, v_{i2}, \dots, v_{in_i})$, où n_i représente la longueur du processus i .

La méthode heuristique proposée se compose de quatre étapes: premièrement, la hiérarchisation qui détermine la profondeur $y(v)$ des sommets $v \in H$ dans l'arbre. Deuxièmement, la phase d'agrégation qui permet d'incorporer H et les processus G_i dans un seul graphe orienté que nous noterons H_G , de sommets V_G et d'arcs E_G . La troisième phase consiste à appliquer un algorithme pour déterminer l'ordre $o(v)$ des sommets dans chaque couche de H_G en vue de minimiser les intersections entre arcs. Finalement, la dernière étape consiste à déterminer la position horizontale $x(v)$ exacte des sommets pour l'affichage à l'écran. Il est à remarquer que les étapes une, trois et quatre composent l'algorithme de Sugiyama (Sugiyama *et al.*, 1981) dans sa version classique. L'originalité de la méthode que nous proposons ici réside, d'une part, dans le problème qu'elle résout et, d'autre part, dans la technique utilisée pour agréger les différents graphes. En effet, comme nous le verrons sous peu, notre approche permet de modéliser des contraintes de proximité (ou d'adjacence) pour des nœuds se trouvant dans la même couche de H . Ces contraintes, qui ne peuvent pas être prises en compte dans la méthode classique, sont nécessaires pour visualiser les processus linéaires de façon, elle aussi, linéaire.

2.3.2.1. Hiérarchisation

La première étape de notre méthode consiste à déterminer la position verticale $y(v)$ des nœuds $v \in H$. La profondeur de tout nœud calculée à cette étape sera maintenue lors des étapes suivantes: la position verticale d'un nœud v est donc la même dans H et H_G . Plusieurs méthodes peuvent être utilisées pour attribuer un nœud à une couche dans un arbre orienté (Battista *et al.*, 1999); nous optons pour la méthode de la profondeur minimale, c'est-à-dire, attribuer chaque nœud à la couche la plus proche de la racine tout en se trouvant en-dessous de ses prédécesseurs. Soit

$$\delta_H(i, j) = \begin{cases} 1 & \text{si } (i, j) \in E \\ 0 & \text{sinon.} \end{cases} \quad (1)$$

Si $\delta_H(i, j) = 1$, cela veut donc dire qu'il existe un arc allant des sommets i vers j . La résolution du programme linéaire suivant permet obtenir les coordonnées verticales des nœuds de H :

$$\begin{aligned} \min \quad & \sum_{i \in V} y(i) \\ \text{s.t.} \quad & y(j) - y(i) \geq 1 \quad \text{si } \delta_H(i, j) = 1 \\ & y(i) \geq 1 \quad \forall i \in V. \end{aligned} \quad (2)$$

2.3.2.2. Agrégation et modélisation de contraintes structurelles

L'objectif des contraintes structurelles que nous allons décrire dans cette section est double: premièrement, celle de pouvoir visualiser un ou plusieurs G_i en minimisant le nombre d'intersections entre arcs. Deuxièmement, de visualiser le processus

de manière linéaire, c'est-à-dire de modéliser des contraintes de «voisinage» entre des sommets ayant les mêmes coordonnées verticales. Le but de ces contraintes est d'éviter d'avoir des arcs traversant la hiérarchie de part et d'autre pour connecter les sommets dans un processus. Par exemple, à la figure 2, en haut à gauche, la superposition de la hiérarchie et du processus impliquent tant des intersections, que de tracer un arc traversant (du sommet B à E). Soient la hiérarchie H et les I processus G_i avec $(v_{i1}, v_{i2}, \dots, v_{in_i})$. Le graphe composé H_G sera construit en I étapes: au départ de H pour intégrer séquentiellement (mais dans n'importe quel ordre) chacun des I graphes G_i . La modélisation des contraintes se fait selon le pseudocode décrit ci-dessous:

```

for  $k = 1$  to  $n_i$  do
  if  $y(v_{i,k}) \neq y(v_{i,k+1})$  then
    if  $y(v_{i,k}) < y(v_{i,k+1})$  then
       $\delta_{H_G}(v_{i,k}, v_{i,k+1}) = 1$ 
    else
       $\delta_{H_G}(v_{i,k+1}, v_{i,k}) = 1$ 
    end if
  else
    ajouter le sommet temporaire  $v_t$  à  $H_G$ 
     $y(v_t) = y(v_{i,k}) + 1$ 
     $\delta_{H_G}(v_{i,k}, v_t) = 1$ 
     $\delta_{H_G}(v_{i,k+1}, v_t) = 1$ 
  end if
end for

```

Si deux sommets consécutifs dans le processus G_i ne sont pas dans la même couche, ils sont connectés dans H_G avec un arc qui va dans le sens de la profondeur de l'arbre H . Pour les sommets se trouvant dans deux couches non-adjacentes, il faut couper cet arc en segments (en ajoutant un sommet temporaire par couche interposée) pour pouvoir appliquer l'algorithme de minimisation d'intersections avec succès. Si deux sommets consécutifs dans G_i ont la même coordonnée verticale, on ajoute une contrainte visant à placer les deux sommets l'un à côté de l'autre. Comme illustré en haut et à droite de la figure 2, cette contrainte est modélisée par le biais d'un sommet temporaire v_t , qui est connecté aux deux sommets du processus et se place en dessous de ceux-ci.

2.3.2.3. Minimisation des intersections

En quoi est-ce que ces contraintes dites structurelles permettent réellement de tracer les processus de manière visuellement linéaire? Les contraintes à elles seules ne suffisent pas à obtenir cet effet: l'objectif est atteint par l'utilisation conjointe de contraintes structurelles avec des algorithmes de minimisation d'intersection d'arcs (*crossing minimization* en anglais). En effet, le fait de connecter les sommets du processus par des sommets et arcs temporaires dans une hiérarchie H_G , puis d'y appliquer un algorithme de minimisation d'intersections, force la proximité des sommets du processus. On peut facilement concevoir que, plus deux sommets connectés se trouvent «loin» (séparés par d'autres sommets), plus le nombre d'intersections entre arcs est

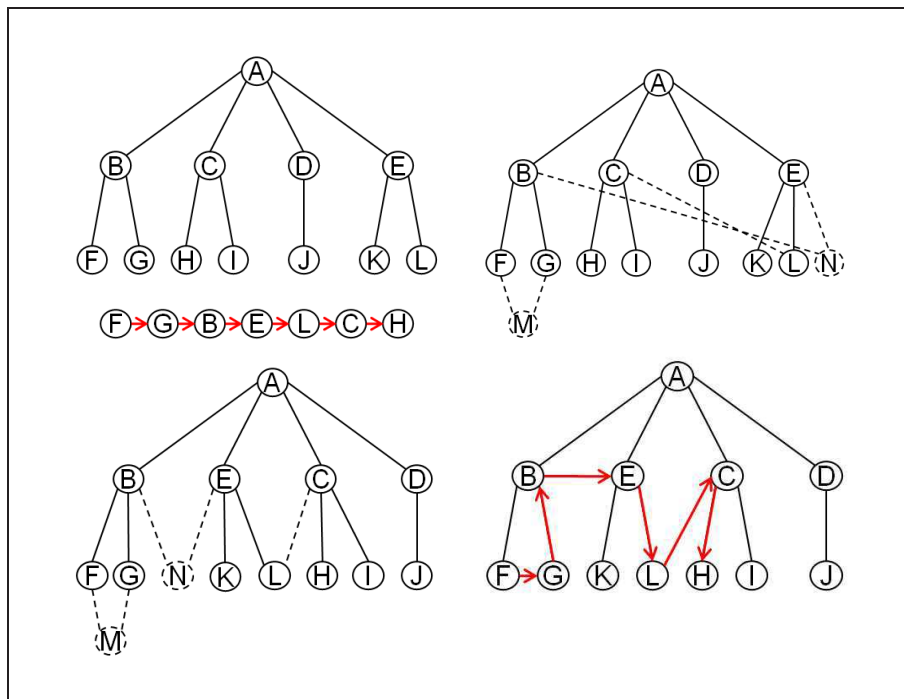


Figure 2. Exemple de hiérarchie et processus linéaire. Notre méthode utilise des contraintes structurales, avec l'ajout de sommets et arêtes temporaires (en pointillés), pour minimiser le nombre d'intersections. Dans le sens de la lecture on trouve ci-dessus, en haut à gauche, la hiérarchie et les processus, les contraintes structurales, l'optimisation et enfin, en bas à droite, le résultat final.

important. Ce principe est illustré à la figure 2 par les parties en haut à droite et en bas à gauche, qui montrent, respectivement, un exemple avant et après cette phase. La minimisation du nombre d'intersections entre les arcs d'un graphe hiérarchique est habituellement approché en traitant deux couches consécutives, c'est-à-dire, dans notre cas, entre les ensembles de sommets $\{v \in V_G \mid y(v) = k\}$ et $\{v \in V_G \mid y(v) = k+1\}$ pour $k = 1, \dots, y_{\max} - 1$, où y_{\max} dénote la profondeur maximale de H_G . Ce sous-problème, qui est connu en anglais sous le nom de *two-layer crossing problem*, est NP-complet, ce qui explique l'existence de nombreuses méthodes heuristiques destinées à le résoudre. L'objectif est de déterminer l'ordre dans lequel doivent être placés les sommets des deux couches pour minimiser le nombre d'intersections entre les arcs qui les relient. Deux méthodes rapides et bien documentées sont la *median heuristic* et la *swap heuristic*, qui existent en plusieurs variantes et exigent souvent l'utilisation des sommets fictifs (temporaires) pour bien fonctionner. Notre implémentation inclut les deux techniques, la première étant plus rapide, mais fournissant des résultats visuellement moins agréables. Le lecteur est invité à consulter (Battista *et al.*, 1999), ainsi

que les références qui y sont citées, pour une description détaillée de ces méthodes et de leur fonctionnement.

2.3.2.4. Détermination des coordonnées horizontales

Une fois l'ordre des sommets de H_G minimisant le nombre d'intersections entre arcs déterminé, il reste en réalité deux étapes distinctes: premièrement, retirer tous les sommets et arcs fictifs introduits au cours des étapes précédentes. Deuxièmement, il est important d'affiner les coordonnées horizontales des sommets restants (c'est-à-dire ceux de l'ensemble V), sans changer leur ordre, pour obtenir une distribution plus uniforme à l'écran. Plusieurs méthodes existent à cette fin, dont certaines sont détaillées dans (Battista *et al.*, 1999). La figure 2 dans sa partie basse et à droite montre un exemple de résultat final de l'application de la méthode. Une fois ces étapes terminées, la position de chaque nœud dans le plan est déterminée. La visualisation interactive des diverses relations est possible, car lors d'une sélection (ou désélection) de relations, il suffit d'afficher (resp. cacher) les nœuds et arcs correspondants.

2.3.2.5. Résultats

Nous avons comparé les résultats produits par notre méthode à la technique de la superposition, c'est-à-dire celle qui consiste à tracer l'arbre avec l'algorithme de Sugiyama puis d'y superposer les processus. Les tests ont été effectués sur des arbres (10 à 250 nœuds) et des processus (5 à 35 nœuds) générés aléatoirement. Pour chaque valeur de n et n_i , nous avons calculé le nombre moyen d'intersections, puis fait le ratio de ces moyennes pour obtenir une valeur relative. La méthode proposée est particulièrement efficace avec des arbres de petite taille (jusqu'à 50 nœuds) permettant en moyenne de réduire le nombre d'intersections de moitié. Pour des arbres plus grands (50 à 250 nœuds), l'amélioration est en moyenne de 25%, indépendamment du nombre de nœuds dans les processus. Pour améliorer les résultats sur des grands graphes, nous envisageons dans des travaux futurs de combiner notre méthode avec l'utilisation de techniques de filtrage, comme celles proposées par (Shen *et al.*, 2006) ou (Sun *et al.*, 2009), pour réduire la taille du graphe au préalable.

3. Réseaux Sociaux Externes

Depuis leur apparition, les blogs et les réseaux sociaux suscitent un intérêt croissant pour l'observation et la modélisation de l'opinion, comme le démontre, notamment, la session spéciale que leur consacrent les conférences TREC depuis l'édition de 2006 (Ounis *et al.*, 2006). La détection de sujets d'actualité brûlante fut ainsi une des tâches de l'édition 2009 de cette session blogs (Macdonald *et al.*, 2009). De même, les réseaux sociaux tels Facebook et Twitter, de par leurs fonctions de partage et de transfert de l'information devraient aussi permettre d'observer quasiment en temps réel la formation des opinions et, permettre ainsi de détecter les tendances. Ainsi (O'Connor *et al.*, 2010) ont montrés qu'il existait une corrélation très importante entre trois indices existants, calculés via des enquêtes quotidiennes et les opinions formulées sur Twitter à propos de ces sujets. Les recherches déjà menées sur Twitter portent sur la

structure du réseau (Lerman, Ghosh, 2010), sur l'usage des hash-tags, de leur diffusion et du filtering (Romero *et al.*, 2011 ; Huang *et al.*, 2010), mais ces réseaux sociaux se transforment à l'usage en véritables médias sociaux (Kwak *et al.*, 2010) et sont donc des lieux d'observation privilégiés pour de la formation des opinions, notamment à propos d'un sujet choisi, que cela soit une personne (personnal branding), une institution officielle ou un opérateur industriel. Si la surveillance du changement de l'opinion à déjà fait l'objet Par d'articles (Akcora *et al.*, 2010), il y a encore peu de choses sur le monitoring de l'e-réputation sur un réseau tel que Twitter. Dans le cas de l'e-réputation, l'observation du buzz et plus particulièrement du buzz négatif (bad buzz) concernant le sujet choisi est particulièrement importante. Nous avons testé pour la ville d'Antibes un prototype de monitoring d'e-réputation sur Twitter.

3.1. Twitter et Micro-Blogging

Twitter a été créé en 2006 dans le but de permettre à ses utilisateurs de partager facilement de courts messages textuels appelés *tweets*². Le système ayant été initialement conçu pour partager les tweets via SMS, une limite de 140 caractères a été fixée à ces messages courts. Bien que le système soit aujourd'hui massivement utilisé via le web et via des applications développées pour ordinateurs ou smartphones, cette contrainte de 140 caractères n'a jamais été levée. Ses principes de base sont les suivants: un utilisateur peut, avec son compte Twitter, générer ou transmettre de l'information via un champ de saisie, et un utilisateur A peut suivre les tweets d'un utilisateur B en le signalant via l'interface ad hoc, et ce, sans que B ne doive suivre les tweets de A en retour.

Les utilisateurs qui suivent un compte Twitter A s'appellent ses *abonnés* ou *followers*, alors que les utilisateurs que A suit s'appellent ses *abonnements* ou *followings*. L'ensemble des tweets des abonnements d'un compte s'appelle *timeline*. De par son principe de «micro-publications», Twitter permet un partage et une diffusion très rapides de l'information ainsi que des opinions concernant cette dernière. L'intense activité actuelle du réseau se traduit par une très grande réactivité par rapport aux faits d'actualité, et cette réactivité se révèle extrêmement intéressante pour l'analyse de la formation d'opinions.

Un certain nombre de pratiques et conventions ont cours sur Twitter, nous allons relever maintenant la principale, nécessaire à la compréhension minimale de la plateforme. Cette convention est la pratique du *retweet* ou *RT*. Lorsqu'un utilisateur voit dans sa timeline une information qu'il veut partager avec ses abonnés, il utilisera la fonction retweet du service (web ou application) qu'il utilise, comme illustré dans l'exemple de la figure 3. Bien que la majorité des retweets soient annoncés par «RT @», d'autres variantes et pratiques coexistent, comme cela a été très bien analysé dans (Boyd *et al.*, 2010). Par exemple, certains utilisateurs éditent le retweet en ajoutant, par exemple un (*via @*) à la fin de celui-ci, comme dans l'exemple ci-dessus. Cette

2. Gazouillis en anglais.

possibilité d'éditer un retweet, associée à la contrainte des 140 caractères crée ce que nous appellerons le *polymorphisme de l'information relayée* sur Twitter. Ce polymorphisme est illustré dans la figure 3, ainsi dans le premier groupe de tweets, le tweet n°1 est retweeté tel quel (retweet n°1.1), mais aussi avec modification (retweet n°1.2 où le RT est remplacé par un via). Ce dernier retweet modifié étant lui-même retweeté avec ajout. De même dans le troisième groupe de tweets, le tweet n°3 est aussi retweeté sans modifications (retweet n°3.1) mais aussi (retweet n°3.2) avec un ajout et deux modifications sur la forme, mais pas sur le fond («plus» remplacé par «+» et «Marineland» remplacé par le nom d'utilisateur twitter de ce dernier @Marineland_Fr). Ce polymorphisme pose un réel problème lorsque l'on veut mesurer la popularité

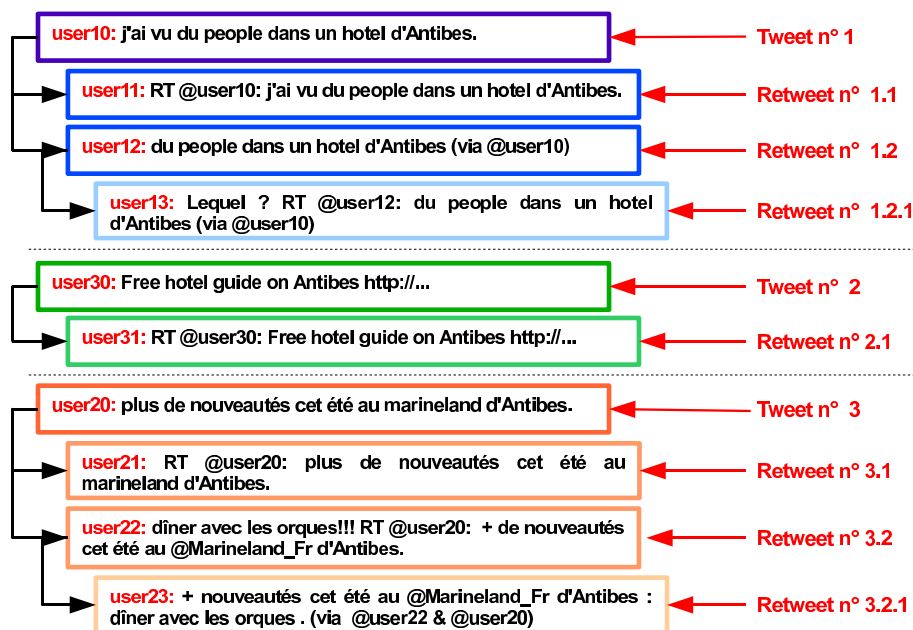


Figure 3. Illustration du polymorphisme de l'information relayée.

d'une information, car si l'on se limite à compter le nombre de retweets non-modifiés, on omettra de comptabiliser une partie des retweets, qui malgré leurs modifications véhiculent essentiellement la même information.

Néanmoins, si l'on fait abstraction des mots de liaisons et des signes de ponctuations (en l'occurrence dans notre exemple: plus, +, de, des, ...), que l'on se restreint aux mots signifiants, alors on peut représenter l'information relayée sous forme ensembliste, comme l'illustre la figure 4. On voit dans cette représentation que les inclusions successives permettent de visualiser la partie commune à tous les tweets, le noyau de l'information relayée, qui dans ce cas est le mot clé «Antibes», mais on y voit aussi les parties communes à des sous-groupes de tweets. Ainsi on voit que les tweets du premier groupe (tweets n°1, 1.1, 1.2 et 1.2.1), malgré les ajouts, véhiculent toujours la même information que l'on peut résumer en trois mots : «Antibes»,

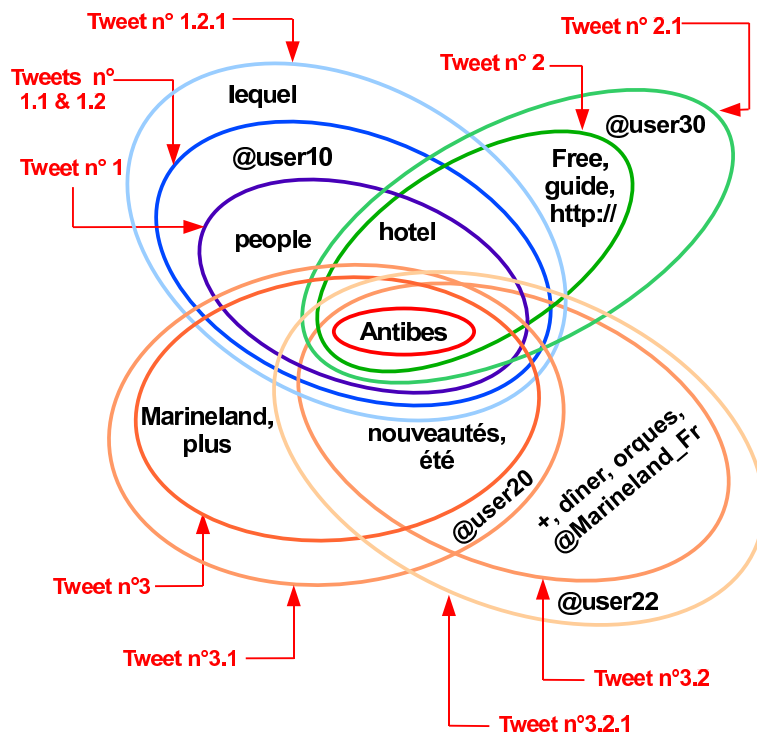


Figure 4. Vue ensembliste du polymorphisme de l'information relayée.

«people» et «hotel». De même pour les tweets du second groupe (tweets n°2 et 2.1), pour lesquels le noyau de l'information est contenu dans les mots «Antibes», «hotel», «free», «guide» et une url non reprise en détail ici. Par contre pour les tweets du troisième groupe (tweets 3, 3.1, 3.2 et 3.2.1), se subdivise en deux sous-groupes qui n'ont en commun que les mots «Antibes», «nouveau» et «été», car même si une partie de l'information est équivalente sur le fond, elle est différente sur la forme («plus» vs «+», «Marineland» vs «Marineland_Fr»). Si cette représentation ensembliste permet de bien visualiser les parties communes des différents groupes de tweets, elle ne permet pas de regrouper visuellement ces tweets. Or les inclusions intrinsèques à ces groupes définissent pourtant une relation d'ordre partiel qui, en étant représentée par un diagramme de Hasse comme dans la figure 5, permettent de voir les regroupement de tweets. L'établissement d'un tel diagramme permet d'établir, à la fois, quels sont les mots communs à un ensemble de tweets, ainsi que les différentes formes sous lesquelles une même information a été relayée. L'analyse formelle de concepts et les treillis de Galois permettent exactement de structurer l'information sous cette forme.

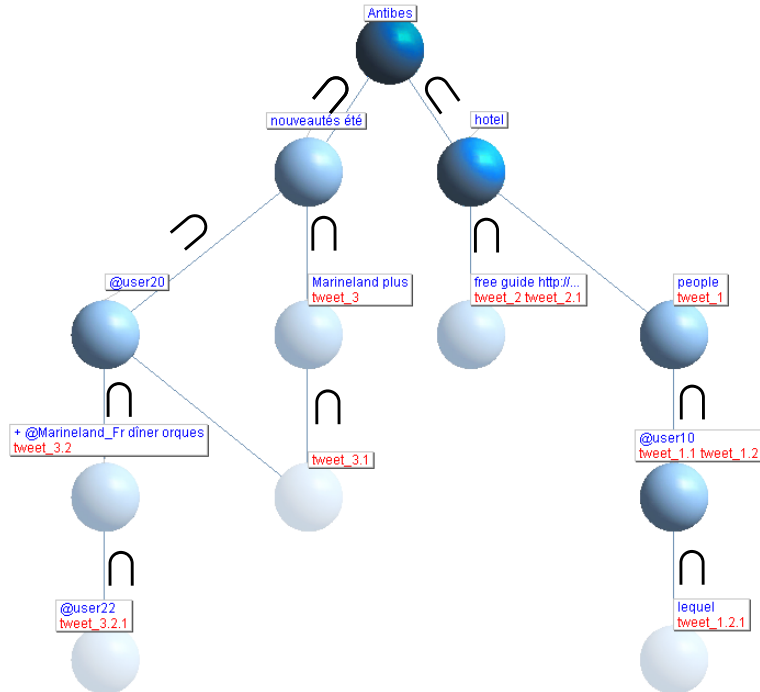


Figure 5. Analyse du polymorphisme du retweet de l'information.

3.2. Analyse Formelle de Concepts - Treillis de Galois

L'analyse formelle de concept, en anglais *Formal Concept Analysis*, FCA, (Wille, 1980) est basée sur les *Treillis de Galois* (Barbut, Monjardet, 1970), qui peuvent être utilisés pour la classification conceptuelle (Wille, 1984). Un treillis de Galois permet de regrouper, de façon exhaustive, des objets en classes, appelées *concepts*, en utilisant leurs propriétés partagées. Un treillis est classiquement basé sur une matrice booléenne, appelée *matrice de contexte* et notée C , dont les lignes représentent un ensemble d'*objets* O que l'on souhaite décrire, et les colonnes, un ensemble d'*attributs* A que ces objets ont ou non et, qui permettent donc la description des dits objets. Nous utiliserons, pour introduire ces treillis, notre exemple des tweets de la figure 3. Le tableau 1 reprend les éléments fondamentaux pour établir un treillis de Galois tel qu'illustré en figure 5, à savoir un ensemble d'objets O et un ensemble d'attributs A . Dans ce cas précis les objets, disposés en lignes, sont les tweets interceptés, et les attributs sont les mots signifiants, c'est-à-dire sans tenir compte des mots de liaisons par exemple³. Nous considérerons qu'un tweet o possède un attribut a , si ledit tweet contient le mot correspondant. La possession de la propriété $a \in A$ par l'objet

3. Pour limiter la taille du tableau à afficher dans cet article, nous avons aussi regroupé dans une seule colonne les mots qui apparaissent toujours dans le même groupe de tweets.

Tableau 1. Table de contexte pour les tweets.

| Mots Tweets | hotel | people | @user10 | free, guide, http: | Marin... plus | nouveautés, été | @user20 | +, dîner, orques, @Marin... | @user22 | Antibes |
|-------------|-------|--------|---------|--------------------|---------------|-----------------|---------|-----------------------------|---------|---------|
| 1 | x | x | | | | | | | | x |
| 1.1 | x | x | x | | | | | | | x |
| 1.2 | x | x | x | | | | | | | x |
| 1.2.1 | x | x | x | | | | | | | x |
| 2 | x | | | x | | | | | | x |
| 2.1 | x | | | x | | | | | | x |
| 3 | | | | | x | x | | | | x |
| 3.1 | | | | | x | x | x | | | x |
| 3.2 | | | | | | x | x | x | | x |
| 3.2.1 | | | | | | x | x | x | x | x |

$o \in O$ traduit l'existence d'une relation I entre eux: aIo . L'existence de cette relation I entre O et A est matérialisée dans la matrice de contexte C par, soit la valeur «vrai» (et «faux» sinon) soit par une marque quelconque (et rien sinon). Le triplet $K = (O, A, I)$ est appelé un *contexte formel* ou simplement un contexte.

L'*intention* d'un ensemble $X \subset O$ est l'ensemble des attributs possédés conjointement par tous les objet de X et, est donnée par la fonction f :

$$f(X) = \{a \in A \mid \forall o \in X, oIa\}. \quad (3)$$

Inversement l'*extension* d'un ensemble $Y \subset A$ est l'ensemble des objets à posséder conjointement tous les attributs de Y et, est donnée par la fonction g :

$$g(Y) = \{o \in O \mid \forall a \in Y, oIa\}. \quad (4)$$

Le couple (f, g) est appelé une *connexion de Galois*, et un *concept* est tout couple $C = (X, Y) \subset O \times A$, tel que *les objets de X soient les seuls à posséder tous les attributs de Y* , en d'autres termes $X \times Y$ forme, à deux permutations près de O et de A , un rectangle maximal dans C , c'est-à-dire que

$$f(X) = Y \ \& \ g(Y) = X. \quad (5)$$

Pour illustrer cette notion de concept, on peut observer dans la table 1 que l'ensemble $X = \{Antibes, hotel, people\}$ donne un concept car $f(X) = \{tweet_1, tweet_1.1, tweet_1.2, tweet_1.2.1\} = Y$ et $g(Y) = X$, et ce concept est donc $(\{tweet_1, tweet_1.1, tweet_1.2, tweet_1.2.1\}, \{Antibes, hotel, people\})$, alors que l'ensemble $X' = \{tweet_1.1, tweet_1.2\}$ ne donne pas un concept car $f(X') = \{Antibes, hotel, people, @user10\} = Y'$ and $g(Y') = \{tweet_1.1, tweet_1.2, tweet_1.2.1\}$.

Le *treillis de Galois* est le *poset* de concepts L muni de l'ordre partiel suivant \leq :

$$(X_1, Y_1) \leq (X_2, Y_2) \Leftrightarrow X_1 \subseteq X_2 \text{ (or } Y_1 \supseteq Y_2). \quad (6)$$

Le treillis de Galois est noté $T = (L, \leq)$ et, est représenté à l'aide d'un *diagramme de Hasse* comme dans la figure 5. Deux types d'affichage existent pour les labels des concepts, l'affichage plein et l'affichage réduit. Dans l'affichage plein, tous les objets

et attributs d'un concept sont affichés, alors que dans l'affichage réduit, les attributs et les objets ne sont affichés qu'une seule fois, la première fois qu'ils sont rencontrés en parcourant le treillis, en partant du haut pour les attributs, et en partant du bas pour les objets, comme c'est le cas dans la figure 5. Ce type d'affichage permet de réduire considérablement la quantité d'information à afficher, mais c'est au prix d'un parcours visuel du treillis pour retrouver tous les objets et les attributs d'un concept. Ainsi pour trouver le concept recouvert par la sphère affichée *tweet_3* dans la figure 5, il faut parcourir tous les éventuels chemins qui relient ce concept au concept supérieur affiché *Antibes* pour trouver tous les attributs, et parcourir tous les éventuels chemins qui relie ce concept aux concepts les plus bas auquel il est relié dans le treillis, ici *tweet_3.1*. En partant du concept *tweet_3* et allant vers le haut, on trouve successivement les attributs *Marineland*, *plus*, *nouveauté*, *été* et *Antibes*. Et en parcourant le treillis vers le bas on trouve l'objet *tweet_3.1*, ce qui donne au final le concept $(\{tweet_3, tweet_3.1\}, \{Antibes, Marineland, nouveautés, plus, été\})$.

Le calcul du treillis peut être effectué à l'aide, par exemple, de l'algorithme de (Bordat, 1986), qui calcule récursivement tous les concepts à partir du concept $(\emptyset, f(\emptyset))$, en calculant pour chaque concept trouvé l'ensemble de ses concepts-fils. Pour voir une revue des autres algorithmes pour la génération de treillis, voir l'article de (Kuznetsov, Obedkov, 2001) qui effectue aussi une comparaison des performances. Un des avantages non négligeable de la classification basée sur les treillis est que pour une table de contexte donnée le treillis résultant est unique (pas d'instabilité à l'exécution), et il est exhaustif (tous les concepts existants s'y trouvent). Dans le cas qui nous occupe, cette classification va nous permettre de retrouver tous les groupes de mots présents dans un groupe de tweets et de les représenter de façon similaire à la figure 5.

3.3. E-buzz Monitoring

Dans ce qui suit, nous proposons d'analyser un groupe de tweets, afin d'y repérer les termes et groupes de termes les plus tweetés dans le groupe considéré. Pour ce faire nous proposons une démarche en quatre étapes:

1. Récupération des tweets contenant un ou plusieurs mots clés.
2. Nettoyage des tweets (suppression des mots de liaisons, ponctuations,...).
3. Établissement d'une table de contexte où les objets sont les tweets et, les attributs les mots contenus dans ces derniers.
4. Calcul du treillis de Galois correspondant.
5. Visualisation des résultats.

Pour illustrer notre technique nous utiliserons toujours l'exemple de la figure 3. L'application des étapes 1 à 4 ne pose pas de problème sur un treillis de taille modeste, alors que la cinquième, la visualisation des résultats, elle, peut se révéler moins évidente. Dans la figure 6, nous affichons l'ensemble du treillis en n'affichant que les attributs (c'est-à-dire les mots signifiants) et en donnant à chaque concept une surface qui soit proportionnelle au nombre de tweets qu'il contient. On constate que malgré

sa taille très modeste (12 concepts trouvés), il est difficile d'afficher tout les concepts proportionnellement à leur taille et, simultanément, l'ensemble de leurs attributs de façon lisible, et ce même en utilisant l'affichage réduit des attributs. Cela est dû au fait que les attributs de notre treillis sont formés de mots ou groupes de mots plus ou moins longs. Un autre type de visualisation peut donc s'avérer nécessaire pour afficher de manière plus lisibles les concepts trouvés. Nous proposons donc d'afficher

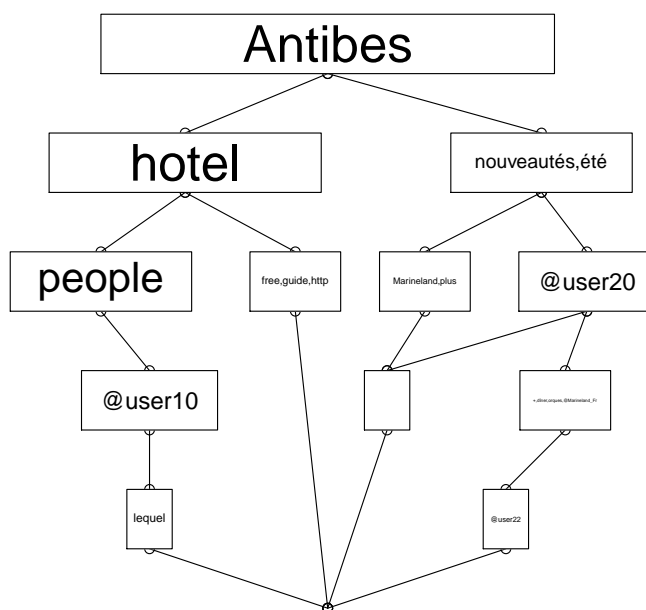


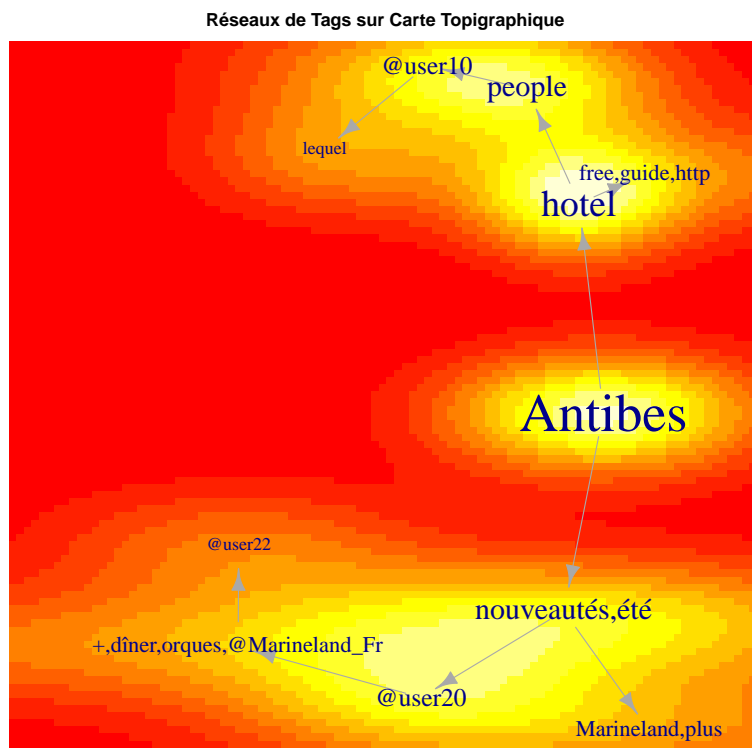
Figure 6. Le treillis de Galois des tweets.

les concepts les plus importants sous forme de nuage de tags, proportionnels et en réseau, pour matérialiser les liaisons entre les différents concepts avec des arrêtes. Ces arrêtes seront dirigées, allant du concept vers ses sous-concepts. Nous effectuons le placement des nœuds à l'aide de la méthode (Fruchterman, Reingold, 1991), car cette méthode optimise l'écartement des nœuds et permet ainsi d'augmenter la lisibilité des tags. Enfin, pour renforcer une lecture allant du plus général au plus particulier, nous avons décidé d'ajouter une allégorie topographique, similairement aux cartes topographiques proposées dans (Fujimura *et al.*, 2008). Pour ce faire, à chaque point du graphique résultant, nous ajoutons un niveau, les différents niveaux étant représentés à l'aide des courbes de niveaux classiques. Pour la mise aux points des niveaux, nous utilisons un mélange de gaussiennes à deux dimensions transformées, auxquelles nous donnons comme centres les coordonnées des centres des tags, et comme écart-

types les largeurs et hauteurs des dits tags. Enfin, pour donner des hauteurs qui soient proportionnelles aux tailles des concepts exactement aux centres des tags, nous normalisons les hauteurs des gaussiennes en les multipliant par les écart-types et ensuite par hauteurs désirées.

Le résultat final peut être vu en figure 7. Dans cette figure, à partir du concept représentant le mot clé de départ *{Antibes}*, on voit que les sous-concepts les plus importants sont *{hotel}* et *{nouveautés, été}*. On voit ensuite que le mot *hotel* amène vers deux concepts *{people}* d'une part et *{free, guide, http:}* d'autre part. Dans cette dernière partie, on constate que les concepts *{@user10}* et *{lequel}* sont de moindre importance. A partir du concept *{nouveautés, été}* on voit que les deux concepts recouvrent la même information, à savoir *Marineland* et plus.

L'idée de cette visualisation est de laisser glisser le regard du lecteur des «sommets» (les concepts les plus généraux) vers les «vallées» (les concepts les plus particuliers). La constructions et les affichages des treillis et des cartes de tags en réseau topographique ont été effectuées dans l'environnement statistique (R Development Core Team, 2010), et pour les parties treillis, à l'aide du package *galois* développé par nos soins.



4. Conclusions et Perspectives

Dans cet article, nous avons évoqué deux des axes d'intégration de réseaux sociaux dans une administration dans le cadre du projet ARSA.

Le premier axe vise la mise en place et l'utilisation d'un réseau social au sein même de l'administration, et ce à l'aide de l'outil SNA de SAP. Le prototype adapté de SNA, proposant des fonctionnalités pour le travail collaboratif et l'exploration du réseau, a été déployé durant le deuxième semestre 2011. Une évaluation de l'outil, basée sur des questionnaires de satisfaction envoyés à un échantillon de 40 employés de la ville d'Antibes, est en cours. Les résultats préliminaires de cet enquête sont très positifs, l'outil étant jugé intuitif, simple d'utilisation et permettant de favoriser la collaboration entre les équipes. Parmi les pistes d'amélioration du prototype demandées par l'utilisateur, nous avons approfondi la problématique de l'exploration visuelle de réseaux sociaux comportant plusieurs types d'entités et relations. La méthode heuristique proposée dans cet article donnant de meilleurs résultats sur des graphes de petite et moyenne taille, nous envisageons de poursuivre nos recherches en la combinant avec des techniques de filtrage de graphes voir, par exemple, (Shen *et al.*, 2006 ; Sun *et al.*, 2009) Une autre piste de recherche importante envisagée est le développement d'algorithmes dynamiques qui, à chaque modification du graphe ou des sous-graphes, recalculeraient la disposition (visuellement) optimale, tout en minimisant la différence entre deux visualisations successives.

Le second axe étudié, également à la demande de l'utilisateur, est une ouverture de l'administration vers les réseaux sociaux externes, non encore à des fins de dialogue directe, mais comme outil de jauge de l'opinion publique à son égard. Ce monitoring de l'opinion a été expérimenté sur le réseau Twitter, à l'aide d'un prototype nommé EVARIST et basé sur les treillis de Galois. Bien qu'EVARIST n'en soit encore à l'état de prototype, un certain nombre d'améliorations sont envisageables et envisagées. La première de ces améliorations c'est l'interactivité et ce, afin de permettre à l'utilisateur de sélectionner le sous-concept qu'il désire développer. On envisage ensuite de «développer» les URL réduites (bit.ly, is.gd, tinyURL...) afin de ne pas comptabiliser sous deux concepts différents une même URL qui aurait été réduite via deux services différents. Dans une prochaine étape, nous prévoyons aussi d'utiliser l'analyse de sentiments pour évaluer la positivité ou négativité des concepts, notions importantes dans le cadre de la surveillance de la réputation électronique.

Bien que chacun des outils proposés soient sujets aux améliorations évoquées ci-dessus, une autre voie naturelle d'évolution est bien entendu l'interconnexion des deux outils. Et ce afin de permettre d'une part l'interaction d'un utilisateur d'un réseau social externe directement, sous la contrainte du respect de droits d'accès, avec un élément du réseau social interne, et d'autre part ce dernier devrait, idéalement, permettre aux services de l'administration d'être informé de toute information voyageant sur le réseau externe les concernant directement. Ce saut qualitatif constitue un axe de recherche privilégié pour le futur d'ARSA.

Remerciements

Les auteurs souhaitent remercier la DGCIS, qui a contribué au financement du projet ARSA, ainsi que la ville d'Antibes et son interlocuteur dans le cadre de ce projet, M. Patrick Duverger.

Bibliographie

- Akcora C. G., Bayir M. A., Demirbas M., Ferhatosmanoglu H. (2010). Identifying breakpoints in public opinion. In *Soma at sigkdd workshop on social media analytics, washington dc, usa*. Washington, DC, USA.
- Barbut M., Monjardet B. (1970). *Ordre et classification, algèbre et combinatoire, tome 2*. Hachette.
- Battista G. di, Eades P., Tamassia R., Tollis I. G. (1999). *Graph drawing: Algorithms for the visualization of graphs*. Upper Saddle River, Prentice Hall.
- Bertault F., Miller M. (1999). An algorithm for drawing compound graphs. In *Proceedings of the 7th international symposium on graph drawing (gd'99)*, p. 197-204.
- Bordat J. (1986). Calcul pratique du treillis de galois d'une correspondance. *Mathématique, Informatique et Sciences Humaines*, vol. 24, n° 94, p. 31-47.
- Boyd D., Golder S., Lotan G. (2010). Tweet, tweet, retweet: Conversational aspects of retweeting on twitter. In *Proceedings of the 43rd hawaii international conference on social systems (hicss)*, p. 1-10.
- Bughin J. (2008). The rise of enterprise 2.0. *Journal of Direct Data and Digital Marketing Practice*, vol. 9, p. 251-259.
- Burch M., Fritz M., Beck F., Diehl S. (2010). Timespidertrees: A novel visual metaphor for dynamic compound graphs. In C. D. Hundhausen, E. Pietriga, P. Díaz, M. B. Rosson (Eds.), *Proceedings of the ieee symposium on visual languages and human-centric computing vl/hcc 2010*, p. 168-175.
- Cao N., Sun J., Lin Y.-R., Gotz D., Liu S., Qu H. (2010). Facetatlas: Multifaceted visualization for rich text corpora. *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, p. 1172-1181.
- Eiglsperger M., Siebenhaller M., Kaufmann M. (2005). An efficient implementation of sugiyama's algorithm for layered graph drawing. *Journal of Graph Algorithms and Applications*, vol. 9, p. 305-325.
- Forster M. (2002). Applying crossing reduction strategies to layered compound graphs. In *Proceedings of the 10th international symposium on graph drawing (gd'02)*, p. 276-284.
- Fruchterman T., Reingold E. (1991). Graph drawing by force-directed placement. *Software - Practice and Experience*, vol. 21, n° 11, p. 1129-1164.
- Fujimura K., Fujimura S., Matsubayashi T., Yamada T., Okuda H. (2008). Topigraphy: visualization for large-scale tag clouds. In *Proceeding of the 17th international conference on world wide web*, p. 1087-1088. New York, NY, USA, ACM. Consulté sur <http://doi.acm.org/10.1145/1367497.1367669>

- Han J., Kamber M. (2006). *Data mining: Concepts and techniques*. San Francisco, Morgan Kaufmann.
- Heer J., Boyd D. (2005). Vizster: Visualizing online social networks. In *Proceedings of the 2005 IEEE Symposium on Information Visualization*, p. 32-39.
- Herman I., Melançon G., Marshall S. (2000). Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, vol. 6, p. 24-43.
- Holten D. (2006). Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, p. 741-748.
- Huang J., Thornton K. M., Efthimiadis E. N. (2010). Conversational tagging in twitter. In *Proceedings of the 21st ACM conference on hypertext and hypermedia (ht)*, p. 173-178. New York, NY, USA.
- Kuznetsov S. O., Obedkov S. A. (2001). Comparing performance of algorithms for generating concept lattices. In *Concept lattices-based theory, methods and tools for knowledge discovery in databases (clkdd'01)*. Stanford, July 30, 2001.
- Kwak H., Lee C., Park H., Moon S. (2010). Haewoon kwak, changhyun lee, hosung park, and sue moon. In *Proceedings of the 19th international conference on world wide web (www '10)*. New York, NY, USA.
- Landesberger T. von, Kuijper A., Schreck T., Kohlhammer J., Wijk J. van, Fekete J.-D. *et al.* (2011). *Visual analysis of large graphs: State-of-the-art and future research challenges*. *Computer Graphics Forum*, vol. 30, p. 1719-1749.
- Lerman K., Ghosh R. (2010). *Information contagion: an empirical study of the spread of news on digg and twitter social networks*. In *Proceedings of 4th international conference on weblogs and social media (icwsm)*.
- Macdonald C., Ounis I., Soboroff I. (2009). *Overview of the trec 2009 blog track*. In *NIST special publication 500-278: The eighteenth text retrieval conference proceedings (trec 2009)*.
- McAfee A. (2006). *Enterprise 2.0: The dawn of emergent collaboration*. *MIT Sloan Management Review*, vol. 47, p. 21-28.
- McAfee A. (2009). *Shattering the myths about enterprise 2.0*. *Harvard Business Review*, vol. 87, p. 1-6.
- Mueller C., Ma K.-L. (2008). *A treemap based method for rapid layout of large graphs*. In *Proceedings of visualization symposium, 2008. pacificvis '08. IEEE Pacific*, p. 231 -238.
- Nieto M. J., Santamaría L. (2007). *The importance of diverse collaborative networks for the novelty of product innovation*. *Technovation*, vol. 27, p. 367-377.
- O'Connor B., Balasubramanyan R., Routledge B. R., Smith N. A. (2010). *From tweets to polls: Linking text sentiment to public opinion time series*. In *Proceedings of the fourth international AAAI conference on weblogs and social media, Washington, DC*.
- Ounis I., Rijke C. M. M. de, Mishne G., Soboroff I. (2006). *Overview of the trec 2006 blog track*. In *NIST special publication 500-272: The fifteenth text retrieval conference proceedings (trec 2006)*, vol. 272, p. 17-31.
- Purchase H. (2000). *Effective information visualization: a study of graph drawing aesthetics and algorithms*. *Interacting with Computers*, vol. 13, p. 147-162.

- Purchase H., Hoggan E., Görg C. (2006). *How important is the “mental map”? an empirical investigation of a dynamic graph layout algorithm.* In Proceedings of 14th international symposium on graph drawing (gd’06), p. 184-195.
- R Development Core Team. (2010). *R: A language and environment for statistical computing Manuel de logiciel.* Vienna, Austria. Consulté sur <http://www.R-project.org> (ISBN 3-900051-07-0)
- Raitner M. (2004). *Visual navigation of compound graphs.* In Proceedings of the 12th international symposium on graph drawing (gd’04), p. 403-413.
- Richter A., Riemer K. (2009). *Corporate social networking sites - modes of use and appropriation through co-evolution.* In Proceedings of the 20th australasian conference on information systems.
- Romero D. M., Meeder B., Kleinberg J. (2011). *Differences in the mechanics of information diffusion across topics: Idioms, political hashtags, and complex contagion on twitter.* In Proceedings of the 20th international conference on world wide web (www). New York, NY, USA.
- Saffrey P., Purchase H. (2008). *The “mental map” versus “static aesthetic” compromise in dynamic graphs: a user study.* In Proceedings of the 9th conference on australasian user interface, p. 85-93.
- Sandhu R., Samarati P. (1994, September). *Access control: Principles and practice.* IEEE Communications Magazine, vol. 32, p. 40-48.
- Shen Z., Ma K.-L., Eliassi-Rad T. (2006). *Visual analysis of large heterogeneous social networks by semantic and structural abstraction.* IEEE Transactions on Visualization and Computer Graphics, vol. 12, p. 1427 -1439.
- Singh J. (2005). *Collaborative networks as determinants of knowledge diffusion patterns.* Management Science, vol. 51, p. 756-770.
- Sugiyama K., Misue K. (1991). *Visualization of structural information: Automatic drawing of compound digraphs.* IEEE Transactions on Systems, Man, and Cybernetics, vol. 21, p. 876-892.
- Sugiyama K., Tagawa S., Toda M. (1981). *Methods for visual understanding of hierarchical system structures.* IEEE Transactions on Systems, Man, and Cybernetics, vol. 11, p. 109-125.
- Sun J., Papadimitriou S., Lin C.-Y., Cao N., Liu S., Qian W. (2009). *Multivis: Content-based social network exploration through multi-way visual analysis.* In Proceedings of the siam international conference on data mining sdm 2009, p. 1063-1074.
- Wattenberg M. (2006). *Visual exploration of multivariate graphs.* In Proceedings of the sigchi conference on human factors in computing systems chi ’06, p. 811–819.
- Wille R. (1980). *Restructuring lattice theory, ordered sets i.* Rival.
- Wille R. (1984). *Line diagrams of hierarchical concept systems.* Int. Classif., vol. 11, p. 77-86.