14e édition des Journées francophones

EGC 2014

Extraction et Gestion des Connaissances

28-31 Janvier

RENNES

IRISA & Centre Inria Rennes - Bretagne Atlantique

Campus de Beaulieu, Rennes

Journées Ateliers/Tutoriels

FDC : Fouille de données complexes

11ème Atelier sur la Fouille de Données Complexes (FDC)

Organisateurs :
Cyril de Runz (CReSTIC, Université de Reims Champagne Ardenne)
Cécile Favre (ERIC, Université Lyon 2)
Germain Forestier (MIPS, Université de Haute-Alsace)

Le groupe de travail "Fouille de Données Complexes"

La onzième édition de l'atelier sur la fouille de données complexes est organisée par le groupe de travail EGC "Fouille de Données Complexes". Ce groupe de travail rassemble une communauté de chercheurs et d'industriels désireux de partager leurs expériences et problématiques dans le domaine de la fouille de données complexes telles que le sont les données non-structurées (ou faiblement), les données obtenues à partir de plusieurs sources d'information ou plus généralement les données spécifiques à certains domaines d'application et nécessitant un processus d'extraction de connaissance sortant des itinéraires usuels de traitement.

Les activités du groupe de travail s'articulent autour de trois champs d'action progressifs :

- l'organisation de journées scientifiques une fois par an (vers le mois de juin) où sont présentés des travaux en cours ou plus simplement des problématiques ouvertes et pendant lesquelles une large place est faite aux doctorants,

- l'organisation de l'atelier "Fouille de Données Complexes" associé à la conférence EGC qui offre une tribune d'expression pour des travaux plus avancés et sélectionnés sur la base d'articles scientifiques par un comité de relecture constitué pour l'occasion,

- la préparation de numéros spéciaux de revue nationale, dans lesquels pourront être publiées les études abouties présentées dans un format long et évaluées plus en profondeur par un comité scientifique. Le troisième numéro spécial est en cours de préparation.

Contenu scientifique de l'atelier

Nous avons reçu cette année 16 propositions, chacune d'elle a été relue par au moins deux évaluateurs. Pour la majorité des propositions nous avons été en mesure de proposer trois rapports d'experts afin d'offrir un processus scientifique constructif aux auteurs. Nous avons retenu 11 propositions en fonction de leur qualité et de l'intérêt des discussions qu'elles pouvaient susciter au sein de l'atelier.

Les articles qui vous sont proposés cette année dans les actes qui suivent explorent une grande variété de complexités, aussi bien dans les données que dans les processus de fouille envisagés.

Remerciements

Les responsables de l'atelier souhaitent remercier vivement toutes les personnes ayant contribué à la tenue de cet atelier. En particulier :

- les auteurs pour la qualité de leurs contributions constituant la base essentielle de discussions fructueuses,
- les membres du comité de programme et plus généralement tous les relecteurs de cet atelier dont le travail d'évaluation était crucial pour assurer la qualité de l'atelier,
- les organisateurs d'EGC 2014 qui ont mis en place l'environnement et les moyens pour la réussite des ateliers.

Nous remercions enfin vivement les présidents : Chantal Reynaud la présidente du comité de programme, Arnaud Martin et René Quiniou les co-présidents du comité d'organisation d'EGC 2014.

Cyril de Runz
CReSTIC
Univ. Reims Champagne Ardenne
cyril.de-runz@univ-reims.fr

Cécile Favre
ERIC
Univ. Lyon 2
cecile.favre@univ-lyon2.fr

Germain Forestier
MIPS
Univ. de Haute-Alsace
germain.forestier@uha.fr

Membres du comité de lecture

Le Comité de Lecture est constitué de :

- Hanane Azzag, Université Paris 13
- Alexandre Blansche, Université de Lorraine
- Omar Boussaid, Université Lyon 2
- Guillaume Cleuziou, Université d'Orléans
- Cyril De Runz, Université de Reims Champagne-Ardenne
- Mounir Dhibi, Institut Supérieur des Sciences Appliquées et de Technologie de Gafsa - Tunisie (relecteur additionnel)
- Sami Faiz, Université de Jendouba - Tunisie
- Cécile Favre, Université Lyon 2
- Germain Forestier, Université de Haute-Alsace
- Pierre Gançarski, Université de Strasbourg
- Michel Herbin, Université de Reims Champagne-Ardenne
- Mehdi Kaytoue, INSA de Lyon
- Camille Kurtz, Université Paris Descartes
- Mustapha Lebbah, Université Paris 13
- Arnaud Martin, Université Rennes 1
- Florent Masséglia, INRIA
- Yoann Pitarch, Université Toulouse 1
- Mathieu Roche, CIRAD
- Anna Stavrianou, XEROX
- Abdelmalek Toumi, ENSTA Bretagne
- Cédric Wemmert, Université de Strasbourg
- Djamel Zighed, Université Lyon 2

TABLE DES MATIÈRES

Partie 1

Partie 2

Partie 3

Partie 4

Méthodologie 3-way d'extraction d'un modèle articulatoire de la parole à partir des données d'un locuteur.

Martine Cadot*, Yves laprie**

*LORIA
martine.cadot@loria.fr,
http://www.loria.fr/ cadot
**LORIA
yves.laprie@loria.fr,
http://www.loria.fr/ laprie

Résumé. Pour parler, le locuteur met en mouvement un ensemble complexe d'articulateurs : la mâchoire qu'il ouvre plus ou moins, la langue à laquelle il fait prendre de nombreuses formes et positions, les lèvres qui lui permettent de laisser l'air s'échapper plus ou moins brutalement, etc. Le modèle articulatoire le plus connu est celui de Maeda (1990), obtenu à partir d'Analyses en Composantes Principales faites sur les tableaux de coordonnées des points des articulateurs d'un locuteur en train de parler. Nous proposons ici une analyse 3-way du même type de données, après leur transformation en tableaux de distances. Nous validons notre modèle par la prédiction des sons prononcés, qui s'avère presque aussi bonne que celle du modèle acoustique, et même meilleure quand on prend en compte la co-articulation.

1 Introduction

Construire un modèle articulatoire de la parole, c'est être capable d'indiquer les mouvements des articulateurs (mâchoires, lèvres, etc.) à l'origine de celle-ci (voir figure 1). Des applications pratiques d'un tel modèle ont déjà été mises en œuvre par les enseignants/chercheurs de l'équipe Parole du Loria, par exemple, pointer pour les étudiants en "Français Langue É trangère" les les articulateurs en jeu lors de la prononciation des sons, doubler les enregistrements vidéo pour les malentendants par une "tête parlante" plus réaliste.

Nous exposons dans cet article comment nous avons extrait un modèle articulatoire à partir de données recueillies auprès d'un locuteur. Ce travail se situe dans la lignée des travaux initiés par Maeda (1990). Il a construit son modèle articulatoire (voir figure 2) au moyen d'analyses en composantes principales sur des données de même type. Puis il l'a évalué de façon acoustique en comparant les sons réels aux sons produits par un synthétiseur de sons piloté par son modèle. La nouveauté de notre démarche consiste en l'utilisation d'une méthode d'analyse 3-way pour extraire le modèle, et de méthodes d'apprentissage supervisé pour le valider. Notre évaluation se fait en comparant de façon phonétique les sons prédits aux sons réels (pour un exemple de sons transcrits en phonétique, voir le tableau 1). L'acoustique intervient de surcroît

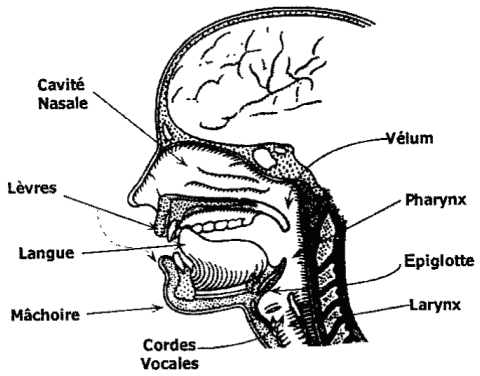


FIG. 1 – *Schéma de l'anatomie du conduit vocal (d'après Flanagran 1972).*

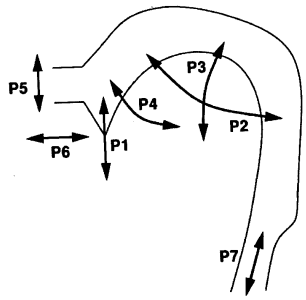


FIG. 2 – *Modèle articulatoire à 7 paramètres de Maeda.*

dans notre évaluation car nous mettons en parallèle les performances de notre modèle et celles du modèle acoustique formé des coefficients cepstraux.
La démarche relatée dans cet article complète et enrichit celle d'un travail précédent de Busset et Cadot (2013). Nous avions alors un corpus de taille inférieure, avec peu de sons différents, et une certaine répétition des phrases. Le modèle étant de petite taille, nous l'avons validé de façon experte, en interprétant un à un ses éléments. Ce premier essai étant probant, nous sommes passés à l'échelle supérieure avec des données plus riches et un modèle plus important, validé de façon automatique.

Notre exposé comporte quatre parties. Nous décrivons dans la première partie la construction du jeu de données numériques, dans la deuxième partie la méthodologie d'extraction du modèle articulatoire que nous avons choisie, dans la troisième l'évaluation par apprentissage automatique de ce modèle, et nous faisons le bilan dans la dernière.

2 Description et signification des données

Les données sont recueillies dans le but de construire un modèle, nous décrivons donc dans une première sous-section le type de modèle que nous visons. Dans la deuxième sous-section, nous exposons le recueil des données, et dans la dernière sous-section comment elles ont été transformées en les données numériques que nous avons traitées.

2.1 Motivation du recueil des données

Dans le modèle articulatoire de la parole construit par Maeda (voir figure 2), le conduit vocal, zone interne allant de l'arrière de la gorge aux lèvres, est schématisé en coupe sagittale, ainsi que les déformations que lui font subir les articulateurs. Elles sont résumées en 7 mouvements, qui sont les 7 paramètres du modèle de Maeda : P1, la mâchoire qui va de haut en bas, P2, P3, P4 la langue qui se déforme dans 3 directions, P5 et P6 les lèvres qui s'ouvrent et se ferment, s'avancent et reculent, et P7 pour le mouvement du larynx. En affectant différentes valeurs à ces 7 paramètres, on obtient différentes formes du conduit vocal, dont on déduit différents sons à l'aide du synthétiseur. Ce modèle est une représentation réaliste de la parole car ce sont les déformations du conduit vocal qui modulent l'air venu des poumons et produisent les différents sons de la parole. Toutefois, avec ses 7 paramètres, c'est une simplification forte de la réalité, et certains sons sont plus difficiles à simuler ainsi que d'autres.

2.2 La vidéo à l'origine des données

Une série de radiographies de la tête a été réalisée pendant qu'un locuteur prononçait quelques phrases courtes, recopiées dans le tableau 1 (pour plus de détail sur ces données, se reporter à Sock et al. (2011)). Puis des contours ont été dessinés sur ces images afin de représenter au mieux les articulateurs (voir figure 1), et de repérer le plus finement possible leurs mouvements.

On dispose aussi de la correspondance entre sons et images. Les 27 symboles suivants ont été choisis pour représenter les différents sons présents dans les données, selon le tableau 1 :

@ 9 A ã b d e E f g i j k l m n o õ p R s S t u w z Z

Ces sons ne sont pas utilisés dans la phase de construction du modèle, mais dans la phase d'évaluation.

2.3 L'obtention des données numériques

La qualité du modèle extrait des données dépend de la qualité des données elles-mêmes. Le repérage des contours des articulateurs s'est appuyé sur toute une série d'outils mis en oeuvre au sein de l'équipe Parole du LORIA. Cette méthodologie d'annotation semi-automatique (voir Laprie et Busset (2011)), a débouché sur plusieurs méthodes qui ont été appliquées à une première bande vidéo, puis évaluées et comparées de diverses façons avant d'être testées à l'aide d'un synthétiseur proche de celui de Maeda (voir la thèse de Busset (2013) pour plus de détails). Ces méthodes ont été appliquées sur une deuxième bande vidéo, plus riche, et les données produites ont été retravaillées lors du stage de Clément (2013). Ce sont ces données, de grande qualité, que nous avons utilisées ici.

Transcription orthographique	Transcription acoustique
Il a pas mal.	ilApAmAl@
Les attablés.	lezAtAble
Très acariâtre.	tREzAkARijAt
Il zappe pas mal.	ilzApAmAl@
Des abat-jour.	dezAbAZuR
Il l'a datée.	ilAdAte
Crabe bagarreur.	kRAbAgAR9R
Trois sacs carrés.	tRwsAkARe
Pas de date précise.	pAdAtpResiz9
Blague garantie.	blAgARãti
Nous palissons.	nupAlisõ
Il a pourri.	ilApuRi
Couds ta chemise.	kutASmiz@
Elle a tout faux.	ElAtufo
Pour tout casser.	puRtukAse

TAB. 1 – *Les 15 phrases successives prononcées par le locuteur et leur transcription acoustique*

Elles se présentent sous la forme de 11 contours formés d'un nombre fixe de points pour les uns, et variable pour les autres. On dispose pour chaque image des coordonnées 2D des points de chaque contour, comme indiqué dans le tableau 2.

3 Extraction du modèle articulatoire

Dans cette section, nous exposons d'abord comment Maeda a utilisé des ACP pour l'extraction de son modèle articulatoire, et les inconvénients de ce type d'analyse. Nous détaillons ensuite la méthode factorielle 3-way, que nous avons choisi d'utiliser, et enfin le traitement des données.

3.1 Utilisation d'Analyses en Composantes Principales

Pour extraire son modèle articulatoire, Maeda (1990) a utilisé des ACP à partir de données similaires. Par exemple, pour obtenir les 3 paramètres de la langue, P2, P3 et P4 (voir figure 2), le contour de la langue a été repéré sur chaque radiographie par une centaine de points. Puis les coordonnées des points ont été disposées séquentiellement dans un tableau ayant autant de lignes que d'images, et les trois premières composantes d'une ACP ont donné les paramètres recherchés. De nombreuses variantes de ce modèle ont été proposées par la suite, portant essentiellement sur la création des tableaux de données soumis à des ACP. Par exemple, Laprie et Busset (2011) ont utilisé une grille polaire adaptative ainsi que des coordonnées curvilignes pour mieux placer les points de la langue qui ont formé un premier tableau soumis à une ACP. Et avant de procéder à des ACP sur les tableaux de données des articulateurs suivants, ils les ont nettoyés de leur liens avec les articulateurs précédents par soustraction des corrélations.

déformable	nb min	nb max	indéformable	nb de points
voile du palais	69	107	os hyoide	30
epiglotte	52	74	plancher de la langue	19
larynx	46	79	palais	39
lèvre supérieure	11	35	machoîre inférieure	50
lèvre inférieure	13	45	machoîre supérieure	23
langue	18	44		

TAB. 2 – *Nombre de points des 11 contours dessinés dans les 1021 images : à gauche les articulateurs déformables, et à droite les indéformables*

Les ACP ont montré leur efficacité dans la construction du modèle articulatoire, mais aussi leurs limites. Il s'est avéré difficile d'extraire de nombreuses composantes d'un seul tableau de données : quand il contenait les points d'un seul articulateur, on atteignait la quasi-totalité de la variance expliquée avec 1, 2 et au maximum 3 composantes, et regrouper tous les points dans un même tableau ne permettait pas de dépasser les 7 composantes du modèle princeps. De plus à l'examen des contributions des points aux axes, on a constaté que les abscisses et les ordonnées d'un nombre non négligeable de points se retrouvaient sur des axes différents, ce qui rendait délicate leur interprétation. Ces problèmes sont inhérents à la méthode d'analyse choisie et nous ont conduits à en chercher une plus adaptée à la fois aux données et au type de modèle recherché.

3.2 Méthode de 3-way MDS

Le MDS (MultiDimensional Scaling, et en français "positionnement multidimensionnel") fait partie des méthodes d'analyses factorielles des données, et est particulièrement adapté à l'analyse des données de type *dissimilarités* [1], non mesurables objectivement, correspondant à des impressions ressenties, recueillies par des questions comme celle-ci :

Du point de vue de l'acidité, quelle distance ressentez-vous entre :
- *la limonade A et la limonade B ?*
- *la limonade B et la limonade C ?*
- *la limonade A et la limonade C ?*

Le MDS a fait l'objet de nombreux articles et ouvrages. Pour plus de détails sur le MDS, nous renvoyons les lecteurs intéressés à Borg et Groenen (1997).

Les principes du MDS. La méthode d'analyse MDS est capable de positionner des objets dans un espace de dimension p de telle sorte que leurs distances deux à deux soient les plus proches possibles de leurs dissimilarités initiales. L'écart entre les deux tableaux de distances est appelé STRESS, et l'ajustement du modèle aux données est d'autant meilleur qu'il est proche de zéro.

La formulation mathématique à la base des MDS est la suivante : si pour deux objets numérotés par i et j, on note δ_{ij} leur dissimilarité initiale, d_{ij} leur distance dans l'espace

1. On appelle *dissimilarité* une *distance affaiblie*, notamment elle n'est pas astreinte à vérifier l'*inégalité triangulaire* qui impose pour tout triplet de points (x, y, z) la relation $d(x, z) \leq d(x, y) + d(y, z)$.

euclidien de dimension p, et f une fonction monotone de ces dissimilarités, le *stress* brut est donné par la formule

$$Stress = \sum_{1 \leq i < j \leq n} (f(\delta_{ij}) - d_{ij})^2$$

C'est par le choix de la fonction f que le *stress* est minimisé.

La mise en oeuvre du MDS. Au départ de l'algorithme, les points sont placés dans une position quelconque de l'espace de dimension p, et l'algorithme consiste à les déplacer un à un pour faire diminuer le *stress*, jusqu'au moment où sa valeur est jugée suffisamment petite. Comme pour l'ACP, le nombre de dimensions p doit être fixé au départ par l'utilisateur. L'exemple traditionnel de cette méthode est son application à un tableau des distances en kilomètres entre des paires de villes. La méthode produit une carte 2D qui s'avère assez proche de la réalité, mis à part quelques distorsions liées au relief montagneux.

Dans notre cas, le fait d'utiliser les distances entre points au lieu de leurs coordonnées permet d'éviter que les abscisses et ordonnées d'un même point ne se retrouvent sur deux axes factoriels différents.

Les differents types de 3-way. Le MDS a été étendu pour pouvoir prendre en considération plusieurs tableaux de dissimilarités au lieu d'un seul. Ce qui est le cas par exemple si on obtient autant de tableaux de dissimilarités que de sujets d'un groupe de q sujets. Aux deux dimensions du tableau s'ajoute une troisième dimension, qui est le sujet, d'où le nom 3-way MDS, donné à la méthode MDS permettant de le traiter en prenant en compte les différences entre sujets. INDSCAL (pour INdividual Difference SCALing) est une des méthodes développées par Carroll et Chang (1970) pour traiter ce type de données. Sa méthode se situe entre deux variantes 3-way extrêmes du MDS (voir Carroll, 1972, page 106) :

- "identically" ; même espace X de dimension p pour chaque sujet k avec une fonction f_k différente pour chacun,
- "idiosyncratic" ; un espace X_k propre à chaque sujet k, avec la même fonction f pour tous.

Pour le détail de ces méthodes de 3-way MDS, nous renvoyons les lecteurs interessés au chapitre 21 de Borg et Groenen (1997).

3.3 Le traitement des données

Compte tenu de la taille des données à traiter nous avons choisi le logiciel R à toutes les étapes de ce travail : création des tableaux de données en entrée, création de tableaux de résultats en sortie, traitement des données "cepstrales" et par 3-way MDS, évaluation quantitative des résultats par arbres de décision et M-SVM. Pour l'examen des données d'entrée comme de sortie (parcours des tableaux de données pour contrôle des valeurs erronées, croisement de variables et graphiques), nous avons utilisé le tableur Excel pour ses possibilités de manipulations interactives.

La création du tableau. Nous n'avons pris qu'un nombre réduit de points par articulateur, 3 pour les articulateurs indéformables sauf le palais qui en a 4, 3 pour chaque lèvre, 4 pour l'épiglotte, 5 pour le velum comme pour le larynx et 10 pour la langue. Pour réaliser cela, les

contours de chaque articulateur ont été découpés en autant de parties que de points souhaités, en veillant à découper plus finement les parties les plus déformables, ou susceptibles de contacts. Par exemple pour la langue, les zones autour de l'apex (pointe), de la racine et du dos ont des points plus rapprochés que le reste de la langue. Puis on a calculé dans chaque partie la moyenne des coordonnées des points, ce qui a donné les coordonnées du point cherché. Nous sommes arrivés ainsi à un nombre de 46 points pour 11 articulateurs des 1021 images ayant été annotées. On a ensuite calculé pour chaque point la distance euclidienne entre ses positions sur deux images, ce qui a donné un tableau d'un demi-million environ de distances (en ne considérant que la moitié du tableau, car il est symétrique). Ce sont ces 46 tableaux de distances que nous avons analysés avec la méthode de 3-way MDS.

La méthode 3-way MDS. Nous avons choisi la fonction *smacofIndDiff* du package SMACOF (de Leeuw et Mair, 2009), avec le paramètre "idioscal" correspondant à la variante "idiosyncratic" décrite dans le troisième paragraphe de la section 3.2, qui s'est avérée moins gourmande en mémoire vive que la méthode INDSCAL que nous avions utilisée avec succès pour des données 5 fois plus petites dans Busset et Cadot (2013). Et nous nous sommes limités à 200 itérations afin de réduire le temps d'exécution. Malgré cela, nous avons dû rapidement migrer d'un ordinateur 32 bits, double-processeur, 2Gio de RAM vers un ordinateur 64 bits, quadri-processeur, 8 Gio de RAM. Nous avons pu ainsi obtenir les positions des quelque 1000 images dans des espaces allant de 2 dimensions à 18. Pour obtenir les 18 dimensions, les 200 itérations demandées ont duré plus de 48 heures.
Nous appelons "modèles articulatoires" les matrices formées de 1021 lignes (une par image) et q colonnes représentant les coordonnées de ces images dans l'espace à q dimensions (q allant de 2 à 18).

4 Évaluation de notre modèle

Nous exposons maintenant les mesures de qualité de ces représentations dans une seule direction, celle de la discrimination du son. Notre indice de qualité est la proportion de sons correctement identifiés, que nous appelons taux de reconnaissance. Pour pouvoir apprécier le taux de reconnaissance du modèle articulatoire, nous le comparons à celui du modèle acoustique. Chaque évaluation sera donc faite successivement avec les deux modèles. Elle se fera d'abord de façon asynchrone, puis en prenant en compte des décalages temporels.

4.1 Extraction du modèle acoustique

Les coefficients *cesptraux*, obtenus par transformée de Fourier inverse du spectre de parole, fournissent un modèle acoustique centré sur le conduit vocal, largement utilisé en traitement automatique de la parole (Busset, 2013). Nous avons extraits les 20 premiers coefficients cepstraux avec le package tuneR (Ligges, 2011) à partir de l'enregistrement audio réalisé pendant que les radios étaient prises. C'est la matrice obtenue de 2119 lignes et 20 colonnes qui représente le "modèle acoustique" que nous souhaitons comparer aux "modèles articulatoires" formés des 1021 images et q colonnes pour q allant de 2 à 18.

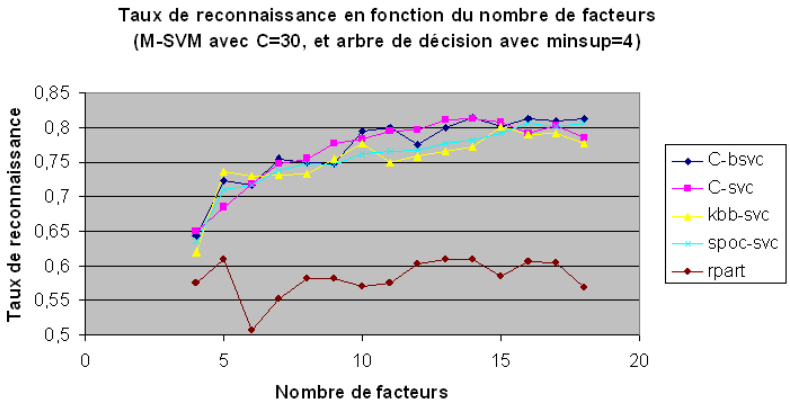


FIG. 3 – *Taux de reconnaissance en fonction du nombre de facteurs.*

4.2 Mesure asynchrone de la qualité de reconnaissance des sons

Transformation des données pour la mesure. Pour le modèle articulatoire, nous avons adjoint aux matrices MDS à q dimensions une colonne avec les sons correspondants à chaque image. Puis nous avons retiré toutes les lignes correspondant à des silences ainsi que celles correspondant à des sons très rares dans les données (w, j et n) c'est-à-dire présents 6 fois ou moins. Les matrices n'ont plus que 732 lignes et la colonne sons n'en a plus que 24 différents.

Pour le modèle acoustique, nous avons également adjoint la colonne de sons correspondants. Nous avons ensuite retiré les lignes de silence et celles des 3 sons retirés dans les matrices de facteurs MDS, afin de pouvoir mieux comparer les capacités de discrimination de sons des deux modèles. La matrice finale de coefficients cepstraux contenant 1450 lignes, nous en avons fait une version réduite en retirant environ une ligne sur deux [2] afin d'avoir la même distribution de sons que la matrice de facteurs MDS. Nous avons ainsi obtenu une deuxième matrice avec seulement 732 lignes.

Les mesures d'évaluation utilisées. Nous avons utilisé le package Rpart (**?**) pour obtenir des arbres de décision comme définis par Quinlan (1986), qui ont l'avantage de fournir des règles explicites de prédiction.

Nous avons complété notre étude en utilisant des méthodes de discrimination par SVM (Support Vector Machine) présentes dans le package KernLab (Karatzoglou et al., 2004). Les SVM sont une méthode de discrimination entre 2 classes. Pour discriminer les 24 sons, nous avons utilisé les M-SVM (*M-* pour multiples) qui en sont une extension à plus de 2 classes. Il en existe plusieurs variantes, décrites dans **?**. Dans ce package, nous avons pu utiliser les 4 types de M-SVM suivants :

2. Nous avons créé non pas une mais deux matrices de 732 lignes à partir de celle de 1450 lignes, obtenues en retirant différemment une ligne sur deux environ : l'une en retirant plutôt la première ligne après le changement de sons, et l'autre en retirant plutôt la deuxième ligne après le changement de sons. Leurs résultats s'étant avérés très proches, nous n'avons transcrit ici que les résultats de la première version.

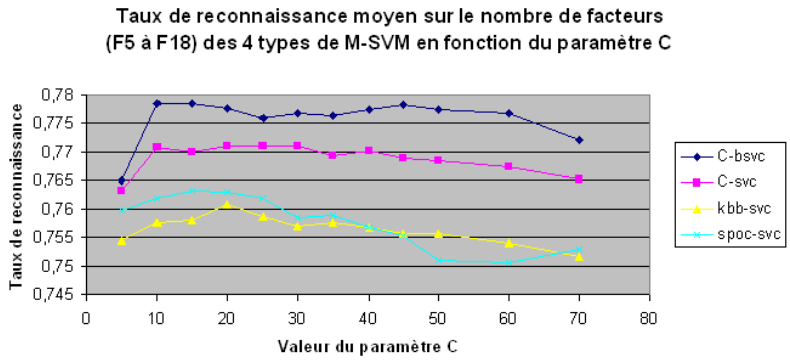


FIG. 4 – *Taux de reconnaissance en fonction de C selon 4 types de M-SVM.*

– C-svc C classification
– C-bsvc bound-constraint svm classification
– spoc-svc Crammer, Singer, native multi-class [3]
– kbb-svc Weston, Watkins, native multi-class

Parmi les options, nous avons choisi la plus courante, qui est le noyau gaussien (option *rbfdot*) et la possibilité de ne saisir qu'un seul paramètre, C, que nous avons fait varier entre 1 et 100. Pour l'apprentissage, nous avons découpé au hasard les lignes de données en quatre parties de tailles équivalentes, l'*ensemble d'entrainement* correspondant à trois de ces parties, et *l'ensemble test* à la quatrième, et nous avons mis dans une colonne les sons prédits pour chaque ligne de la partie test. La colonne de sons prédits a été remplie au bout des 4 itérations pendant lesquelles chaque partie est devenue à son tour l'ensemble test.

Dans le graphique de la figure 3, on a représenté les 5 méthodes choisies pour les matrices de facteurs MDS, avec un nombre de facteurs allant de 4 à 18. On voit que la méthode par arbre de décision n'est pas très stable. L'examen des valeurs prédites montre que tous les sons ne sont pas prédits, le nombre de sons prédits augmentant avec le nombre de facteurs, sans jamais atteindre 24, qui est le nombre total de sons. Parmi les méthodes M-SVM, ce sont les "faux" M-SVM, c'est-à-dire utilisant des stratégies de type "une classe contre toutes les autres" qui donnent les meilleures prédictions, non seulement pour C=30, mais aussi pour les autres valeurs de C, comme l'établit la figure 4. L'examen des logs d'apprentissage sur les 4 sous-ensembles montre que les chutes de performances portent parfois sur un seul ensemble test.

On peut conclure que le taux de reconnaissance croît quand le nombre de facteurs MDS passe de 4 à 14, puis il stagne autour de 0,81 pour plus de 14 facteurs.

Les résultats. Les mêmes méthodes appliquées aux matrices de données cepstrales donnent des résultats de même nature que ceux que nous venons de décrire. Par rapport aux matrices MDS, les taux de reconnaissance sont dans l'ensemble moins bons (maximum 0,76, pour un nombre de coefficients compris entre 16 et 20) avec les matrices cepstrales de taille 732, mais

3. Pour les non "native multi-class", chaque classe est comparée à l'ensemble de toutes les autres.

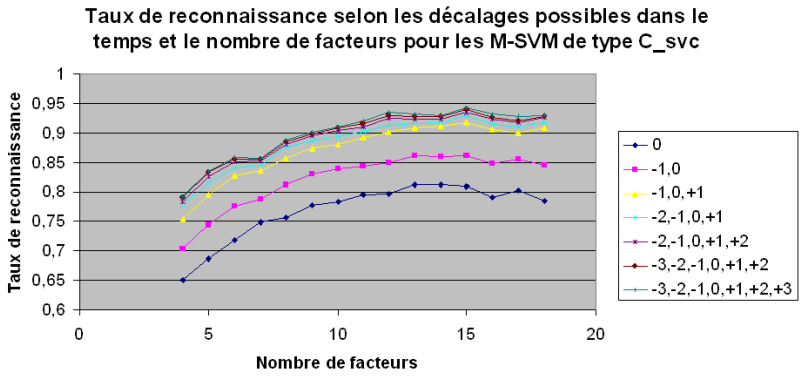


FIG. 5 – *Taux de reconnaissance MDS avec décalages pour M-SVM type C-bcsv.*

meilleurs (maximum 0,86 pour 20 coefficients) avec celles de taille 1450 (voir lignes en traits pleins, figure 6).

4.3 Prise en compte des proximités temporelles

Jusqu'ici, l'aspect temporel n'a pas été pris en compte, pas plus que la co-articulation. Nous le faisons ici en opérant des décalages dans la reconnaissance : si le son prédit pour une image correspond au son de l'image précédente, le décalage est noté -1, s'il correspond à celui de 2 images plus loin, il est noté +2. Et pour un ensemble de décalage donné, par exemple (-2, -1, 0, 1), on juge que le son prédit est juste s'il est le même qu'attendu pour la même image, pour l'image précédente, ou celle encore avant, ou pour l'image suivante. On voit dans la figure 5 que le taux augmente ainsi jusqu'à plus de 0,94, soit une amélioration de 0,13 quand on autorise des décalages allant jusqu'à 3 images avant ou après.

Ce phénomène se retrouve pour les matrices cepstrales, mais avec une ampleur moindre. Dans la figure 6, les scores sans décalage ont été représentés par un trait plein, et un trait en pointillés représente les décalages de -3 à 3. On voit que les décalages font gagner moins de 0,07 en moyenne, que ce soit pour les matrices de 732 lignes (en bleu) ou celles de 1450 lignes (en rouge). Ce qui peut s'expliquer par le fait que le phénomère de résilience des sons doit être moins prégnant que celui de coarticulation.

Au final, en prenant en compte les décalages, le modèle articulatoire prédit légèrement mieux les sons que le modèle acoustique.

5 Discussion, conclusion, perspectives

Nous venons d'exposer comment nous avons utilisé une méthode 3-way MDS dans le but d'extraire un modèle articulatoire performant des données dont nous disposons. Nous avons rencontré un certain nombre de difficultés lors de l'application de cette méthode, nous les listons ici :

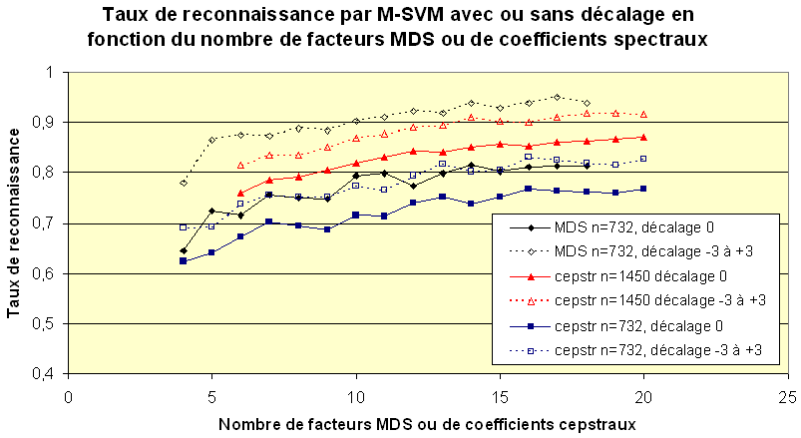


FIG. 6 – *Taux de reconnaissance avec décalages de M-SVM type C-bcsv.*

1. le choix du nombre de points à prendre par image pour que les programmes tiennent en mémoire, sans trop de perte d'informations sur les mouvements des articulateurs,
2. la difficulté d'interprétation des dimensions MDS, due au remplacement de INDSCAL par IDIOSCAL pour les mêmes raisons de place insuffisante en mémoire,
3. la distribution déséquilibrée des sons qui gêne le fonctionnement de certains discriminateurs,
4. le choix un peu artificiel de sons séparés pour la reconnaissance d'un modèle articulatoire : prononcer "la" est-il équivalent à prononcer "l" puis "a" ?
5. Où placer la prise en compte de l'aspect temporel dans l'analyse : dans la construction du modèle ou dans son évaluation ?

Malgré ces difficultés nous arrivons à un modèle articulatoire qui contient autant d'informations sur les sons que le modèle acoustique. Ces bons résultats nous invitent à continuer dans cette voie, en tentant d'améliorer dans différentes directions :

– revoir la programmation des fonctions R utilisées pour solutionner les points 1 et 2, en prenant plus de points par image, et en réutilisant INDSCAL au lieu d'IDIOSCAL,
– essayer de changer de discrimination pour répondre aux points 3 et 4,
– essayer d'incorporer les dépendances temporelles dans le modèle 3-way MDS pour le point 5.

Références

Borg, I. et P. Groenen (1997). *Modern Muldtidimensional Scaling*. Springer series in Statistics. New York: Springer-Verlag.

Busset, J. (2013). *Inversion acoustique articulatoire a partir de coeffcients cepstraux*. Thèse de doctorat, Université de Lorraine.

Busset, J. et M. Cadot (2013). Fouille d'images animées : cinéradiographies d'un locuteur. In *Atelier FOuille de données Spatio-Temporelles et Applications - FOSTA*, Toulouse, France, pp. 1–12.

Carroll, D. (1972). Individual differences and multidimensional scaling. In R. Shepard, A. Romney, et S. Nerlove (Eds.), *Multidimensional Scaling*, Volume 1: Theory, pp. 105–155. Seminar Press.

Carroll, D. et J. Chang (1970). Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition. *Psychometrica 35*, 283–319.

Clément, R. (2013). Construction de modèles articulatoires du conduit vocal pour la production de la parole. Rapport de stage, Université de Lorraine, Master Ingénierie de la Mesure et de l'Image, Spécialité Mesure, Performance et Certifcation.

de Leeuw, J. et P. Mair (2009). Multidimensional scaling using majorization : SMACOF in R. *Journal of Statistical Software 31*(3), 1–30.

Karatzoglou, A., A. Smola, K. Hornik, et A. Zeileis (2004). kernlab – an S4 package for kernel methods in R. *Journal of Statistical Software 11*(9), 1–20.

Laprie, Y. et J. Busset (2011). A curvilinear tongue articulatory model. In *International Seminar on Speech Production 2011 - ISSP'11*, Montréal, Canada.

Ligges, U. (2011). tuneR: Analysis of music. Technical report, Department of Statistics, University of Dortmund, Germany.

Maeda, S. (1990). Compensatory articulation during speech : Evidence from the analysis and synthesis of vocal-tract shapes using an articulatory model. Volume 4, pp. 131–149. Kluwer Academic Publisher, Amsterdam.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning 1*, 81–106.

Sock, R., F. Hirsch, Y. Laprie, P. Perrier, B. Vaxelaire, G. Brock, F. Bouarourou, C. Fauth, V. Ferbach-Hecker, L. Ma, J. Busset, et J. Sturm (2011). An X-ray database, tools and procedures for the study of speech production. In *Proceedings of the 9th International Seminar on Speech Production (ISSP2011)*, Montréal, Canada, pp. 41–48.

Summary

For speaking, a speaker sets in motion a complex set of articulators: the jaw that opens more or less, the tongue which takes many shapes and positions, the lips that allow him to leave the air escaping more or less abruptly, etc.. The best-known articulary model is the one of Maeda (1990), derived from Principal Component Analysis made on arrays of coordinates of points of the articulators of a speaker talking. We propose a 3-way analysis of the same data type, after converting tables into distances. We validate our model by predicting spoken sounds, which prediction proved almost as good as the acoustic model, and even better when co-articulation is taken into account.

Un système d'exploration de données médicales complexes pour la recherche des cas "typiques" et "atypiques"

Afshan Nourizadeh[*,**,***], Amine Ait-Younes[*], Frédéric Blanchard[*],
Brigitte Delemer[***] et Michel Herbin[*]

[*]Laboratoire CReSTIC (EA 3804),
IUT RCC Chemin des rouliers,
51687 REIMS Cedex 2
[**]AXON'CABLE SAS,
2 route de Châlons-en-Champagne
51210 Montmirail - France
[***]Service Endocrinologie-Diabétologie-Nutrition,
CHU de Reims - Hôpital Robert Debré
Avenue du Général Koenig 51092 Reims Cedex

Contact : Michel Herbin,
Université de Reims Champagne-Ardenne,
IUT, chaussée du port,
51000 Châlons-en-Champagne
michel.herbin@univ-reims.fr
http://crestic.univ-reims.fr/

Résumé. L'analyse de données nécessite une phase exploratoire pour effectuer des rapprochements et extraire les informations pertinentes. Dans le domaine médical où les cas se traitent de façon individualisée, sans exclure les cas rares ou atypiques, la modélisation thérapeutique est souvent complexe. Dans cette communication, nous proposons une approche par graphes adaptée à l'exploration de ces données médicales hétérogènes. Dans une première étape, nous fuzzifions les données multidimensionnelles qualitatives ou quantitatives. Par une opération d'agrégation, nous définissons les indices de similarités entre données. Ensuite un indice de représentativité dans l'échantillon est obtenu par une nouvelle agrégation portant cette fois sur les similarités. Enfin, le voisinage de chaque donnée étant défini par α-coupes, nous connectons la donnée à son voisin le plus représentatif pour construire un graphe dépendant du seuil α utilisé. Ceci nous permet de faire apparaitre les cas "typiques" et "atypiques" ou cas rares. Cette approche est appliquée lors d'une étude sur l'insulinothérapie du diabète de type 2 des personnes âgées.

1 Introduction

En médecine, la décision thérapeutique se base sur la connaissance des experts et l'expérience acquise à travers des cas cliniques réels. Chaque nouveau cas est comparé aux éléments d'une base de cas réels connus ou appris. Les éléments les plus similaires sont alors pris en compte pour estimer ou déduire un traitement thérapeutique individualisé et adapté au nouveau cas. Cette approche de la modélisation de l'expérience des experts par un raisonnement à base de cas Proceeding of ICCBR (2012), Marling et al. (2009) est un outil pragmatique pour résoudre des problèmes complexes d'aide à la décision thérapeutique. Avant la décision, l'analyse des données passe par une phase exploratoire pour déterminer les cas les plus "typiques" et les cas rares ou "atypiques" qui vont constituer notre base de cas pour permettre ensuite de déduire le traitement le plus adapté.

Les méthodes classiques reposent sur les techniques de clustering qui vont extraire des prototypes généralement "typiques" de l'échantillon des données. Hélas, ces approches classiques souffrent de nombreux inconvénients :

- les informations recueillies sont parfois incomplètes,
- les données aberrantes et le nombre de cas atypiques ou rares perturbent les résultats,
- l'appartenance d'un cas à son groupe ou cluster n'est pas certaine,
- le nombre de clusters est généralement inconnu,
- le mécanisme de classification dépend étroitement du classifieur utilisé,
- un effectif minimal est nécessaire pour permettre la prédiction et la décision thérapeutique.

Dans nos travaux précédents Nourizadeh et al. (2013a), nous avions utilisé quelques unes des méthodes classiques d'analyse exploratoire et nous avons précisé les contraintes de ces méthodes dans ce contexte strictement exploratoire. Dans cette communication, nous proposons une méthodologie nouvelle pour essayer de s'affranchir de ces contraintes.

Dans ce contexte, la logique floue est souvent mise à contribution pour gérer l'incertitude et l'imprécision. Dans cette communication, elle nous permet aussi une évaluation plus flexible des similarités entre données de l'échantillon (Nourizadeh et al., 2013b). Dans une première étape, nous présenterons brièvement la fuzzification des données multidimensionnelles qualitatives ou quantitatives. Par une opération d'agrégation, nous définirons les indices de similarités entre données. Ensuite un indice de représentativité dans l'échantillon sera obtenu par une nouvelle agrégation portant cette fois sur les similarités. Enfin, le voisinage de chaque donnée étant défini par α-coupes, nous connecterons la donnée à son voisin le plus représentatif pour construire un graphe dépendant du seuil α utilisé. Nous présenterons ensuite une interprétation des résultats qui nous permet de mettre en évidence des cas "typiques" et "atypiques" avant de conclure cette communication.

2 Fuzzification de données multi-dimensionnelles dans un échantillon

Soit E un échantillon de n données défini par : $E = \{X_i \ / \ 1 \leq i \leq n\}$. Les données appartiennent à un espace de dimension p. Autrement dit, chaque donnée X appartenant à E

a p composantes. Ainsi la donnée X est définie par : $X = (x_r)_{1 \leq r \leq p}$. Les composantes d'une donnée X sont :

- soit quantitatives et définies dans un intervalle de $\mathbb{R}$. La valeur quantitative x_r appartient alors à un domaine D_r où : $D_r = [a_r, b_r]$.
- soit qualitatives. La composante x_r appartient alors à un domaine D_r avec $D_r = \{1, 2, 3, ..., v\}$ où v est le nombre de valeurs que peut prendre x_r.

Le domaine de définition de l'échantillon E est alors défini par : $\Omega = \prod_{1 \leq r \leq p} (D_r)$ où $\prod$ est le produit cartésien des p domaines des composantes.

Une valeur d'une composante de X, qu'elle soit quantitative ou qualitative, est souvent imprécise et incertaine et il est classique de la représenter par un nombre flou ou par une quantité floue. Dans cette communication nous proposons une méthode de fuzzification de la donnée multidimensionnelle X sur E en considérant que chaque attribut ou composante de X est flou ou préalablement fuzzifié.

Soit une donnée X de l'échantillon E. Chacune des p composantes x_r de X peut être représentée par un sous-ensemble flou de son domaine D_r. La fonction d'appartenance à ce sous-ensemble flou est alors telle que :

$$\begin{array}{rccc} \mu_{x_r} : & D_r & \longrightarrow & [0,1] \\ & t & \longmapsto & \mu_{x_r}(t) \end{array} \tag{1}$$

Dans cet article, ces sous-ensembles flous sont normés avec : $\mu_{x_r}(x_r) = 1$.

En utilisant une classique méthode d'agrégation (voir par exemple (Detyniecki, 2001) ou (Dubois et Prade, 2004)), nous proposons de définir X comme une donnée floue de E dont la fonction d'appartenance μ_X est définie par :

$$\begin{array}{rccc} \mu_X : & E & \longrightarrow & [0,1] \\ & Y & \longmapsto & \mu_X(Y) \end{array} \tag{2}$$

X et Y étant deux observations de E. Si $X = (x_r)$ et $Y = (y_r)$ avec $1 \leq r \leq p$, nous proposons de définir μ_X sur E par :

$$\mu_X(Y) = agreg(\mu_{x_r}(y_r)) \tag{3}$$

avec $agreg$ comme opérateur d'agrégation des p degrés d'appartenance. Pour illustrer cette communication, nous utilisons une approche très empirique où $agreg$ est simplement la moyenne arithmétique.

3 Les indices de similarité et l'indice de représentativité

La fuzzification de la donnée X donne lieu à plusieurs remarques. L'observation X est considérée comme une donnée floue sur E et non sur le domaine Ω. Ce sous-ensemble flou de E est normé car $\mu_X(X) = 1$ pour la méthode $agreg$. Soient deux observations X et Y de E, le sous-ensemble flou associé à X définit une relation valuée de comparaison de Y avec X et :

- Y est similaire à X si $\mu_X(Y) = 1$,
- et Y est dissimilaire à X si $\mu_X(Y) = 0$.

Ainsi μ_X définit un *indice de similarité* à X. La relation de similarité induite sur E n'est pas nécessairement symétrique car $\mu_X(Y)$ n'est pas toujours égal à $\mu_Y(X)$. Pour chaque donnée X, il y a donc une indice de similarité sur E.

En agrégeant les données floues de E, on définit un sous-ensemble flou dont la fonction d'appartenance est :

$$\begin{array}{rccc} \mu: & E & \longrightarrow & [0,1] \\ & X & \longmapsto & \mu(X) \end{array} \tag{4}$$

avec :

$$\mu(X) = agreg(\mu_{X_i}(X)) \tag{5}$$

pour $1 \leq i \leq n$ avec la méthode d'agrégation notée $agreg$. Le couple (E, μ) définit alors un échantillon flou de données (sous-ensemble flou de E). Dans cette communication, nous utilisons de nouveau la moyenne arithmétique comme opérateur d'agrégation.

Deux remarques découlent des propriétés élémentaires des opérateurs d'agrégation. Plus l'observation X est similaire aux autres observations de E, plus $\mu(X)$ est proche de 1. Si X était similaire à toutes les données de E, alors on aurait $\mu(X) = 1$. Plus l'observation X est dissimilaire des autres observations de E, plus $\mu(X)$ est proche de 0. Si X était dissimilaire à toutes les données de E, alors nous aurions $\mu(X) = 0$. La valeur $\mu(X)$ devient alors un indicateur de similarité de X avec l'ensemble E dans sa globalité. Dans cet article $\mu(X)$ est appelé *un indice de représentativité* de X dans E.

4 Graphe de similarité

Soit X une observation de E, elle est considérée comme une donnée floue de fonction d'appartenance μ_X. Une α-coupe définit un voisinage de X dans E par :

$$V_\alpha(X) = \{Y \in E / \mu_X(Y) \geq \alpha\} \tag{6}$$

$V_\alpha(X)$ n'est pas vide et contient au moins X.

En utilisant l'indice de représentativité dans le voisinage de X on va rechercher l'élément le plus représentatif. Pour chaque valeur de α, on définit alors un graphe sur E en connectant chaque donnée X à la donnée voisine Z_X ayant la plus grande représentativité :

$$Z_X = \underset{Y \in V_\alpha(X)}{\arg\max} \left(\mu(Y)\right) \tag{7}$$

Ce graphe a plusieurs composantes connexes. Nous noterons par k le nombre de composantes connexes. On remarque que, dans chaque composante connexe, il existe une et une seule donnée qui est connectée à elle-même. Ces données connectées à elles-mêmes sont appelées représentants de E. Si $\alpha = 1$, alors $k = n$, il y a n représentants dans E. Si $\alpha = 0$, alors $k = 1$, il y a un seul représentant dans E (aux cas d'égalités près). Dans cette approche exploratoire de E, α permet d'abord de définir les représentants de E à une échelle donnée par α.

5 Application

Nous avons appliqué cette approche à des données médicales extraites d'une étude en cours au CHU de Reims (Nourizadeh et al., 2013a). Cette étude porte sur le diabète de type 2 chez

des sujets âgés sous traitement insulinique. L'échantillon E recueillis lors de cette étude est composé de 44 observations (44 patients) et 11 attributs (11 variables physiologiques et biologiques, quantitatives ou binaires : âge, poids, sexe, glycémie,...).

Tout d'abord, nous avons fuzzifié chacun des attributs par un nombre flou triangulaire ou trapézoïdale et les attributs binaires sont modélisés avec deux valeurs 0 ou 1. Chaque attribut d'une donnée X définit alors un sous-ensemble flou sur E. Par agrégation de ces attributs flous, nous fuzzifions la donnée X qui devient un sous-ensemble flou de E. Nous obtenons ainsi un indice de similarité vis à vis de X pour chaque donnée X de E. Après cette étape de fuzzification de chaque donnée, nous procédons au calcul de l'indice de représentativité puis à la construction du graphe de similarité pour différentes valeurs du paramètre α. Chaque noeud du graphe représente un patient diabétique que nous identifions par un numéro.

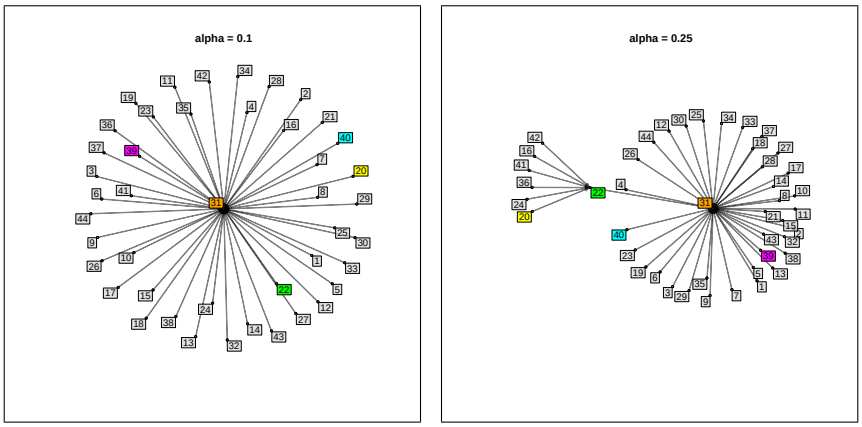
Lorsque α augmente, la taille des voisinages diminue et le nombre de représentants (i.e. le nombre de composantes connexes du graphe) augmente. On constate alors la propriété triviale suivante :

- lorsque $\alpha = 1$, chaque individu est un représentant et il y a n représentants,
- lorsque $\alpha = 0$, il a un seul représentant dans tout l'ensemble.

Dans cette application, nous ne disposons pas d'information *a priori* nous permettant de choisir une valeur de α. Empiriquement, nous avons fixé $\alpha = 0.42$, il y a alors 3 représentants pour l'échantillon (les patients numéro 22, 31 et 39, voir la figure 1d) Au delà de ce seuil de 0.42, une faible augmentation de α provoque un morcellement du graphe et donc une augmentation du nombre de représentants. Il y a par exemple 9 représentants lorsque $\alpha = 0.50$ (figure 2b) et 28 représentants lorsque $\alpha = 0.60$ (figure 2d). Au delà de 0.42, le graphe de similarité devient trop peu structuré pour dégager des regroupements de données stable. Ce n'est pas un problème, car nous ne recherchons pas des regroupements ou clusters. En effet, nous souhaitons extraire de E un sous échantillon avec à la fois des données "typiques" indiquant un regroupement potentiel ou une "généralité" et des données "atypiques" n'entrant pas dans le cadre de regroupements évidents.

Pour mieux comprendre la structuration de l'échantillon de données, nous avons fait varier α de 0.00 à 0.99 par pas de 0.01 et nous avons observé les 100 graphes de similarité obtenus. Pour rappel, si $\alpha = 0.00$ alors tous les noeuds du graphe se connectent sur le patient numéro 31 qui est le seul représentant de E, racine de la seule composante connexe du graphe. Si $\alpha = 1.00$ alors tous les noeuds du graphe se connectent à eux-même, il y a alors 44 représentants isolés dans E (constitués d'un seul sommet) et donc 44 composantes connexes dans le graphe. Etudions en fonction de α, l'état des connexion entrantes et sortantes pour chaque donnée X de E. Quand α est supérieur à 0.42, le nombre de représentants (ou racines des arbres du graphe) augmente, le nombre de points isolés sans connexion entrante ou sortante augmente et les chainages des arbres s'allongent.

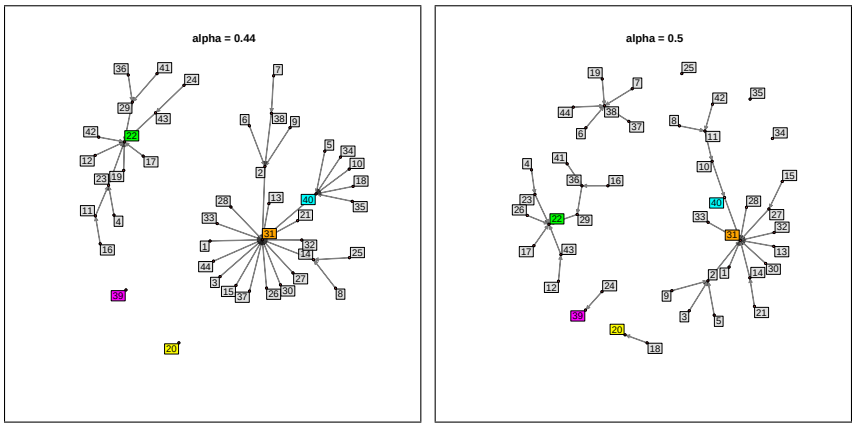
Le patient numéro 39 a un comportement particulier puisqu'il est le premier représentant isolé qui apparait lorsque $\alpha = 0.42$. Quand α augmente, 39 demeure représentant, il ne se connecte à aucun autre noeud mais il apparait une connexion entrante de la part du numéro 24. Le numéro 20 est aussi un représentant isolé précoce, il est isolé lorsque $\alpha = 0.44$. Quand $\alpha = 0.55$, un chainage est construit dont le représentant demeure le numéro 20 avec une seule connexion entrante constitué du numéro 18. Ce type d'information permet d'extraire les patients dont la situation n'est pas assimilable à celle d'autres sujets puisqu'ils sont précocement déconnectés dans E et ne reçoivent que très peu de connexion (une ou deux connexions



(a) $\alpha = 0.10$ (b) $\alpha = 0.25$

(c) $\alpha = 0.35$ (d) $\alpha = 0.42$

FIG. 1: Évolution du graphe de similarité en fonction de α, le nombre de composantes connexes et le nombre de données isolées augmentent avec α.



(a) $\alpha = 0.44$ (b) $\alpha = 0.50$

(c) $\alpha = 0.55$ (d) $\alpha = 0.60$

FIG. 2: Évolution du graphe de similarité en fonction de α, le nombre de composantes connexes et le nombre de données isolées augmentent avec α.

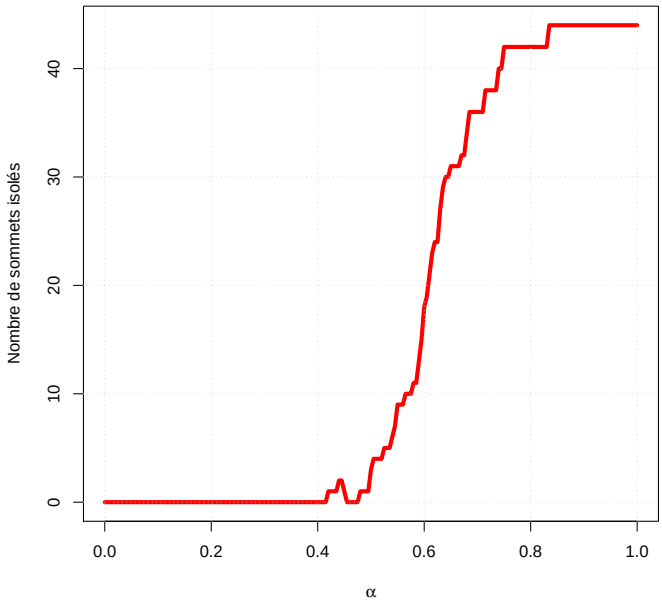


FIG. 3: Nombre de sommets isolés en fonction de α.

entrantes). Ils constituent des cas "atypiques" ayant par exemple des caractéristiques physiologiques différentes ou une prescription insulinique particulière qu'il convient de considérer différemment.

L'étude de ces individus isolés peut être complétée en observant l'évolution de leur nombre et de l'état d'isolement lorsque α varie. La courbe de la figure 3 représente l'évolution du nombre de sommets isolés en fonction de la valeur de α. L'allure de cette courbe est proche de celle du nombre de composantes connexes des graphes, ce nombre augmente d'autant plus que le nombre de sommets isolés croît.

Quand la valeur de α est entre 0.00 et 0.14, le noeud 31 est le seul représentant, il ne se connecte à aucun autre cas (aucune connexion sortante). Ce noeud 31 a une capacité particulière à agréger ou regrouper les autres données. Pour cette raison nous parlerons alors d'une donnée "typique" de l'échantillon E. A partir de 0.59, ce noeud 31 perd beaucoup de connexions entrantes mais il ne se connecte à aucune autre donnée.

Quand le nombre de connexion entrantes augmente, cela indique une plus forte capacité à regrouper les autres données. Quand $\alpha = 0.15$, on constate que le noeud 22 commence à indiquer cette capacité de regroupement par une connexion entrante (FIGURE). En effet le noeud 41 se connecte à 22 et 41 n'est plus connecté directement à 31. Le nombre de connexions reçues par 22 augmente lorsque α varie de 0.15 à 0.33. A 0.33, le patient 22 devient un représentant de E sans connection sortante (racine d'un des arbres du graphe). Nous consid-

érons que le noeud 22 est aussi un représentant "typique"" de E. Plus tardivement quand α plus grand que 0.30, le noeud 40 lui aussi accepte des connexions entrantes développant des capacité de regroupement de données. Ce noeud 40 est aussi considéré comme un cas "typique" de E.

On définit ainsi les profils des connexions entrantes et sortantes de chaque donnée dans E pour définir les cas typiques et atypiques de notre échantillon.

6 Conclusion

On a défini, d'une part, un graphe à partir d'un échantillon de données multidimensionnelles qualitatives ou quantitatives et, d'autre part, une méthode pour extraire des représentants de cet échantillon et définir un sous-échantillon composé de cas typiques et atypiques. La méthode met à contribution le concept de flou pour définir un indice de similarité entre données. La méthode permet de :

- structurer l'échantillon E par un graphe,
- sous-échantillonner E,
- faciliter la compréhension de E à l'aide de quelques cas qui ne sont pas des prototypes virtuels.

Ainsi on peut définir une typologie à l'intérieur de l'échantillon E. Cette approche sans contrainte préalable de clustering ne nécessite pas d'effectif important et s'adapte à l'initialisation d'une base de cas pour le démarrage à froid d'un raisonnement à base de cas ou d'un système de recommandation.

Cette approche exploratoire d'un échantillon de données est illustrée par l'utilisation de données médicales en vue de déterminer les différents modèles de traitements insuliniques dans le cas du diabète de type 2 chez des sujets âgés. Ce cas d'étude est complexe car ces patients souffrent souvent de multiples pathologies rendant difficile l'extraction d'une typologie insulinique pour le diabète de type 2. Si l'insulinothérapie est si difficile à modéliser dans le type 2, cela tient à la fois à la variabilité biologique intrinsèque et à la conjugaison des phénomènes d'insulino-résistance et d'insulino-sensibilité spécifiques de chaque patient. L'approche que nous proposons est une première étape exploratoire vers une modélisation de l'aide à l'insulinothérapie pour des patients diabétiques âgés par une approche à base de cas.

Références

Detyniecki, M. (2001). Mathematical aggregation operators and their application to video querying. Technical report, LIP6, Paris.

Dubois, D. et H. Prade (2004). On the use of aggregation operations in information fusion processes. *Fuzzy Sets and Systems 142*, 143–161.

Marling, C., J. Shubrook, et F. Schwartz (2009). Toward case-based reasoning for diabetes management: A preliminary clinical study and decision support system prototype. *Computational Intelligence 25*, 165–179.

Nourizadeh, A., F. Blanchard, A. Ait-Younes, B. Delemer, et M. Herbin (2013a). Analyse exploratoire de données d'insulinothérapie du diabète de type 2. In *JETSAN 2013*.

Nourizadeh, A., F. Blanchard, B. Delemer, et M. Herbin (2013b). Données multidimensionnelles floues et graphe de similarité. In *LFA 2013, Actes des 22ème rencontre francophone sur la logique floue et ses applications*, pp. 259–265.

Proceeding of ICCBR (2012). In B. D. Agudo et I. Watson (Eds.), *Case-Based Reasoning Research and Development*. Lyon, September 2012.

Summary

Data analysis requires an exploratory phase to perform reconciliations and extract relevant information. In the medical field where cases are treated on an individual basis , without excluding rare cases or atypical , the therapeutic modeling is often complex. In this communication, we propose an approach based on exploration these heterogeneous medical data. In a first step, we fuzzify both qualitative and quantitative multidimensional data. By aggregation operation, we define the index of similarities between data. Then an index of representativeness in the set is obtained by new aggregation operator within data samples. Finally, the neighborhood of each data is defined by α-cuts, we connect each data to its most representative neighbor to build a graph depends on the threshold α used. This allows us to bring up the "typical" case and "atypical" or rare cases. This approach is applied in a study on insulin therapy in type 2 diabetes in the elderly.

Développement d'une application de recommandation d'offres d'emploi aux utilisateurs de Facebook et LinkedIn

Mamadou Diaby[*,**] Emmanuel Viennet[*]

[*]Université Paris 13, Sorbonne Paris Cité, L2TI, F-93430, Villetaneuse, France
{mamadou.diaby, emmanuel.viennet}@univ-paris13.fr,
[**]Work4, 3 Rue Moncey, 75009, Paris, France
https://www.work4labs.com

Résumé. Ce papier présente les différents systèmes de recommandation que nous avons développés pour proposer des offres d'emploi pertinentes aux utilisateurs de Facebook et LinkedIn en utilisant les données contenues dans leurs profils et les descriptions des offres d'emploi.

La première partie de ce papier est consacrée à un système de recommandation utilisant la similarité cosinus sur les vecteurs des utilisateurs et des offres d'emploi obtenus en agrégeant les données de leurs différents champs textuels.

Nous avons ensuite présenté une méthode pour estimer l'importance des différents champs pour la recommandation d'offres d'emploi. L'application des poids obtenus a amélioré de façon significative les résultats obtenus dans la première partie.

Enfin, la troisième partie est consacrée à l'utilisation d'un algorithme d'apprentissage (SVM) afin d'améliorer les résultats obtenus dans les deux premières parties de ce papier. Nos résultats montrent que l'utilisation de cette procédure d'apprentissage permet d'améliorer les performances par rapport l'utilisation d'une heuristique comme la similarité cosinus.

1 Motivations

Les systèmes de recommandation sont devenus très populaires depuis les années 1990 (date d'apparition des premiers articles sur ces systèmes selon (Adomavicius et Tuzhilin, 2005)), car ils présentent à la fois des intérêts académiques (Wang et Blei, 2011; Ma et al., 2009) et commerciaux (Linden et al., 2003). (Xiao et Benbasat, 2007) définit les systèmes de recommandation comme des logiciels qui calculent les intérêts que porteraient des consommateurs à des items ou produits, pour leur recommander les plus pertinents.

Durant la dernière décennie, nous avons assisté à un développement rapide du web et des réseaux sociaux, ce qui a permis l'essor du commerce et l'exploitation des données des utilisateurs. Les informations personnelles publiées par les utilisateurs d'un réseau social dans leurs profils (descriptions personnelles, messages, notes, likes, liens sociaux, etc.) peuvent être exploitées par un système de recommandation.

Cet article présente trois systèmes de recommandation d'offres d'emploi aux utilisateurs de Facebook et LinkedIn. Une variante de ces systèmes de recommandation est utilisée par l'entreprise Work4, le leader mondial des technologies de recrutements sur Facebook. Pour des raisons de protection de la vie privée, nos systèmes de recommandation utilisent uniquement les données dont les utilisateurs des réseaux sociaux ont explicitement autorisé l'accès. Nos systèmes doivent traiter des données complexes, bruitées et incomplètes issues des utilisateurs de Facebook et de LinkedIn et des offres d'emploi postées par les clients de l'entreprise Work4.

Le reste de cet article est organisé comme suit : nous avons résumé les travaux effectués sur les systèmes de recommandation au cours des deux dernières décennies dans la section 2 ; la section 3 présente nos différents systèmes de recommandation ; dans la section 4, nous présentons nos différentes expérimentations et les résultats obtenus. Nous terminerons par une discussion dans la section 5.

2 Travaux antérieurs

Les systèmes de recommandation (Adomavicius et Tuzhilin, 2005; Jannach et al., 2011) aident les utilisateurs à faire face à la surcharge informationnelle en leur proposant des items pertinents pour eux. Ils sont principalement liés à la recherche d'information (Baeza-Yates et Berthier, 1999; Salton et al., 1975), l'apprentissage artificiel (de Campos et al., 2010; Salakhutdinov et Mnih, 2008a), au data mining (Séguela, 2012) et à d'autres champs de recherche qui sortent du cadre de cette étude. Ils sont généralement classés en trois catégories (Adomavicius et Tuzhilin, 2005; Bobadilla et al., 2013) : les méthodes basées sur le contenu, les méthodes de filtrage collaboratif et les approches hybrides.

Les systèmes de recommandation basés sur le contenu utilisent les descriptions des utilisateurs et des items (quand elles sont disponibles), ainsi que les notes que les utilisateurs ont donné à certains items (Adomavicius et Tuzhilin, 2005; Rocchio, 1971). Dans cette étude, nous nous sommes intéressés aux systèmes basés sur le contenu.

Contrairement aux systèmes basés sur le contenu, les méthodes de filtrage collaboratif utilisent l'opinion d'une communauté d'utilisateurs similaires à l'utilisateur actif pour lui pour recommander des items pertinents (Jannach et al., 2011; Salakhutdinov et Mnih, 2008b).

Les systèmes de recommandation hybrides combinent un système basé sur le contenu et une technique de filtrage collaboratif en un seul modèle (Adomavicius et Tuzhilin, 2005; Claypool et al., 1999; Wang et Blei, 2011).

Dans les systèmes de recommandation, le contenu textuel d'un document (utilisateur ou item) est souvent représenté par un vecteur de couples (terme, poids du terme) ; ce vecteur est généralement construit en utilisant la méthode « sac de mots » et une fonction de pondération tout en supprimant les mots vides.

Dans la méthode « sac de mots » on suppose que l'ordre des termes dans un document n'a pas beaucoup d'importance dans le processus de catégorisation des documents. Pour supprimer les mots vides, plusieurs méthodes existent dans la littérature (Séguela, 2012) : utiliser une liste de mots vides, supprimer certaines catégories grammaticales ou les mots les plus fréquents et les moins fréquents.

Les fonctions de pondération calculent l'importance d'un terme pour un document donné ; (Séguela, 2012) les a classé en trois catégories : les fonctions de pondération locales, globales et hybrides. Une fonction de pondération locale calcule l'importance d'un terme au sein

d'un document tandis qu'une fonction de pondération globale calcule l'importance d'un terme dans un corpus (ensemble de documents). Les fonctions de pondération classiques sont : TF (Fréquence des Termes), Log-TF ($\log$(TF)), Booléenne (0/1 selon la présence ou non du terme dans le document), TF-IDF, log-Entropie, etc. Okapi qui est une variante du TF-IDF, donne également de bons résultats (Claveau, 2012).

Nous avons utilisé une fonction de pondération hybride pour développer nos systèmes de recommandation : le TF-IDF qui est définit comme suit :

$$\text{TF-IDF}_{t,d} = \text{TF}_{t,d} \times \text{IDF}_t \tag{1}$$

où $\text{TF}_{t,d} = \frac{f_{t,d}}{\max_k f_{k,d}}$ est la fonction de pondération fréquence normalisée (pondération locale) et $\text{IDF}_t = 1 + \log\left(\frac{N}{n_t}\right)$ est une fonction de pondération globale, $f_{t,d}$ la fréquence du terme t dans le document d, N le nombre total de documents dans le corpus et n_t le nombre de documents qui contiennent le terme t.

Dans la littérature, les systèmes de recommandation utilisent plusieurs fonctions de similarité pour calculer la similarité entre les utilisateurs, entre les items ou entre utilisateurs et items (Adomavicius et Tuzhilin, 2005; Jannach et al., 2011) : la similarité cosinus, le coefficient de corrélation de Pearson, la distance Euclidienne, etc. Nous avons utilisé la similarité cosinus et une méthode d'apprentissage artificiel : le SVM (Cortes et Vapnik, 1995) dans nos systèmes de recommandation.

La similarité cosinus est définie comme suit :

$$\cos(u, v) = \frac{\sum_k u_k v_k}{\sqrt{\sum_k u_k^2}\sqrt{\sum_k v_k^2}} \tag{2}$$

où u et v sont des vecteurs des utilisateurs ou des offres d'emploi.

Les mesures de performance classiques utilisées pour mesurer la performance des recommandations (Omary et Mtenzi, 2010) sont : la précision, le rappel, la F-mesure. Nous avons opté pour l'AUC-ROC comme critère de performance pour nos systèmes de recommandation car la précision, le rappel et le F-mesure dépendent d'un seuil et la détermination d'un seuil optimal est souvent difficile. L'AUC-ROC est l'aire sous la courbe ROC qui est obtenue en traçant le taux de vrais-positifs en fonction du taux de faux-positifs (Omary et Mtenzi, 2010).

3 Méthodes

3.1 Modélisation des documents

Les utilisateurs de Facebook et LinkedIn sont définis par deux types de données : leurs propres données et les données de leurs amis sur le réseau social. Dans ce papier nous nous sommes uniquement intéressés aux données des utilisateurs ; les données propres des utilisateurs sont à la fois les données qu'ils ont publiées sur le réseau social et celles liées à leur utilisation de la plateforme (lectures de posts, likes, annotations, etc.). Nos systèmes de recommandation utilisent uniquement les données dont les utilisateurs ont explicitement autorisé l'accès.

Les utilisateurs de Facebook ont autorisé les applications de Work4 à accéder aux données de 5 champs : *Work*, *Education* , *Bio*, *Interests* et *Quotes*. Ceux de LinkedIn ont seulement autorisés 3 champs : *Headline*, *Educations* et *Positions*. La description de nos offres d'emploi contient 3 champs : *Title*, *Description* et *Responsibilities*.

Nos utilisateurs et nos offres d'emploi sont vus comme des documents ; pour filtrer les mots vides, nous avons construit une liste de mots vides. Le vecteur qui représente le profil d'un document donné (un utilisateur ou une offre d'emploi) est construit comme suit :

- dans Engine-1, le premier système de recommandation proposé (voir la section 3.2), les textes des différents champs d'un document sont concaténés ; le texte obtenu est transformé en un vecteur en utilisant la fonction de pondération TF-IDF (c.f. eq (1)) et en supprimant les mots vides. On remarque qu'ici, tous les champs sont supposés avoir la même importance sur les scores de recommandation.
- dans Engine-2 et Engine-3 (voir les sections 3.4 et 3.5), les textes des différents champs sont concaténés en tenant compte de l'importance de leur champ d'origine (c.f. section 4.2.1). La transformation des textes concaténés en vecteurs se fait en utilisant la fonction de pondération TF-IDF (c.f. eq. (1)) comme dans Engine-1.

Dans nos travaux préliminaires, nous avons comparé différentes fonctions de pondération (TF, log-TF, Booléenne, TF-IDF et Log-Entropy) avec différentes fonctions de similarité (cosinus, coefficient de corrélation de Pearson et distance euclidienne) : nous avons obtenu de meilleurs résultats pour le couple TF-IDF/cosinus.

3.2 Premier système de recommandation : Engine-1

Le premier système de recommandation proposé est Engine-1 : les utilisateurs et les offres d'emploi sont représentés comme des vecteurs en utilisant le modèle « sac de mots » avec TF-IDF (c.f. eq. (1)) comme fonction de pondération tout en supposant que tous les champs ont la même importance dans le processus de recommandation d'offres d'emploi. Nous avons utilisé la similarité cosinus (c.f. eq. (2)) comme fonction de similarité.

3.3 Importances des différents champs

Nous savons qu'il est très peu probable que tous les champs aient la même importance pour la recommandation d'offres d'emploi : certains sont plus importants que d'autres. Nous avons ainsi proposé le problème d'optimisation suivant pour déterminer les importances (poids) des différents champs.

Le vecteur $u(\alpha)$ d'un utilisateur pour les importances α des champs des utilisateurs est défini comme la somme pondérée des vecteurs des différents champs :

$$u(\alpha) = \sum_{f=1}^{f_u^0} \alpha_f^0 u_f^0 + \sum_{f=1}^{f_u^1} \alpha_f^1 u_f^1 \tag{3}$$

où $\alpha = (\alpha^0, \alpha^1)$, $\alpha^0 = (\alpha_1^0, \ldots, \alpha_{f_u^0}^0)$ et $\alpha^1 = (\alpha_1^1, \ldots, \alpha_{f_u^1}^1)$ sont respectivement les importances des champs des utilisateurs de Facebook et de LinkedIn, f_u^0 et f_u^1 sont respectivement le nombre de champs pour les utilisateurs de Facebook et LinkedIn dans nos bases de données, u_f^0 et u_f^1 sont respectivement les vecteurs du champ f de l'utilisateur de Facebook et de

LinkedIn et α_f^0 et α_f^1 sont respectivement les importances du champs f pour les utilisateurs de Facebook et de LinkedIn.

Le vecteur $v(\beta)$ d'une offre d'emploi pour les importances β des champs des offres d'emploi est défini comme la somme pondérée des vecteurs des différents champs :

$$v(\beta) = \sum_{f=1}^{f_j} \beta_f v_f \tag{4}$$

où $\beta = (\beta_1, \ldots, \beta_{f_j})$, f_j est le nombre de champs pour les offres d'emploi dans nos bases de données, v_f est le vecteur du champ f de l'offre d'emploi v et β_f est l'importance du champ f.

La similarité $\hat{y}_{uv}(\alpha, \beta)$ entre un utilisateur u et une offre d'emploi v en utilisant les importances des champs α et β est calculée comme $\cos(u(\alpha), v(\beta))$ (c.f. eq. (2)).

Nous avons minimisé une version modifiée de l'erreur quadratique moyenne $E_\Gamma(\alpha, \beta, c_0, c_1)$ pour apprendre α et β sur une base d'apprentissage Γ :

$$E_\Gamma(\alpha, \beta, c_0, c_1) = \frac{1}{|\Gamma|} \sum_{(u,v,y)\in\Gamma} c_y \cdot (y - \hat{y}_{uv}(\alpha, \beta))^2 \tag{5}$$

Chaque entrée de Γ est un triplet (u, v, y) où u, v et $y \in \{0, 1\}$ représentent respectivement un utilisateur, une offre d'emploi et un label, c_0 et c_1 sont les coûts respectifs des classes 0 et 1. Le label $y = 1$ signifie que l'offre d'emploi v correspond à l'utilisateur u (matching) tandis que $y = 0$ si l'utilisateur et le l'offre d'emploi ne sont pas assortis.

Pour tout $\lambda_1 > 0$ et $\lambda_2 > 0$, on remarque que $\hat{y}_{uv}(\lambda_1\alpha, \lambda_2\beta) = \hat{y}_{uv}(\alpha, \beta)$, ce qui signifie que si le problème d'optimisation admet une solution, alors elle n'est pas unique. Nous avons alors utilisé les contraintes suivantes pour diminuer le nombre de solutions possibles : $\|\alpha^0\| = 1$, $\|\alpha^1\| = 1$ and $\|\beta\| = 1$.

3.4 Deuxième système de recommandation : Engine-2

Dans ce modèle les vecteurs TF-IDF des utilisateurs et des offres d'emploi sont obtenus en agrégeant les données des différents champs tout en tenant compte du poids du champ (les poids obtenus dans la section 4.2.1). Comme dans Engine-1, nous avons utilisé la similarité cosinus comme fonction de similarité (c.f. eq. (2)).

3.5 Troisième système de recommandation : Engine-3

Nous avons utilisé les SVMs (Support Vector Machines) (Cortes et Vapnik, 1995) pour apprendre une fonction de similarité entre les utilisateurs des réseaux sociaux et les offres d'emploi, à partir de nos données. Les SVMs donnent généralement de très bons résultats dans la catégorisation des documents textuels (Joachims, 1998) ; ils sont plus stables que les réseaux de neurones et ne nécessitent pas d'hypothèse d'indépendance des termes comme dans les réseaux Bayésiens naïfs.

Le vecteur d'entrée $I_{SVM}(u, v)$ pour le SVM correspondant au couple de vecteur u et v est construit comme suit :

$$I_{SVM}(u, v) = (u, v) = (u_1, \ldots, u_K, v_1, \ldots, v_K) \tag{6}$$

où K est le nombre total de termes distincts dans nos documents et u et v sont respectivement les vecteurs TF-IDF d'un utilisateur et d'une offre d'emploi obtenus en utilisant les poids optimaux des champs (voir les résultats de la section 4.2.1).

Nous avons 218533 termes distincts (toutes langues confondues), donc nous travaillons en très grande dimension ; dans la littérature, certains pensent qu'en très grande dimension, les données sont facilement linéairement séparables (théorème de Cover (Cover, 1965)), raison pour laquelle nous avons utilisé un modèle linéaire. Afin de gérer efficacement les bases d'apprentissage déséquilibrées, nous avons utilisé des coûts différents pour les deux classes : c_0 et c_1 respectivement pour la classe 0 (label = 0) et la classe 1 (label = 1). Nous avons utilisé la librairie Libsvm (Chang et Lin, 2011) pour apprendre nos modèles SVMs.

4 Expérimentations

Nous avons utilisé l'AUC-ROC (voir Section 2) comme mesure de performance pour nos différentes expérimentations ; nous avons calculé les intervalles de confiance empiriques à 95% ($[2.5\%$ quantile, 97.5% quantile$]$) à l'aide du bootstrapping (100 exécutions) à chaque fois.

4.1 Différents jeux de données

Nous avons évalué la performance de nos systèmes de recommandation d'offres d'emploi (proposés à la section 3) sur 6 jeux de données collectés par l'entreprise Work4. Chaque entrée de nos jeux de données est un triplet (u, v, y) où u et v sont respectivement les vecteurs d'un utilisateur et d'une offre d'emploi donné et $y \in \{0, 1\}$ est leur label associé. Pour rappel, $y = 1$ signifie que l'offre d'emploi v correspond à l'utilisateur u (matching) tandis que $y = 0$ si l'utilisateur et l'offre d'emploi ne sont pas assortis. Voici la description des 6 jeux de données collectés :

1. Candidate : les utilisateurs peuvent utiliser les applications de Work4 pour postuler à des offres d'emploi ; nous supposons que les utilisateurs postulent uniquement aux offres d'emploi qui leur correspondent. Ce jeu de données contient ainsi les données des candidatures.
2. Feedback : contient les retours (feedback) des utilisateurs des applications de Work4.
3. Random : cette base de données contient des couples d'utilisateurs et d'offres d'emploi qui ont été tirés au hasard de nos bases de données et annotés manuellement.
4. Review : contient les recommandations faites par les systèmes de Work4 qui ont été vérifiées par des validateurs et l'équipe de Work4.
5. Validation : contient les recommandations faites par les systèmes de Work4 qui ont été vérifiées par des validateurs.
6. ALL : un sixième jeu de données a été artificiellement créé : c'est l'union des cinq jeux de données précédents.

Le tableau 1 présente les différentes statistiques de nos jeux de données. Après avoir filtré les mots vides en utilisant une liste de mots vides, nous avons obtenu un dictionnaire avec 218533 termes (mots) distincts. Le tableau 2 montre le pourcentage de champs vides dans les différents champs de nos jeux de données. Il est intéressant de remarquer que les utilisateurs de

	Jeux de données					
	ALL	Candidate	Feedback	Random	Review	Validation
Facebook	77.579	29.487	62	2.211	10.762	35.057
LinkedIn	126.411	692	184	687	21.679	103.169
Label 0	160.461	0	86	2.872	24.787	132.716
Label 1	43.529	30.179	160	26	7.654	5.510
Total	**203.990**	**30.179**	**246**	**2.898**	**32.441**	**138.226**

TAB. 1: Nombre total d'entrées dans chaque jeu de données ; nous avons également les distributions des entrées liées aux utilisateurs de Facebook et LinkedIn et des labels 0 et 1 dans les différents jeux de données.

Facebook ne remplissent pas complètement les champs qui nous intéressent ; nos systèmes de recommandation doivent donc traiter des données incomplètes (et qui contiennent beaucoup de bruits).

4.2 Résultats

4.2.1 Importances des champs

Pour calculer les importances des différents champs, le problème d'optimisation sous contraintes présenté dans la section 3.3 a été résolu en utilisant la fonction *minimize* (du module Scipy.optimize) avec la méthode SLSQP (Sequential Least Squares Quadratic Programming). Les coûts c_0 et c_1 des deux classes (labels 0 et 1) ont été définis comme dans Anand et al. (2010) $c_0 = \frac{1}{n_0}$ et $c_1 = \frac{1}{n_1}$ où n_0 et n_1 sont respectivement le nombre d'entrées avec le label 0 et label 1 dans l'échantillon d'apprentissage.

La figure 1 montre les importances des différents champs des utilisateurs de Facebook, de Linekdin et des offres d'emploi. Elle suggère également que les champs les plus importants dans le processus de recommandation d'offres d'emploi sont :

- *Work* (antécédents de travail) pour les utilisateurs de Facebook,
- *Headline* (le résumé de carrière) pour les utilisateurs de LinkedIn,
- *Title* (le titre) pour les offres d'emploi.

4.2.2 Comparaison entre Engine-1 et Engine-2

Nous avons procédé à une comparaison entre Engine-1 et Engine-2 pour voir si les poids des differents champs obtenus dans la section 4.2.1 permettent d'améliorer nos résultats. La figure 2 montre que l'application des poids des champs a permis d'améliorer de façon significative les résultats en terme d'AUC sur tous les jeux de données sauf le jeu de données Random ; La dégradation des résultats constatée sur ce jeu de données n'est pas significative vue la taille des intervalles de confiance : donc Engine-2 est meilleur que Engine-1.

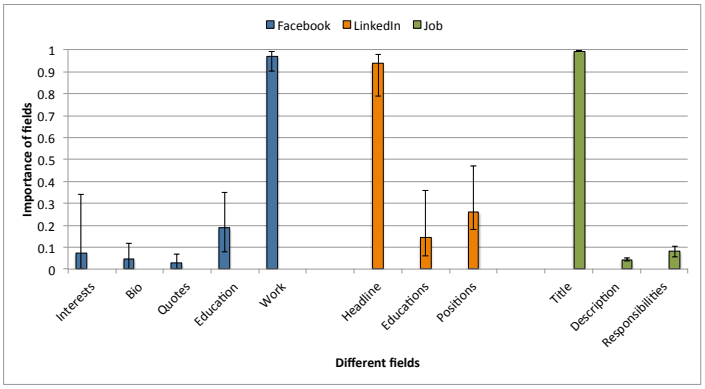


FIG. 1: Estimation des poids (importances) ($\in [-1, +1]^{11}$) des différents champs des utilisateurs de Facebook, de LinkedIn et des offres d'emploi ; plus le poids est élevé, plus le champs correspondant est important dans le processus de recommandation d'offres d'emploi.

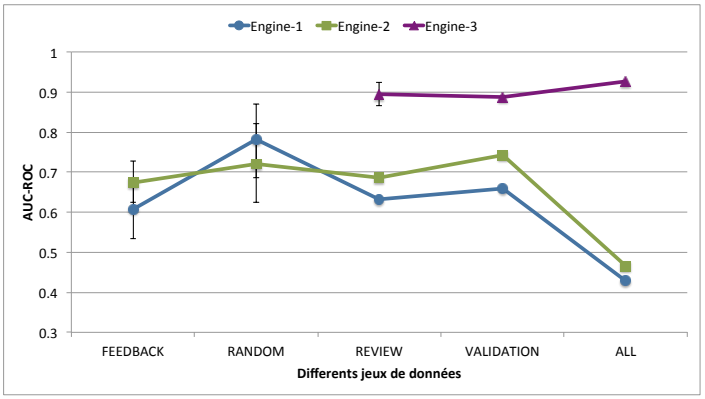


FIG. 2: Comparaison entre Engine-1, Engine-2 et Engine-3 sur tous nos jeux de données ; pour Engine-3, nous avons utilisé de la validation croisée 10-fold sur les 3 plus grands jeux de données (Review, Validation et ALL).

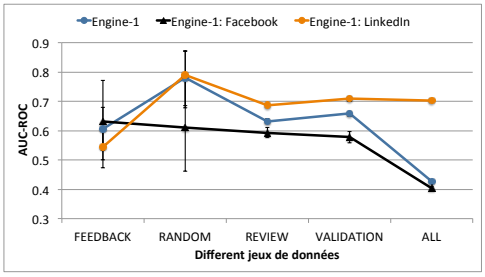


FIG. 3: Comportement de Engine-1 pour les utilisateurs de Facebook et de LinkedIn.

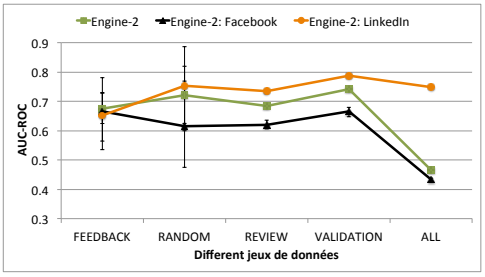


FIG. 4: Comportement de Engine-2 pour les utilisateurs de Facebook et de LinkedIn.

		Jeux de données					
	Fields	ALL	Candidate	Feedback	Random	Review	Validation
Facebook	Bio	**66,5**	**81,3**	**61,8**	**85,0**	**69,4**	48,4
	Education	**81,6**	**88,2**	**52,7**	**81,4**	**78,3**	**73,9**
	Interests	**76,9**	**79,1**	**90,9**	**94,5**	**79,7**	**72,7**
	Quotes	**93,6**	**87,7**	**92,7**	**95,9**	**97,5**	**98,8**
	Work	**60,8**	**73,3**	23,6	**80,1**	45,5	46,2
LinkedIn	Educations	14,1	12,9	8,4	12,5	13,1	14,0
	Headline	0,3	1,6	0,0	0,3	0,4	0,2
	Positions	0,2	3,8	0,0	1,0	0,2	0,1
Offres d'emploi	Description	0,0	0,0	0,0	0,5	0,0	0,0
	Responsibilities	**57,6**	**52,8**	**78,6**	**74,7**	**65,4**	**82,0**
	Title	0,0	0,0	0,0	0,0	0,0	0,1

TAB. 2: Pourcentage des champs vides dans les différents jeux de données ; en gras, nous avons les champs vides pour plus de 50% des entrées.

Les figures 3 et 4 montrent respectivement les comportements de Engine-1 et Engine-2 sur Facebook et LinkedIn ; on remarque que les données de LinkedIn sont de meilleures qualités pour la recommandation d'offres d'emploi que celles de Facebook avant et après l'application des poids des différents champs estimés dans la section 4.2.1.

4.2.3 Comparaison entre Engine-2 et Engine-3

Les coûts des classes c_0 et c_1 ont été calculés comme dans la section 4.2.1. Nous avons fait varier progressivement la proportion de la base d'apprentissage de 0 à 0.8 (validation croisée holdout), en calculant la performance du système appris pour chacune de ces valeurs ; nous également utilisé de la validation croisée 10-fold sur les 3 plus grands jeux de données (Review, Validation et ALL) car les jeux de données Feedback et Random sont trop petits pour apprendre une fonction de similarité. La figure 2 montre que les fonctions de similarité apprises par le SVM permettent d'améliorer nos résultats par rapport à Engine-2 : Engine-3 est meilleur Engine-2. Les figures 5 , 6 et 7 montrent les évolutions des performances de Engine-3 en fonction de la taille de la base d'apprentissage pour les jeux de données ALL, Review et Validation.

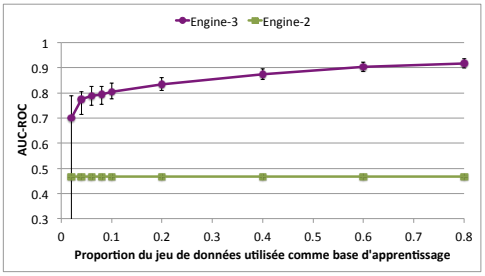


FIG. 5: Comparaison entre Engine-2 et Engine-3 sur le jeu de données ALL.

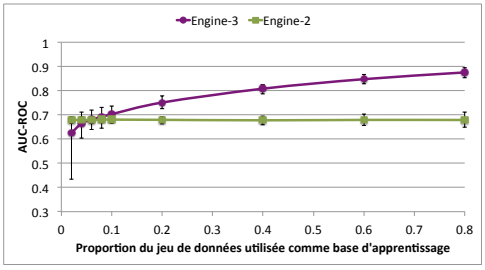


FIG. 6: Comparaison entre Engine-2 et Engine-3 sur le jeu de données Review.

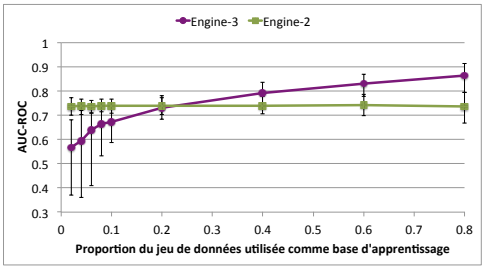


FIG. 7: Comparaison entre Engine-2 et Engine-3 sur le jeu de données Validation.

5 Conclusion et discussions

Nous avons mené de nombreuses expériences sur des données réelles collectées par l'entreprise Work4 pour tester nos systèmes de recommandation.

Nos études ont monté que les champs les plus intéressants pour la recommandation d'offres d'emploi sont *Work* (antécédents de travail) pour les utilisateurs de Facebook et *Headline* (le résumé de carrière) pour les utilisateurs de LinkedIn et *Title* (le titre) pour les offres d'emploi. Ce résultat semble cohérent, puisque étant donné un utilisateur quelconque de Facebook ou de LinkedIn et la description d'une offre d'emploi, pour dire si cet utilisateur correspond à ce poste, on comparera sans doute en premier ses antécédents de travail ou son résumé de carrière au titre de l'offre d'emploi...

L'application des poids des différents champs a permis d'améliorer de façon significative nos résultats. L'application du SVM donne des résultats encore meilleurs. Les résultats du SVM sont possiblement biaisés car une partie des offres d'emploi d'une même entreprise se retrouve en apprentissage et le reste en test (mais cela correspond à un besoin de l'entreprise Work4), sachant que les offres d'emploi d'une même entreprise sont généralement similaires. Ce biais pourra être corrigé en mettant les entrées liées aux offres d'emploi d'une même entreprise soit en apprentissage, soit en test ; mais cette procédure n'est pas applicable à nos jeux de données actuels (présence de grosses pages d'offres d'emploi d'une même entreprise).

Actuellement, nous travaillons sur deux nouveaux systèmes de recommandation : l'un utilisant une ontologie (prise en compte de la sémantique des termes, ce qui n'est pas le cas des systèmes de recommandation présentés dans ce papier) et l'autre utilisant les données des amis

des utilisateurs pour améliorer nos performances. L'approche par ontologie donnera une nouvelle représentation des profils des offres d'emploi et des utilisateurs de Facebook et LinkedIn, qui représentera certainement une alternative crédible au TF-IDF.

Références

Adomavicius, G. et A. Tuzhilin (2005). Toward the next generation of recommender systems : A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng. 17*(6), 734–749.

Anand, A., G. Pugalenthi, G. B. Fogel, et P. N. Suganthan (2010). An approach for classification of highly imbalanced data using weighting and undersampling. *Amino Acids 39*(5), 1385–91.

Baeza-Yates, R. A. et R.-N. Berthier (1999). *Modern Information Retrieval*. Boston, MA, USA : Addison-Wesley Longman Publishing Co., Inc.

Bobadilla, J., F. Ortega, A. Hernando, et A. Gutiérrez (2013). Recommender systems survey. *Knowledge-Based Systems 46*, 109–132.

Chang, C.-C. et C.-J. Lin (2011). Libsvm : A library for support vector machines. *ACM Trans. Intell. Syst. Technol. 2*(3), 27 :1–27 :27.

Claveau, V. (2012). Vectorisation, okapi et calcul de similarité pour le tal : pour oublier enfin le tf-idf (vectorization, okapi and computing similarity for nlp : Say goodbye to tf-idf) [in french]. In *Actes de la conférence conjointe JEP-TALN-RECITAL 2012, volume 2 : TALN*, Grenoble, France, pp. 85–98. ATALA/AFCP.

Claypool, M., A. Gokhale, T. Miranda, P. Murnikov, D. Netes, et M. Sartin (1999). Combining content-based and collaborative filters in an online newspaper.

Cortes, C. et V. Vapnik (1995). Support-vector networks. In *Machine Learning*, pp. 273–297.

Cover, T. M. (1965). Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition. *Electronic Computers, IEEE Transactions on EC-14*(3), 326–334.

de Campos, L. M., J. M. Fernández-Luna, J. F. Huete, et M. A. Rueda-Morales (2010). Combining content-based and collaborative recommendations : A hybrid approach based on bayesian networks. *Int. J. Approx. Reasoning 51*(7), 785–799.

Jannach, D., M. Zanker, A. Felfernig, et G. Friedrich (2011). *Recommender Systems : An Introduction*. Cambridge University Press.

Joachims, T. (1998). Text categorization with suport vector machines : Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*, ECML '98, London, UK, UK, pp. 137–142. Springer-Verlag.

Linden, G., B. Smith, et J. York (2003). Amazon.com recommendations : Item-to-item collaborative filtering. *IEEE Internet Computing 7*(1), 76–80.

Ma, H., I. King, et M. R. Lyu (2009). Learning to recommend with social trust ensemble. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, New York, NY, USA, pp. 203–210. ACM.

Omary, Z. et F. Mtenzi (2010). Machine learning approach to identifying the dataset threshold for the performance estimators in supervised learning. *International Journal for Infonomics (IJI) 3*.

Rocchio, J. J. (1971). Relevance feedback in information retrieval. In G. Salton (Ed.), *The SMART Retrieval System : Experiments in Automatic Document Processing*, Prentice-Hall Series in Automatic Computation, Chapter 14, pp. 313–323. Prentice-Hall, Englewood Cliffs NJ.

Salakhutdinov, R. et A. Mnih (2008a). Bayesian probabilistic matrix factorization using markov chain monte carlo.

Salakhutdinov, R. et A. Mnih (2008b). Probabilistic matrix factorization.

Salton, G., A. Wang, et C. Yang (1975). A vector space model for information retrieval. *Journal of the American Society for Information Science 18*(11), 613–620.

Séguela, J. (2012). *Fouille de données textuelles et systèmes de recommandation appliqués aux offres d'emploi diffusées sur le web*. Ph. D. thesis, Conservatoire National des Arts et Métiers (CNAM), Paris, France.

Wang, C. et D. M. Blei (2011). Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, New York, NY, USA, pp. 448–456. ACM.

Xiao, B. et I. Benbasat (2007). E-commerce product recommendation agents : use, characteristics, and impact. In *MIS Quarterly*, Volume 31, pp. 137–209. Society for Information Management and The Management Information Systems Research Center Minneapolis, MN, USA.

Summary

This paper presents the content-based recommender systems we have developed in order to propose relevant jobs to Facebook and LinkedIn users using their profiles' data and the descriptions of jobs. The profiles of our social networks users and the description of our jobs are divided into several parts called fields.

The first part of this document is dedicated to the study of a recommender system that uses cosine similarity to compute the similarity between a user's and a job's vectors (obtained by concatenating data from different fields).

We then present a method to estimate the importance of different fields of our users and jobs in the task of job recommendation. The application of the computed importance (weights) significantly improved the results.

The third part is devoted to the use of a machine learning algorithm to improve the results with similarity measure: we learnt a linear SVM (Support Vector Machines). Our results show that the use of this learning method increases the performance of our recommender system.

Evaluation de la pertinence dans un système de recommandation sémantique de nouvelles économiques

David Werner*, Christophe Cruz* et Aurélie Bertaux*

*LE2I UMR 6306
Faculté des sciences Mirande
BP 47870, 21078 Dijon Cedex, France
david.werner,christophe.cruz,aurelie.bertaux@u-bourgogne.fr

Résumé. De nos jours dans les secteurs commerciaux et financiers la veille électronique d'articles économiques est cruciale. Maintenir une veille efficace implique de cibler les articles à consulter, car la charge d'information est importante. Pour répondre à cette problématique, nous proposons un système novateur de recommandation d'articles, car il s'appuie sur l'intégration d'une description sémantique des items et des profils basés sur une modélisation ontologique des connaissances. Nous appuyons notre système de recommandation sur un modèle vectoriel intrinsèquement efficace que nous avons perfectionné pour pallier la confusion native de ces modèles entre les notions de similarité et de pertinence qui ne permet pas de prendre en compte les effets de la différence dans la précision des descriptions sémantiques des profils et articles, sur la perception de la pertinence pour l'utilisateur. Nous présentons donc dans cet article une nouvelle méthode d'évaluation de la pertinence adaptée au modèle vectoriel.

1 Introduction

Afin de rester en phase avec les tendances actuelles du marché, le processus de prise de décision dans le domaine économique nécessite la centralisation et l'apport de grandes quantités d'informations. Pour cela, les hommes d'affaires, les entrepreneurs et les vendeurs doivent parfaitement connaître leur environnement. Cela signifie qu'il faut maintenir une veille économique constante facilitant l'identification des perspectives d'affaires, permettant de décrocher de nouveaux contrats. Cette veille incontournable est cependant complexe à assurer, c'est pourquoi nous proposons un outil efficace de recommandation d'articles régionaux d'actualités économiques utilisant des représentations sémantiques communes des connaissances afin de décrire les besoins de l'utilisateur et les informations permettant d'y répondre.

Notre approche s'établit sur l'adéquation de la recommandation aux besoins des utilisateurs, ainsi avons-nous mené une enquête auprès des clients (lecteurs) afin de définir les critères qui pourraient permettre la personnalisation du contenu de la revue. Les résultats de l'enquête ainsi que la connaissance des experts du domaine ont permis de mettre en avant trois critères principaux : les *Thèmes* (principaux évènements économiques traités dans l'article), les *Secteurs économiques* dont traitent l'article et les *Localisations*. Cette étude nous permet

de baser de manière adéquate notre système de recommandation sur le contenu des articles économiques. Pour cela des méthodes de Traitement Automatisé du Langage Naturel, ainsi qu'un travail manuel d'indexation sont mis en œuvre afin d'affecter au mieux les trois vocabulaires contrôlés décrivant ces critères principaux d'indexation. Ces critères sont communs aux descriptions des profils des lecteurs et des articles, nous permettant de rapprocher ces deux indexations selon des mesures de *similarité* et de *pertinence*.

Ces deux indicateurs de similarité et pertinence sont au cœur de cet article, car comme nous le montrons dans l'état de l'art en section 2, un amalgame est fait entre ces deux notions. Nous les distinguons donc et les définissons en section 4. Par ailleurs, le second point important de notre démarche est l'*expansion* des profils des lecteurs et des articles par notre base de connaissances ontologique que nous présentons en section 3. Enfin, nous évaluons ces deux axes : l'expansion ontologique de nos vecteurs d'une part et l'intégration d'une mesure de pertinence appelée *Relevancy measure* d'autre part avant de conclure et de présenter les travaux qui étendent cette approche.

2 Etat de l'art

Deux principaux systèmes de recommandation sont distingués : les systèmes dits de *filtrage collaboratif* et les systèmes *basés sur le contenu*. La synthèse de (Rao et Talwar, 2008) propose une comparaison générale des principaux avantages et inconvénients de ces deux systèmes de recommandation. Notre besoin de recommandation rapide d'articles économiques, chaque jour nouveaux, exclut la solution de systèmes à base de filtrage collaboratif. En effet, ces systèmes nécessitent qu'un nombre suffisant d'utilisateurs aient lu chacun des articles avant d'être capables de les recommander. Ces systèmes peinent donc à recommander de nouveaux items. En outre, nous devons être en mesure de recommander des profils d'utilisateurs très particuliers, car certains besoins clients peuvent être uniques. Cela n'est pas possible avec un système de filtrage collaboratif car la recommandation des items se fait en fonction des appétences de personnes ayant un profil similaire. Nous nous sommes donc tournés vers les systèmes basés sur le contenu, plus adaptés à nos besoins. Les avantages des systèmes de recommandation basés sur le contenu dans le cadre de la recommandation d'articles d'actualité sont également développés dans (Liu et al., 2010).

Il existe de nombreux systèmes de recommandation qui fonctionnent sans utilisation de connaissances supplémentaires (Liu et al., 2010), (Billsus et Pazzani, 1999) et (Resnick et al., 1994) mais (IJntema et al., 2010) ont montré que l'utilisation de connaissances extérieures peut améliorer la recommandation. Ils parlent alors de systèmes de recommandation basés sur la *sémantique* pour qualifier des systèmes basés sur le contenu utilisant des connaissances externes. Les systèmes de recommandation basés sur la sémantique utilisent des connaissances lexicales (Getahun et al., 2009), comme WordNet (Fellbaum, 1998), ou alors des connaissances de domaine (Middleton et al., 2004), voire une combinaison des deux (IJntema et al., 2010) dans l'objectif d'améliorer les performances du système. Les ontologies utilisées par ces systèmes existent déjà ou sont créées à la main, et maintenues. Contrairement à ces systèmes, notre base de connaissances est utilisée comme index, les articles et les profils y sont définis sémantiquement.

Le modèle vectoriel (Salton, 1971) consiste en la représentation des items à recommander (dans notre cas des articles), ainsi que parfois du besoin (requêtes, profils) sous la forme

de vecteurs. Cette présentation permet l'utilisation de différentes métriques afin de les comparer. Dans cet article nous utilisons les similarités cosinus et Jaccard ainsi que la distance Euclidienne. Beaucoup de systèmes de recommandation basés sur le contenu les utilisent afin de réaliser des comparaisons, que ce soit entre items, ou entre item et profil ((IJntema et al., 2010) (Middleton et al., 2004) (Getahun et al., 2009) (Billsus et Pazzani, 1999) (Ahn et al., 2007)). Cette méthode d'algèbre linaire présente deux principaux avantages : non seulement elle fournit un résultat non binaire, permettant donc d'ordonner les résultats des systèmes de recommandation, mais elle permet également des calculs rapides et une bonne résistance à la montée en charge.

Par ailleurs, des méthodes de recherche d'information peuvent être utilisées afin de prendre en compte cette connaissance tout en utilisant une modélisation vectorielle. L'approche proposée par (Voorhees, 1994) utilise la base de connaissance lexicale WordNet (Fellbaum, 1998) afin d'améliorer la gestion de l'hétérogénéité du langage naturel et donc d'améliorer la compréhension des besoins de l'utilisateur. L'idée est d'ajouter de l'information aux requêtes des utilisateurs (*expansion de requêtes*). Cette méthode permet d'augmenter le *rappel* [1] dans l'objectif d'améliorer les performances globales du système.

Nous avons transposé cette méthode aux systèmes de recommandation. (Middleton et al., 2004) utilise cette méthode sans la nommer. (IJntema et al., 2010) y a également recours, mais contrairement à Middleton, il utilise d'autres relations ontologiques que " is_a " pour étendre le profil de l'utilisateur.

Nous avons constaté que les notions de similarité et de pertinence sont généralement confondues dans les systèmes utilisant une modélisation vectorielle. Définir la pertinence comme une similarité ne permet pas de prendre en compte les différents degrés de spécificité dans la description du besoin des utilisateurs. Cette description est pourtant rendue possible par l'utilisation de connaissances externes au système. Nous proposons donc une mesure d'évaluation de la pertinence, *Relevancy measure*, utilisant les notions de similarité, mais prenant en compte la perception de la pertinence par l'utilisateur.

3 Vectorisation

Notre objectif concerne la recommandation d'articles économiques produits par une société auprès de ses lecteurs abonnés. Afin de répondre à cette problématique, nous proposons une approche en deux phases. La première consiste à créer une représentation des besoins des utilisateurs et du contenu des articles. La seconde étape s'attache à la recommandation de ces articles auprès de ces utilisateurs. Dans cette partie nous abordons la première étape de notre démarche : la création des vecteurs utilisateurs et articles, puis leur expansion sémantique.

3.1 Génération des vecteurs

Afin de définir le contenu des articles ainsi que les profils des lecteurs, un système d'indexation a été développé. Il permet l'utilisation d'un référentiel commun d'indexation pour les articles et profils, facilitant leur comparaison et donc la recommandation.

1. Nombre d'articles correctement considérés pertinents par rapport au nombre d'articles réellement pertinents.

3.1.1 Indexation d'articles

L'indexation est l'étape où le lien est fait dans la base de connaissances entre l'article et les connaissances qui lui sont associées, cela permet la création d'une représentation compréhensible par la machine du contenu de chaque article. Les articles sont indexés de façon semi-automatique, après leur rédaction de façon supervisée par leurs auteurs. La plate-forme GATE (Cunningham, 2002) est utilisée pour l'analyse des articles et l'extraction des informations. Les informations non structurées contenues dans les articles sont analysées. Deux types d'information peuvent être distingués : les informations *explicites* (par exemple les lieux, les personnes, les organisations, etc) et les informations *implicites* (par exemple, le thème de chaque article ou les secteurs économiques concernés). Les résultats de cette analyse sont ensuite vérifiés, corrigés et validés manuellement par leur rédacteur. Les vecteurs de description de chaque article utilisés lors de la comparaison avec les profils contiennent les instances des critères avec lesquels ils sont en relation directe dans la base de connaissances.

3.1.2 Indexation des profils des lecteurs

L'indexation des profils de lecteurs s'opère lors de leur inscription. Les vendeurs de la société en charge de la compréhension des besoins de chaque client, proposent une période d'essai gratuite qui permet à un expert de créer un premier profil à chacun des clients. Les profils sont indexés en fonction des mêmes critères que les articles. Ils sont décrits dans la même base de connaissances que les articles afin de faciliter leur rapprochement. Une interface permet à l'expert la création manuelle des profils, cela permet d'éviter le problème du démarrage à froid, commun aux systèmes de recommandation basés sur le contenu (Rao et Talwar, 2008). De façon similaire aux articles, les vecteurs de description de chaque profil utilisés lors de la comparaison avec les articles contiennent les instances des critères avec lesquels ils sont en relation directe dans la base de connaissances.

3.2 Expansion de vecteurs

Définition d'Ontologie. Nous caractérisons une ontologie suivant la définition de (Ehrig et al., 2004) :

$$O = (C, T, \leq_C, \leq_T, R, A, \sigma_A, \sigma_R, \leq_R, \leq_A)$$

avec C, T, R, A des ensembles disjoints de concepts, types de données, relations et attributs, $\leq_C, \leq_T, \leq_R, \leq_A$ les hiérarchies de classes, type de données, relation et attributs, et σ_A, σ_R des fonctions qui produisent une signature pour chaque $\sigma_A : A \rightarrow C \times T$ attribut et $\sigma_R : R \rightarrow C \times C$ relation.

Définition de Base de connaissances. Nous définissons une base de connaissances par :

$$K = (C, T, R, A, I, V, i_C, i_T, i_R, i_A)$$

avec C, T, R, A, I, V des ensembles disjoints de concepts, type de données, relations, attributs, instance et valeurs de données. i_C la fonction d'instanciation des classes $i_C : C \rightarrow 2^I$. i_T est la fonction d'instanciation des types de données $i_T : T \rightarrow 2^V$. i_R est la fonction d'instanciation des relations $i_R : R \rightarrow 2^{I \times I}$. i_A est la fonction d'instanciation des attributs $i_A : A \rightarrow 2^{I \times V}$.

Conception de la base de connaissances. Une fois les vecteurs d'articles et de profils constitués, nous les enrichissons grâce à notre base de connaissances, figure 1, composée de plusieurs ontologies selon les principes des ontologies modulaires (d'Aquin et al., 2009) :

- une ontologie de *domaine* instanciant les secteurs d'activité et événements économiques,
- une ontologie *générale* ayant pour charge dans un premier temps des paramètres de géolocalisation et la temporalité,
- une ontologie du *système* instanciant les profils et les articles.

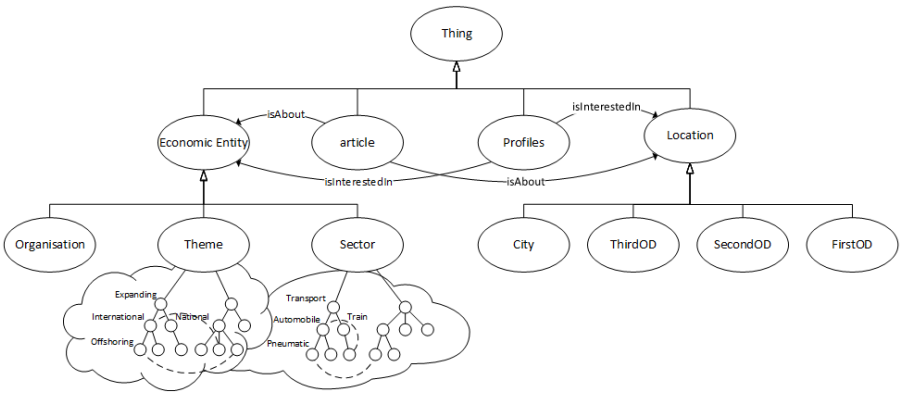


FIG. 1 – *Présentation d'une partie de la structure de l'ontologie.*

L'ontologie du système est peuplée par les instances de profils et d'articles ainsi que les relations « isAbout » et «isInterestedIn » qui permettent d'associer respectivement les profils et les articles aux instances des critères définis dans la base de connaissances. L'objectif étant la création d'une représentation sémantique, compréhensible par la machine, des besoins et intérêts de chaque utilisateur ainsi que du contenu de chaque article.

Expansion de vecteurs. Notons, que lors de la vectorisation, nous ne tenons pas compte des connaissances externes. Ainsi si un article concerne Dijon et qu'un profil s'intéresse à la Bourgogne, le système, ne sachant pas que Dijon est en Bourgogne, ne considérera pas l'article. La modélisation vectorielle ne permet en pratique pas de prendre en compte la relation qui existe entre Dijon et Bourgogne dans notre exemple. En effet, (Voorhees, 1994) montre que toutes les dimensions sont orthogonales dans le modèle vectoriel et qu'ainsi, tous les éléments de chaque vecteur sont considérés comme indépendants. Nous nous intéressons donc à prendre en compte cette connaissance externe par une expansion des vecteurs.

Dans les travaux sur les systèmes de recherche d'informations de (Voorhees, 1994) les requêtes des utilisateurs sont représentées sous forme de vecteurs. Ces vecteurs sont étendus par l'ajout de synonymes et de méronymes [2]. Plus récemment cette méthode a été adaptée aux systèmes de recommandations par (IJntema et al., 2010). Elle prend la forme d'expansion de vecteurs de profil. Dans ces travaux seuls les vecteurs décrivant le besoin d'information sont étendus, cela dans l'objectif d'augmenter les performances des systèmes en augmentant les

2. Désigne une sous partie, par exemple *toit* est un méronyme de *maison*.

mesures de rappel. Dans ces systèmes, les informations ajoutées aux vecteurs sont des informations en relation directe dans les bases de connaissances externes utilisées avec les informations déjà présentes dans les vecteurs. L'ajout d'instances en relation directe avec les instances déjà présentes dans le vecteur profil ne nous permettrait pas à partir de Bourgogne d'ajouter Dijon, mais seulement Côte d'Or et les autres départements de la région. La pertinence de l'article pour le profil serait donc toujours nulle. L'ajout des instances en relation via transitivité avec les instances déjà présentes dans le vecteur profil, permet alors d'ajouter Dijon au vecteur et ainsi de prendre en compte que l'article traite bien d'un contenu en relation avec le contenu souhaité par l'utilisateur. Seulement, l'ajout par transitivité ajoute non seulement Dijon, mais aussi les quatre mille autres communes de la région. La pertinence de l'article pour le profil ne serait donc pas nulle, mais tout de même très faible, alors qu'elle devrait être relativement forte. Afin de pallier à ce problème, nous étendons les vecteurs articles en plus des vecteurs profils. De plus, afin de limiter la taille du vecteur, nous ajoutons les ancêtres de l'instance sélectionnée et non les descendants. Notre méthode se rapporte aux méthodes de recherche de la profondeur de l'ancêtre commun dans un graphe pour l'évaluer la distance sémantique entre deux nœuds.

4 Similarité versus pertinence

La tâche de recommandation est basée sur la comparaison entre profils et articles et s'appuie donc sur leur index commun défini dans la base de connaissances. Les méthodes classiques de recherche d'information ou de recommandation utilisant une modélisation vectorielle déduisent directement la pertinence de la mesure de similarité entre le vecteur représentant le profil et celui représentant l'article. La base théorique, est que le profil peut être considéré comme un article idéal, donc plus un article est similaire à cet article idéal (profil) plus il est pertinent, plus il correspond aux intérêts et besoins de l'utilisateur.

Cependant, nous introduisons ici notre notion de précision entre un profil et un article. En effet, afin de décrire le contenu d'un article, ou les besoins d'un profil, les descripteurs utilisés pour chaque critère peuvent être plus ou moins précis.

Le fait d'avoir, pour un critère donné, un article proposant un contenu plus précis que celui recherché par un profil, n'a pas les mêmes répercussions sur la perception de la pertinence que le fait d'avoir un profil intéressé pour un critère donné par un contenu plus précis que celui proposé par un article. Cela doit donc être pris en compte lors de l'évaluation de la pertinence.

Pour exposer la distinction que nous opérons entre similarité et précision, nous rappelons tout d'abord une définition de similarité, puis nous introduisons notre définition de pertinence.

4.1 Similarité

Puisque nous travaillons avec des modèles vectoriels, nous pouvons utiliser la définition suivante de mesure de similarité entre articles et profils :

$$SimilariteF(\overrightarrow{a}, \overrightarrow{p}) = \frac{\sum \omega_c Similarite_c(\overrightarrow{a_c}, \overrightarrow{p_c})}{\sum \omega_c}$$

$Similarite_c(\overrightarrow{a}, \overrightarrow{p})$ étant la mesure de similarité entre le profil $\overrightarrow{p}$ et l'article $\overrightarrow{a}$ pour le critère spécifique c, tel que $c \in \{Themes, Secteur, Location\}$. Nous utiliserons, lors de

l'évaluation, alternativement les mesures : Cosinus, Jaccard et Euclide en tant que mesure de similarité $Similarite_c(\overrightarrow{a}, \overrightarrow{p})$. ω_c est le coefficient de pondération défini pour le critère c. $\forall i_{x,c} \in I'_c,\ \overrightarrow{p_c} =< i_{1,c}, i_{2,c}, ..., i_{n,c} >$ et $\overrightarrow{a_c} =< i_{1,c}, i_{2,c}, ..., i_{m,c} >$ avec n et $m \in [0; |I'_c|]$. Un ou plusieurs concepts peuvent être définis pour chaque critère. Pour le critère localisation par exemple, nous avons les concepts, pays région, département et ville. Ainsi donc dans la liste des instances disponibles pour la description des articles et profils sur chaque critère il est possible d'utiliser toutes les instances de tous ces concepts. $Similarite_c(\overrightarrow{a}, \overrightarrow{p})$ étant la mesure de similarité entre le profil $\overrightarrow{p}$ et l'article $\overrightarrow{a}$ pour le critère spécifique c, tel que $c \in \{Themes, Secteur, Location\}$. Nous utiliserons, lors de l'évaluation, alternativement les mesures : Cosinus, Jaccard et Euclide en tant que mesure de similarité $Similarite_c(\overrightarrow{a}, \overrightarrow{p})$. ω_c est le coefficient de pondération défini pour le critère c. $\forall i_{x,c} \in I'_c,\ \overrightarrow{p_c} =< i_{1,c}, i_{2,c}, ..., i_{n,c} >$ et $\overrightarrow{a_c} =< i_{1,c}, i_{2,c}, ..., i_{m,c} >$ avec n et $m \in [0; |I'_c|]$. Un ou plusieurs concepts peuvent être définis pour chaque critère. Pour le critère localisation par exemple, nous avons les concepts, pays région, département et ville. Ainsi donc dans la liste des instances disponibles pour la description des articles et profils sur chaque critère il est possible d'utiliser toutes les instances de tous ces concepts.

4.2 Pertinence pour un critère

Ici, nous apportons la distinction entre *pertinence* et *similarité*. Nous nous concentrons notamment sur la gestion de la différence de précision entre la définition des profils et des articles et son influence sur la mesure pertinence. Par exemple, Dijon étant pour le critère localisation, plus précis que Bourgogne, si un article traite de Dijon et un profil montre un intérêt pour la Bourgogne, la pertinence doit être plus élevée que dans le cas inverse. Afin d'intégrer ce paramètre, nous utilisons un vecteur intermédiaire pour chaque critère. Le sous-vecteur $\overrightarrow{s_c}$ est composé des instances communes entre le vecteur de l'article $\overrightarrow{a_c}$ et celui du profil $\overrightarrow{p_c}$.

Dans la hiérarchie des concepts, les concepts les plus généraux englobent d'autres concepts plus spécifiques, il en va de même dans la hiérarchie d'instances. Les instances hautes dans la hiérarchie sont moins précises que les instances basses.

Si un article traite d'instances de bas niveau et qu'un profil s'intéresse à des instances de haut niveau (de la même branche) la pertinence de l'article pour le profil doit être plus élevée que dans le cas contraire. Car si l'article est plus général que le profil alors il y a une perte de précision par rapport au besoin de l'utilisateur et qui doit être répercutée par une perte de pertinence. Ainsi, nous définissons la pertinence pour un critère c par :

$$Pertinence_c(\overrightarrow{a_c}, \overrightarrow{p_c}) = \frac{\omega'_{1,c} \times Similarite_c(\overrightarrow{a_c}, \overrightarrow{s_c}) + \omega'_{2,c} \times Similarite_c(\overrightarrow{p_c}, \overrightarrow{s_c})}{\omega'_{1,c} + \omega'_{2,c}}$$

Avec S_c le sous-ensemble commun d'éléments de l'ensemble d'instances en relation à la fois avec le profil $I'_{p,c}$ et l'article $I'_{a,c}$; $S_c = I'_{p,c} \cap I'_{a,c}$. $\forall i_{x,c} \in S_c$ le vecteur $\overrightarrow{s_c}$ est composé des éléments de l'ensemble S_c ; $\overrightarrow{s_c} =< i_{1,c}, i_{2,c}, ..., i_{t,c} >$.

Avec cette méthode, il est possible de pondérer de plusieurs façons la différence de précision entre profils et articles, afin de l'adapter aux besoins. Dans notre cas, nous utilisons $\omega'_{1,c} = 1$ et $\omega'_{2,c} = 4$ car nous considérons que la perte de précision du profil par rapport à

l'article ne doit pas influencer plus de 20% du résultat. Par contre, la perte de précision de l'article par rapport au profil doit influencer fortement le résultat, ici 80%. Il est toutefois possible de modifier ces valeurs, et il est aussi possible de les gérer de façon distincte selon le critère considéré.

4.3 Pertinence globale : *Relevancy measure*

La pertinence globale $Relevancy(\overrightarrow{a}, \overrightarrow{p})$ est la somme des mesures de pertinence pour chacun des critères, éventuellement pondérées. Cette mesure est utilisée dans notre prototype pour trier les résultats (articles) proposés à l'utilisateur en fonction de son profil :

$$Relevancy(\overrightarrow{a}, \overrightarrow{p}) = \frac{\sum \omega_c * Pertinence_c(\overrightarrow{a_c}, \overrightarrow{p_c})}{\sum \omega_c}$$

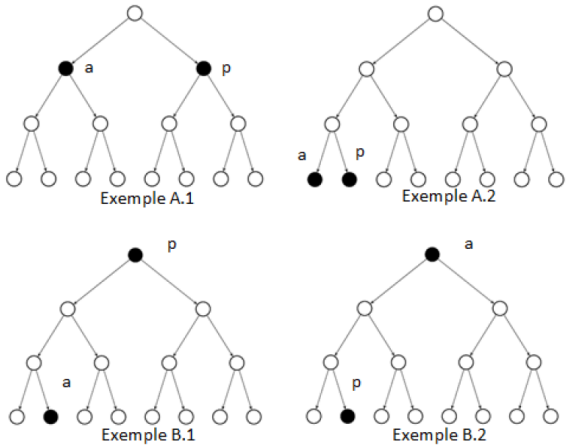


FIG. 2 – *Exemples de profils p et d'articles a pour un critère donné*

Pour illustrer notre propos, considérons la figure 2. Grâce à notre approche, la valeur de pertinence pour le cas A.2 est plus élevée que dans le cas A.1, car dans le cas A.2, a et p ont des ancêtres communs plus généraux que dans le cas A.2. En outre, les cas B1 et B2 illustrent le problème de la précision. Avec notre méthode asymétrique, la valeur de similarité entre a et p dans le cas B.1 est plus élevée que dans le cas B.2. Les besoins des utilisateurs sont plus spécifiques que les informations de l'article (sur ce critère), donc l'article est moins pertinent pour l'utilisateur.

5 Expérimentations

Nous avons défini une méthode couplant une expansion des vecteurs de profils et d'articles et une prise en compte de la différence de précision entre les descriptions fournies par ces vecteurs. Nous évaluons donc ici ces deux supports fondamentaux de notre approche.
Pour cela nous avons élaboré un jeu de test comportant 10 profils de lecteurs et 70 articles

(ce qui correspond à la production quotidienne d'articles pour cette société). Ce jeu de données est suffisamment conséquent pour répondre aux besoins de l'évaluation, mais de taille raisonnable pour permettre à un expert d'établir une recommandation manuelle de référence. Les mesures de pertinence expérimentée ici sont appliquées dans un espace vectoriel permettant l'utilisation de nombreuses méthodes d'évaluation de la similarité. Nous avons donc établi notre benchmark sur trois des plus classiques : Similarité Cosinus, Similarité Jaccard et distance euclidienne.

5.1 Méthodes d'évaluation

Evaluation Binaire Pour évaluer la recommandation de ces algorithmes nous nous basons sur les mesures classiques de *précision* [3] et de *rappel*.
Cette évaluation étant binaire, nous avons besoin de résultats binaires à partir d'algorithmes de recommandation. Or ils fournissent des articles de façon triée à l'aide d'une valeur de pertinence (entre 0 et 1). Nous avons donc défini un seuil au-delà duquel un item est recommandé et en dessous duquel il ne l'est pas. Le seuil de 0,5 a été choisi pour l'évaluation de la pertinence binaire.

Afin de considérer à la fois la précision [4] et le rappel, nous utilisons, la F-mesure proposé par (Rijsbergen, 1979). Cette métrique produit des scores allant de 0 à 1.

Evaluation de rang. Pour évaluer l'ordre des articles recommandés par les algorithmes, nous utilisons les deux mesures de corrélation linéaire de rang les plus populaires : le rho de Spearman et le tau de Kendall. Ces deux métriques produisent des scores allant de -1 à 1. 0 étant l'absence de similitude, 1 la similitude complète et -1 l'inverse.

5.2 Evaluation de l'expansion des vecteurs

Dans cette partie nous évaluons l'intérêt l'expansion des vecteurs profils et articles. Pour cela nous restons dans un contexte où la pertinence d'un article pour un profil est déduite directement de leur similarité. Nous confrontons deux algorithmes : celui utilisant le modèle vectoriel classique sans expansion (méthode *C*), et avec expansion des deux vecteurs articles et profils par l'ajout d'instances (méthode *B*).

Algorithmes	Précision	Rappel	F1-mesure	Tau de Kendall	Rho de Spearman
COSINUS B	**0.916**	0.453	**0.607**	**0.830**	**0.894**
COSINUS C	0.883	0.181	0.301	0.713	0.694
JACCARD B	0.883	0.150	0.256	0.819	0.886
JACCARD C	0.883	0.150	0.256	0.712	0.693
EUCLIDE B	0.396	**0.985**	0.565	0.649	0.734
EUCLIDE C	0.549	0.495	0.521	0.549	0.615

TAB. 1 – *Comparaison vecteurs étendus et non étendus par des mesures d'évaluation binaires et de corrélation de rang.*

3. Nombre d'articles correctement considérés pertinents par rapport au nombre d'articles considérés pertinents.
4. Dans cette section, *précision* désigne la mesure de la précision de la recommandation.

Les résultats de l'évaluation de la recommandation en utilisant comme mesure de pertinence la similarité directe entre les vecteurs classiques et les vecteurs étendus sont présentés dans la table 1. Pour chaque algorithme d'évaluation de la pertinence par mesure de similarité (Jaccard, Cosinus, Euclide), la F1-mesure montre que lorsque les vecteurs sont étendus afin de prendre en compte les connaissances de la base de connaissances (méthode *B*), les résultats sont au moins aussi bons qu'avec les vecteurs classiques (méthode *C*). L'évaluation de l'ordre des articles rangés par les différents algorithmes montre les mêmes résultats. Par ailleurs nous pouvons observer une perte de précision avec la distance euclidienne. Ce problème de perte de précision a déjà été expliqué par (Voorhees, 1994) avec sa propre méthode d'expansion du vecteur. En effet, l'expansion des vecteurs vise l'amélioration du rappel et comme le montrent les résultats, cela peut avoir un coût en précision. Nous confirmons ici les résultats de (Middleton et al., 2004) et (IJntema et al., 2010) quant à l'intérêt de l'expansion de vecteurs, et montrons que notre approche d'expansion ontologique s'inscrit dans ce constat.

5.3 Evaluation de la *Relevancy measure*

Nous nous intéressons ici à l'évaluation de l'apport fourni par la prise en compte de la différence de précision entre la description des profils et des articles lors de la mesure de la pertinence. La métrique *Relevancy measure*, permet de prendre en compte lors de l'évaluation de la pertinence d'un article son adéquation avec le degré de spécificité par rapport à celui souhaité par le lecteur. Ainsi nous comparons dans cette section les résultats fournis lors de l'utilisation de vecteur étendu par une mesure de pertinence directement déduite de la similarité des vecteurs (méthode *B*) et par notre métrique *Relevancy measure* (méthode *A*).

Algorithmes	Précision	Rappel	F1-mesure	Tau de Kendall	Rho de Spearman
COSINUS A	0.856	0.971	**0.910**	**0.836**	**0.898**
COSINUS B	0.916	0.453	0.607	0.830	0.894
JACCARD A	**0.928**	0.588	0.720	0.836	0.896
JACCARD B	0.883	0.150	0.256	0.819	0.886
EUCLIDE A	0.566	0.971	0.715	0.728	0.817
EUCLIDE B	0.396	**0.985**	0.565	0.649	0.734

TAB. 2 – *Comparaison des mesures de pertinence avec et sans prise en compte des différences de précisions des descriptions par évaluation binaires et de corrélation de rang.*

Les résultats de l'évaluation de la recommandation permettant de distinguer d'une part la méthode de mesure de la pertinence basée sur la similarité directe et notre méthode *Relevancy measure* d'autre part, sont présentées dans la table 2. Elles utilisent toutes les deux des vecteurs étendus.

Les deux méthodes d'évaluation (Tau de Kendall ou Rho de Spearman), indiquent que la méthode A fournit un meilleur classement d'articles. En ce qui concerne la F1-mesure, elle indique aussi que la méthode A fournit les meilleurs résultats, c'est à dire proposant le meilleur rapport entre précision et rappel.

En conclusion, ces évaluations témoignent de la pertinence de notre approche et mettent en avant que les paramètres les plus efficaces pour la recommandation d'articles sont d'effectuer une expansion des vecteurs profils et articles, et de prendre en compte les différences de pré-

cision entre l'expression du besoin et la description du contenu des articles comme le permet notre méthode.

6 Conclusion et travaux futurs

Dans cet article, nous avons présenté l'adaptation d'un système basé sur le modèle vectoriel de recommandation à notre méthode spécifique d'indexation qui définit sémantiquement les articles et les profils dans une base de connaissances par l'intermédiaire des relations avec les connaissances du domaine prédéfinies dans celle-ci. Nous avons présenté notre approche qui répond aux manques de l'état de l'art sur les points de gestion du degré de precision du besoin et de leur comparaison avec l'offre.
Nous avons présenté notre approche qui intègre une *expansion ontologique* des vecteurs d'articles et de profils d'utilisateurs. Nous avons distingué et défini les notions de *similarité* et de *pertinence*. Enfin, nous avons évalué nos algorithmes en utilisant à la fois une méthode d'évaluation binaire et de corrélation de rang afin de montrer les apports de notre approche. Cette évaluation montre que notre approche fournit le meilleur classement d'articles, notamment quand elle est utilisée avec une mesure de comparaison de vecteurs *cosinus*.
Nous projetons d'évaluer notre méthode en intégrant les comportements de l'utilisateur lors de l'utilisation de l'outil de recommandation. De plus, nous enrichissons la base de connaissances afin d'améliorer encore la pertinence.

Notre démarche étant expérimentalement validée, le prolongement de ces travaux s'intéresse à son passage à l'échelle. Cependant une autre méthode que la méthode vectorielle devra être appliquée car selon nos premiers travaux, elle génère des effets de bord lorsque le volume de données devient conséquent.
Par ailleurs, nous souhaitons améliorer la recommandation en nous appuyant sur des algorithmes provenant de l'étude des graphes afin de calculer la pertinence entre les profils et les articles. L'information que nous utilisons est déjà structurée dans une ontologie (plus efficace dans la transmission sémantique que le modèle vectoriel), il semble donc inutile de restructurer l'information sous forme de vecteurs pour effectuer des comparaisons entre les articles et les profils à moins que cela n'apporte un réel gain en temps de calcul. Certaines approches permettant de comparer des instances dans une base de connaissances existent déjà (Albertoni et Martino, 2006), (Ehrig et al., 2004) et sur lesquelles nous nous appuierons.

Références

Ahn, J.-w., P. Brusilovsky, J. Grady, D. He, et S. Y. Syn (2007). Open user profiles for adaptive news systems : help or harm ? pp. 11. ACM Press.

Albertoni, R. et M. D. Martino (2006). Semantic similarity of ontology instances tailored on the application context. In *Lecture Notes in Computer Science Volume 4275*, pp. 1020–1038. Springer.

Billsus, D. et M. J. Pazzani (1999). A personal news agent that talks, learns and explains. pp. 268–275. ACM Press.

Cunningham, H. (2002). GATE, a general architecture for text engineering. *Computers and the Humanities 36*(2), 223–254.

d'Aquin, M., A. Schlicht, H. Stuckenschmidt, et M. Sabou (2009). Criteria and evaluation for ontology modularization techniques. In *Modular Ontologies*, Volume 5445, pp. 67–89. Springer Berlin Heidelberg.

Ehrig, M., P. Haase, M. Hefke, et N. Stojanovic (2004). Similarity for ontologies - a comprehensive framework. In *In Workshop Enterprise Modelling and Ontology : Ingredients for Interoperability, at PAKM 2004*.

Fellbaum, C. (1998). *WordNet : an electronic lexical database*. Cambridge, Mass : MIT Press.

Getahun, F., J. Tekli, R. Chbeir, M. Viviani, et K. Yetongnon (2009). Relating RSS News/Items. In *Web Engineering*, Number 5648 in Lecture Notes in Computer Science, pp. 442–452. Springer Berlin Heidelberg.

IJntema, W., F. Goossen, F. Frasincar, et F. Hogenboom (2010). Ontology-based news recommendation. pp. 1. ACM Press.

Liu, J., P. Dolan, et E. R. Pedersen (2010). Personalized news recommendation based on click behavior. pp. 31. ACM Press.

Middleton, S. E., N. R. Shadbolt, et D. C. De Roure (2004). Ontological user profiling in recommender systems. *ACM Transactions on Information Systems 22*(1), 54–88.

Rao, K. et V. Talwar (2008). Application domain and functional classification of recommender Systems—A survey. *DESIDOC Journal of Library and Information Technology 28*(3), 17–35.

Resnick, P., N. Iacovou, M. Suchak, P. Bergstrom, et J. Riedl (1994). GroupLens : an open architecture for collaborative filtering of netnews. pp. 175–186. ACM Press.

Rijsbergen, C. J. V. (1979). *Information Retrieval* (2nd ed.). Newton, MA, USA : Butterworth-Heinemann.

Salton, G. (1971). The SMART retrieval system - experiments in automatic document processing.

Voorhees, E. M. (1994). Query expansion using lexical-semantic relations. In *SIGIR '94*, pp. 61–69. Springer London.

Summary

Today in the commercial and financial sectors, staying informed about economic news is crucial and involves targeting good articles to read, because the huge amount of information. To address this problem, we propose an innovative article recommendation system, based on the integration of a semantic description of articles and on a knowledge ontological model. We support our recommendation system on an intrinsically efficient vector model that we have perfected to overcome the confusion existing in models between the concepts of similarity and relevancy that does not take into account the effects of the difference in the accuracy of the semantic descriptions precision between profiles and articles, on the perceived relevancy to the user. We present in this paper a new evaluation of the relevancy adapted to vector model.

Une approche Web sémantique et combinatoire pour un système de recommandation sensible au contexte appliqué à l'apprentissage mobile

Fayrouz Soualah Alila[*,**], Christophe Nicolle[*] et Florence Mendes[*]

[*]LE2I, UMR CNRS 6306, Université de Bourgogne, Dijon, France
{fayrouz.soualah-alila, christophe.nicolle, florence.mendes}@checksem.fr

[**]CrossKnowledge, 4 Port aux Vins avenue, 92150 Suresnes, Paris, France
fayrouz.soualah-alila@crossknowledge.com

Résumé. Au vu de l'émergence rapide des nouvelles technologies mobiles et la croissance des offres et besoins d'une société en mouvement en formation, les travaux se multiplient pour identifier de nouvelles plateformes d'apprentissage pertinentes afin d'améliorer et faciliter le processus d'apprentissage à distance. La prochaine étape de l'apprentissage à distance est naturellement le port de l'e-learning (apprentissage électronique) vers les nouveaux systèmes mobiles. On parle alors de m-learning (apprentissage mobile). La recherche d'informations dans le domaine du m-learning peut être définie comme une activité dont la finalité est de localiser et de délivrer des contenus d'apprentissage à un apprenant en fonction de son besoin en informations et de son contexte. Jusqu'à présent l'environnement d'apprentissage était soit défini par un cadre pédagogique soit imposé par le contenu d'apprentissage. Maintenant, nous cherchons, à l'inverse, à adapter le cadre pédagogique et le contenu d'apprentissage au contexte de l' apprenant.
Nos travaux de recherche portent sur le développement d'une nouvelle architecture pour le m-learning. Dans cette communication, nous présentons une définition du problème de recommandation de parcours de formation dans un contexte mobile. Nous présentons par la suite note approche pour développer un système de recommandation pour l'optimisation de l'offre m-learning.

1 Introduction

Ces dernières années ont été marquées par l'essor de l'apprentissage mobile ou m-learning, favorisé par le développement continu des nouvelles technologies mobiles. L'apprentissage devient situé, contextuel, et personnel. Ce phénomène pousse à l'évolution des méthodes d'apprentissage pour s'adapter à ce nouveau type d'apprentissage.
De nouveaux usages apportés dans le domaine de l'apprentissage se sont multipliés sous différentes modalités. Dans le cadre de l'apprentissage au sein des entreprises, nous cherchons à

développer un système m-learning dont les principaux enjeux sont : (1) l'apprentissage au travail quel que soit l'heure, le lieu, le dispositif de délivrance, les contraintes technologiques des processus d'apprentissage et adapté au profil de l'apprenant ; (2) l'apprentissage sans rupture au travers des différents contextes. Dans le cadre de nos travaux, nous proposons une approche pour un système m-learning contextuel et adaptatif intégrant des stratégies de recommandation de scénarios de formations sans risque de rupture.
Dans l'objectif de développer un tel système m-learning, nous commençons par identifier différents niveaux d'hétérogénéité : hétérogénéité sémantique et hétérogénéité d'usage :
D'un côté, en e-learning les ressources sont conçues et développées par des organisations et des formateurs différents, constituant généralement des contenus d'apprentissage autonomes mais aussi hétérogènes au niveau sémantique. En effet, des conflits sémantiques surviennent puisque les systèmes n'utilisent pas la même interprétation de l'information qui est définie différemment d'une organisation à l'autre. Les besoins immédiats demandent l'application de standards en vigueur pour rendre les contenus d'apprentissage réutilisables pour assurer l'interopérabilité sémantique des plateformes e-learning hétérogènes.
D'un autre côté, les apprenants qui sont les principaux acteurs d'une plateforme d'apprentissage, ont des connaissances et des objectifs différents et se situent dans des contextes d'apprentissages différents (hétérogénéité d'instant, hétérogénéité de durée, hétérogénéité de support visuel, hétérogénéité de niveau sonore, etc.). Il faut dans ce cas avoir une meilleure connaissance du contexte d'apprentissage et s'interroger efficacement sur les stratégies pédagogiques à mettre en place pour répondre au mieux aux besoins de chaque apprenant.

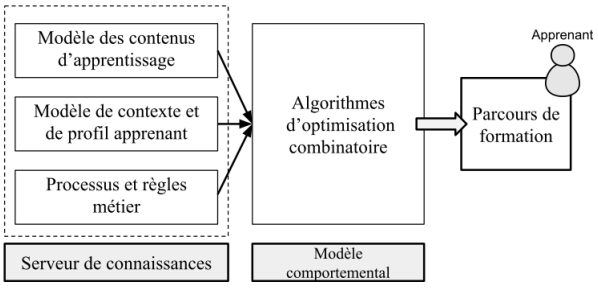


FIG. 1 – *Architecture générale du système m-learning*

Ce papier est dédié à la présentation de l'architecture d'un système m-learning dont la principale caractéristique réside dans sa capacité à construire des parcours de formation selon des contraintes contextuelles de l'apprenant. L'enjeu du système à construire est de répondre à la fois au verrou d'hétérogénéité sémantique (aspect statique du système) et au verrou d'hétérogénéité d'usage (aspect dynamique et adaptatif du système). Nous proposons une architecture d'une plateforme pour le m-learning articulée en deux parties : la première partie est constituée d'un serveur de connaissances où les données et les processus métiers sont modélisés par une ontologie évolutive et des règles métier, et la deuxième partie est basée sur des algorithmes de métaheuristiques permettant d'analyser les règles métier et l'ontologie pour permettre une bonne combinaison des contenus d'apprentissage (Fig 1).
Coupler les techniques de modélisation sémantique des contenus pédagogiques et du contexte

apprenant avec des algorithmes performants issus du domaine de l'optimisation combinatoire, constituent notre système de recommandation pour l'optimisation de l'offre m-learning.

2 Conception de l'ontologie des Learning Objects

2.1 Learning Object

Un contenu d'apprentissage est une instanciation d'objets pédagogiques, ou LOs (Learning Objects). L'idée fondamentale derrière la création des LOs est la possibilité de construire un parcours de formation autour de composants de petite taille qui peuvent être réutilisés plusieurs fois dans différents contextes d'apprentissage.Selon LTSC [1], un LO est définit comme "toute entité, sur un support numérique ou non, pouvant être utilisée, réutilisée et référencée au cours d'un processus de formation". Cette définition est complétée par (Abel, 2007) considérant qu'un LO est "un matériel qui peut être sélectionné, combiné avec d'autres LOs selon les besoins des apprenants. C'est aussi un matériel qui peut être recherché et indexé facilement".
Seulement ces LOs sont souvent conçus et développés par des organisations et des auteurs différents constituant généralement des contenus d'apprentissage autonomes et sémantiquement hétérogènes. Ils sont ainsi difficilement réutilisables car ils n'ont pas été conçus à cet effet. Il est alors indispensable de penser à une modélisation partagée des LOs en vue de les rendre facilement accessibles, exploitables, réutilisables et sémantiquement interopérables.
À l'instar de tout système d'information requérant un mécanisme de description formelle de ses ressources par des éléments de métadonnées, les systèmes d'apprentissage se démarquent aussi par la conception et la mise en place de ses propres modèles de description des LOs. Les métadonnées sont devenues l'un des mécanismes essentiels pour décrire, référencer et localiser des ressources dans le système. Nous proposons pour modéliser les LOs d'utiliser des informations issues des schémas de description des ressources pédagogiques par des métadonnées. Une solution peut provenir des initiatives de normalisation qui visent à établir des règles qui faciliteront le partage et la réutilisation des contenus pédagogiques(Grandbastien et al., 2008) (Ghebghoub et al., 2009).
Différentes normes ont été définies pour aider à l'élaboration de systèmes d'apprentissage, des LOs associés, leur représentation et leur interrelation. L'application de ces normes, garantit non seulement l'interopérabilité mais également la qualité du système (Grandbastien et al., 2008). Parmi ces normes, on peut citer LOM (lom), SCORM (sco) et IMS-LD (ims). LOM s'intéresse à la description des contenus d'apprentissage, SCORM à la structure des contenus, et IMS-LD au scénario d'apprentissage. Nous nous intéressant dans notre cas au standard LOM.
LOM (Learning Object Metadata) est un standard élaboré par le consortium IEEE. Il définit la structure d'une instance de métadonnées pour un LO. Il est constitué d'un ensemble de 80 éléments divisés en 9 catégories accomplissant chacune une fonction différente :

1. General : Cette catégorie regroupe les informations générales qui définissent l'objet pédagogique dans son ensemble. Eléments : titre, langue, mots-clés, etc.
2. Lifecycle : Cette catégorie décrit l'histoire et l'état actuel de l'objet pédagogique et des entités qui ont eu des répercussions sur l'objet pédagogique lors de son évolution. Eléments : version, statut, contribution, etc.

1. http ://ieeeltsc.org

3. Meta-Metadata : Cette catégorie décrit comment cette instance de métadonnées peut être identifiée, qui a créé cette instance de métadonnées, comment, quand et avec quelles références. Eléments : schéma de métadonnées, langue des métadonnées, etc.
4. Technical : Cette catégorie décrit les spécifications et les caractéristiques techniques de l'objet pédagogique. Eléments : format, taille du fichier, exigences techniques, etc.
5. Educational : Cette catégorie décrit les caractéristiques essentielles de l'objet pédagogique en matière d'éducation et de pédagogie. Eléments : type d'interactivité, type de contenu, rôle, âge et langue de l'utilisateur, etc.
6. Rights : Cette catégorie décrit les droits de propriété intellectuelle et les conditions d'usage de l'objet pédagogique. Eléments : copyright, droits, etc.
7. Relation : Cette catégorie définit les liens existants entre l'objet pédagogique et d'autres objets pédagogiques.
8. Annotation : Cette catégorie permet aux utilisateurs de partager leurs appréciations ou commentaires sur l'objet pédagogique.
9. Classification : Cette catégorie décrit comment l'objet pédagogique entre dans un système de classification spécifique.

Les descripteurs de LOM peuvent être utilisés dans la conception des systèmes m-learning pour l'indexation des LOs. Nous avons besoin dans ce cas d'implémenter ces descripteurs dans un langage structuré.

2.2 Ontologie des Learning Objects

La représentation du modèle abstrait dans un format spécifique est appelé binding. Aujourd'hui il existe 2 binding du schéma LOM : soit du binding XML, soit du binding RDF (Ghebghoub et al., 2009) :
Le binding XML est facile à implémenter, cependant il reste insuffisant pour la représentation de tous les éléments de LOM puisqu'il ne permet pas d'exprimer la sémantique de ces éléments.
Le binding RDF définit un ensemble de constructions RDF qui facilitent l'introduction des métadonnées de LOM dans le web, et il est complété par RDFS pour la définition des classes, des propriétés, etc. L'avantage de ce deuxième type de binding c'est qu'il rajoute de la sémantique aux éléments de LOM, sauf qu'il n'est pas assez expressif pour définir toutes les contraintes de LOM. Prenons l'exemple des éléments "Title" et "Entry" de la catégorie "General" qui sont des éléments obligatoires dans le LOM. En utilisant RDF et RDFS on ne peut préciser qu'une propriété est obligatoire ou contraindre son utilisation à une seule fois pour une ressource (Bourda, 2002). Comme deuxième exemple, RDF et RDFS ne permettent pas d'exprimer l'inverse d'une relation : ainsi, dire qu'un LO x "has part" un LO y, ne permettra pas d'induire que le LO y "is part of" LO x. Ce manque d'expressivité nous mène à penser à l'utilisation d'un autre formaliste plus puissant.
Afin de déterminer quel langage est le plus approprié pour résoudre le problème d'expressivité, nous nous somme penché sur l'identification de la logique de description ($\mathcal{LD}$) nécessaire. La $\mathcal{LD}$ est une famille de formalismes pour représenter les connaissances d'une façon structurée et formelle. Une caractéristique fondamentale de ces langages est qu'ils ont une sémantique descriptive formelle. Nous partons d'une logique minimale $\mathcal{ALC}$ et nous rajoutons à cette logique

les constructeurs nécessaires pour définir toutes les contraintes de LOM. Nous avons ainsi besoin des constructeurs suivants :

- Nominal $\mathcal{O}$: Exemple, l'élément "Statues" de la catégorie "Lifecycle" doit absolument avoir l'une de ces valeurs {draft, final, revised, unavailable}.
- Fonctionnalité $\mathcal{F}$: Exemple, restreindre l'utilisation de l'élément "Title" de la catégorie "General" à une seule fois.
- Restriction de nombre qualifié $\mathcal{Q}$: Exemple, fixer une cardinalité minimale de 1 pour l'utilisation de l'élément "Keyword" de la catégorie "General".
- Hiérarchie des relations $\mathcal{H}$: Exemple, l'élément "Type" de la catégorie "Relation" est traduit pas la relation "has type relation" et ses sous relations "is part of", "is version of", etc.
- Transitivité des relations $\mathcal{R}^*$: Exemple, LO x "is part of" LO y et LO y "is part of" LO z alors LO x "is part of" LO z.
- Relations inverses $\mathcal{I}$: Exemple, indiquer que la relation "is part of" est l'inverse de la relation "has part".

On peut ainsi conclure que notre $\mathcal{LD}$ appartient à la famille de description $\mathcal{SROIQ}^{(*)}$. Cette $\mathcal{LD}$ correspond au langage ontologique OWL, est plus particulièrement à du OWL-Full.
Utiliser une ontologie du LOM pour indexer les ressources pédagogiques permet une meilleure compréhension des éléments et des valeurs proposées et en conséquence faciliter leurs descriptions. Les travaux de recherche de (Ghebghoub et al., 2009) proposent un ensemble de règles pour transformer le schéma LOM en une ontologie. Nous appliquons ces règles pour la description des LOs.
La figure 2 illustre un aperçu simplifié du binding LOM/OWL (la figure 2 n'illustre que les relations hiérarchiques entre les différents concepts, et pas les relations sémantiques).
Cette modélisation partagée basée sur un squelette d'ontologie est complétée par une description du contexte mobile de l'apprenant (localisation spatiale, localisation temporelle, description de son profil et du support d'apprentissage). Cette modélisation permet ainsi l'organisation de contenus d'apprentissage autour de petites pièces de LOs sémantiquement annotées (enrichis). Les LOs peuvent être ainsi facilement organisés en des parcours d'apprentissage (rapide et juste à temps) et livrés à la demande à l'apprenant selon son profil et son contexte (pertinence).

3 Contexte mobile et modélisation

Le m-learning est souvent considéré comme une extension du e-learning. Cette extension n'est pas sur support mobile uniquement, mais c'est aussi une extension à de nouvelles formes intégrées à l'environnement d'apprentissage que le e-learning ne permet pas. Le contexte d'apprentissage est un aspect crucial dans l'apprentissage mobile. Il faut donc déterminer selon le contexte quelles ressources à envoyer, de quelle manière, à quel moment, sur quelle interface, etc. Tout le processus d'apprentissage doit s'adapter à ces changements de contexte. Cependant, la contextualisation dans l'apprentissage n'est pas facile à atteindre. La diversité des technologies mobiles et la dynamique dans des environnements mobiles compliquent le processus de contextualisation.

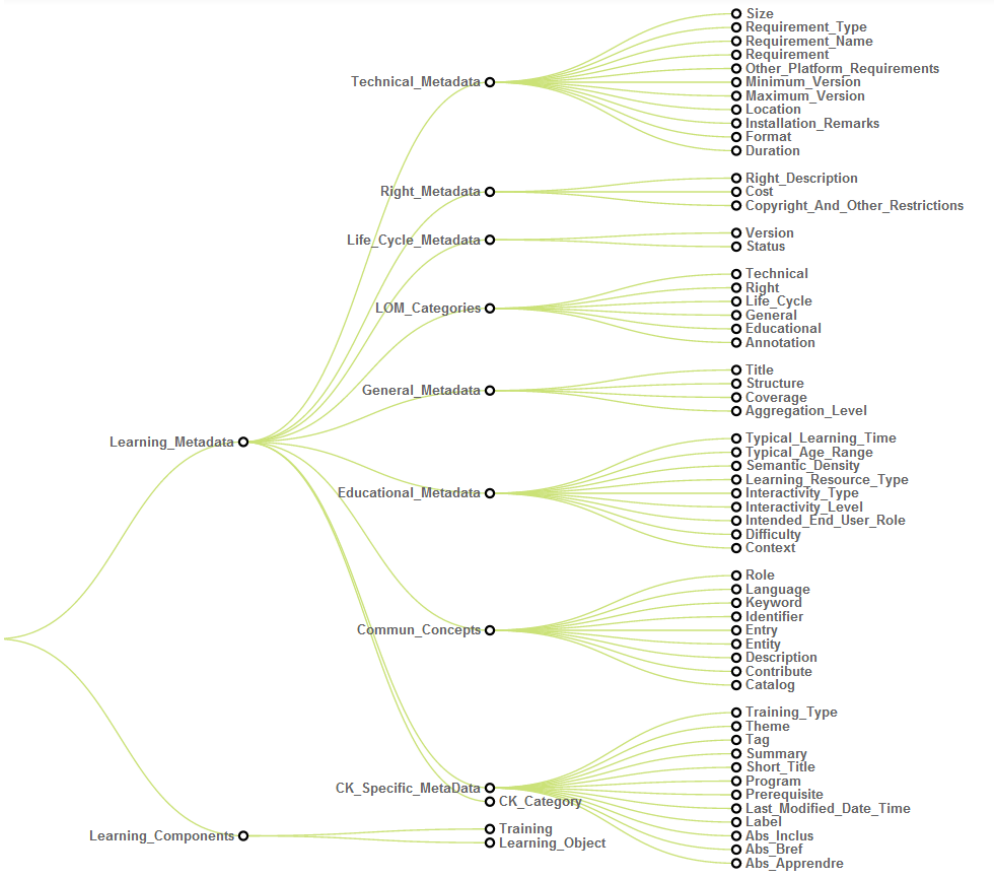


FIG. 2 – *Aperçu simplifié du binding LOM/OWL2*

3.1 Notion de contexte

L'informatique sensible au contexte est apparue dans le milieu des années quatre vingt dix impulsée par les travaux de (Schilit et Theimer, 1994). Ce terme fait référence à des systèmes capables de percevoir un ensemble de conditions d'utilisation, le contexte, afin d'adapter en conséquence leur comportement en termes de délivrance d'informations et de services (Cheverst et al., 2002) (Dey, 2001). On comprend donc qu'avec l'avènement des technologies mobiles, la sensibilité au contexte est devenue un caractère incontournable des systèmes qui permettent une utilisation de type nomade.
Pour bien comprendre et appliquer cette sensibilité au contexte, il est plus simple de passer par une catégorisation des variables du contexte. Selon (Schilit et Theimer, 1994) le contexte se décompose en trois sous classes où chacune des variables répond à l'une des questions "où suis-je ?", "avec qui suis-je ?", "Quelles sont les ressources de mon environnement proche ?". (Ryan et al., 1998) catégorisent le contexte en identité de l'utilisateur, ressources de l'environ-

nement proche, localisation de l'utilisateur et période temporelle d'exécution de l'interaction. Dans le cadre du m-learning, pour avoir une meilleure visibilité et une meilleure compréhension du contexte d'apprentissage, nous proposons d'organiser les données qui constituent ce dernier en différentes dimensions. Nous présentons ici les dimensions qui s'appliquent au domaine du m-learning (Pham Nguyen, 2010) :

- **Dimension spatiale** : La localisation de l'apprenant mobile est un facteur d'une grande importance afin de proposer des contenus adaptés aux propriétés de l'environnement dans lequel il se situe. Souvent, la localisation de l'utilisateur est considérée comme un concept seulement caractérisé par les coordonnées physiques par rapport à un système de coordonnées géographiques. Cependant, les coordonnées physiques, même étant pertinentes pour caractériser une localisation, ne sont pas les seules caractéristiques que nous pouvons considérer lorsqu'on définit une localisation. Supposons que nous essayons de nous localiser sans l'utilisation de technologies de détection de position, tout simplement en essayant de répondre à la question "où je suis ?". Nous pouvons imaginer une multitude de façons pour répondre à cette simple question. En effet, tel que discuté dans (Dobson, 2005), il existe différentes façons plausibles pour caractériser la localisation d'un utilisateur mobile : relative (à côté de, loin de, etc.), nom de la place (Champ de Mars, dans un restaurant, etc.), type de la place (dynamique, fixe, public, privé, etc.), propriétés de la place (niveau de bruit, confort, etc.).
- **Dimension temporelle** : Dans notre travail nous considérons que le contexte temporel a une influence sur l'activité de recherche de l'apprenant mobile. Par exemple, imaginons qu'un apprenant émet une requête "apprendre dans un métro avec un trajet d'une durée de 15 min", nous pouvons dans ce cas décliner une préférence pour une vidéo dont la durée est approximative ou inférieure à 15 min. Nous tentons donc d'exploiter l'information temporelle en vue de décliner les centres d'intérêts de l'apprenant mobile selon cette dimension du contexte.
- **Dimension utilisateur** : Un profil apprenant est une collection de données personnelles associées à un apprenant spécifique. Un profil apprenant est essentiellement décrit par un ensemble de données statiques (nom, prénom, date de naissance, etc.) et un ensemble de données dynamiques (but, préférences, connaissances, compétences, centres d'intérêt, etc.). Un profil joue un rôle important dans un système d'apprentissage (Brusilovsky, 1996) (Rety et al., 2003) pour adapter l'apprentissage aux spécificités de chaque profil.
- **Dimension device** : Afin d'adapter un contenu pédagogique à la technologie mobile censée délivrer l'information il est nécessaire de connaitre les propriétés caractérisant ses technologies. Par exemple, si on a une formation à la souris, un Smartphone n'est pas adapté car il n'a pas de souris, de même si une formation contient des vidéos elle ne sera pas adaptée à des appareils mobiles tels que les MP3.

Pour réaliser un système m-learning sensible à ces différentes dimensions du contexte, le cycle de vie du processus de contextualisation a été étudié.

3.2 Modèle de contexte pour le m-learning

La gestion du contexte est constituée par un processus itératif qui utilise des informations contextuelles au niveau du système à partir de la détection et de l'acquisition du contexte. Il s'agit de capturer les données du contexte, de les stocker et de distribuer les LOs à l'apprenant

selon les informations contextuelles stockées. (Pham Nguyen, 2010) définit les étapes nécessaires dans le cycle de vie d'un système sensible au contexte (Fig 3) :

- L'acquisition des données contextuelles : Il s'agit de capturer toutes les informations contextuelles qui sont disponibles.
- Stockage : Les données capturées sont stockées de façon significative et compréhensible pour l'utilisation envisagée.
- Traitement : Dans notre cas, le traitement des informations du contexte consiste à sélectionner des LOs à partir d'une requête et appliquer une méthode d'optimisation pour raffiner les LOs sélectionnés.

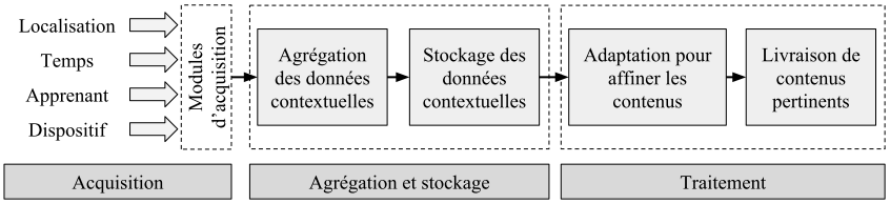


FIG. 3 – *Processus de gestion du contexte*

Pour prendre en compte le contexte dans un système m-learning, il est nécessaire de trouver un moyen de représenter le contexte dans ce dernier. Cette représentation doit fournir un cadre cohérent pour mémoriser et traiter les informations du contexte pour réagir aux changements de l'environnement. Il en résulte alors le modèle de contexte.
Il existe plusieurs méthodes de représentation du modèle contextuel : modèle à base de schéma XML (Henricksen et al., 2003), modèle graphique UML (Strang et Linnhoff-Popien, 2004), modèle topic maps (Sielis et al., 2012), etc. Cependant aucun de ces modèles n'assure l'interopérabilité des données au niveau sémantique. De plus une représentation du contexte doit permettre d'effectuer des raisonnements en vue d'une adaptation. Nous soutenons qu'une modélisation à base d'un squelette ontologique est plus appropriée pour définir le modèle de contexte. Ceci est principalement dû aux propriétés formelles des ontologies et des moteurs d'inférence associés. En ce qui concerne les changements dynamiques du contexte, les ontologies permettent d'assurer l'interopérabilité au niveau sémantique et ainsi il est plus aisé de faire des modifications en assurant la cohérence sémantique des données. La figure 4 présente un aperçu simplifié du modèle de contexte.
Le modèle de contexte vient compléter l'ontologie des Learning Objetcs pour former ainsi une ontologie de domaine du m-learning. Une fois le choix du modèle effectué, les règles de construction de l'ontologie définies et le formalisme de représentation identifié, nous avons créé l'ontologie du m-learning avec l'éditeur Protégé [2].L'ontologie est peuplée avec des LOs venant de la base de données de CrossKnowledge [3] en utilisant l'outil d'intégration de donnée Talend [4]. L'ontologie est par la suite sauvegardé dans un triple store de type Sesame OWLIM [5]. L'objectif maintenant est d'appliquer dessus des techniques de raffinement et d'adaptation des

2. http ://protege.stanford.edu/
3. http ://www.crossknowledge.com/
4. http ://www.talend.com/
5. http ://www.ontotext.com/owlim

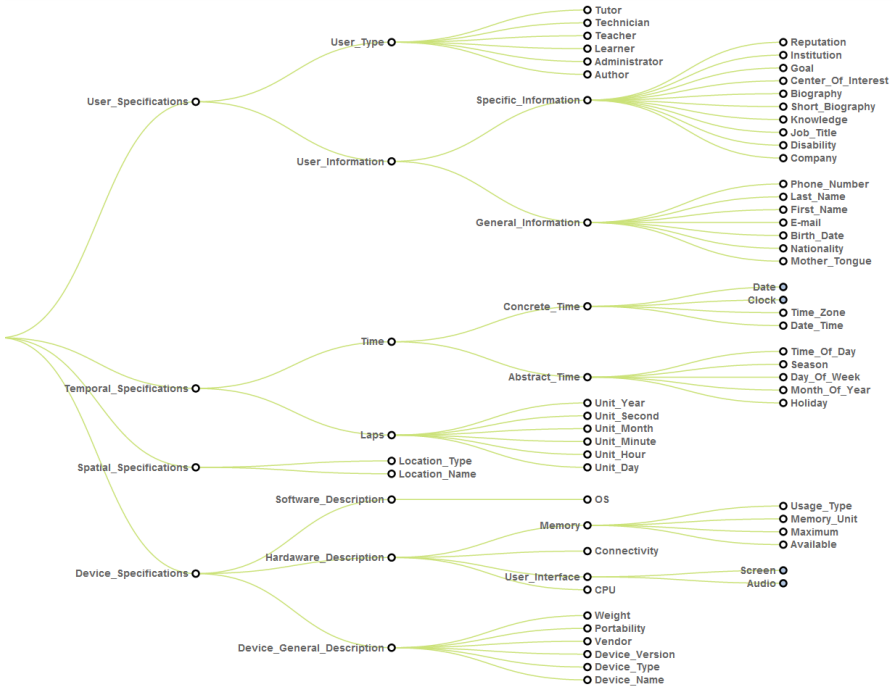


FIG. 4 – *Aperçu simplifié du modèle de contexte*

LOs pour livrer à l'apprenant un parcours de formation pertinent et optimisé selon son contexte.
L'ontologie de domaine du m-learning est disponible à l'url :
`http://checksem.u-bourgogne.fr/WebServices/graphOntology/`.

4 Approche adaptative et combinatoire pour le m-learning

4.1 Filtrage des Learning Objects

Afin d'implémenter un système d'apprentissage mobile contextualisé, chaque contexte apprenant est sauvegardé dans l'ontologie m-learning. Selon ce contexte, le système doit proposer à l'apprenant un ensemble de LOs. Une méthode pour filtrer les LOs est d'appliquer un ensemble de règles métier, indiquant quel LO utiliser dans quel contexte. Ces règles synthétisent les connaissances du domaine et des contraintes métiers qui doivent être respectées par le système. Les règles métier sont traduites en SWRL (Semantique Web Rule Language) afin de les intégrer au triple store Sesame Owlim et par la suite pouvoir raisonner dessus. Exemples :

– Règle 1 : Limiter l'accès à un LO sur un dispositive de type Smartphone :
Learner(?x)∧Device(Smartphone)∧has learning device(?x,Smartphone)∧Learning Object(?z)→has access(?x, ?z)

– Règle 2 : Limiter l'accès aux LOs pour un apprenant sourd :
Learner(?x)∧Disability(deaf)∧has disability(?x,deaf)∧Learning Object(?y)∧
Format(text)∧has format(?y,text)→has access(?x, ?y)

Les règles métier sont définies par les experts du domaine de l'apprentissage. Comme ces experts n'ont pas nécessairement une connaissance du langage de règles SWRL, nous avons donc développé un outil de génération de règles permettant de manipuler facilement les données de l'ontologie de domaine du m-learning et générer automatiquement les règles en SWRL.

4.2 Problème d'optimisation combinatoire

Dans le domaine du e-learning, un LO représente le plus bas niveau de granularité d'un parcours de formation pouvant faire l'objet d'un suivi. Pour valider une formation, un apprenant doit absolument avoir dans son parcours de formation un ensemble de LOs de type objectif. Des règles de précédence entre les LOs sont définies, pour préciser que certaines notions doivent impérativement être assimilées avant d'autres. Chaque parcours de formation contient ainsi un ensemble de LOs destinés à être délivrés sans interruption, et respectant les règles de précédence de manière à former un ensemble cohérent qui va permette de remplir l'objectif de formation, et qui correspond à chaque étape au contexte de l'utilisateur, notamment aux supports de délivrance qui sont à sa disposition.
Si chaque LO était accessible sur chaque support de formation, il serait aisé de choisir à tout instant le meilleur support permettant de délivrer l'enseignement de la manière la plus adaptée au contexte de l'apprenant. Les cas réels que nous avons étudiés nous ont montré au contraire une grande hétérogénéité de supports disponibles selon les LOs. Les cours proposés ont une structure et une durée différente en fonction du support, ce qui interdit de changer de support de délivrance en cours de formation sans risquer la redondance de certaines briques de contenu, ou la présence de contenus absents de l'objectif de formation.
Dans notre cas, le problème peut se ramener à un problème de recherche de plus court chemin multimodal. Ce problème difficile, très étudié ces dernières années consiste à rallier un point B à partir d'un point A en empruntant divers moyens de transport, avec des temps de parcours, des itinéraires et des coûts de transport différents. Nous pouvons faire le rapprochement en considérant que le parcours de formation optimal est égal au plus court chemin pour rallier l'objectif de formation par différents moyens de transport (différents supports de formation). Tout comme deux trajets peuvent suivre des itinéraires différents selon le moyen de transport, deux parcours de formation peuvent comporter des LOs différents. Tout comme le temps de parcours entre deux points varie en fonction du moyen de transport utilisé (parcours à pied plus long qu'en bus), le temps nécessaire pour parcourir un ensemble de briques de d'enseignement peut varier en fonction du support de diffusion (cours présentiel plus long que la lecture du même cours sur papier). Enfin, la disponibilité de chaque support de formation varie dans le temps, tout comme la disponibilité des moyens de transport.
Le problème général qui nous est posé est de proposer à un apprenant un panel de LOs correspondant à son contexte actuel et permettant d'optimiser son expérience d'apprentissage. Cette optimisation intervient sur différents plans : la minimisation de la durée de la formation, la maximisation du gain de compétences et la pertinence des supports de formation par rapport au contexte actuel de l'utilisateur.
Malgré l'évolution permanente des calculateurs, il existera certainement toujours, pour un problème difficile, une taille critique au-dessus de laquelle même une énumération partielle des

solutions admissibles devient prohibitive en temps de calcul. Compte tenu de ces difficultés, la plupart des spécialistes de l'optimisation combinatoire ont orienté leur recherche vers le développement de méthodes heuristiques. Une métaheuristique est souvent définie comme une procédure exploitant au mieux la structure du problème considéré, dans le but de trouver une solution de qualité raisonnable en un temps de calcul aussi faible que possible(Widmer, 2001; Nicholson, 1971).
Dans nos futurs travaux, nous proposons de comparer l'efficacité de certaines heuristiques proposées pour la résolution du problème de recherche du plus court chemin avec une métaheuristique inspirée du recuit simulé déjà utilisée avec succès pour un problème de recommandation de séjour touristique [18].

5 Conclusion

Dans ce papier, nous avons présenté une approche pour un système de recommandation appliqué au domaine du m-learning combinant les technologies du Web sémantique et des algorithmes d'optimisation combinatoire. Ce système est composé d'une partie statique représentant à la fois les contenus d'apprentissage et le contexte de l'apprenant et une partie adaptative et dynamique contenant des règles de comportement dans un contexte de mobilité et des métaheuristiques d'optimisation combinatoire. Notre approche permet aux formateurs de représenter leur savoir-faire en utilisant des règles métier et une ontologie pour assurer une hétérogénéité des connaissances. Ensuite, dans un environnement de mobilité, elle permet de prendre en compte les contraintes de l'environnement et les contraintes utilisateur. Enfin, la partie métaheuristique de notre proposition permet une combinaison dynamique de morceaux de la formation en fonction de ces contraintes.

Références

Draft standard for learning object metadata, ieee 1484.12.1-2002. `http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf`,.

Ims learning design information model, version 1.0 final specification. `http://www.imsglobal.org/learningdesign/ldv1p0/imsld_infov1p0.html`,.

Scorm 2004 handbook. `http://203.183.1.152/aen/content/act2005eg/data/txt1.pdf`,.

Abel, M. (2007). Apport des memoires organisationnelles dans un contexte d'apprentissage.

Bourda, Y. (2002). Des objets pédagogiques aux dossiers pédagogiques (via l'indexation). In *Document numérique*, pp. 115–128.

Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. pp. 87–129.

Cheverst, K., K. Mitchell, et N. Davies (2002). The role of adaptive hypermedia in a context-aware tourist guide. *Commun. ACM 45*(5), 47–51.

Dey, A. (2001). Understanding and using context. *Personal Ubiquitous Comput. 5*(1), 4–7.

Une approche Web sémantique et combinatoire.

Dobson, S. (2005). Leveraging the subtleties of location. In *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence : innovative context-aware services : usages and technologies*, New York, NY, USA, pp. 189–193. ACM.

Ghebghoub, O., M. Abel, et C. Moulin (2009). Lomonto : Une ontologie pour l'indexation d'objets pédagogiques. In *AFIA platform workshop : Constructions d'ontologies : Vers un guide de bonnes pratiques*, Hammamet, Tunisia.

Grandbastien, M., B. Huynh-Kim-Bang, et A. Monceaux (2008). Les ontologies du prototype luisa, une architecture fondée sur des web services sémantiques pour les ressources de formation. In *Actes des 19es Journées Francophones d'Ingénierie des Connaissances (IC 2008)*, Nancy, France, pp. 61–72.

Henricksen, K. M., J. Indulska, et A. Rakotonirainy (2003). Generating context management infrastructure from high-level context models. In *Proceedings of the Fourth International Conference on Mobile Data Management*. Monash University.

Nicholson, T. (1971). *Optimization in industry*.

Pham Nguyen, C. (2010). *Conception d'un système d'apprentissage et de travail pervasif et adaptatif fondé sur un modèle de scénario*. Ph. D. thesis, Télécom Bretagne.

Rety, J., J. Martin, C. Pelachaud, et N. Bensimon (2003). *Coopération entre un hypermédia adaptatif éducatif et un agent pédagogique*. Paris : Hermes & Lavoisier.

Ryan, N., J. Pascoe, et D. Morse (1998). Enhanced reality fieldwork : the context-aware archaeological assistant. In *Computer Applications in Archaeology 1997*, Oxford.

Schilit, B. et M. Theimer (1994). Disseminating active map information to mobile hosts. *IEEE Network 8*, 22–32.

Sielis, G. A., C. Mettouris, A. Tzanavari, et G. A. Papadopoulos (2012). *Context-aware recommendations using topic maps technology for the enhancement of the creativity process*.

Strang, T. et C. Linnhoff-Popien (2004). A context modeling survey. In *Workshop Proceedings, First International Workshop on Advanced Context Modelling, Reasoning And Management at UbiComp*.

Widmer, M. (2001). Les métaheuristiques : des outils performants pour les problèmes industriels. In *3e Conférence Francophone de MOdélisation et SIMulation "Conception, Analyse et Gestion des Systèmes Industriels". MOSIM'01*.

Summary

Given the rapid emergence of new mobile technologies and the growth of needs of a moving society in training, works are increasing to identify new relevant educational platforms to improve distant learning. The next step in distance learning is porting e-learning to mobile systems. This is called m-learning. So far, the learning environment was either defined by an educational setting, or imposed by the educational content; in our approach, in m-learning, we change the paradigm where the system adapts learning flow to the context of the learner. In this paper, we present a new approach to develop a recommender system for optimizing learning in mobile context.

Langage communautaire, confiance et recettes de cuisine

Damien Leprovost*, Thierry Despeyroux*
Yves Lechevallier*

*Inria – Rocquencourt, Équipe-projet AxIS, BP 105 – 78153 Le Chesnay Cedex, France
{prénom.nom}@inria.fr

Résumé. De nos jours, les sites de partage de connaissance communautaires représentent une part importante et grandissante du Web. Sur ces sites, les utilisateurs échangent des connaissances, en étant à la fois auteurs et lecteurs du contenu. Dans de telles circonstances, la communauté se structure autour d'une sémantique empirique qui lui est propre, et qui peut différer grandement des standards académiques des domaines concernés. L'analyse de cette sémantique à partir des bases de connaissance de référence traditionnelles peut alors se révéler insuffisamment pertinente pour prendre en compte ces comportements utilisateurs.

Dans cet article, nous présentons une méthode pour construire notre propre compréhension de la sémantique des contributions des utilisateurs, sans recours à une base de connaissance externe. Cette compréhension est rendue possible par une extraction de la connaissance présente dans les contributions analysées. Nous proposons une évaluation de la confiance imputable à cette compréhension déduite, afin d'évaluer la qualité du contenu de l'utilisateur. Ce taux de qualité ainsi calculé peut être considéré comme la mesure avec laquelle le contenu est compréhensible par la globalité des utilisateurs de la communauté. Nous illustrons notre travail en analysant des recettes de cuisine fournies par des utilisateurs sur des sites Web de partage de recettes de cuisine.

1 Introduction

Avec l'essor du Web 2.0, les internautes sont désormais au centre de l'utilisation du Web, étant à la fois consommateurs et contributeurs du contenu du réseau. Au sein de ce Web social, de nombreux sites de partage de connaissance entre utilisateurs, appelés sites de partage de connaissance communautaires, ont vu le jour et gagnent sans cesse en popularité. Le plus célèbre d'entre eux, l'encyclopédie Wikipédia, est même devenue aujourd'hui le sixième site Web le plus consulté du Web mondial [1]. Sur ces sites, les utilisateurs déposent des connaissances, assimilent celles des autres utilisateurs, en discutent et se structurent en communautés. Il est alors observable que ces communautés sont régies par des modes de fonctionnement qui leurs sont propres et issues des habitudes générales des membres de la communauté. Dans ce contexte, les usages deviennent normes, indépendamment d'éventuels standards préexistants.

1. http ://www.alexa.com/topsites

Ce constat s'applique également à la sémantique des échanges, qui dépend donc uniquement des habitudes propres aux utilisateurs impliqués dans l'échange de connaissance. Dans le cadre de la recherche traditionnelle de fouille de la connaissance, cette évolution en relative autonomie peut se révéler problématique. Car dans de telles circonstances, il n'existe aucune garantie que l'évolution des échanges de la communauté se structure autour d'une sémantique qui soit en adéquation avec les bases de connaissance de référence traditionnellement identifiées. La pertinence des conclusions issues de ces mêmes références peut alors ne pas ou peu refléter l'évolution réelle du comportement des utilisateurs et de la sémantique de leurs échanges.

Afin de contourner cette faiblesse des bases de connaissance de référence, nous proposons dans cet article une méthode pour construire notre propre compréhension des contributions des utilisateurs, basée uniquement sur les données de celles-ci. Le fait de s'affranchir des bases de connaissance externes permet de ne plus être soumis aux faiblesses qui y sont liées. En effet, face à un système social en permanente évolution, une base statique de connaissance présente un fort risque d'obsolescence. De plus, nombre d'entre-elles ont bien souvent un caractère ciblé, avec un vocabulaire précis et spécialisé, certes très pertinent mais parfois peu adapté au vocabulaire commun de la majorité des échanges utilisateurs. Notre approche se base uniquement sur le vocabulaire du jeu de données, ce vocabulaire est donc directement issu des utilisateurs. Ces derniers étant à la fois contributeurs et consommateurs de ce vocabulaire, nous visons donc à obtenir la même compréhension que celle que l'utilisateur du système possède lorsqu'il manipule ces données. Afin d'évaluer la qualité du contenu utilisateur analysé, nous évaluons la compréhension que nous obtenons de ces données. Nous attributions pour chaque compréhension déduite une valeur de confiance qui représente dans quelle mesure les utilisateurs de la communauté témoignent d'une utilisation — et donc d'une compréhension — commune de ces éléments. Nous plaçons notre travail dans le contexte des recettes de cuisine, dont les sites de partage communautaires sont nombreux et très populaires sur le Web français comme mondial. Nous illustrons notre travail en analysant des recettes de cuisine fournies par des utilisateurs de ces mêmes sites.

Cet article est organisé comme suit : la section 2 présente un état de l'art de l'utilisation de recettes de cuisine dans le domaine de la gestion des connaissances, puis la section 3 introduit notre modèle d'acquisition de la connaissance. Notre évaluation de la confiance de cette acquisition est décrite dans la section 4, et notre modèle d'expérimentation dans la section 5. La section 6 conclue et présente nos orientations futures.

2 État de l'art

La recette de cuisine est un type de données particulier, composé d'un ensemble d'ingrédients et de procédures d'exécution. Les tentatives de prise en compte des spécificités de ce type de données existent dans la littérature, notamment dans le domaine des systèmes de recommandation. Le *Cooking Assistant* (Sobecki et al., 2006) définit un système de recommandation démographique de recettes de cuisine, basé sur une inférence à logique floue. Raisonnant à partir de métadonnées annotées manuellement, cette méthode est efficace pour fournir une réponse globale à un besoin général. Mais la généralisation des caractéristiques des recettes conduit à une recommandation également généralisée. Il apparaît un besoin de prise en compte des caractéristiques propres aux ingrédients. Freyne et Berkovosky utilisent pour cela dans de multiples travaux (Freyne et Berkovsky, 2010a,b; Berkovsky et Freyne, 2010) la rela-

tion de composition qui existe entre ingrédients et recettes pour propager des évaluations. Par le biais du logiciel d'apprentissage Weka (Hall et al., 2009) et en utilisant l'algorithme d'arbre de décision M5P (Quinlan, 1992) qui y est implémenté, les auteurs déterminent un comportement utilisateur (Freyne et al., 2011). L'ensemble de ces approches nécessitent néanmoins une phase constante de normalisation, un travail d'expert consistant à vérifier ou annoter les ingrédients afin qu'ils correspondent à une liste de référence connue à l'avance.

Bien qu'elles ne soient pas directement liées à nos travaux, il existe également dans le domaine du raisonnement à partir de cas plusieurs approches intéressantes relatives à ce type particulier de données que sont les recettes de cuisines. Le système *CHEF* (Hammond, 1986) est un système d'adaptation par la critique dans le domaine des recettes de cuisine du Sichuan. Cette approche permet notamment de prendre en compte la spécificité du type de données qu'est l'ingrédient, en relevant par exemple les problèmes découlant d'une substitution d'ingrédient, quand bien même ceux-ci aurait été très proches. En revanche, comme nombre de systèmes du genre, une importante phase d'apprentissage est requise. Le système *MIKAS* (Khan et Hoffmann, 2003) pour *Menu construction using Incremental Knowledge Acquisition System*, propose de contourner ce besoin d'apprentissage initial par un recours à l'expert en fonction des besoins d'exploitation tout au long de l'utilisation. Cette aspect de la transmission de connaissance de l'expert au système par l'expérience plutôt que par le déclaratif est vu comme plus efficace et plus adapté à l'utilisation en conditions réelles, plus robuste aux cas inhabituels et imprévus. Il ne permet toutefois pas une évaluation indépendante des contenus, car dépendant des connaissances propres de l'expert.

3 Extraction de l'information semi-structurée

Au sein d'une recette de cuisine, une structure est identifiable : *a minima*, la recette se compose d'un titre, d'une liste d'ingrédients et d'instructions de réalisation. Au sein de ces éléments en revanche, il n'existe pas de structure contrainte. Notamment dans la cas des recettes de cuisine saisies par les utilisateurs, les modes d'expressions peuvent êtres divers et variés. Dans le cadre de nos travaux, nous nous intéressons aux informations relatives aux ingrédients utilisés, à partir des données brutes issues de ces utilisateurs. Ces données sont donc composées de lignes d'ingrédients librement saisies par des auteurs multiples. Pour exploiter ces données, nous recherchons ce que nous définissons comme étant la structure présumée par les utilisateurs. En effet, bien que les formes et les manières de présenter un ingrédient dans une ligne brute sont nombreuses et variées, nous observons toujours un fort consensus sur le mode le plus simple d'expression. Par exemple, s'il est tout à fait possible de trouver un ingrédient décrit comme suit : « un morceau de bœuf d'environ 250g », la majorité des utilisateurs écrivent simplement : « 250g de bœuf ». À partir de ce constat, nous définissons la structure présumée comme étant : *quantité – unité – ingrédient*, où *quantité* est le nombre d'éléments impliqués, *unité* est l'unité de mesure associée à la quantité, et *ingrédient* l'ingrédient lui-même. Pour chaque ligne correspondant au modèle, il est possible de retrouver l'élément vide pour un ou plusieurs des emplacements de structure précédemment défini. Le tableau 1 illustre ce découpage structurel.

Bien sûr, toutes les lignes d'ingrédients ne correspondent pas à ce modèle. Les éléments constituant une ligne peuvent ne pas être dans le même ordre, ou être quantité variable (par exemple, la ligne « 2 ou 3 pommes »). Nous discriminons donc nos lignes d'ingrédient en

TAB. 1 – *Structure présumée*

ligne	quantité	unité	ingrédient
250g de bœuf	250	g	bœuf
3 pommes	3	ø	pommes
pincée de sel	ø	pincée	sel
ketchup	ø	ø	ketchup

quatre classes : les lignes *quantité–unité–ingrédient*, les lignes *quantité–ingrédient*, les lignes *quantificateur–ingrédient* [2] et les lignes non comprises, pour l'instant supposées comme un ingrédient unique non-identifié.

Nous procédons ensuite à une phase d'apprentissage, où toutes les lignes bien formées permettent de comprendre les autres. Cet apprentissage se fait en deux étapes principales :

– Tout d'abord, nous cherchons des incohérences dans les éléments identifiés. La présence d'un ingrédient complexe dans une ligne bien formée permet de mettre en évidence une erreur de détection du même ingrédient dans une ligne plus simple. Par exemple, « 500g de corned beef » identifie clairement « 500 » comme étant la *quantité*, « g » comme étant l'*unité* et « corned beef » comme étant l'*ingrédient*. En revanche, en présence d'une ligne simple « corned beef », basiquement, « corned » sera identifié comme étant un *quantificateur* et « beef » comme étant l'*ingredient*. La connaissance de la ligne complète mentionnée précédemment nous permet alors de comprendre « corned beef » en tant qu'ingrédient dans son ensemble et donc de considérer cette ligne comme étant une *ligne à ingrédient seul*.
– Une fois l'ensemble des incohérences traitées, pour toutes les *lignes à ingrédient seul* restantes, nous cherchons à les faire correspondre aux cas précédemment rencontrés. À partir des *quantités*, *unités* et *ingrédients* précédemment rencontrés, nous les distinguons en *lignes quantité–unité–ingrédient*, *lignes quantité–ingrédient* et *lignes quantificateur–ingrédient*, ou simplement en tant que *lignes à ingrédient seul déduites* si la ligne est simplement un ingrédient seul, dont l'existence a déjà été rencontrée précédemment. À défaut, les lignes restantes sont considérées comme étant des *lignes à ingrédient seul supposées*.

Le tableau 2 présente les classes de lignes ainsi obtenues. Alors que la première étape permet de valider les placements des éléments dans les classes 1, 2 et 3 ; la seconde étape permet de ventiler la classe 5 (qui contient initialement les lignes pour lesquelles aucune compréhension n'est ressortie) dans les autres classes. La figure 1 illustre leur distribution sur le jeu de données Marmiton ainsi que l'effet de la phase d'apprentissage sur le peuplement de ces classes. Dans cet exemple, la taux de lignes non-validées passe de 15% à 1,9% lors de traitement.

4 Évaluation de la confiance

De l'exploitation de la structure présumée précédemment décrite, nous identifions une liste d'ingrédients manipulés par les utilisateurs lors de la rédaction de recettes. Bien que la saisie de ces ingrédients soit entièrement libre, nous observons un effet longue traîne : un petit nombre

2. Une *unité* sans présence de *quantité* est alors appelé *quantificateur*.

TAB. 2 – *Classes de lignes analysées*

Class 1	ligne quantité–unité–ingrédient
Class 2	ligne quantité–ingrédient
Class 3	ligne quantificateur–ingrédient
Class 4	ligne lignes à ingrédient seul déduites
Class 5	ligne lignes à ingrédient seul supposées

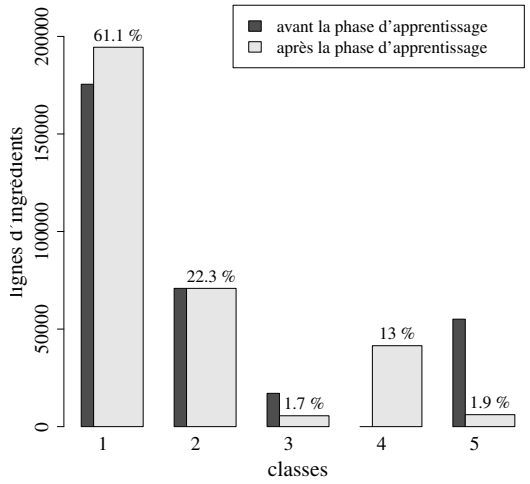


FIG. 1 – *Distribution des classes sur Marmiton*

d'éléments concentre un grand nombre d'occurrences, alors qu'une grande majorité d'éléments ne représente qu'une toute petite partie des utilisations. Ce phénomène est commun à bon nombre de sites sociaux à usages libres — et même au delà — où la distribution suit une loi de puissance. Ce principe de convergence sociale valide notre approche de recherche de motifs majoritaire et guide notre processus d'évaluation de la confiance des ingrédients : une formulation fortement utilisée par l'ensemble des utilisateurs aura nécessairement de bonnes chances d'être bien compris par les utilisateurs et devra donc être gratifiée d'une confiance élevée. À l'inverse un terme extrêmement rare et très peu utilisé ne présente aucune garantie quant à sa compréhension par la communauté et devra recevoir une valeur de confiance faible. La figure 2 illustre la répartition en fréquence cumulée des ingrédients sur le jeu de données de Marmiton. Les valeurs d'*unités*/*quantificateurs* suivent le même schéma de distribution.

Application du principe de Pareto

Eu égard à la distribution en loi de puissance de nos données sociales, nous appliquons à notre modèle le principe de Pareto (ou loi des 80-20), où 80 % des effets sont le produit de 20 % des causes. Appliqué à notre modèle, cela signifie que 80 % des lignes de recettes rédigées

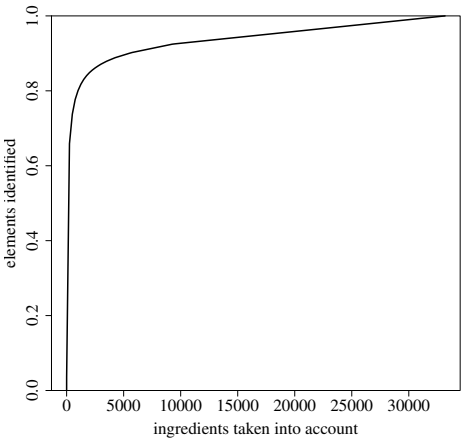


FIG. 2 – *Répartition des ingrédients sur Marmiton*

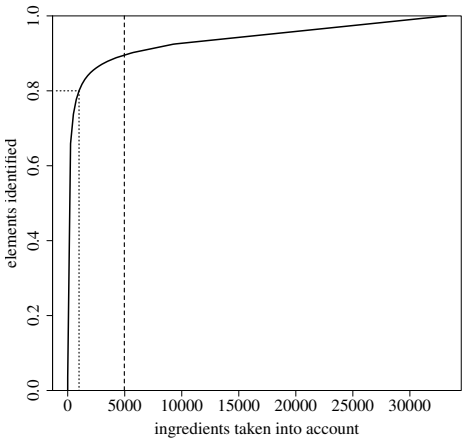


FIG. 3 – *Application du principe de Pareto sur Marmiton*

librement par les utilisateurs sont issus de 20 % de l'ensemble des ingrédients existants. Dans le cas d'une répartition encore plus prononcée (voir figure 2), où moins de 20 % des ingrédients existants représentent plus de 80 % des lignes, cela signifie qu'un certain nombre d'ingrédients ne sont pas significatifs et ne doivent pas être considérés comme tels. Pour discriminer le jeu d'ingrédients suffisamment significatifs à conserver de l'ensemble total des ingrédients renseignés, nous calculons la taille idéale de ce jeu significatif. Nous définissons le nombre a minimal d'ingrédients possible pour représenter 80 % des lignes utilisateurs et définissons ce nombre comme représentant 20 % des ingrédients du jeu significatif. Le jeu significatif est alors l'ensemble d'ingrédient ayant un cardinal b égal à cinq fois ce nombre a qui maximise la représentation des lignes d'ingrédients. La figure 3 illustre cette coupe opérée sur l'ensemble initial.

Confiance par ingrédient Une valeur de confiance est alors attribuée à chaque ingrédient, en fonction de sa fréquence d'utilisation. Cette valeur est maximisée pour le sous-ensemble de tête que sont les 20 % d'ingrédients représentant 80 % des lignes. Cette maximisation est justifiée par la position majoritaire qu'ils occupent dans le système, témoignant d'une utilisation — et donc d'une maîtrise — suffisamment forte par les utilisateurs. Afin de ne pas découper brutalement l'ensemble des ingrédients en deux groupes, le reste des éléments de l'ensemble se voient attribuer une confiance variable en fonction de la fréquence d'utilisation respective de chacun de ses éléments. Pour tout élément i, la valeur de confiance C_i ainsi attribuée est de :

$$C_i = \begin{cases} 1 & \text{si} \quad i < a \\ \frac{N_i - N_b}{N_a - N_b} & \text{si} \quad a < i < b \\ 0 & \text{si} \quad i > b \end{cases} \tag{1}$$

où a est l'ensemble des 20% d'ingrédients les plus utilisés représentant 80% des lignes et N_a le nombre de ses occurrences, b l'ensemble significatif tel que $5 * a = b$ et N_b son nombre d'oc-

curences, et N_i le nombre d'occurrence de i. La confiance est nécessairement contenue dans l'ensemble $[0; 1]$. L'utilisation de valeurs fixes aux extrémités se justifie par la volonté de ne pas sur-nuancer les ingrédients totalement assimilés par la communauté d'une part (confiance à 1), et de ne pas attribuer de valeurs négatives aux ingrédients non-reconnus afin de ne pas sur-impacter le calcul des recettes (confiance à 0). La figure 4 illustre la confiance ainsi calculée des ingrédients de Marmiton.

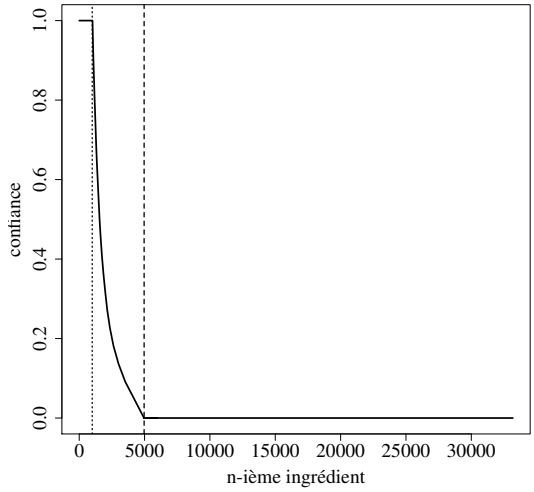


FIG. 4 – *Confiance calculée sur Marmiton*

Confiance par unité/quantificateur Nous considérons à présent l'ensemble des unités — appelés quantificateur en cas d'unité sans quantité — pour leur attribuer également une valeur de confiance. Toutefois, une unité ou quantificateur ne saurait avoir une cohérence universelle et donc une valeur de confiance absolue. En effet et par exemple, si associé à l'ingrédient « lait », l'unité « ml » est fréquente et par conséquent devrait jouir d'une confiance élevée, la déclaration « 200 ml de pommes » n'a aucune cohérence, et donc ne devrait se voir attribuer aucune confiance. Nous identifions donc un lien total et direct entre la cohérence d'une unité ou d'un quantificateur et l'ingrédient avec lequel il est utilisé. Nous considérons donc l'ensemble des unités et quantificateurs en fonction de chaque ingrédient. Pour chaque ingrédient, nous établissons un ensemble de fréquence par quantificateurs, et une valeur de confiance sur le même modèle que pour les ingrédients. Nous remplissons ainsi une matrice de confiance entre ingrédients et unités/quantificateurs. Cette matrice nous permet d'associer à tout couple d'ingrédient x et d'unité/quantificateur i une valeur de confiance $C_{x,i}$, comme étant l'évaluation la cohérence de rencontrer ces deux éléments associés dans une ligne d'ingrédient. Il est important de noter que l'unité/quantificateur vide est répertorié comme toute autre valeur, sa présence pouvait être plus ou moins justifiée en fonction des ingrédients. À titre d'exemple, l'unité vide jouit d'une confiance forte associée à l'ingrédient « poivre » (très rarement lié à

une unité par les utilisateurs), alors que sa confiance est nulle associée à l'ingrédient « riz » (très fréquemment lié à une unité par les utilisateurs).

Confiance par recette À partir des valeurs de confiance des ingrédients et des unités ou quantificateurs par ingrédient précédemment calculées , nous attribuons une valeur de confiance par recette. Cette confiance globale est représentée comme la moyenne des confiances par ligne, qui elle est le produit des confiances de ses composants. La confiance C_x d'une recette x est telle que :

$$C_x = \frac{\sum_{i \in I_x} (C_i * C_{x,i})}{||I_x||}$$

où I_x est l'ensemble des ingrédients de x, C_i la confiance de l'ingrédient i et $C_{x,i}$ la confiance de l'unité associée à l'ingrédient i dans la recette x.

Chaque valeur de confiance ainsi calculée est donc comprise entre 0 (confiance nulle) et 1 (confiance totale). Cette confiance représente donc l'évaluation du degré de certitude qu'un utilisateur du système, auteur comme lecteur, sera en mesure de comprendre et d'utiliser ces ingrédients. La figure 5 présente les valeurs de confiance calculées par recettes sur Marmiton, dont l'expérimentation est détaillée ci-après.

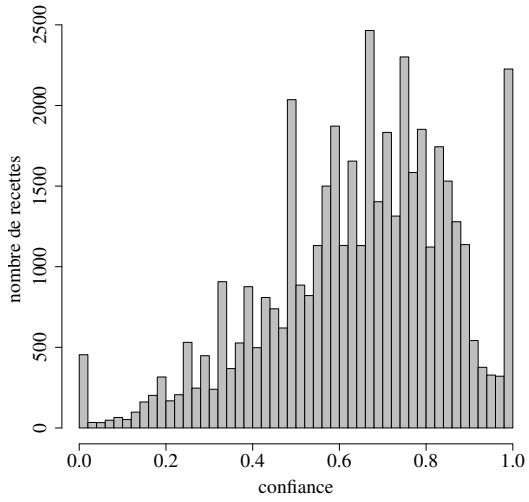


FIG. 5 – *Confiance des recettes du site Marmiton*

5 Expérimentation

Pour valider notre approche et illustrer notre méthode, nous avons réalisé une expérimentation complète sur le site français de partage communautaire de recettes Marmiton.org.

Extraction des données En premier lieu, nous avons développé un robot d'aspiration dédié. Comme il est d'usage dans les sites Web de ce type, chaque page de recette présente un nombre important de liens vers d'autres pages de recettes du site. Nous relevons donc les adresses Web de l'ensemble des pages de recettes présentes sur la page d'accueil du site. Pour chacune d'entre elle, nous indexons le contenu, et relevons l'ensemble des liens vers des pages de recettes que nous ne connaissons pas encore qu'elle contient, et continuons ainsi jusqu'à épuisement de la liste des pages de recettes identifiées. La présence d'éléments de navigation, comme les propositions de menus, les dernières recettes publiées ou les recettes au hasard nous permettent de ne jamais risquer de s'enfermer dans une partie plus réduite du site. En outre, cette approche est adaptée à la philosophie du site, qui veut qu'un maximum de recettes soit visible en un minimum d'effort par ses utilisateurs. Par cette méthode, nous avons collecté 44 169 recettes distinctes.

Normalisation des données extraites Dans un second temps, nous avons utilisé un analyseur syntaxique développé spécifiquement pour parcourir et normaliser l'ensemble des recettes collectées. Cette normalisation au format XML nous permet de structurer les différents types de données que contiennent les recettes, et notamment les lignes d'ingrédients. Par ce traitement, nous avons identifié 354 856 lignes d'ingrédients. Après traitements et corrections d'erreurs mineures, nous avons relevé 33 177 ingrédients distincts dans ces 354 856 lignes.

Calcul de la confiance et résultats Nous avons alors appliqué notre méthode comme présentée dans les sections 3 et 4. La figure 5 présente la distribution des valeurs de confiance par recette calculées sur le jeu de données. Cette distribution suit globalement une loi normale, avec des pics aux extrémités, conséquence des recettes très populaires et très simples d'une part (comme faire une pâte) et incomprises d'autre part [3]. La table 3 résume les principales mesures de cet ensemble des valeurs de confiance des recettes.

TAB. 3 – *Confiances des recettes sur Marmiton*

mesure	**valeur**
premier quartile	0,800
médiane	0,667
troisième quartile	0,507
moyenne	0,658

Outre les résultats présentés tout au long de l'article, nous avons également mené en fin d'expérimentation une première exploration de croisement de ces résultats avec les divers métadonnées additionnelles que propose Marmiton. La figure 6 présente l'amplitude de confiance des recettes en fonction de leur type déclaré.

On remarque une confiance nettement supérieure des desserts, cohérent avec le fait que nombre d'entre-eux partagent des ingrédients très communs, limitant ainsi les apports ésotériques qui grèvent la compréhension, et donc la confiance. Le phénomène inverse s'observe pour les boissons, où la base d'éléments communs est beaucoup plus réduite et où en conséquence une beaucoup plus forte diversité d'ingrédients s'exprime — ingrédients que l'on ne

3. Il s'agit notamment de recettes mal rédigées pour lesquelles l'analyse des ingrédients ne fournit aucun résultat

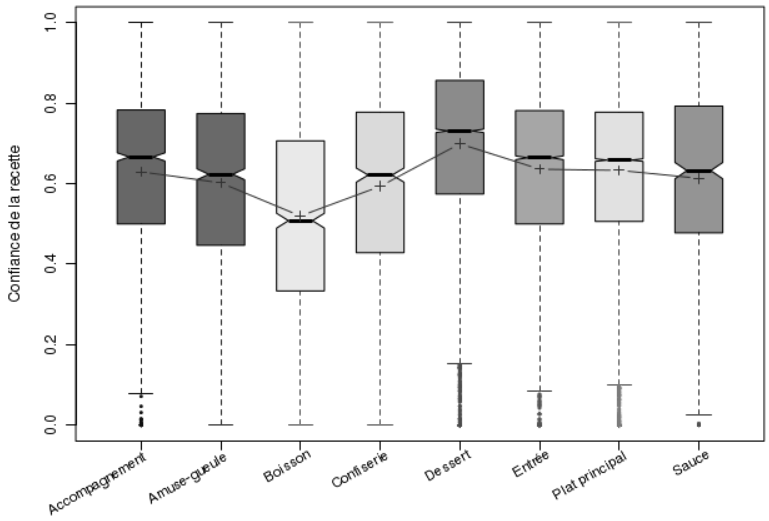


FIG. 6 – *Amplitudes de confiance des recettes par type sur Marmiton*

retrouvera bien souvent pas dans les autres recettes du site — ce qui conduit globalement à une confiance plus faible pour les recettes de ce type.

6 Conclusion et travaux futurs

Dans ce papier, nous avons présenté une méthode pour évaluer la confiance attribuable à une publication utilisateur, comme étant la probabilité qu'un autre utilisateur du système comprenne la sémantique de sa contribution. Nous utilisons une approche indépendante de toute base de connaissance externe, afin de raisonner directement sur les termes manipulés par les utilisateurs. Cette méthode présente également l'avantage de ne pas être dépendant de la langue, ni de souffrir des problèmes de pertinence ou de couverture relatifs aux bases de connaissance de référence, tout en construisant une base de connaissance propre à la communauté.

Pour nos travaux futurs, nous projetons d'exporter la connaissance extraite de cette compréhension des contributions utilisateurs, ce qui permettra de définir sans apport extérieur l'ontologie du système analysé, ou d'enrichir une base extérieure pour améliorer sa pertinence et lutter contre son obsolescence. Enfin, l'application de méthode de partitionnement de fouille de données, guidées par nos mesures de confiance, permettra prochainement d'évaluer une structure interne de la sémantique du système et des relations déductibles qui existent entre les différents ingrédients (proches ou dérivés) ou recettes (variantes, alternatives).

Références

Berkovsky, S. et J. Freyne (2010). Group-based recipe recommendations : analysis of data aggregation strategies. In *Proceedings of the 2010 ACM Conference on Recommender Systems*, Barcelona, Spain, pp. 111–118. ACM.

Freyne, J. et S. Berkovsky (2010a). Intelligent food planning : personalized recipe recommendation. In *Proceedings of the 2010 International Conference on Intelligent User Interfaces*, Hong Kong, China, pp. 321–324. ACM.

Freyne, J. et S. Berkovsky (2010b). Recommending food : Reasoning on recipes and ingredients. In *User Modeling, Adaptation, and Personalization, 18th International Conference*, Big Island, HI, USA, pp. 381–386. Springer.

Freyne, J., S. Berkovsky, et G. Smith (2011). Recipe recommendation : Accuracy and reasoning. In *User Modeling, Adaptation, and Personalization, 19th International Conference*, Girona, Spain, pp. 99–110. Springer.

Hall, M., E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, et I. H. Witten (2009). The weka data mining software : an update. *SIGKDD Explor. Newsl. 11*(1), 10–18.

Hammond, K. J. (1986). Chef : A model of case-based planning. In *AAAI*, pp. 267–271.

Khan, A. S. et A. Hoffmann (2003). Building a case-based diet recommendation system without a knowledge engineer. *Artificial Intelligence in Medicine 27*(2), 155–179.

Quinlan, R. J. (1992). Learning with continuous classes. In *5th Australian Joint Conference on Artificial Intelligence*, Singapore, pp. 343–348. World Scientific.

Sobecki, J., E. Babiak, et M. Slanina (2006). Application of hybrid recommendation in web-based cooking assistant. In *Knowledge-Based Intelligent Information and Engineering Systems, 10th International Conference, Proceedings, Part III*, pp. 797–804. Springer.

Summary

Today, websites sharing community knowledge are an important and growing part of the Web. On these sites, users share knowledge, being both authors and readers of the content. In such circumstances, the community is structured around an empirical semantics of its own, and may differ greatly from the academic standards of the areas concerned. The analysis of the semantic knowledge bases from traditional reference can then be insufficiently relevant to take into account the user behavior.

In this paper, we present a method to build our own understanding of the semantics of user contributions, without the use of any external knowledge base. This understanding is performed thanks to the knowledge extracted from the same analyzed user contributions. We propose in this method an evaluation of the trust attributable to the deduced understanding, in order to evaluate the quality of user content. This computed quality rate can be viewed as the extent to which the content is understandable by the community of users. We illustrate our work by focusing on the cooking recipes provided by users on sharing websites.

Dynamique des communautés par prévision d'interactions dans les réseaux sociaux

Blaise Ngonmang[*] [**] [***], Emmanuel Viennet[*]

[*]Université Paris 13, Sorbonne Paris Cité,
L2TI (EA 3043), F-93430, Villetaneuse, France. firstname.lastname@univ-paris13.fr
[**]UMI 209 UMMISCO, Université de Yaoundé I, B.P. 337 Yaoundé, Cameroun
[***]LIRIMA, Equipe IDASCO, Faculté des Sciences,
Département d'Informatique,
B.P. 812 Yaoundé, Cameroun

Résumé. Les travaux sur la détection de communautés dans les réseaux sociaux ont pendant longtemps uniquement considéré l'aspect statique du réseau: une image du réseau est considérée à un instant donné. Les communautés sont ensuite calculées sur cet instantané. Parce que les réseaux sociaux sont dynamiques par nature, des travaux sur la détection de communautés en tenant compte de cette dynamique ont commencé à voir le jour ces dernières années.

Un des problèmes actuellement peu explorés dans la littérature est la prévision de communautés: connaissant l'évolution du réseau jusqu'au temps t, peut-on prévoir les communautés au temps $t + 1$?

Dans cet article, nous proposons une approche générale de prévision des communautés basée sur un modèle d'apprentissage automatique pour la prévision des interactions. En effet, nous pensons que, si on peut prévoir avec précision la structure du réseau, alors on a juste à rechercher les communautés sur le réseau prévu.

Des expérimentations sur des jeux de données réels montrent la faisabilité de cette approche.

1 Introduction

Les réseaux sociaux sont dynamiques par nature : de nouveaux nœuds et/ou de nouveaux liens arrivent et des nœuds et/ou des liens existants disparaissent. La détection de communautés a longtemps considéré uniquement l'aspect statique : une image du réseau est prise à un instant donné et les communautés y sont calculées. Récemment, des travaux sur la dynamique des communautés ont vu le jour. Certains auteurs essayent de suivre l'évolution des communautés dans le temps (Palla et al., 2007; Tantipathananandh et al., 2007; Asur et al., 2007), d'autre proposent de mettre à jour les communautés existantes en fonction des nouveaux évènements qui se produisent (ajout ou suppression de nœuds et/ou de liens) (Nguyen et al., 2011). Enfin les derniers essayent de trouver des communautés présentes sur plusieurs intervalles de temps (Aynaud et Guillaume, 2011).

Dans de nombreuses applications (Ngonmang et al., 2012b), on s'intéresse surtout à la communauté à laquelle appartient un nœud donné du réseau. Pour cette raison des méthodes de détection de communautés locales ont été proposées : en partant d'un nœud, ces méthodes explorent de proche en proche le voisinage et incluent à la communauté locale le nœud extérieur qui maximise le gain de la fonction objectif définie par la méthode. Des exemples de méthodes locales sont présentées dans (Bagrow, 2008; Ngonmang et al., 2012a). Avec ces méthodes, le suivi d'une communauté devient trivial : on a juste besoin de la re-calculer à la tranche de temps suivante.

Un des problèmes non encore explorés dans la littérature sur la dynamique des communautés est la prévision : connaissant l'évolution du réseau jusqu'au temps t, peut-on prévoir les communautés au temps $t + 1$? Dans cet article, nous proposons une approche générale de prévision de communautés basée sur la prévision des interactions dans les réseaux complexes. Dans cette approche, étant donné l'évolution du réseau jusqu'au temps t, les interactions sont prévus pour le temps $t + 1$ et les communautés sont ensuite calculées sur ce réseau prévu. L'hypothèse qui soutient cette démarche est la suivante : si on est capable de prévoir l'évolution du réseau avec précision, alors on peut utiliser le réseau prévu pour d'autres tâches (ici la prévision des communautés).

Les méthodes de prévision de liens (Lu et Zhou, 2011; Kamga et al., 2013) supposent généralement que le réseau est croissant : de nouveaux liens sont créés mais les liens existants sont définitifs. Toutefois, dans les réseaux réels, certains liens disparaissent avec le temps. Par exemple, dans un graphe d'appels, les liens d'un contact particulier évoluent avec le temps : il peut commencer de nouvelles relations (nouveaux collègues) et en terminer d'autres (expetites amies par exemple). Les méthodes de prévision de liens ne sont donc pas adaptées à ce contexte.

Dans cet article, le problème de prévision des communautés formalisé, ensuite, nous proposons des modèles simples et flexibles pour prévoir l'évolution des communautés. Cet article est organisé comme suit : la section 2 présente les définitions et notations de base. La section 3 présente les différentes approches de détection de communautés dans les réseaux dynamiques. La section 4 rappelle le problème de prévision de liens. La section 5 présente les modèles proposés pour la prévision d'interactions. Les performances de ces modèles et leur évaluation pour la prévision des communautés sont présentées en section 6. Finalement, la section 7 met en évidence les conclusions et quelques perspectives.

2 Définitions et notations de base

Une interaction se définit comme une action, effectuée à un instant donné, entre deux nœuds. Par exemple un utilisateur a envoie un message à un utilisateur b sur une plate-forme de messagerie instantanée, ou il écrit sur le mur de b dans un réseau social. Les interactions observées durant une tranche de temps constituent les liens du graphe. Ces liens peuvent être pondérés en fonction du nombre d'interactions.

Une communauté est un ensemble de nœuds qui présentent une forte densité de liens entre eux et une faible densité avec l'extérieur. Lorsque la communauté est identifiée en partant d'un nœud sans connaissance globale du réseau, on parle de communauté locale.

Une tranche de temps est un intervalle durant lequel le réseau est observé. Un réseau dynamique G avec un ensemble de nœuds V, un ensemble de liens E et n intervalles de temps

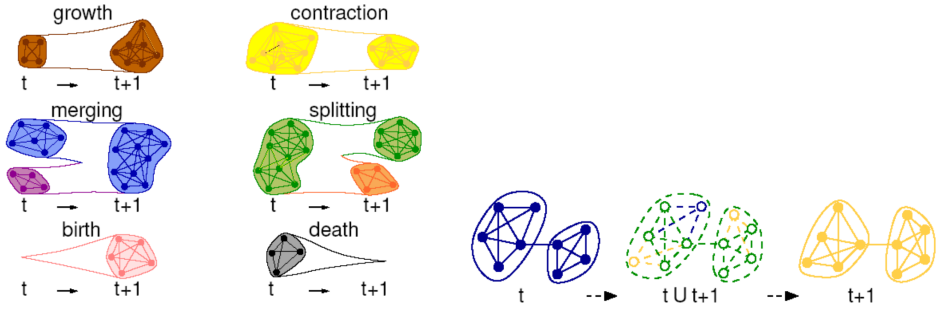


FIG. 1 – *Évolutions possibles des communautés (a) et étapes de détection de communautés dynamique de la méthode de (Palla et al., 2007) (b)*

est noté $G = (G_1, G_2, ..., G_n)$ avec $G_i =< V_i \in V, E_i \in E >$ le réseau observé pendant la tranche de temps i. Il est à noter que les tranches de temps peuvent être cumulatives (pour $j > i$, tous les liens de tranches t_i sont inclus dans la tranche t_j) ou non. Les modèles proposés dans la section 5 ne considèrent que des tranches de temps non cumulatives.

3 Détection de communautés dynamiques

Lorsqu'on parle de détection de communautés dans les réseaux dynamiques, il n'y a actuellement pas de consensus ni de définition universelle de ce qu'est une communauté. Nous présentons ici quelques méthodes pour le suivi de communautés, la mise à jour des communautés et la détection de communautés à long-terme (qui sont présentes sur plusieurs tranches de temps).

3.1 Suivi de communautés

L'idée générale de cette classe de méthodes est qu'on peut calculer les communautés à chaque intervalle de temps indépendamment et ensuite les faire correspondre entre différents couples de tranches de temps consécutives. Entre deux intervalles de temps consécutifs, pour une communauté, les différents événements suivants sont possibles : la continuation, la fusion, la division, la naissance, la mort. La figure 1 (a) présente ces différents événements.

Les méthodes de cette catégorie diffèrent, d'une part, par l'algorithme utilisé pour détecter les communautés à chaque tranche de temps et par la méthode utilisée pour mettre en correspondance les communautés des différentes tranches de temps, d'autre part. Palla et al. (2007) par exemple ont proposé d'utiliser la méthode de percolation de cliques (Palla et al., 2005) pour détecter les communautés à chaque tranche de temps. Pour faire la correspondance entre les communautés des tranches de temps consécutives, ces auteurs proposent de calculer les communautés sur le graphe union (voir fig. 1 (b) pour l'illustration). La correspondance est effectuée de la manière suivante : si une communauté du graphe union contient uniquement une communauté de chacune des tranches de temps consécutives alors c'est une continuation. Sinon, la correspondance est effectuée dans l'ordre décroissant des nœuds communs aux com-

munautés. Le principal inconvénient de cette méthode est qu'elle est basée sur des propriétés particulières de la méthode de percolation de cliques (Palla et al., 2005) (qui est très lente sur de grands graphes) et est donc difficilement utilisable avec d'autres algorithmes.

Greene et al. (Greene et al., 2010) ont proposé une méthode plus générale qui peut (en principe) être utilisée avec n'importe quelle méthode de détection de communautés statiques. La méthode proposé par ces auteurs consiste dans un premier temps à choisir une méthode de détection de communautés statiques et à l'utiliser pour détecter les communautés de chaque tranche de temps. Ensuite la correspondance entre les tranches est réalisée en utilisant la similarité de Jaccard. Cependant, cette méthode n'est pas adaptée aux algorithmes de détection de communautés non déterministes. Ce non déterministe peut être limité en utilisant les cœurs de communautés comme décrit dans (Seifi et al., 2013).

Tantipathananandh et al. (Tantipathananandh et al., 2007) ont défini le problème de suivi de communautés comme un problème de coloration de graphes. En dépit de cette formulation originale, la méthode de résolution proposée est coûteuse en temps de calcul.

3.2 Mise à jour de Communautés

Le principe de ces méthodes est de calculer les communautés uniquement à un instant de référence t_0 et ensuite de les mettre à jour avec les événements élémentaires de modification du réseau qui se produisent dans les tranches de temps suivantes. Ces événements élémentaires sont : l'ajout d'un lien, la suppression d'un lien, l'ajout d'un nœud avec k liens, la suppression d'un nœud avec k liens. Dans leur travaux, Nguyen et al. (2011) proposent des heuristiques pour gérer ces événements.

3.3 Communautés à long-terme

Cette dernière classe de méthodes consiste à détecter les communautés qui sont consistantes sur plusieurs tranches de temps. Aynaud et Guillaume (2011) ont proposé deux méthodes basées sur l'optimisation de la modularité. (Newman et Girvan, 2004). La première méthode consiste à construire un *réseau somme* et ensuite à utiliser un algorithme statique de détection de communautés sur ce réseau. La seconde méthode consiste à définir une modularité moyenne sur toutes les tranches de temps. Cette modularité moyenne est ensuite optimisée en utilisant un algorithme similaire à celui de Louvain (Blondel et al., 2008).

Mitra et al. (2011) ont proposé une méthode adaptée aux réseaux de type *citations*. Un réseau résumé est construit de la manière suivante : un nœud A_i est crée si l'auteur A a fait une publication au temps i, et un lien est créé entre deux nœud A_i et B_j si l'article de l'auteur A publié au temps i cite l'article publié par l'auteur B au temps j. Un algorithme de détection de communautés dans les réseaux statiques est ensuite utilisé sur ce réseau construit.

Ces méthodes sont souvent difficiles à évaluer expérimentalement car on a rarement accès à une vérité terrain donnant les communautés "réelles".

4 Prévision de liens

La prévision de liens est un sujet largement étudié dans la littérature sur la dynamique des réseaux (Lu et Zhou, 2011). Le problème de prévision de liens peut se définir comme suit : étant

donné la structure du réseau jusqu'au temps t, quels sont les nouveaux liens qui apparaîtront dans le futur ? Deux classes principales de méthodes existent : les méthodes basées sur la similarité et les méthodes d'apprentissage supervisé. Les méthodes basées sur la similarité calculent un score entre les paires de nœuds. Un seuil est ensuite choisi pour décider si oui ou non le lien doit être recommandé. Les méthodes classiques de cette catégorie sont décrites dans (Liben-Nowell et Kleinberg, 2003). Les méthodes supervisées définissent le problème de prévision de liens comme un problème de classification binaire et utilisent les méthodes existantes pour le résoudre. Un exemple de méthode supervisée est présenté dans (Hasan et al., 2006).

La prévision de liens n'est pas suffisante pour analyser la dynamique des communautés dans les réseaux sociaux. En effet, elle suppose que les liens crées existeront indéfiniment. Or, dans la réalité, des liens apparaissent tandis que d'autres créés disparaissent. Pour réellement prévoir la dynamique des communautés dans ce contexte, on a besoin de modèles plus généraux de prévision d'interactions.

Il convient tout de même de noter que, dans le cas d'un réseau strictement croissant, les méthodes de prévision de liens conviennent parfaitement. Dans la suite, nous allons donc nous limiter au cas de réseaux d'interactions.

5 prévision des interactions

Dans cette section, le problème de prévision des interactions est formalisé, puis un modèle basé sur la similarité et une approche par apprentissage supervisé sont proposés.

5.1 Définition du problème

Le problème de prévision des interactions peut être défini comme suit : étant donné un réseau dynamique $G = (G_1, ..., G_n)$ dont les tranches de temps sont non cumulatives (les liens correspondent aux interactions de la tranche de temps uniquement) quelle sera la structure du réseau (G_{n+1}) correspondant à la tranche de temps $n + 1$? Ce problème peut être vu comme une généralisation de la prévision de liens : on ne se limite pas aux liens non existants mais on vérifie aussi que les liens existants resterons présents. De ce fait, les mêmes classes de méthodes peuvent être utilisées pour le résoudre. Dans ce qui suit nous présentons une solution basée sur la similarité et une solution par apprentissage supervisée. Dans les approches proposées, le temps joue un rôle important.

5.2 Modèle basés sur la similarité

Nous commençons par présenter un modèle simple basé sur une mesure de similarité entre deux nœuds. Ce modèle sera ensuite utilisé comme point de comparaison (*baseline*). Cette mesure de similarité prend en compte le temps, les interactions existantes et le voisinage de chaque paire de nœuds considérée. La forme générale de cette mesure de similarité est :

$$Sim(i,j) \quad = \quad \sum_{t \in T} f(t) \ \times \ (\alpha W[i,j] \ + \ \beta g(neigh(i), neig(j)) \ + \ \theta h(i,j)) \quad (1)$$

Dans l'équation 1, les fonctions f et g et les paramètres α et β sont à définir. W est la matrice des poids. f est la fonction temporelle, elle permet de prendre en compte l'âge des interactions (donner plus d'importance aux relations récentes par exemple). g est la fonction de similarité topologique qui mesure la proximité dans le graphe social. h est la fonction de similarité entre les attributs (lorsqu'ils sont disponibles). Enfin $neigh(i)$ est une fonction de voisinage. Des exemples de voisinage qui peuvent être considérés sont : les voisins, les voisins et leurs voisins, la communauté (locale).

Les paramètres de ce modèle peuvent facilement optimisés en utilisant par exemple la méthode ***random restart Hill Climbing*** (Russell et Norvig, 2003). Le critère d'optimisation choisi dans nos expérimentations est l'aire sous la courbe ROC (Receiver Operating Characteristic)(Bradley, 1997).

Ce modèle est très intuitif. Cependant, il ne peut pas modéliser une large classe de relation possible entre les variables d'entrée et la variable à prévoir. Pour cette raison, nous proposons dans la suite un modèle plus général basé sur l'apprentissage supervisé.

5.3 Modèle d'apprentissage supervisé

Pour l'approche supervisée, nous proposons de procéder comme suit : pour chaque tranche de temps t de la période d'apprentissage, les attributs suivants dont calculés pour chaque paires de nœuds candidate :

- le nombre de voisins communs ;
- le nombre de membre communs dans la communauté locale (Ngonmang et al., 2012a) ;
- le nombre interactions pour cette tranche de temps ;
- le degré de chaque nœud ;
- le coefficient de clustering de chaque nœud ;
- les attributs des nœuds (si disponibles et éventuellement après transformation).

Les classes réelles sont obtenues sur la période de test. Il est à noter que, pour réduire la complexité, ne sont considérées que les paires de nœud dont les scores sont tous non nuls.

Une méthode d'apprentissage supervisé peut alors être utilisée pour construire le modèle de prévision. Tous les attributs sont numériques et peuvent être normalisés entre -1 et 1. Les expérimentations sont basées sur les Machines à Vecteurs de Support (SVM) en utilisant l'outil ***LIBSVM*** (Chang et Lin, 2011). Pour prendre en compte les relations non linéaires, un noyau Gaussien est utilisé. Pour compenser le déséquilibre entre les classes, un échantillonnage permet de retenir le même nombre d'exemples de chaque classes. Enfin les paramètres du modèle SVM sont sélectionnés en utilisant la méthode d'optimisation *grid search* (Chang et Lin, 2011).

Cette méthode est plus générale que celle présentée en section 5.2 et plus flexible si on veut ajouter d'autres attributs.

6 Evaluation et discussion

Dans cette section, les jeux de données utilisés sont d'abord décrits. Ensuite l'évaluation de la prévision des interactions est présentée. Enfin, l'évaluation de l'application à la prévision des communautés est présentée.

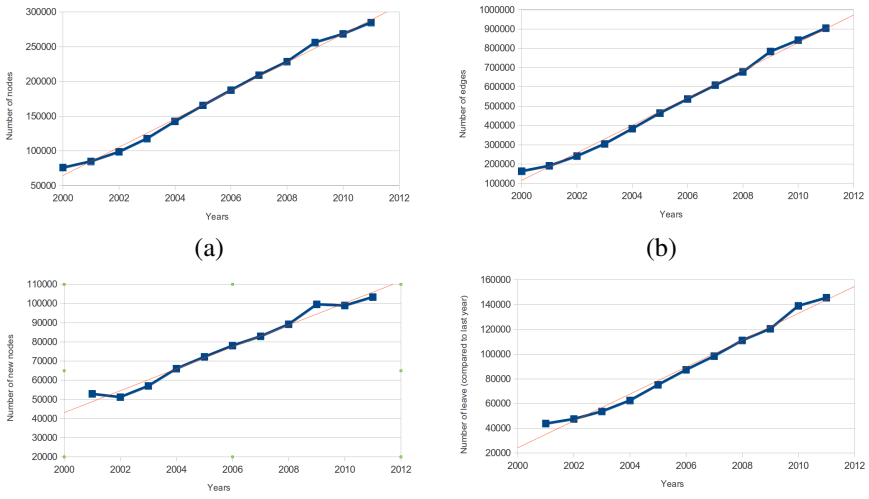


FIG. 2 – *Quelques statistiques sur DBLP.*

6.1 Description des jeux de données

Les jeux de données utilisés pour les évaluations sont *DBLP* et *Facebook wall*. DBLP est un réseau de collaboration entre auteurs indexés sur `http://dblp.uni-trier.de/`. Pour chaque tranche de temps (années), une interaction existe entre deux auteurs s'ils ont au moins une publication commune. Les liens sont pondérés par le nombre de publications communes. La figure 2 présente quelques statistiques sur ce jeux de données.

Le jeux de données Facebook wall (`http://konect.uni-koblenz.de/networks/facebook-wosn-wall`) est un réseau construit à partir d'un sous-ensemble d'utilisateurs de la New Orleans. Pour chaque année, un lien existe entre deux utilisateurs si l'un d'eux a publié sur le mur de l'autre. Les liens sont pondérés par le nombre de publications. La figure 3 présente quelques statistiques sur ce jeu de données. Ces statistiques montrent que ce réseau est très dynamique et que les nombres de nœuds et de liens croissent très rapidement avec le temps.

6.2 Évaluation de la prévision des interactions

Dans cette sous-section, les résultats sur la prévision des interactions sont présentés. En raison du déséquilibre entre les classes, l'aire sous la courbe ROC (Receiver Operating Characteristic) noté AUC (Area Under Curve) (Bradley, 1997) est utilisée pour évaluer les performances des approches.

Pour chaque jeu de données, les résultats du modèle basé sur la similarité sont présentés puis ceux du modèle supervisé.

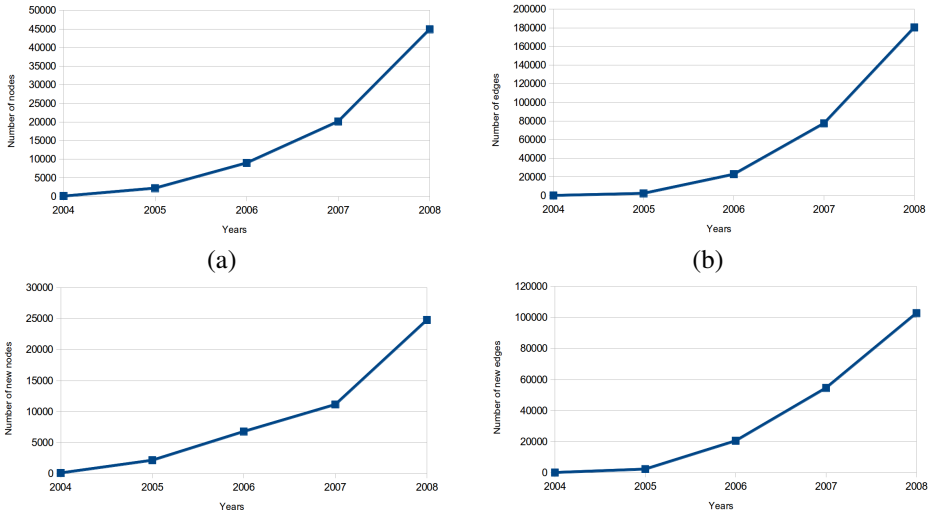


FIG. 3 – *Quelques statistiques sur Facebook Walls.*

6.2.1 DBLP Dataset

Le modèle basé sur la similarité présenté dans la section 5.2 est général. Pour chaque jeu de données, les paramètres sont à déterminer. L'équation utilisée est :

$$Sim_D(i,j) = \sum_{t \in T} (1/(n-t+1)) \times (0.67 \times ||W[i,j]|| + (0.33 \times Jaccard(com(i), com(j)))) \quad (2)$$

avec n le nombre d'intervalles de temps, $||x||$ la normalisation de la variable x entre 0 et 1 et $Jaccard(com(i), com(j))$ la similarité de Jaccard entre la communauté locale de i et celle de j. Les communautés locales sont calculées comme décrit dans Ngonmang et al. (2012a). Comme montré dans Ngonmang et al. (2012b), les communautés locales constituent un bon compromis entre le premier et le second voisinage.

Ce modèle produit une AUC de $0,69$ et est déjà meilleur qu'un modèle aléatoire dont l'*AUC* serait de $0,50$.

Le modèle supervisé pour ce jeu de données est construit comme décrit ci-dessus. Sans tenir compte des attributs des nœuds, il produit une AUC de $0,87$.

Dans ce jeu de données on dispose des titres des articles et les noms des conférences ou journaux dans lesquels ils sont publiés. Pour chaque auteur et chaque année, on peut donc construire un vecteur *TF-IDF* (Salton et McGill, 1986) relatif aux mots contenus dans les titres (en prenant soin de supprimer les mots vides (*"stop words"*)) de ses publications et les noms des conférences et/ou journaux dans lesquels il a publié. En tenant compte de ses attributs on obtient une valeur de l'*AUC* de $0,88$ très légèrement supérieure au modèle sans attributs. Nous

	DBLP	Facebook wall
Modèle aléatoire	0,50	0,50
Modèle basé sur la similarité	0,69	0,84
Approche supervisée sans attributs	**0,87**	**0,92**
Approche supervisée avec attributs	**0,88**	-

TAB. 1 – *Évaluation (AUC) des modèles de prévision d'interactions*

pensons que le modèle aurait été meilleur si nous avions à disposition des attributs plus riches comme les résumés des articles.

6.2.2 Jeux de données Facebook walls

Comme pour le jeu de données DBLP, nous avons obtenu par optimisation des paramètres le modèle suivant :

$$Sim_F(i,j) = \sum_{t \in T} (1/(n-t+1)) \times (0.74 \times ||W[i,j]|| + (0.24 \times Jaccard(com(i), com(j)))) \quad (3)$$

Avec ce modèle on obtient une AUC de $0,84$, largement supérieure à celle d'un modèle aléatoire.

Avec le modèle supervisé, on obtient sur ce jeu de données une AUC de $0,92$. Puisqu'on n'a pas à disposition les attributs pour ce jeu de données, c'est ce modèle qui sera utilisé pour la prévision des communautés.

La table 1 résume l'évaluation de la prévision des interactions.

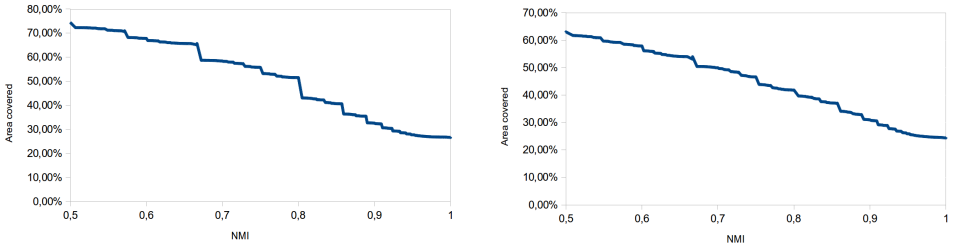
6.3 Évaluation de la prévision des communautés locales.

Dans cette section, les modèles de prévision des interactions sont utilisés pour prévoir les communautés.

L'indice de performance utilisé est l'Information Mutuelle dans sa version Normalisée (*NMI*) (Bagrow, 2008). Cet indice permet de comparer deux partitionnements en quantifiant l'information qu'ils partagent.

La méthode de détection de communautés locales utilisée dans les évaluations est décrite dans Ngonmang et al. (2012a). Cet algorithme optimise une fonction de qualité et est capable de détecter la structure de communauté recouvrante à laquelle appartient un nœud donné du réseau.

Pour l'évaluation, les communautés locales sont calculées dans le réseau réel et dans le réseau prévu. Les deux résultats sont ensuite comparés en utilisant le *NMI*. Cette évaluation ne prend en compte que les communautés locales des nœuds qui apparaissent dans le réseau réel et dans le réseau prévu.



(a) (b)

FIG. 4 – *Évaluation de la prévision des communautés locales.*

6.3.1 DBLP dataset

Les résultats de l'évaluation pour ce jeu de données sont présentés sur la figure 4 (a). On peut constater que pour plus de 30% des nœuds on a une prévision parfaite avec une valeur de $NMI = 1$. Plus de 50% des nœuds produisent un *NMI* supérieur à $0,8$ et enfin, plus de 70% des nœuds produisent un $NMI > 0,6$. Il est à noter que dans l'évaluation les nœuds qui apparaissent uniquement dans la période cible sont pris en compte dans le calcul du *NMI* (ils ne sont pas considérés comme nœuds de départ mais peuvent être membres des communautés). Néanmoins, on obtient de bons scores en moyenne.

6.3.2 Jeu de données Facebook walls

Comme pour DBLP, les résultats sont présentés dans la figure $4(b)$. Dans plus de 25% des cas on a une prévision parfaite avec un *NMI* de 1 et dans plus de 50% des cas on a un *NMI* supérieur à $0,7$.

7 Conclusions et perspectives

Récemment, de nombreux travaux sur la détection des communautés dans les réseaux dynamiques ont commencés. Un des problèmes encore non exploré est la prévision des communautés. Dans cet article, nous avons dans un premier temps proposé des modèles pour la prévision des interactions (basées sur la similarité et par apprentissage supervisé). Ensuite, nous avons utilisé ces modèles pour prévoir les communautés. Des tests sur des jeux de données réels montre la faisabilité de notre approche.

En perspective, nous pensons prendre en compte l'arrivée de nouveaux nœuds. Nous pensons aussi tester cette approche sur d'autres jeux de données.

Remerciement

Ce travail est partiellement financé par le projet FUI français AMMICO.

Références

Asur, S., S. Parthasarathy, et D. Ucar (2007). An event-based framework for characterizing the evolutionary behavior of interaction graphs. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, New York, NY, USA, pp. 913–921. ACM.

Aynaud, T. et J.-L. Guillaume (2011). Multi-Step Community Detection and Hierarchical Time Segmentation in Evolving Networks. In *Fifth SNA-KDD Workshop Social Network Mining and Analysis, in conjunction with the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2011)*.

Bagrow, J. P. (2008). Evaluating local community methods in networks. In *Journal of Statistical Mechanics*, pp. 05001.

Blondel, V. D., J. Guillaume, R. Lambiotte, et E. Lefebvre (2008). Fast unfolding of communities in large networks. In *Journal of Statistical Mechanics : Theory and Experiment*, pp. 10008.

Bradley, A. P. (1997). The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition 30*(7), 1145–1159.

Chang, C.-C. et C.-J. Lin (2011). LIBSVM : A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology 2*, 27 :1–27 :27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Greene, D., D. Doyle, et P. Cunningham (2010). Tracking the evolution of communities in dynamic social networks. In *Proceedings of the 2010 International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '10, Washington, DC, USA, pp. 176–183. IEEE Computer Society.

Hasan, M. A., V. Chaoji, S. Salem, et M. Zaki (2006). Link prediction using supervised learning. In *In Proc. of SDM 06 workshop on Link Analysis, Counterterrorism and Security*.

Kamga, V., M. Tchuente, et E. Viennet (2013). Prévision de lien dans les graphes bipartites avec attributs. *Revue des Nouvelles Technologies de l'Information (RNTI-A6)*, 67–85.

Liben-Nowell, D. et J. Kleinberg (2003). The link prediction problem for social networks. In *Proceedings of the twelfth international conference on Information and knowledge management*, CIKM '03, New York, NY, USA, pp. 556–559. ACM.

Lu, L. et T. Zhou (2011). Link prediction in complex networks : A survey. *Physica A : Statistical Mechanics and its Applications 390*(6), 1150 – 1170.

Mitra, B., L. Tabourier, et C. Roth (2011). Intrinsically dynamic network communities. *CoRR abs/1111.2018*.

Newman, M. et M. Girvan (2004). Finding and evaluating community structure in networks. In *Phy. Rev.*, Volume 69, pp. 026113.

Ngonmang, B., M. Tchuente, et E. Viennet (2012a). Local communities identification in social networks. *Parallel Processing Letters 22*(1).

Ngonmang, B., E. Viennet, et M. Tchuente (2012b). Churn prediction in a real online social network using local community analysis. In *IEEE/ACM Int. Conf. on Advances in Social Networks Analysis and Mining (ASONAM'12)*, pp. 282–290.

Nguyen, N., T. Dinh, Y. Xuan, et M. Thai (2011). Adaptive algorithms for detecting community structure in dynamic social networks. In *INFOCOM, 2011 Proceedings IEEE*, pp. 2282 – 2290.

Palla, G., I. Derényi, et T. Vicsek (2005). Uncovering the overlapping community structure of complex networks in nature and society. In *Nature*, Volume 435, pp. 814–818.

Palla, G., A. lászló Barabási, T. Vicsek, et B. Hungary (2007). Quantifying social group evolution. Volume 446, pp. 2007.

Russell, S. J. et P. Norvig (2003). *Artificial Intelligence : A Modern Approach*. Pearson Education.

Salton, G. et M. J. McGill (1986). *Introduction to Modern Information Retrieval*. New York, NY, USA : McGraw-Hill, Inc.

Seifi, M., I. Junier, J.-B. Rouquier, S. Iskrov, et J.-L. Guillaume (2013). Stable Community Cores in Complex Networks. In *Complex Networks*, Volume 424 of *Studies in Computational Intelligence*, pp. 87–98. Springer Berlin Heidelberg.

Tantipathananandh, C., T. Berger-Wolf, et D. Kempe (2007). A framework for community identification in dynamic social networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, New York, NY, USA, pp. 717–726. ACM.

Summary

Until recently all the works done on community detection in complex network have only considered static networks: a snapshot of the network is taken at a particular time. The communities are then computed on that constructed network. Because real networks are dynamic by nature, investigations on community detection in dynamic networks have started these last years.

One problem actually unexplored in community dynamic is the prediction: knowing the evolution of the network until the time-step t, can we predict the communities at the time-step $t + 1$?

In this paper, we propose a general approach for communities prediction based on a machine learning model predicting interaction in social networks. In fact, we believe that if one is able to predict the structure of the network with a high precision, then one just need to compute the communities on this predicted network to have the prediction of the community structure.

Evaluation on real datasets (DBLP and Facebook walls) shows the feasibility of the approach.

Détection d'opinions sur des lieux touristiques dans des tweets

Caroline Collet*,**, Alexandre Pauchet* Khaled Khélif**

*Laboratoire LITIS - EA 4108, {nom.prenom}@insa-rouen.fr
**Cassidian SAS - Office France{nom.prenom}@cassidian.com

Résumé. Depuis son lancement en 2006, Twitter ne cesse de croître en popularité et ainsi de fournir une source d'informations importante sous la forme de brefs messages. La détection d'opinions sur ce réseau social permet, par exemple, de suivre l'avis de consommateurs ou l'opinion de l'électorat. Dans cet article, nous décrivons une méthode permettant de déterminer l'opinion d'un tweet en détectant dans un premier temps sa subjectivité, puis sa polarité. Nous proposons d'utiliser une extraction de motifs séquentiels fréquents pour détecter la subjectivité et un SVM pour la détection de polarité. Sur ce dernier, plusieurs caractéristiques différentes en entrée ont été testées. Les performances des deux étapes ont été évaluées sur des corpus de test constitués manuellement.

1 Introduction

Avec l'avènement des réseaux sociaux tels Twitter ou Facebook, la fouille d'opinions sur Internet, jusqu'alors limitée aux forums et commentaires, s'est considérablement développée. Les institutions étatiques, tout comme les entreprises s'appuie souvent sur l'opinion publique pour orienter leurs décisions stratégiques. Savoir par exemple ce que la majorité d'une population pense d'un nouveau produit, permet de décider de la politique marketing d'une entreprise ou de la consolider.

Le réseau social Twitter offre aux utilisateurs la possibilité de poster publiquement des messages (tweets) limités à 140 caractères, dont le principe n'est pas sans rappeler celui des SMS. Avec l'arrivée des smartphones, c'est d'ailleurs de cette manière que beaucoup d'utilisateurs considèrent les tweets. Ainsi, Twitter est une révolution dans le domaine du Big data en offre une source de données considérable, notamment pour l'analyse des opinions qui y sont contenues.

La principale caractéristique des tweets est qu'ils sont extrêmement courts, il est donc le plus souvent impossible de cerner leur contexte d'écriture. La détection d'opinions dans un tweet est rendue plus difficile encore par le faible niveau de langue, le style phonétique et la quantité de fautes présentes. En effet à l'heure actuelle, la plupart des outils TALN sont spécialisés dans l'exploitation de textes dans un niveau de langue courant et une orthographe correcte. Les tweets n'en restant pas moins des textes, des méthodes classiques de détection d'opinions peuvent leur être appliquées, en les adaptant à leur taille réduite et leur style.

Une opinion, dans un tweet comme dans un texte plus long, peut être définie de deux manières différentes. Soit une opinion est considérée comme positive, négative ou neutre (Pak

et Paroubek, 2010), la détection d'opinions se ramenant alors à un problème de classification en 3 classes distinctes. Un second point de vue consiste à considérer qu'une opinion ne peut pas être neutre et donc qu'un texte n'est qu'objectif (équivalent à neutre), ou subjectif avec une polarité positive ou négative (Barbosa et Feng, 2010; Pang et Lee, 2004). Cette seconde approche, que nous adopterons, permet de décomposer le problème de détection d'opinions en une première étape d'évaluation de l'objectivité (ou de la neutralité), suivie dans le cas d'un texte subjectif, d'une seconde étape de détection de sa polarité (positive ou négative).

Ces travaux de recherche s'inscrivent dans le cadre d'un projet européen dont le domaine central est le tourisme. L'objectif de cet article est ainsi de présenter une méthode permettant de détecter si un tweet, écrit en anglais ou en français, présente une opinion sur un lieu touristique et de détecter la polarité de cette opinion (positive ou négative).

Dans la suite de cet article, nous présentons section 2 un état de l'art sur la détection d'opinions dans des contenus textuels. La section 3 décrit notre méthode de détection de subjectivité et son évaluation sur des tweets. La section 4 est dédiée à la détection de polarité. Enfin, la section 6 résume et analyse les résultats obtenus et propose des pistes pour nos travaux futurs.

2 La détection d'opinions dans des textes

Les tâches d'évaluation de la subjectivité et de la polarité d'un texte sont très similaires et peuvent être résolues avec des approches identiques. À l'heure actuelle, trois grandes classes de méthodes peuvent être considérées : les méthodes symboliques qui s'appuient sur des règles manuelles ou extraites automatiquement pour classer des textes, les méthodes statistiques qui se basent sur les fréquences d'apparition de mots pour effectuer une classification et les méthodes hybrides qui combinent approches symboliques et statistiques.

2.1 Méthodes symboliques pour la détection d'opinion

Les méthodes symboliques consistent en la constitution d'un ensemble de règles de décision servant à classer un texte dans une catégorie (ici objectif/subjectif ou positif/négatif). Ces règles peuvent être éditées de manière manuelle ou semi-manuelle par un expert humain et sont directement appliquées aux textes à évaluer. Un spécialiste du domaine (par exemple un linguiste) doit donc définir les règles grammaticales permettant de classer les textes. L'inconvénient de cette approche est qu'elle est très coûteuse et ne peut malheureusement pas être exhaustive. De plus, les règles sont souvent très précises et engendrent une bonne précision mais un rappel généralement faible.

Afin de diminuer l'investissement humain, les règles définies peuvent être plus générales. Par exemple pour déterminer la valence d'un texte, il est possible de s'appuyer sur une ressource linguistique comme Sentiwordnet (Baccianella et al., 2010; ¸Serban et al., 2012) et d'effectuer un calcul de valence sur les phrases. Il existe plusieurs formules allant de la somme algébrique (Hamouda et Rohaim, 2011) à des algorithmes utilisant des arbres de dépendance (Yuanbin Wu et Wu, 2009), en passant par l'information mutuelle (Hu et Liu, 2004) par exemple. Ces approches restent néanmoins peu précises et ne permettent pas la détection d'opinions pour des structures linguistiques compliquées.

Une alternative à la création manuelle de règles de décision est l'utilisation d'algorithmes permettant d'apprendre des règles humainement compréhensibles et réutilisables pour classer

les textes. Les algorithmes génétiques, entre autres, le permettent et peuvent même être très efficaces. Par exemple, Das et S. Bandyopadhyay (Das et Bandyopadhyay, 2010) les ont testés sur des textes de news anglaises et ont obtenu d'excellents scores de précision et de rappel avec le corpus MPQA (Wiebe et al., 2005). Une vérification humaine est cependant nécessaire et de plus les algorithmes génétiques nécessitent souvent un temps de traitement très long.

2.2 Les méthodes statistiques

Les méthodes statistiques s'appuient sur des vecteurs de caractéristiques encodant les textes (par exemple le nombre d'occurrences ou la présence/absence de mots) afin d'en extraire des propriétés par apprentissage.

Cet apprentissage peut s'effectuer sans intervention humaine (apprentissage non supervisé), par une extraction automatique de classes (clustering). Ces méthodes sont très pratiques puisqu'elles ne nécessitent pas de corpus annoté pour l'apprentissage et permettent même de préciser le nombre de classes à rechercher. Ces méthodes sont cependant à utiliser avec méfiance car si les caractéristiques choisies ne sont pas pertinentes, les classes obtenues ne correspondent pas aux catégories désirées (Hatzivassiloglou et McKeown, 1997).

Lorsqu'une intervention humaine est nécessaire, par exemple pour étiqueter le corpus d'apprentissage suivant les catégories recherchées, on parle alors d'apprentissage supervisé. Ce type de méthodes est couramment utilisé en analyse d'opinions, et en particulier Bayes Naïf et Support Vector Machine (SVM). SVM est plus précis et efficace la plupart du temps mais est très dépendant du corpus d'apprentissage. Généralement, les caractéristiques utilisées sont les lemmes des mots associés à leur catégorie grammaticale. Des tests ont aussi été effectués avec la fréquence des mots mais il a été démontré que cette méthode est moins pertinente (Pang et Lee, 2008). Pour améliorer la performance du modèle, il est aussi possible d'utiliser un SVM polynomial. Par exemple, Roy de Groot (de Groot, 2010) a testé une méthode à base de SVM sur des tweets avec un vecteur contenant les mots les plus fréquents et un vecteur contenant les mots à caractère sentimental. Il s'avère qu'il est plus judicieux d'utiliser les mots les plus présents.

2.3 Les méthodes hybrides

Les méthodes hybrides, mélangeant apprentissage statistique et édition de règles -le plus souvent manuelles-, sont actuellement en pleine expansion. Par exemple Dziczkowskid (DZICZKOWSKI, 2008) a combiné 3 méthodes différentes : un classifieur SVM, une méthode basée sur des règles et une méthode de Clustering. Le résultat de ces trois méthodes est combiné dans un réseau de neurones. La sélection de motifs séquentiels les plus fréquents (Serrano et al., 2012) peut aussi être classée dans la catégorie des méthodes hybrides.

Les méthodes hybrides semblent efficaces à condition d'effectuer les bonnes combinaisons.

2.4 Discussion

En l'absence de ressources humaines suffisantes pour procéder à une édition manuelle de règles, les méthodes statistiques ou hybrides doivent être considérées. Les méthodes d'apprentissage supervisé sont les plus efficaces mais nécessitent la création d'un corpus annoté. En

l'absence de corpus, il est possible d'utiliser des méthodes hybrides alternatives telles que l'extraction de motifs séquentiels fréquents (Serrano et al., 2012).

Dans le cas de la détection de subjectivité, à notre connaissance, il n'existe pas de corpus de tweets objectifs (ou subjectifs) annotés. N'ayant pas les ressources pour en construire un manuellement, ni de méthode pour le construire automatiquement, nous proposons une approche pour la détection de subjectivité nécessitant un corpus d'apprentissage mais non annoté. Nous avons choisi de tester une méthode hybride utilisée par le Greyc, basée sur l'extraction automatique de motifs séquentiels les plus fréquents dans un texte et sur la sélection manuelle de ces motifs afin de ne conserver que les plus pertinents (Serrano et al., 2012).

Dans le cas de la détection de polarité, un corpus annoté peut être constitué (Go et al., 2009). Nous proposons d'appliquer dessus un apprentissage supervisé de type SVM, réputé plus efficace.

3 Détection de subjectivité par motifs séquentiels fréquents

La méthode des motifs séquentiels fréquents se charge d'extraire les motifs les plus fréquents retrouvés dans un corpus d'apprentissage et un vérificateur humain s'occupe ensuite de sélectionner, dans les motifs retournés, les plus pertinents par rapport au contexte d'utilisation (ici la subjectivité). L'idée est de combiner les performances d'un système de détection automatique de règles, qui obtiendra une couverture importante de l'ensemble des règles que l'humain précisera en sélectionnant les règles à conserver.

3.1 Description de l'extraction des motifs séquentiels fréquents

Trois concepts de base sont utilisés dans cette méthode : les items (une information sur un mot, par exemple le mot lui-même, son lemme ou sa catégorie grammaticale), les itemsets (un ensemble d'items représentant les informations sur un mot de la phrase) et les séquences (un ensemble d'itemsets qui représente généralement une phrase). Dans notre cas, nous avons considéré qu'un tweet ne possède qu'une seule séquence. Les textes utilisés pour l'apprentissage doivent être en premier lieu transformés en séquences d'itemsets.

Voici un exemple de séquence :

{Je je SUJET}{suis être VB}{en en PREP}{France France NP}

Chaque groupe de mots entouré d'accolades correspond à un itemset. Les items sont les éléments contenus dans l'itemset. Ainsi, {Je je SUJET} correspond au premier itemset et est composé de trois items. Ici, le premier item correspond au mot d'origine, le second à son lemme et le troisième à sa catégorie grammaticale.

La méthode permet d'extraire des motifs, c'est-à-dire des combinaisons d'items ou de suites d'items retrouvées fréquemment dans l'ensemble des textes. Pour générer nos séquences, nous avons placé, pour chaque mot du texte, 3 items dans chaque itemset : le mot, sa catégorie grammaticale, son lemme (pour le français) ou son stem (pour l'anglais). Pour diminuer le bruit induit par les URL, les hashtags et autres informations propres à Twitter, ont été identifiés et remplacés par une annotation particulière : une annotation pour les URL, une pour les hashtags, une pour les noms d'utilisateurs et une par émoticônes.

Nous avons établi une liste de termes subjectifs basée sur Sentiwordnet (Baccianella et al., 2010) en ne conservant que les verbes et les adjectifs. En effet, nous émettons l'hypothèse que les autres catégories grammaticales sont dépendantes du contexte et risqueraient donc d'apporter du bruit. Afin de ne conserver que les termes marquant une forte subjectivité, nous avons filtré automatiquement cette liste en ne conservant uniquement que les adjectifs et verbes avec une valence absolue dépassant les 75%. Cette liste a ensuite été stemmée ou lemmatisée pour pouvoir retrouver ses éléments dans les tweets à traiter. Enfin, si un terme de la phrase appartient à cette liste, nous ajoutons le suffixe -SUBJ à chacun de ses items.

Dans la méthode des motifs séquentiels, plusieurs paramètres peuvent être faits varier afin d'affiner l'extraction des motifs. Tout d'abord, il faut préciser le nombre minimum et le nombre maximum d'itemsets à utiliser pour construire les motifs. Étant dans le cas de tweets, ceux-ci peuvent être extrêmement courts. Nous avons donc considéré 3 itemsets au minimum pour être sûrs de couvrir des opinions et non des émotions, et un maximum de 100 afin d'être certains de pouvoir englober le tweet complet. Dans un second temps, il faut préciser le support, c'est à-dire la fréquence minimum des motifs à garder. Devant par la suite sélectionner manuellement les motifs les plus pertinents, nous n'avons conservé que 1000 motifs environ (support entre 100 et 200). Enfin, il est possible de fixer le gap, c'est-à-dire le nombre d'itemsets à ignorer lors de la génération des motifs (par défaut, les motifs sont générés en combinant les itemsets situés côte-à-côte dans le texte). Nous avons choisi de tester notre méthode sur l'anglais et le français, avec à chaque fois un gap variant de 0 à 2 : 0 pour conserver la structure de la phrase et 2 pour apporter plus de flexibilité aux motifs.

3.2 Description des corpus d'apprentissage et de test

Pour extraire des motifs subjectifs, un corpus d'apprentissage de tweets subjectifs doit être fourni au système. Pour ce faire, nous avons utilisé l'API de rapatriement de Twitter et nous avons extrait 300 000 tweets pour le français et 300 000 pour l'anglais. Nous avons suivi le principe de (Go et al., 2009) qui considèrent qu'un tweet contenant un émoticône est forcément subjectif. Nous avons donc récupéré uniquement des tweets présentant un émoticône en le précisant comme critère de recherche dans l'API. Travaillant dans le cadre d'un projet touristique, notre but est de vérifier la subjectivité par rapport à un lieu touristique donné. Tous les tweets du corpus doivent donc faire référence à un lieu touristique. À l'aide d'un service de détection d'entités nommées développé par Cassidian, chaque entité nommée détectée dans le texte d'un tweet a permis l'ajout d'une métadonnée. Nous avons ainsi pu filtrer et supprimer du corpus tous les tweets ne contenant pas d'entité nommée de type lieu touristique. Le corpus final d'apprentissage est ainsi constitué de 5 000 tweets pour l'anglais et 4 000 pour le français.

Nous avons également construit par la même méthode, un corpus de test. 20 lieux touristiques célèbres (Bali, tour Eiffel, ...) ont alors été utilisés comme critère de recherche dans l'API pour collecter des tweets. Moins de tweets ayant été collectés en français qu'en anglais, un corpus de test de 700 tweets pour le français et 800 pour l'anglais, ont ainsi été constitués et annotés manuellement (quantité suffisamment petite pour être annotée manuellement et suffisamment grande pour obtenir des résultats exploitables).

3.3 Résultats obtenus

Après sélection automatique des motifs avec un support fixé entre 100 et 200 pour obtenir une quantité de motifs analysable manuellement, nous obtenons environ 1 000 motifs pour chacune des deux langues. Nous pouvons ainsi sélectionner manuellement les motifs permettant d'extraire la subjectivité.

Le tableau 1 montre des exemples de motifs obtenus pour le français et pour l'anglais.

anglais	français
{NN}{VBG|SUBJ}{to}	{PRO}{NOM}{SMILEY}
{PRP}{will}{VB|SUBJ}	{NOM}{VER}{NOM}{SMILEY}
{ !}{UH}{.}	{ !}{ !}{ !}
{.}{ !}{ !}{.}	

TAB. 1 – *Exemples de motifs séquentiels fréquents obtenus. UH = interjection, NN = nom commun, VBG = verbe au participe présent, VB = base verbale, PRP = pronom possessif, PRO = pronom, SMILEY = émoticône, VER = verbe*

Dans la suite de cet article, trois métriques seront utilisées pour présenter nos résultats :

– la précision : $\frac{nombre\ de\ vrais\ positifs}{nombre\ de\ vrais\ positifs+nombre\ de\ faux\ positifs}$

– le rappel : $\frac{nombre\ de\ vrais\ positifs}{nombre\ de\ vrais\ positifs+nombre\ de\ faux\ negatifs}$

– l'accuracy : $\frac{nombre\ de\ vrais\ positifs+nombre\ de\ vrais\ negatifs}{nombre\ de\ documents}$

Les résultats obtenus pour l'anglais sont présentés dans la table 2 et les résultats obtenus pour le français dans la table 3.

Les résultats obtenus sont mitigés. De notre point de vue, cette méthode ne permet pas de détecter la subjectivité d'un tweet avec suffisamment de précision. En étudiant manuellement les motifs extraits, nous avons observé que dans le plus souvent seules les catégories grammaticales, la ponctuation et les émoticônes y sont présents. Or, ces informations ne sont pas suffisantes pour détecter la subjectivité. Par exemple, nous obtenons des motifs tels que : {.}{ !}{ !}{.} ou encore {NOM}{VER}{NOM}{SMILEY}.

Les motifs retournés sont donc essentiellement basés sur la ponctuation et la présence d'émoticônes comme marque de la subjectivité, ce qui n'est pas suffisamment discriminant.

4 Détection de polarité

Bien que notre méthode de détection de subjectivité ait été jugée insatisfaisante, nous nous sommes également intéressé à la détection de polarité. Pour ce faire, nous une méthode d'apprentissage statistique de type SVM pour effectuer une classification. (Pang et Lee, 2008) ayant montré que la présence des mots était une information plus pertinente à utiliser comme caractéristique d'entrée que leur fréquence ou le score TF-IDF, nous avons donc encodé les tweets à classer à l'aide de vecteurs représentant la présence/absence de mots d'un index. Nous avons donc commencé par créer notre index en stemmant l'ensemble du corpus anglais et lemmatisé

	Précision	Rappel	Accuracy
gap 0	62.5%	61.5%	69%
gap 1	62.5%	62%	66%
gap 2	64%	62%	67%

TAB. 2 – *Résultats obtenus sur le corpus anglais.*

	Précision	Rappel	Accuracy
gap 0	61.5%	58.5%	62%
gap 1	58%	69.5%	65%
gap 2	65%	66%	66%

TAB. 3 – *Résultats obtenus sur le corpus français.*

le corpus français. Nous avons associé aux stems ou aux lemmes leur catégorie grammaticale afin de limiter les confusions entre mots. Nous avons ensuite remplacé les URL et les noms d'utilisateurs par une annotation pour qu'ils n'apparaissent qu'une seule fois dans l'index. Nous avons également supprimé certains mots vides tels que les déterminants.

4.1 Description des corpus d'apprentissage et de test

(Go et al., 2009) sont partis du principe qu'un tweet contenant un émoticône positif formulait une opinion positive et inversement pour ceux contenant des émoticônes négatifs. Ainsi, ils sont parvenus, en ne collectant que des tweets possédant un émoticône à construire un corpus d'opinions annoté. Ils ont par la suite supprimé les émoticônes pour éviter de biaiser l'apprentissage. Nous sommes partis de leur corpus de 1,6 millions de tweets pour l'anglais et nous en avons constitué un corpus de 300 000 tweets pour le français en appliquant la même méthode. Nous avons par contre conservé une partie des émoticônes dans le corpus d'apprentissage en respectant leur fréquence (24% des tweets seulement présentent un émoticône) pour que le système puisse s'en servir pour générer ses règles d'apprentissage. Ainsi, nous avons collecté 50 000 tweets aléatoirement, sans effectuer de sélection sur les émoticônes avant de n'en conserver que la même proportion pour les tweets positifs et pour les tweets négatifs. Finalement, nous avons filtré et conservé dans le corpus uniquement les tweets contenant une entité nommée de type lieu touristique, de la même manière que lors de la constitution du corpus subjectif.

Le corpus de test de la phase de détection de subjectivité, constitué respectivement de 700 et 800 tweets en français et en anglais, a été réutilisé et annoté manuellement selon la valence de chaque tweet.

4.2 Résultats obtenus

Nous avons effectué une sélection des termes à conserver dans l'index en se basant sur leur fréquence d'apparition dans les textes du corpus d'apprentissage. Pour cela, nous avons

calculé le nombre de termes apparaissant dans l'index en fonction de la fréquence des mots. Ainsi, pour l'anglais, nous avons pu apercevoir deux sauts, le premier à 50 000 termes dans l'index et un second à 1 000 termes. Pour le français, nous avons également observé deux sauts, l'un à 10 000 termes et l'autre à 2 000.

Pour l'anglais, nous avons dans un premier temps effectué un test avec 1 000 termes (caractéristiques) sur une portion réduite du corpus (300 000 documents) pour réduire le temps d'apprentissage. Nous avons ensuite testé sur l'ensemble des documents représentés par ces mille termes, c'est-à-dire 900 000. Et pour finir, nous avons effectué un test avec 50 000 caractéristiques sur le corpus complet.

Pour le français, nous avons effectué deux tests sur le corpus complet.

Les résultats obtenus pour l'anglais sont présentés dans la table 4

	Précision	Rappel	Accuracy
300 000 documents/1 000 termes	71.5%	65%	70.8%
900 000 documents/1 000 termes	74%	73%	73%
900 000 documents/50 000 termes	62%	62%	62%

TAB. 4 – *Premiers résultats obtenus sur le corpus anglais.*

De même, les résultats obtenus pour le français sont présentés dans la table 5

	Précision	Rappel	Accuracy
2 000 termes	58%	58%	59%
10 000 termes	62%	61%	63%

TAB. 5 – *Premiers résultats obtenus sur le corpus français.*

En observant ces premiers résultats sur le corpus anglais, à caractéristiques égales, les performances sont meilleures lorsque l'on considère le corpus complet. En revanche, on s'aperçoit que les performances sont moins bonnes lorsque l'on augmente le nombre de caractéristiques jusqu'à 50 000. Ainsi, il semblerait que les caractéristiques supplémentaires aient apporté du bruit à la place d'information pertinente. En ce qui concerne le corpus français, les performances, contrairement à l'anglais, sont bien meilleurs avec 10 000 caractéristiques. Ainsi, au lieu d'apporter du bruit, les caractéristiques supplémentaires apportent de l'information manquante dans le cas des 2000 caractéristiques.

En analysant manuellement les tweets classés par le système, nous avons pu nous apercevoir que deux pistes d'améliorations sont possibles. En effet, le système gère assez mal la négation. Cette faiblesse a pour conséquence d'inverser totalement la polarité de certaines phrases. Par exemple « I am not sure this is a good idea » va être considéré de manière positive. Il faut donc trouver un moyen pour inclure la négation dans l'apprentissage.

De plus, le système utilise, pour la sélection des termes à conserver dans l'index, une méthode basée uniquement sur la fréquence. Or, dans le cadre de la détection de polarité, certains mots rares peuvent être très pertinents. Par exemple « se délecter » en français, est un terme très positif mais peu usité.

Nous proposons ci-après deux méthodes afin de solutionner ces problèmes.

4.3 Gestion de la négation

Nous pensons que pour pouvoir gérer efficacement la négation, il est nécessaire de prendre en compte l'ordre des mots dans le modèle d'apprentissage. Or, la méthode de (Pang et al., 2002) transforme le document en un sac de mots qui perd complètement la notion d'ordre dans la phrase.

Quelques tentatives ont été effectuées en utilisant des n-grammes, c'est-à-dire des combinaisons de mots. Les n-grammes ont l'avantage de permettre de conserver partiellement l'ordre de la phrase d'origine. En revanche (Pang et al., 2002) a montré que l'accuracy est moins élevée avec des n-grammes que dans le cas d'uni-grammes. Nous pensons pouvoir l'expliquer par le fait que plus les n-grammes sont grands, plus le nombre de combinaisons augmente et donc plus le support diminue. Nous aimerions donc trouver une méthode permettant de prendre en compte la place des mots dans la phrase d'origine, sans avoir à effectuer des combinaisons de mots.

Pour ce faire, nous avons parcouru l'ensemble des documents et pour chaque terme à conserver dans l'index, nous avons conservé la position (ou les positions dans le cas d'un terme apparaissant plusieurs fois dans un même tweet), puis nous avons normalisés ces positions en fonction de la taille de chaque tweet.

Le schéma 1 décrit le fonctionnement de cette méthode.

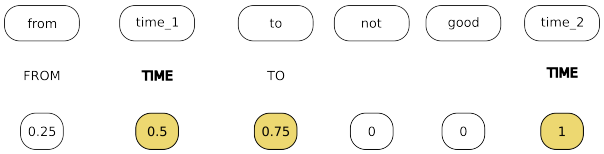


FIG. 1 – *Schéma de la construction du vecteur à partir de l'index*

La première ligne du schéma 1 correspond à l'index, la seconde ligne correspond au document et la troisième au document traduit en vecteur.

Les résultats obtenus pour le français et l'anglais sont présentés dans la table 6.

	Précision	Rappel	Accuracy
anglais	60%	60%	60%
français	62.6%	57%	62%

TAB. 6 – *Résultats obtenus sur le corpus anglais et français avec la méthode de la position des mots.*

On s'aperçoit que les performances pour l'anglais sont moins bonnes que dans le cas d'une simple présence des mots puisque le rappel et la précision ont chuté de presque 10%. En revanche, pour le français, les performances sont similaires aux autres tests. Ayant perdu en performance, il est probable que l'information de position absolue soit trop précise et qu'il faille plutôt considérer une position relative.

4.4 Sélection des termes de l'index

Dans les deux tests précédents, les termes de l'index sont sélectionnés uniquement via la fréquence. Or, certains mots peu fréquents peuvent être fortement discriminants dans le cadre de la détection de polarité (par exemple « se délecter »). Inversement, certains mots fréquents peuvent ajouter du bruit dans le cadre de la détection de subjectivité (tels que « Facebook » ou « Twitter » qui reviennent très fréquemment, mais n'apporte aucune information de polarité). Afin de solutionner ce problème, nous proposons ci-après deux méthodes que nous n'avons malheureusement pas eu le temps de tester.

La première méthode consiste à parcourir l'index généré à l'aide des mots les plus fréquents et de supprimer manuellement ceux n'ayant pas de lien avec le thème de la polarité des opinions. Par exemple le terme « manger » n'a pas d'influence sur la polarité d'une phrase, et pourrait donc être supprimé.

Dans un second temps, nous proposons d'ajouter dans l'index uniquement les termes fortement subjectifs.

5 Évaluation globale

Pour la détection de subjectivité, (Barbosa et Feng, 2010) ont comparé quatre méthodes de détection de subjectivité pour l'anglais. Ils ont utilisé le taux d'erreur comme métrique. La meilleure méthode a un taux d'erreur de 18.1. La moins bonne possède un taux de 32 et en moyenne le taux d'erreur est d'environ 25. En calculant le taux d'erreur dans notre cas, nous obtenons 33 pour le français et 34 pour l'anglais. Quoique proches, nous sommes donc légèrement en dessous des performances obtenues par (Barbosa et Feng, 2010). Attention cependant, il faut tenir compte du fait que le corpus de test est différent dans les 2 cas et donc non réellement comparable, mais l'ordre de grandeur reste le même. En particulier, le corpus que nous avons utilisé a été annoté manuellement avec des taux d'accord inter-annotateur assez bas.

Nous avons obtenu des résultats acceptables dans la cas de la détection de polarité puisque (Go et al., 2009) ont obtenu 82% d'accuracy sur leur corpus de test anglais. Cependant, une fois encore, le corpus de test que nous avons utilisé est différent du leur. Dans le cas du français, nous n'avons pas d'élément de comparaison puisque nous n'avons pas trouvé d'étude dans la littérature portant sur un tel corpus.

6 Conclusion et perspectives

Dans cet article, nous avons proposé une méthode permettant dans un premier temps de détecter la subjectivité d'un tweet et dans un second temps d'en évaluer sa polarité dans le cas d'un tweet subjectif.

Pour la détection de subjectivité, ne possédant pas de corpus annoté, nous avons testé une méthode utilisée dans le domaine de l'extraction d'informations et ne nécessitant pas de corpus : l'extraction de motifs séquentiels fréquents. Les performances obtenues sont moyennement satisfaisantes, mais semblent liées au fait que les motifs extraits ne contiennent pas suffisamment d'information pertinente pour le contexte de la subjectivité.

Une possible solution serait d'utiliser une base de connaissances pour remplacer l'ensemble des mots subjectifs par leur concept englobant. Par exemple, « aimer », « adorer »,

« kiff », « like », « apprécier » deviendraient le concept « amour ». Il faudrait cependant conserver l'information de la classe grammaticale du terme d'origine. Dans le cas contraire, le système pourrait ne conserver dans les motifs que le concept englobant et on ne saurait plus quelle est sa position dans la phrase. Le principe serait donc d'avoir une combinaison CONCEPT-classe grammaticale comme par exemple « AMOUR-VB », « HAINE-NOM ». L'objectif est ainsi, en remplaçant les termes par leur concept, de réduire le support de projection. Plus un item est fréquent, plus la probabilité qu'il soit sélectionné dans un motif est importante. On devrait ainsi pouvoir obtenir l'information nécessaire à une sélection manuelle de motifs séquentiels fréquents pour la détection de subjectivité.

Dans le cas de la détection de polarité, nous avons obtenu des performances raisonnables bien que légèrement inférieurs à ceux de la littérature, ce qui peut s'expliquer par une difficulté à sélectionner les paramètres optimaux du SVM.

Dans le but d'intégrer la négation, nous avons pu voir que prendre en compte dans les vecteurs la position absolue de chaque terme semble dégrader les performances du système. C'est pourquoi il est sans doute préférable de créer une méthode permettant de prendre en compte la position relative des termes. En l'occurrence, seule la position de la négation semblent être une information réellement pertinente. (Pak et Paroubek, 2010) ont créé des bigrammes dans lesquels ils ont inclus une information sur la négation. Ils ont en effet créé des bigrammes avec les termes englobant la négation. Par exemple « I don't like it » donne deux bigrammes : « do-NOT » et « NOT-like ». Cette manière de procéder permet de situer la position de la négation dans la phrase mais ne permet pas de savoir exactement sur quoi porte cette dernière. En effet, elle pourrait très bien s'appliquer à l'ensemble de la phrase comme à un mot unique. Elle n'est donc pas suffisante. Effectuer une analyse lexicale complète de la phrase pour détecter précisément sur quel mot ou groupe de mots se porte la négation et ajouter cette information dans l'index pourrait s'avérer être une bonne alternative.

Références

Baccianella, S., A. Esuli, et F. Sebastiani (2010). Sentiwordnet 3.0 : An enhanced lexical resource for sentiment analysis and opinion minings. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.

Barbosa, L. et J. Feng (2010). Robust sentiment detection on twitter from biased and noisy data. In C.-R. Huang et D. Jurafsky (Eds.), *COLING (Posters)*, pp. 36–44. Chinese Information Processing Society of China.

¸Serban, O., A. Pauchet, A. Rogozan, et J.-P. Pécuchet (7 pages, 2012). Semantic propagation on contextonyms using sentiwordnet. In *Workshop Affects, Compagnons Artificiels et Interaction*, Grenoble, France.

Das, A. et S. Bandyopadhyay (2010). Subjectivity detection using genetic algorithm. *the 1st Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA10)*.

de Groot, R. (2010). *Data Mining for Tweet Sentiment Classification*. Ph. D. thesis.

DZICZKOWSKI, G. (2008). *Analyse des sentiments :système autonome d'exploration des opinionsexprimées dans les critiques cinématographiques*. Ph. D. thesis.

Go, A., R. Bhayani, et L. Huang (2009). Twitter sentiment classification using distant supervision. *Processing*.

Hamouda, A. et M. Rohaim (2011). Reviews classification using sentiwordnet lexicon. *The Online Journal on Computer Science and Information Technology (OJCSIT)*.

Hatzivassiloglou, V. et K. R. McKeown (1997). Predicting the semantic orientation of adjectives.

Hu, M. et B. Liu (2004). Mining and summarizing customer reviews. *Conference on Knowledge discovery and data mining*.

Pak, A. et P. Paroubek (2010). Twitter as a corpus for sentiment analysis and opinion mining.

Pang, B. et L. Lee (2004). A sentimental education : Sentiment analysis using subjectivity summarization based on minimum cuts. In *In Proceedings of the ACL*, pp. 271–278.

Pang, B. et L. Lee (2008). Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*.

Pang, B., L. Lee, et S. Vaithyanathan (2002). Thumbs up ? sentiment classification using machine learning techniques. *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Serrano, L., T. Charnois, S. Brunessau, B. Grilheres, et M. Bouzid (2012). Combinaison d'approches pour l'extraction automatique d'événements. *Actes de la 19e conférence sur le Traitement Automatique des Langues Naturelles (TALN'2012)*.

Wiebe, J., T. Wilson, et C. Cardie (2005). Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*.

Yuanbin Wu, Qi Zhang, X. H. et L. Wu (2009). Phrase dependency parsing for opinion mining. pp. 1533–1541.

Summary

Since 2006, Twitter is growing in popularity and therefore provide an important source of information in the form of short messages. Opinion detection on this social network enables, for instance, to assess consumers' opinion or the views of the electorate. In this paper, we describe a method to determine the opinion of a tweet by detecting its subjectivity and its polarity. We propose to use a sequential pattern mining approach to detect subjectivity and an SVM to compute polarity. On the latter, several input characteristics were tested. The performance of the two stages were evaluated on test corpora constructed manually.

Étude de cas sur DBpédia en français

Jungyeul Park, Mouloud Kharoune, Arnaud Martin

UMR 6074 IRISA, Université de Rennes 1, Lannion, France
{jungyeul.park, mouloud.kharoune, arnaud.martin}@univ-rennes1.fr
http://www.iut-lannion.fr

Résumé. Dans ce papier, nous présentons une étude de cas de DBpédia en français. DBpédia a été considéré comme un concentrateur des données interconnectées. L'état actuel de DBpédia en français est encore à ses débuts, à la deuxième étape de développement selon notre critère . La première étape du projet a été accomplie, c'est-à-dire qu'il est identifié par les URIs et qu'il est décrit par les RDFs qui témoignent des données interconnectées ainsi que des ressources lisibles par les humains. La deuxième étape est à 44,65% de son avancement de l'appariement des ontologies au mois d'octobre 2013. Pour finir, étant donné l'état d'avancement actuel, nous proposons différentes démarches afin d'améliorer le développement du DBpédia en français.

1 Introduction

Le Web sémantique est un concept qui commence à prendre conscience de l'hétérogénéité des données dans le Web et de la difficulté de l'intégration et l'accessibilité entre celles-ci (Berners-Lee et al., 2001). L'objectif principal d'un tel système est de permettre aux machines d'interpréter sémantiquement les données du Web. Le principe est de faciliter le traitement et l'interprétation des informations issues des ressources du Web en les présentant sous des formes compréhensibles par une machine. Pour cela, le W3C a publié un langage d'ontologique pour le Web (OWL : Web Ontology Language) dans lequel on peut publier et partager une ontologie pour la construction et la gestion d'une base de connaissances sous une forme plus avancée (Harman, 2008). De plus, OWL est construit selon le modèle RDF (Resource Description Framework) qui est un format de données de référence pour définir la sémantique et les relations dans les ressources du Web. Dans ce contexte, DBpédia [1] peut être vue comme une source de données interconnectées (*Linked Data*) sous la forme d'une base de connaissances qui utilise une ontologie et le RDF. DBpédia est le résultat de tentatives d'extraction d'informations structurées à partir de Wikipédia, avec un identifiant unique (URI). Par exemple, si l'on accède à la ressource DBpédia sur `http://dbpedia.org/page/François_Hollande`, l'information structurée sous le format RDF est retournée. DBpédia est ainsi en train de jouer le rôle d'un concentrateur (*hub*) pour les données intercon-

1. `http://dbpedia.org`

nectées et est actuellement interconnecté avec d'autres bases de connaissances à grande échelle telles que YAGO2s [2], Freebase [3] etc.

Dans DBpédia, l'extraction de l'information est possible pour 119 langues de Wikipédia. DBpédia est accessible par SPARQL endpoint qui est un service acceptant des requêtes, comparables à SQL, pour interroger une base de connaissances en RDF. Nous constatons, que nous ne pouvons accéder par SPARQL endpoint de DBpédia qu'à seulement une partie parmi elles. Cela signifie que seul ces langues peuvent être utilisées via DBpédia depuis l'extérieur. Ceci est certainement lié à différentes raisons : d'une part, il est difficile de développer une version localisée de DBpédia pour les différentes langues. D'autre part DBpédia a été développé pour l'anglais y compris le cadre d'extraction d'information (DIEF) [4]. Le DIEF qui est un processus essentiel pour le développement de DBpédia, constitue un module d'extraction brute (dump) de Wikipédia. En effet, dans le cas de la langue grecque, des problèmes sont apparus dans le développement dès le début en raison de l'encodage de l'URI avec des caractères non latins mais également en raison de l'encodage des chaînes pour la navigation des ressources (Kontokostas et al., 2012).

DBpédia peut jouer le rôle d'un concentrateur des données interconnectées pour chaque langue, et il peut également produire des données dans les nuages *via* l'interconnexion avec d'autres données. Dans ce papier, nous analysons l'état actuel de DBpédia français en étudiant différentes étapes de progrès que nous définissons, pour proposer ensuite différentes démarches afin d'améliorer le développement de DBpédia français.

2 Données interconnectées

DBpédia a été introduit en 2007 par l'interaction entre l'extraction de données de Wikipédia et une meilleur gestion des bases de connaissances (Auer et al., 2007). Il est fondé sur le concept de données interconnectées proposé par Tim Berners-Lee. Au début, DBpédia a tenté d'extraire et de modifier les données structurées de Wikipédia dans le format RDF, en considérant le titre d'un article de Wikipédia comme une entité. YAGO, apparaissant simultanément à DBpédia, a utilisé la hiérarchie de synset de WordNet pour reconstruire une base de connaissances ontologiques comme un thésaurus avec une taxonomie qui permet de fournir les données les plus propres de Wikipédia (Suchanek et al., 2007, 2008).

D'autre part, une version antérieure de DBpédia a souffert de l'ambiguïté des entités et de l'incohérence des catégories de Wikipédia car elle a utilisé les données de Wikipédia sans modification. Ceci a diminué la fiabilité des données, ce qui a permis la découverte de références incomplètes sur d'autres données ouvertes interconnectées (LOD : Linked Open Data). Pour résoudre ces problèmes, DBpédia a défini l'ontologie de DBpédia en 2009 (Bizer et al., 2009). Ces efforts ont ainsi permis de clarifier les entités nommées et leurs propriétés grâce à l'appariement des ontologies en considérant que le modèle Wikipédia était une ontologie. Fondées sur les idées de DBpédia et YAGO, les ressources linguistiques peuvent aussi être utilisées pour améliorer les ressources les plus utiles du Web (Chiarcos et al., 2011).

Le projet des LOD linguistique (LLOD) a pour objectif principal l'ouverture de ressources linguistiques. Il promeut la recherche grâce au libre accès des ressources linguistiques, per-

2. http://www.mpi-inf.mpg.de/yago-naga/yago
3. http://www.freebase.com
4. https://github.com/dbpedia/extraction-framework/wiki

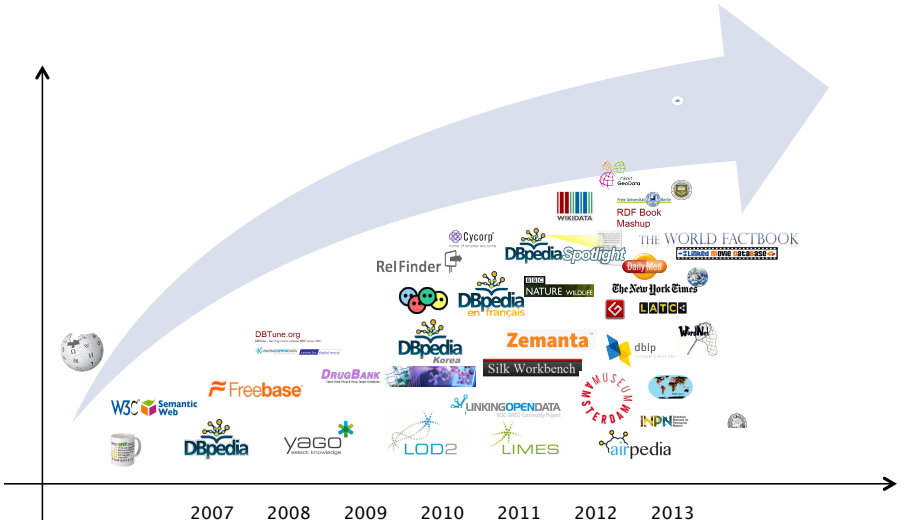


FIG. 1: DBpédia et les ressources interconnectées

mettant également d'interconnecter les autres ressources existantes. Pour réaliser cet objectif, les ressources linguistiques ont été converties en RDF. En particulier, dans les meilleures pratiques pour le Web sémantique le groupe de travail WordNet [5] a fait des efforts pour convertir WordNet [6] en RDF et plusieurs travaux se sont engagés à interconnecter les DBpédia avec la version localisée de WordNet pour chaque langue (van Assem et al., 2006; Huang et Zhou, 2007; Koide et al., en soumission; Lim et al., 2013). DBpédia a constamment effectué l'interconnexion entre ses ressources et les LLOD. Dans ce contexte, les travaux de découvertes de références qui continuent depuis 2011 tels que SILK (Volz et al., 2009) et LIMES (Ngomo et Auer, 2011), se focalisent sur l'interconnexion automatique des ressources.

Récemment les outils d'annotation qui utilisent les ressources et l'ontologie de DBpédia ont été réalisés (Mendes et al., 2011). De plus les méthodes de l'appariement des ontologies sont aussi abordées par (Aprosio et al., 2013a,b). Enfin, la dernière version de DBpédia a montré que beaucoup d'efforts ont été apportés pour améliorer la qualité des données *via* l'extension de l'ontologie (Lehmann et al., en soumission). Ainsi la FIG. 1 montre DBpédia et la croissance de ses ressources interconnectées [7].

5. 'Semantic Web Best Practices : WordNet Task Force', http://www.w3.org/2001/sw/BestPractices/WNET/tf
6. http://wordnet.princeton.edu
7. http://wiki.dbpedia.org/Interlinking

3 DBpédia en français

3.1 État actuel

DBpédia en français a commencé à mettre des données en ligne [8] depuis 2009 (la version 3.2). L'équipe française a donc achevé une extraction des données de Wikipédia en RDF pour le développement de DBpédia en français, qui permet à présent sa consultation sur le site de DBpédia. Depuis 2011, DBpédia en français a effectué l'appariement des ontologies entre DBpédia et Wikipédia. Depuis lors, il est achevé avec la mise en place le SPARQL endpoint et l'interconnexion vers DBpédia en anglais, avec des pages lisibles par les humains. DBpédia devient particulièrement approprié pour les machines et on le considère comme une base de connaissances interconnectées. Pourtant, ces données ne sont pas lisibles par les humains, ce qui est contraire au principe des connaissances interconnectées. C'est pour cela qu'il est important que DBpédia fournisse les informations utiles pour les humains.

Au mois d'octobre 2013, DBpédia en français montre un taux de 10% entre les classes de DBpédia et les modèles de Wikipédia, qui représente 44.65% pour les documents. Tandis que le taux d'appariement des ontologies en français est assez haut, et se place même à la 8ème position parmi les autres langues, le taux d'appariement des modèles y compris leurs occurrences est relativement bas [9]. Une comparaison avec DBpédia en italien dans le tableau 1 le montre bien. Wikipédia en français et en italien contiennent un nombre similaire de documents. DBpédia en français montre effectivement un taux assez haut dans l'appariement des ontologies (10,08%) et un nombre important d'appariements (195). Toutefois, le taux d'appariement des occurrences des modèles n'est que de 44,65%. En ce qui concerne DBpédia en italien, ces taux sont, respectivement, de 5,86% avec 55 appariements, dont 79,41% de documents appariés. Les utilisateurs de Wikipédia en français ont tendance à créer leurs propres modèles au lieu d'utiliser ceux qui sont déjà fournis. Le nombre de modèles dans Wikipédia en français est deux fois plus important que celui en italien (1 935 vs. 939). En outre les modèles les plus utilisés dans Wikipédia en français ne sont pas appariés, comme par exemple `Modèle:Autres projets`, qui est utilisé pour plus de 200 000 de documents [10].

	l'appariement des modèles	l'appariement des documents
français	10,08% (195 de 1 935)	44,65% (579 700 de 1 298 362)
italien	5,86% (55 de 939)	79,41% (905 259 de 1 139 956)

TAB. 1: Comparaison entre DBpédia en français et en italien : taux de l'appariement des modèles et des documents

8. `http://fr.dbpedia.org`
9. `http://mappings.dbpedia.org/server/statistics/fr`
10. `http://fr.wikipedia.org/wiki/Modèle:Autres_projets`

3.2 Interconnexion au-delà de DBpédia en français

Pour améliorer DBpédia en français afin de l'introduire dans l'étape de l'interconnexion au-delà de DBpédia, nous pourrions proposer les points suivants :

– l'addition de l'appariement des ontologies entre Wikipédia et DBpédia en français.
– l'interconnexion avec les autres ressources en *français*.

Dans Wikipédia en français, les quatre modèles parmi les plus utilisés tels que `Modèle:Autres projets`, `Modèle:Ouvrage`, `Modèle:ÉluDébut`, et `Modèle:Lien` ne sont pas encore appariés. Ces modèles pourraient aider à augmenter le taux jusqu'à 13% de documents de Wikipédia.

L'*interconnexion* est en effet un processus essentiel des données interconnectées. Les ressources décrites en français doivent être interconnectées par DBpédia en français pour devenir le vrai centre des données interconnectées. Le projet DBpédia en français et Sémanticpédia [11] pour Wikipédia francophone ont déjà commencé à constituer une avancée majeure pour la politique du ministère de la Culture et de la Communication en faveur de l'accessibilité aux données culturelles [12]. Ils offrent ainsi aux musées, aux bibliothèques et même aux opérateurs culturels, des perspectives de diffusion et de partage des ressources extraits de Wikipédia en français.

L'autre axe de l'interconnexion pour DBpédia en français à développer est celui des ressources linguistiques pour la langue française. Les ressources linguistiques se distinguent par rapport à leur importance pour la communauté du Web sémantique. La fourniture de ces ressources en RDF et en données interconnectées est aujourd'hui une pratique bien établie. Elle pourrait ouvrir une voie des données interconnectées linguistiques pour DBpédia en français. Nous citons d'abord EuroWordNet [13] et WOLF [14] pour WordNet en français. EuroWordNet est un système de réseaux sémantiques pour les langues européennes, fondé sur WordNet. Chaque langue développe son propre WordNet, et elles sont interconnectées avec des liens interlingue dans l'Interlingual Index (ILI). Le WOLF est aussi une ressource lexicale sémantique, et il a été construit à partir de diverses ressources multilingues (Sagot et Fišer, 2008). Le Le*fff* (Lexique des Formes Fléchies du Français) est un lexique morphologique et syntaxique à large couverture (Sagot, 2010) et il peut être couplé avec le WOLF [15]. Les données linguistiques du LADL [16] sont aussi une ressource non négligeable pour la langue française : le système DELA qui comporte un lexique de mots simples (DELAS), un lexique associé de transcriptions phonétiques (DELAP), et un lexique de noms composés (DELAC). Il contient également le lexique-grammaire des phrases élémentaires du français ainsi que des grammaires locales représentant des phrases dans des domaines spécifiques.

11. http://www.semanticpedia.org
12. http://www.culturecommunication.gouv.fr/Actualites/A-la-une/Lancement-de-DBpedia-et-de-Semanticpedia
13. http://www.illc.uva.nl/EuroWordNet
14. http://alpage.inria.fr/~sagot/wolf.html
15. http://alpage.inria.fr/~sagot/lefff.html
16. http://infolingu.univ-mlv.fr

4 Conclusion

Depuis la naissance des données interconnectées, un nombre important de personnes et d'organisations a adopté des données interconnectées comme un moyen de publier leurs propres données. Ils placent ces données interconnectées sur le Web, mais les utilisent aussi dans le Web [17]. Dans ce papier, nous nous sommes intéressés à DBpédia qui a été considéré comme un concentrateur des données interconnectées. Nous avons analysé l'état actuel de DBpédia en français. Il en résulte que pour DBpédia en français, il est important que les ressources décrites en français doivent être interconnectées. Pour cela, nous nous sommes concentrés sur les ressources culturelles et linguistiques. Mais il serait également intéressant d'interconnecter les données gouvernementales de la France qui existent depuis 2011 [18]. Pour améliorer DBpédia y compris en français, l'interconnexion est essentielle car elle pourrait permettre à DBpédia de se développer en tant que base de connaissances au-delà de Wikipédia.

Références

Aprosio, A. P., C. Giuliano, et A. Lavelli (2013a). Automatic Expansion of DBpedia Exploiting Wikipedia Cross-Language Information. *The Semantic Web : Semantics and Big Data, Lecture Notes in Computer Science 7882*, 397–411.

Aprosio, A. P., C. Giuliano, et A. Lavelli (2013b). Automatic Mapping of Wikipedia Templates for Fast Deployment of Localized DBpedia Datasets. In *Proceedings of (i-KNOW 2013)*, Graz, Austria.

Auer, S., C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, et Z. Ives (2007). DBpedia : A Nucleus for a Web of Open Data. *The Semantic Web, Lecture Notes in Computer Science 4825*, 722–735.

Berners-Lee, T., J. Hendler, et O. Lassila (2001). The Semantic Web. *Scientific American 284*(5), 28–37.

Bizer, C., J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, et S. Hellmann (2009). DBpedia - A Crystallization Point for the Web of Data. *Web Semantics : Science, Services and Agents on the World Wide Web 7*(3), 154–165.

Chiarcos, C., S. Hellmann, et S. Nordhoff (2011). Towards a Linguistic Linked Open Data cloud : The Open LinguisticsWorking Group. *Traitement Automatique des Langues (TAL) 52*(3), 245–275.

Harman, G. (2008). DeLanda's Ontology : Assemblage and Realism. *Continental Philosophy Review 41*(3), 367–383.

Huang, X.-x. et C.-l. Zhou (2007). An OWL-based WordNet lexical ontology. *Journal of Zhejiang University SCIENCE A 8*(6), 864–870.

Koide, S., H. Takeda, F. Kato, I. Ohmukai, F. Bond, H. Isahara, et T. Kuribayashi (en soumission). DBpedia and Wordnet in Japanese. *Semantic Web Journal*.

17. http://www.w3.org/2001/tag/doc/selfDescribingDocuments.html
18. http://www.data.gouv.fr

Kontokostas, D., C. Bratsas, S. Auer, S. Hellmann, I. Antoniou, et G. Metakides (2012). Internationalization of linked data. the case of the greek dbpedia edition. *Web Semantics : Science, Services and Agents on the World Wide Web 15*, 51–61.

Lehmann, J., R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, et C. Bizer (en soumission). DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web - Interoperability, Usability, Applicability*.

Lim, K., Y. Hahm, M. Rezk, J. Park, Y. Yongun, et K.-S. Choi (2013). Enrichment of DBpedia by Linking Korean WordNet and Improving Web Resource Accessibility. *Journal of KIISE : Computing Practices and Letters 19*(9), 474–478.

Mendes, P. N., M. Jakob, A. García-Silva, et C. Bizer (2011). DBpedia Spotlight : Shedding Light on the Web of Documents. In *Proceedings of the 7th International Conference on Semantic Systems*, I-Semantics '11, New York, NY, USA, pp. 1–8. ACM.

Ngomo, A.-C. N. et S. Auer (2011). LIMES - A Time-Efficient Approach for Large-Scale Link Discovery on the Web of Data. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, Barcelona, Catalonia, pp. 2312–2317.

Sagot, B. (2010). The Le*fff*, a Freely Available and Large-coverage Morphological and Syntactic Lexicon for French. In N. C. C. Chair), K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, et D. Tapias (Eds.), *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Sagot, B. et D. Fišer (2008). Building a Free French WordNet from Multilingual Resources. In *Proceedings of OntoLex 2008*, Marrakech, Morocco.

Suchanek, F. M., G. Kasneci, et G. Weikum (2007). YAGO : A Core of Semantic Knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, New York, NY, USA, pp. 697–706. ACM.

Suchanek, F. M., G. Kasneci, et G. Weikum (2008). YAGO : A Large Ontology from Wikipedia and WordNet. *Web Semantics : Science, Services and Agents on the World Wide Web 6*(3), 203–217.

van Assem, M., A. Gangemi, et G. Schreiber (2006). Conversion of WordNet to a standard RDF/OWL representation. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy, pp. 237–242.

Volz, J., C. Bizer, M. Gaedke, et G. Kobilaro (2009). Silk – A Link Discovery Framework for the Web of Data. In *Proceedings of WWW2009 workshop : Linked Data on the Web (LDOW2009)*, Madrid, Spain.

Summary

We present a case study on French DBpedia based on its progress on Linked Data. DBpedia has been considered as the hub for Linked Data. We analyze the current state of French DBpedia where it can be uniquely identified by URIs and be described in RDF, which bespeaks a web of data as well as human readable resources. It has also achieved 44.6% of all template

occurrences on October, 2013 for ontology mapping. Based on the current state of progress, we suggest some guidelines of how to improve French DBpedia.

Vers un système collectif et distribué pour la classification consensuelle de données

Rabah Mazouzi*, Lynda Seddiki*
Cyril De Runz**, Herman Akdag*

*LIASD, Université Paris 8, 2 rue de la Liberté - 93526 Saint-Denis cedex
rabah@ai.univ-paris8.fr, lynda.seddiki@ai.univ-paris8.fr, Herman.Akdag@ai.univ-paris8.fr
http://www.ai.univ-paris8.fr/
**CReSTIC, IUT de Reims, Chemin des Rouliers CS30012 51687 REIMS CEDEX 2
cyril.de-runz@univ-reims.fr
http://crestic.univ-reims.fr/

Résumé. De nos jours, au regard des très grands volumes de données mis en jeu, il est devenu nécessaire de recourir à la distribution des données et des traitements associés avec pour objectifs une meilleure qualité des résultats et la réduction du temps de calcul. Dans cet article, nous proposons une nouvelle approche pour la classification supervisée et distribuée de données. Basée sur le paradigme multi-agents, des agents sont répartis dans un réseau de noeuds, où chaque agent construit son classifieur avec ses propres données d'apprentissage. A l'issue de la classification, chaque agent combine son résultat avec ceux de ses voisins en prenant en compte les facteurs de performance. Ainsi, une dynamique se crée au sein du système permettant l'émergence d'un résultat collectif et consensuel. Enfin, nous présentons un modèle pour l'implémentation et l'expérimentation de notre approche dans un cadre de Cloud Computing.

1 Introduction

Parmi les aspects qui ont marqué récemment le domaine de la classification automatique de données est la banalisation des ressources mises à disposition pour mettre en place des solutions distribuées, autre fois très coûteuse et inaccessibles. Aujourd'hui, le développement du Cloud Computing a grandement facilité la construction de systèmes répartis, supportant des solutions distribuées et collaboratives.

La classification distribuée ou plus généralement la fouille de données distribuée ou DDM (*Distributed Data Mining*) ne se limite pas seulement aux faits de réaliser des gains en temps d'exécution, mais ouvre aussi des horizons en matière d'amélioration de la précision de calcul, de la scalabilité, et de la capacité à traiter des données très volumineuses (*Big Data*). En effet, de nombreuses recherches montrent que l'approche collective d'un système de classification améliore la qualité des résultats (Dietterich (2000); Zouari (2004)). Cette approche trouve son implémentation idéale dans l'architecture totalement distribuée, sans entité centrale et sans hiérarchisation (à la manière des réseaux P2P (Peer to Peer), où un noeud joue à la fois le rôle

du client et celui du serveur). Dans cet article, nous essayons de mettre en exergue certains bénéfices de l'utilisation d'une telle approche, notamment pour améliorer la classification de données en précision et en scalabilité.

Plusieurs approches basées Cloud Computing et/ou multi-agent ont récemment été utilisées dans diverses domaines, où l'émergence d'une décision collective au sein du système conduit à la pertinence des résultats globaux. On les trouve notamment dans le cas où le système est naturellement réparti, tel qu'en sécurité des réseaux, où des systèmes de détection d'intrusion distribués sont proposés Zhou et al. (2010). C'est dans ce contexte que nous plaçons notre démarche.

En effet, nous proposons d'utiliser la distribution des données et des traitements afin de réaliser un gain considérable en temps de calcul et de ressources utilisées. Nous souhaitons ainsi tendre vers le traitement de très grands volumes de données (*Big Data*). Pour ce faire, nous faisons recours aux techniques du Cloud computing Gillick et al. (2006) dont l'objectif est de montrer l'impact de l'utilisation des classifieurs massifs sur la qualité des résultats produits par un système Multi-Classifieurs. En effet, ce patron de conception a connu un grand succès, et est largement utilisé comme support de mise en oeuvre pour la distribution de traitement et de données (Gillick et al. (2006)).

Le travail que nous présentons touche à trois axes différents :

- Le premier est relatif à l'amélioration des résultats par l'utilisation d'un système multi-classfieurs large-échelle. Etant donné que la plupart des travaux qui traitent la combinaison des classifeurs se limitent, en moyenne, à une combinaison de trois ou quatre classifieurs (Zouari (2004)), en vue de l'optimisation du temps de calcul. Nous étudierons l'impact de l'utilisation de classifieurs massifs sur la précision des résultats.
- Le deuxième axe consiste à débattre des données d'apprentissage produites par les systèmes informatiques, qui sont de plus en plus volumineuses. Utiliser de telles données volumineuses avec un nombre réduit de classfieurs conduit au sur-apprentissage, ce qui résulte dans l'incapacité des classifeurs obtenus à préserver leur capacité de généralisation (Wann et al. (1990)).
- Le dernier axe est relatif au temps de calcul, qui devrait être considérablement réduit avec l'exécution parallèle dans le cadre du Cloud Computing.

La suite de l'article est organisée comme suit : La section 2 présente des travaux connexes en classification distribuée, et collective de données. Certains travaux utilisant la même philosophie, et qui sont relatifs à la fouille de données sont également présentés. Ensuite, dans la section 3, nous décrivons notre système de classification distribuée et consensuelle. Enfin, nous exposons, dans la section 4, une spécification technique de mise en oeuvre de notre système et des brèves présentations des environnements de développement qui peuvent être utilisés pour son implémentation.

2 Travaux connexes

Nous présentons ici, quelques travaux ayant traité le problème de l'apprentissage distribué, que ce soit pour la classification ou pour le clustering de données. On s'intéresse principalement à ceux qui visent à améliorer la précisons de la classification globale obtenue à partir de multiples classifieurs locaux, entraînés individuellement.

Ping Luo et al. ont proposé une approche collective pour la classification distribuée de données dans un système P2P (Luo et al. (2007)). Selon leur approche, chaque paire construit ses propres classifieurs en utilisant des données locales, et en exécutant l'algorithme d'apprentissage « Pasting bites ». Ensuite, tous les résultats sont combinés en utilisant la technique du vote majoritaire. Il s'agit d'un protocole de vote distribué, basé sur l'échange de messages entre les paires du réseau. Le modèle de distribution proposé dans ce travail ne peut être envisagé dans le cas d'un réseau large échelle, étant donné que dans ce genre de réseaux, le vote majoritaire de tous les paires ne peut pas être envisagé.

Une version distribuée de l'algorithme de clustering K-means, dans un environnement P2P est proposée par Souptik Datta et al. (2009). L'algorithme ne nécessite que l'échange d'information locale. Selon les auteurs, il s'agit du premier algorithme du K-means qui pourra être appliqué dans le cas d'un réseau large-échelle. Chaque noeud du réseau calcule les centroïdes des clusters, et les échange avec ses voisins. Chaque voisin recalcule ses centroïdes, en utilisant ses données locales, et les centroïdes obtenus de ses voisins. Le majeur problème qui résulte de l'approche proposée par les auteurs est la synchronisation des noeuds voisins. Aussi, la dynamique qui permet l'émergence d'un clustering final n'est pas bien explicitée.

En terme de distribution de données volumineuses d'apprentissage sur un réseau de noeuds, plusieurs approches ont été proposées dans la littérature (Moretti et al. (2008)). On distingue quatre méthodes possibles de mise en oeuvre de la distribution des données d'apprentissage, et ce en considérant l'emplacement de ces données sur les noeuds du système :

- La méthode "Streaming" : s'applique au cas de sources de données réparties, où la fonction de partitionnement relie simplement chaque source à un classifieur dans le système via un flux, telle qu'une connexion TCP.
- La méthode "Pull" : la fonction de partitionnement lit les données d'apprentissage à partir d'un noeud et écrit les partitions sur ce même noeud. Chacun des classifieurs des autres noeuds importe une partition.
- La méthode "Push" : la fonction de partitionnement lit la donnée d'un noeud et écrit les partitions directement sur les noeuds distants, où les classifieurs lisent leurs copies en local.
- La méthode "Hybride" : la fonction de partitionnement choisit un ensemble réduit de noeuds intermédiaires rapides, fiables, et d'une capacité suffisante pour écrire les données partitionnées. Lors de l'exécution, chaque noeud lit sa partition à partir de ces noeuds.

3 Une approche distribuée et consensuelle pour la classification

Contrairement aux systèmes multi-classifieurs à base d'une entité centrale, comme l'approche parallèle de la combinaison de classifieurs (Zouari (2004)), qui nécessite l'existence d'une entité centrale de combinaison pour effectuer le traitement de fusion ou sélection, le système décrit dans cet article utilise des agents autonomes repartis sur un réseau de noeuds. Chaque noeud contient un agent qui représente un classifieur élémentaire. Ce dernier reçoit une portion des données d'apprentissage, et l'utilise pour son propre entraînement. Ces portions proviennent des ensembles volumineux de données, ou bien des sources distribuées et

hétérogènes. Contrairement, à certains travaux ayant utilisé des méthodes à base d'agents telles que celle décrite dans (Saidane et al. (2005)), où tous les agents sont mis en interaction, dans notre cas, seul les agents qui sont voisins interagissent entre eux, par l'envoi asynchrone de messages. Ceci permet la construction de systèmes de classification large échelle, avec dynamique complexe. L'autonomie de l'agent, dans le sens où il n'est soumis à aucun contrôle externe, sauf à l'influence de ses voisins et de son environnement, permis d'avoir des systèmes à dynamiques complexes assurant à la fois robustesse et émergence de fonctionnalités globales.

Nous partons de l'hypothèse que, dans les systèmes complexes, les processus locaux doivent être simples et réactifs, et que c'est l'interaction et la dynamique du système qui permet l'émergence d'un résultat global qui est dans notre cas consensuel et meilleur que tous les résultats locaux au niveau des entités distribuées.

Les données à classifier seront mises sur un noeud du réseau, où ce dernier les diffuse à ses noeuds voisins. Au fur et à mesure que ces données sont classifiées par les noeuds auxquelles elles aboutissent, elles se propagent pour atteindre tous les noeuds du réseau.

Dans la suite de la section, nous montrons les différents aspects qui relèvent de notre système, notamment l'apprentissage des classifieurs locaux, et comment la classification distribuée et consensuelle est opérée.

3.1 Apprentissage

Chaque agent procède à l'apprentissage des classifieurs qui sont à sa disposition par un sous-ensemble d'apprentissage qui lui est propre. Plusieurs méthodes peuvent être envisagées pour le partitionnement des données d'apprentissage lorsque ces dernières sont centralisées sur un site particulier. Moretti et al. Moretti et al. (2008) ont cité quelques unes de ses méthodes. Nous adoptons pour notre système celle dite Shuffle, étant donné qu'elle assure à la fois un partitionnement aléatoire et équilibré. Cette tâche s'incombe à l'agent et pour l'accomplir, il n'interagit avec aucun agent de son voisinage, sauf pour le choix de la nature des classifieurs à considérer, et ce pour garantir une diversité de classifieurs dans le système. Chaque agent évalue la précision de ses classifieurs en utilisant un jeu de test, et mémorise ceci à son niveau sous formes de facteurs de performance. Dans le cas de données massives, le jeu d'apprentissage par classifeur doit être réduit au juste nécessaire pour éviter le sur- apprentissage et ne pas perdre la capacité de généralisation.

En ce qui concerne la distribution de données d'apprentissage, la méthode "Hybride" proposée dans Moretti et al. (2008) est la plus appropriée pour accomplir la distribution dans le cas de systèmes large-échelle. En effet, dans de tels systèmes, les données d'apprentissage sont généralement issues de sources multiples, il reste donc à apparier les noeuds de classification à ces sources de données, et ce en utilisant la technologie du Cloud-computing.

3.2 Sélection des classifieurs

A l'issue de l'étape d'apprentissage, l'agent procède à la sélection d'un classifieur qui participe à la classification distribuée, parmi ceux entrainés au niveau du noeud. La sélection se base sur un facteur de performance, relatif à chaque classifieur. Afin d'assurer la diversité des classifieurs au sein du système, les agents peuvent communiquer au sein de leurs voisinages respectifs, dans le but d'utiliser des classifieurs différents (Kuncheva et Whitaker (2003)). Par

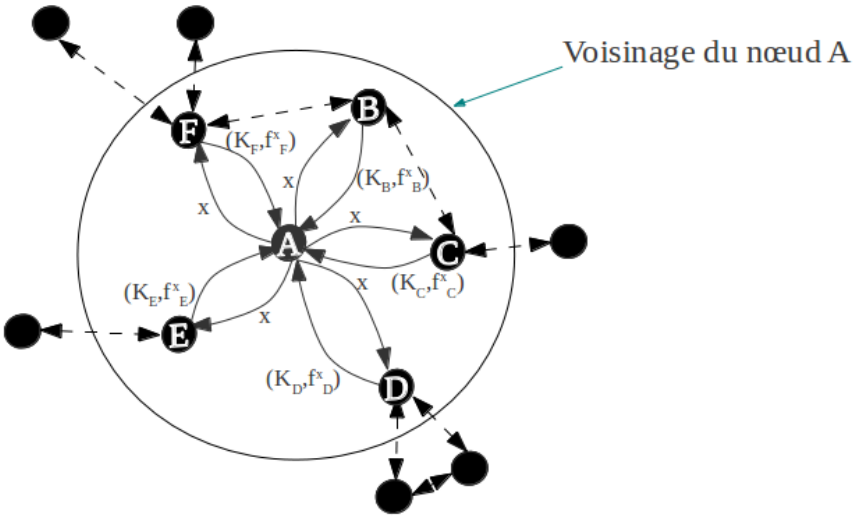


FIG. 1 – *Exemple d'interaction locale au sein du système*

ailleurs, la diversité est assurée en premier lieu par la diversité des sous-ensemble d'apprentissage, car des sous-ensembles d'apprentissage différents nécessitent des classifieurs différents en nature et en structure. Les techniques les plus utilisées pour cela sont essentiellement le bagging et le boosting (Palit et Reddy (2012); Breiman (1996)).

3.3 Classification

Afin d'expliciter le principe de la classification distribuée et consensuelle, proposée dans cet article, nous montrons un scénario d'interaction entre les classifieurs d'un voisinage pour la détermination collective d'une classe (Fig. 1). Soit un noeud A au centre, entouré d'un ensemble V de noeuds voisins : $V = \{B, C, D, E, F\}$. Le classifieur au niveau du noeud A calcule la classe de l'attribut en question (x), et il reçoit les choix ($f_B^x, f_C^x, f_D^x, f_E^x, f_F^x$) de tous ses voisins avec leurs facteurs de performances(K_i) ; $i < |V|$. Ensuite, il essaie de remettre en question son propre résultat de classification en fonction de la classe choisie pour l'attribut x, et les classes choisies par ses voisins (f_i^x) dans une logique qui prend en compte les indicateurs de performance de chaque classifieur : K_i. Le vote pondéré peut être utilisé (Kuncheva (2004)). De cette manière, dans un voisinage donné, tous les classifieurs participant à la classification se mettent d' accord (un consensus) sur un choix unique. Ce scénario se répète pour chaque noeud, et cela dans une dynamique permanente. Ainsi, par propagation et échange des résultats, d'un voisinage à un autre, l'émergence d'un résultat final est obtenue.

L'agent au niveau d'un noeud donné, calcule la classe en utilisant son propre classifieur ; et se met en attente jusqu'à la réception de toutes les classes calculées par ses voisins, ainsi que leurs facteurs de performance. A ce stade, il recalcule la classe, en utilisant ces dernières données, et retransmet le résultat à tous ses voisins.

3.4 Emergence du résultat

Selon le schéma décrit précédemment, un noeud influence le calcul au sein de ses voisins par son propre calcul. Cependant, il est influencé par le calcul des autres. La classe la plus probable est créée dans certains noeuds du réseau. Pour cette classe, les facteurs de performance, sont naturellement les plus hauts. Ces noeuds vont être plus influants que les autres, et la classe qui y calculée va se propager, et devient de plus en plus dominante dans le système, pour représenter à la fin la classe émergente.

Après un certain temps, la dynamique du système se stabilise et le même résultat est présent dans tous les noeuds du système. L'utilisateur récupère le résultat sur le noeud auquel il a envoyé ses données à classifier.

4 Eléments d'implémentation

Afin d'expliciter le processus au sein de chaque noeud, nous le présentons, ci-après, sous forme des pseudo-codes, en utilisant les notations suivantes :

- N_i : noeud i ;
- D_i : données d'apprentissage propres aux noeud i ;
- C_i : classifieur du noeud i ;
- K_i : facteur de performance du classifieur C_i ;
- A : un noeud ;
- x : un attribut à classifier ;
- f_A^x : classe attribuée par A à x ;

4.1 Apprentissage

Au début, on suppose l'existence d'une entité qui apparie les sous-ensembles d'apprentissage aux noeuds du réseau. Cette entité, indique à chaque noeud, quelles sont ses données d'apprentissage (en utilisant la méthode Hybride (Moretti et al. (2008))). Ceci peut être fait simplement en indiquent au noeud en question l'adresse de la machine (domaine), où ses données d'apprentissage sont stockées.

Algorithm 1 Method *Partition()*

for each node N_i **do**
  Assign the subset D_i to the node N_i
end for

La fonction d'apprentissage est exécutée au sein du noeud lui-même, en invoquant la méthode Learning de l'agent local. Chaque agent utilise le meilleur de ses classifieurs en considérant son propre sous ensemble d'apprentissage (resp. de test) D_i.

4.2 Classification

Le processus de classification, commence par la mise sur un noeud d'un attribut x dont on désire déterminer sa classe. L'attribut se propage dans le réseau par broad-casting local.

Algorithm 2 Method *Learning()*

for each classifier C_j within the node N_i **do**
  Learn the classifier C_j by using the subset D_i
  Save K_j
end for
Select C_i having $K_i = Max\{K_j\}$

Chaque noeud qui reçoit l'attribut, calcule sa classe, en utilisant son classifieur sélectionné, et transmet à son tour l'attribut à ses voisins.

En suite, chaque noeud, après réception de toutes les classes calculées par ses voisins, recalcule une nouvelle classe, en incluant sa propre classe, et en utilisant la méthode de vote pondéré. Dans ce cas, le facteur de performance est lui-même utilisé comme valeur de pondération. Le résultat obtenu est transmis aux noeuds voisins et le processus continu ainsi jusqu'à ce que la classe soit constante (classe émergente).

Algorithm 3 Method *A.InitialComputation()*

Compute F_A^x
for each node N_j neighbouring A **do**
  Send x to N_j
end for

Algorithm 4 Method *A.ContinuousComputation()*

Wait
Until Receiving all classes from neighbours $\{N_j\}$ of A
$f_x = Combine(F_A^x, \{f_j^x, K_j\})$
for each node N_j neighbooring A **do**
  Send f^x to N_j
end for

Il est à noter que notre approche est centrée sur le noeud, et aucune entité centrale n'est utilisée, sauf pour le cas de la distribution de données d'apprentissage qui se fait avant que le processus de classification ne commence. Aussi, cette entité se contente de l'assignation des sources de données aux noeuds contenant les classifieurs. Ce caractère de distribution totale, permet ainsi la construction de classifieurs massifs large échelle, qui ne peuvent pas être envisagés avec toute approche nécessitant une entité centrale.

5 Vers une mise en oeuvre d'une plate-forme pour classifieurs massifs

Pour mettre en oeuvre le système décrit dans cet article, un certain nombre d'outils doivent être utilisés : une plate-forme multi-agents (ex : Madkit, NetLogo, etc.), un environnement de distribution de traitements et de données sur un réseau de Cloud computing (ex : GridGain,

Hadoop, etc.), et une boite à outils qui contiendra l'implémentation des algorithmes de classification (Weka, etc.).

5.1 Eléments de mise en oeuvre

5.1.1 Le voisinage

Dans ce travail, la notion de voisinage est prépondérante. Gridgain dispose d'une API (Application Programming Interface) particulière appelée les services SPI (Service Provider Interface). Ces services sont utilisés pour la découverte des noeuds du réseau et les mesures de charges des liaisons entre les noeuds. Ceci permet entre autre à déterminer la disponibilité d'un noeud par rapport à un autre et ainsi on peut à partir d'un noeud trouver ses voisins.

5.1.2 La diversité

Elle est assurée en partie par les sous-ensembles d'apprentissage hétérogènes. Et pour renforcer encore cette diversité, les différents agents dans un même voisinage peuvent interagir après l'étape d'apprentissage et échanger les vecteurs d'erreurs engendrées au niveau de chaque classifieur. Le facteur de diversité dans ce cas est calculé par le principe d'intercorrélation entre les vecteurs d'erreurs.

5.2 Un premier pas

Dans un travail précédent, nous avons étudié la sélection dynamique de classifieurs et l'optimisation du temps de calcul par l'utilisation du Cloud computing comme support d'exécution (Mazouzi (2013)). Le projet se présente sous forme d'un prototype de combinaison parallèle de classifieurs ; développé en utilisant l'API Weka. Le projet consistait à utiliser deux modules : le premier pour la sélection, et le second pour la fusion. Ce dernier contenu dans une entité séparé qu'on a appelé combineur, qui abrite la logique de fusion des réponses des différents classifieurs sélectionnés pour la combinaison (vote et vote pondéré). Ce prototype en forme d'une application Cloud-computing, nous a permis de réaliser une exécution parallèle des classifeurs afin d'effectuer une sélection rapide à partir d'un ensemble vaste de classifeurs.

Les taux de bonne classification réalisés par chaque classifieur de l'ensemble sélectionné, en utilisant 3 sous ensembles de test étaient comme suit : {(trees.ADtree,79,68%), (trees.J48, 84,11%), (functions.RBFNetwork,74,34%)}. le résultat de la combinaison était : 86,34%. En considérant les résultats obtenus, on constate que le taux de bonne classification est nettement amélioré avec l'utilisation d'une méthode de combinaison parallèle par vote pondéré.

En terme de temps de calcul lors de l'apprentissage, nous avons utilisé le module de sélection afin d'entraîner un ensemble de 10 classifieurs, ensuite sélectionner un sous ensemble des meilleures parmi eux. Les données utilisées (diabets.arff) sont fournies avec le framework Weka. Dans un premier temps, le module de sélection est exécuté sans le modèle de grille (exécution en local). Dans un deuxième temps, nous avons utilisé le support de Cloud computing (exécution sur plusieurs noeuds). Chaque classifieur s'exécute sur un noeud (10 classifieurs / 10 noeuds).

Les temps d'exécution de chaque classifieurs (en local), est donné dans le tableau suivant :

ZeroR	JRip	NaiveBayes	BayesNet	J48	ADTree	LibSVM	RBFNetwork	MISVM	SimpleMI
1,5	2,5	1,5	1	3	1	NC	3	NC	NC

Avec une exécution séquentielle, le temps total est 11,5 seconde. Avec l'exécution parallèle en utilisant plusieurs noeuds, le temps d'exécution est 3,5 secondes. Ainsi, par l'utlisation d'une solution de Cloud Computing, nous observons un important gain de temps d'exécution.

Pour notre cas, nous réutiliserons le module de sélection afin d'effectuer la distribution de calcul et des données d'apprentissage. Et comme notre système ne suppose pas l'existence d'une entité autre que les agents autonomes, nous intégrons la logique de fusion dans chaque noeud. Aussi, bien que notre approche est pour l'instant essentiellement formelle, les résultats préliminaires obtenus sont plus qu'encourageant.

6 Conclusion

Dans cet article, nous avons proposé une nouvelle approche de classification distribuée et consensuelle de données. Contrairement à la majorité des approches qui existent dans la littérature, elle se caractérise par son aspect totalement réparti, étant donné qu'aucune entité centrale n'est envisagée dans le système. La classification, se fait d'une manière collective et consensuelle, au sein de chaque voisinage du système. Enfin la dynamique globale du système permet l'émergence de la classe la plus pondérante, en considérant les résultats locaux des classifieurs, et leurs facteurs de performance. Il s'agit dans notre cas, d'un système complexe pour la classification distribuée. Au sein de tels systèmes, l'étude des différentes interactions entre les classifieurs permet l'émergence de nouveaux algorithmes issus de la combinaison automatique de classifieurs collaborants.

Dans l'immédiat, nous implémentons notre approche, en utilisant un système multi-agent, en faisant recours à une plate-forme dédiée, ainsi qu'aux différents outils de mise en oeuvre tels que le cloud-comuting, et Weka.

Références

Breiman, L. (1996). Bagging predictors. *Machine Learning 24*(2), 123–140.

Datta, S., C. Giannella, et H. Kargupta (2009). Approximate distributed k-means clustering over a peer-to-peer network. *IEEE Trans. Knowl. Data Eng. 21*(10), 1372–1388.

Dietterich, T. G. (2000). Ensemble Methods in Machine Learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, MCS '00, London, UK, UK, pp. 1–15. Springer-Verlag.

Gillick, D., A. Faria, et J. Denero (2006). Mapreduce : Distributed computing for machine learning.

Kuncheva, L. I. (2004). *Combining Pattern Classifiers : Methods and Algorithms*. Wiley-Interscience.

Kuncheva, L. I. et C. J. Whitaker (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Mach. Learn. 51*(2), 181–207.

Luo, P., H. Xiong, K. Lü, et Z. Shi (2007). Distributed classification in peer-to-peer networks. In *KDD*, pp. 968–976.

Mazouzi, R. (2013). *Combinaison de Classifieurs :Prototype d'un Framework de combinaison basé sur l'API Weka*. Rapport de master, Université Paris 8.

Moretti, C., K. Steinhaeuser, D. Thain, et N. V. Chawla (2008). Scaling up classifiers to cloud computers. In *ICDM*, pp. 472–481. IEEE Computer Society.

Palit, I. et C. K. Reddy (2012). Scalable and parallel boosting with mapreduce. *IEEE Trans. Knowl. Data Eng. 24*(10), 1904–1916.

Saidane, A., H. Akdag, et I. Truck (2005). Une approche SMA de l'Agrégation et de la Coopération des Classifieurs. In *Conférence Internationale SETIT'2005 Sciences Electroniques, Technologiques de l'Information et des Télécommunication*, pp. 126–126.

Wann, M., T. Hediger, et N. Greenbaum (1990). The influence of training sets on generalization in feed-forward neural networks. In *Neural Networks, 1990., 1990 IJCNN International Joint Conference on*, pp. 137–142 vol.3.

Zhou, C., C. Leckie, et S. Karunasekera (2010). A survey of coordinated attacks and collaborative intrusion detection. *Computers - Security 29*(1), 124–140.

Zouari, H. (2004). *Contribution a l'évaluation des méthodes de combinaison parallèle de classifieurs par simulation*. Thèse de doctorat, Université de Rouen.

Summary

Nowadays, with regard to the very large volumes of data involved, it has become necessary to resort to the distribution of data and associated treatments for obtaining a better quality of results and the computation time reduction. We propose a new approach for the supervised and distributed classification of data. Based on the multi-agent paradigm, agents are deployed on a network of nodes, where each agent built its classifier with its own training data. At the end of the classification, each agent combines its result with those of its neighbors by taking into account the factors of performance. Thus, a dynamic is created within the system allowing the emergence of a collective and consensual outcome. Finally, we present a model for the implementation and testing of our approach within a framework of Cloud Computing.

Algorithme Hybride de Sélection d'attributs pour le Classement des protéines

Faouzi Mhamdi*, Mehdi Kchouk*

*Laboratoire de Recherche en Technologies de l'Information et
de la Communication & Génie Electrique
ENSIT, Université de Tunis, Tunisie
Faouzi.mhamdi@ensi.rnu.tn, mehdi.kchouk@gmail.com

Résumé. Pour faire le classement des protéines nous appliquons le processus d'Extraction de Connaissances à partir des Données (ECD). Dans la phase de prétraitement de ce processus on fait l'extraction d'attributs et le codage de données pour construire le tableau de données qui sera l'entrée des techniques de fouille de données. Ces techniques sont appliquées pour faire de la classification, du regroupement, etc. Vue la grande masse des données biologiques, le nombre d'attributs extraits est énorme. Ce qui influe sur la qualité du tableau de données et par conséquence sur les résultats des algorithmes d'apprentissage. D'où la nécessité de passer par une phase de *sélection d'attributs* afin d'éliminer les attributs non pertinents et/ou redondants. L'élimination de ces attributs permet d'améliorer les performances des algorithmes d'apprentissage en taux d'erreurs et en temps de calcul. Dans ce papier, nous présentons un algorithme hybride de sélection d'attributs. Les expérimentations sur des données biologiques réelles extraites à partir des banques de données biologiques ont donné des bons résultats.

1 Introduction

Les données biologiques sont caractérisées par leurs grandes masses et leurs complexités. Dans cet article on s'intéresse au classement de protéines, en utilisant les techniques de fouille de données (SVM). Pour cela on doit construire un tableau de données à partir des données brutes (séquences de protéines). Pour réaliser ce travail, nous avons appliqués le processus de l'Extraction de Connaissance à partir des Données (ECD, KDD en anglais Ű *Knowledge Descovery and Datamining*) (Fayyad et al., 1996). Ce processus se compose de trois principales phases : (*i*) *prétraitement de données*, (*ii*) *apprentissage ou fouille de données*, (*iii*) *post-traitement de données* (Frawley et al., 1992; Fayyad et al., 1996; Piatetsky-Shapiro et al., 2006).

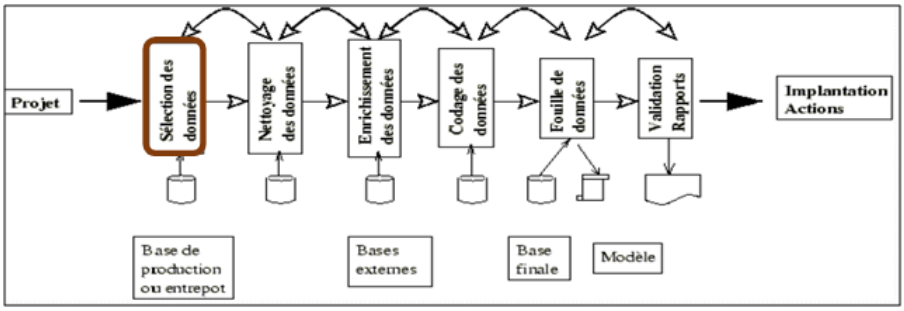


FIG. 1 – *Les phases du processus de l'ECD*

Dans ce travail, on s'intéresse à la phase de prétraitement : D'abord on fait l'extraction d'attributs à partir des séquences de protéines. Les attributs extraits sont des *n*-grammes. Ensuite on construit le tableau de données on utilisant la pondération booléenne. Ce tableau est donné en entrée des techniques de fouille de données pour faire la classification supervisée.

FIG. 2 – *Fichier brute de familles de protéines*

Dans le cas où le nombre d'attributs est trop élevé, les performances des techniques de fouille de données se dégradent le temps d'exécution augmente d'une façon considérable. Dans ce cas le passage par une phase de sélection d'attributs devient une nécessité afin de sélectionner un sous-ensemble d'attributs qui ne contient pas les attributs non pertinents et redondants. En effet, la sélection a pour but : la réduction de la quantité des données, l'amélioration de la qualité des données en se concentrant sur les d'attributs pertinents et augmenter les performances de l'algorithme d'apprentissage. Spécifiquement, la sélection d'attributs consiste à rechercher un sous-ensemble d'attributs afin d'avoir des motifs pertinents et discriminants. En effet, on peut définir la sélection d'attributs comme un processus qui choisit un sous-ensemble minimal de M attributs à partir de l'ensemble original de N attributs, ($M \leq N$), de telle sorte que l'espace des attributs est réduit d'une manière optimale respectant certains critères d'évaluation.

En général, les algorithmes de sélection d'attributs se composent de quatre étapes :*i*) génération d'un sous-ensemble, *ii*) évaluation du sous-ensemble, *iii*) critère d'arrêt *iv*) validation des résultats. Le processus de sélection d'attributs peut s'arrêter selon un des critères suivants : *a*) prédéfinir le nombre d'attributs à sélectionner, *b*) prédéfinir le nombre d'itérations, *c*) lorsque l'ajout/suppression d'un attribut ne peut pas produire un meilleur sous-ensemble, *d*) obtenir un sous-ensemble optimal vérifiant le critère d'évaluation (Sauwens, 2010). Les étapes de génération et d'évaluation de sous-ensemble d'attributs, décrites dans la figure 3, sont itérées jusqu'à ce que le sous-ensemble d'attributs généré soit le meilleur.

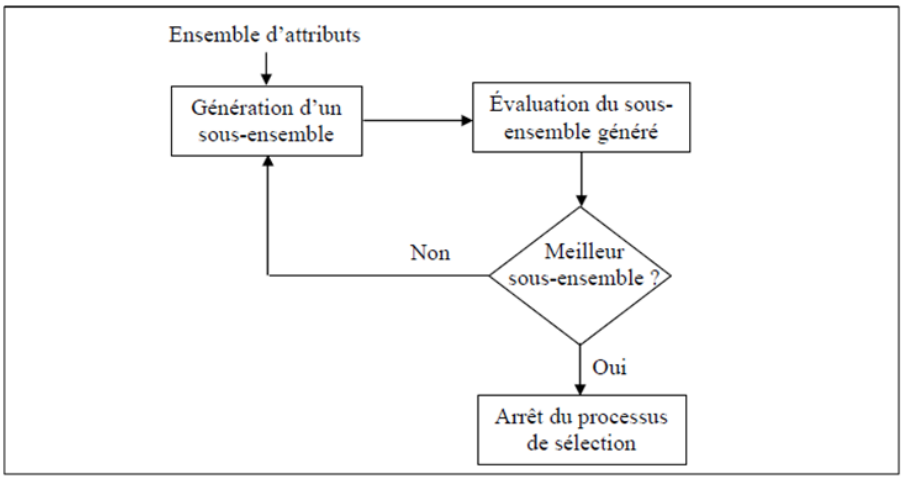


FIG. 3 – *Processus général de sélection d'attributs*

Ils existent trois types d'algorithmes de sélection d'attributs :

– *Les algorithmes filtrants* : Utilisent des critères statistiques pour calculer la pertinence d'une variable avant le processus d'apprentissage. L'avantage de ces algorithmes c'est qu'ils sont rapides.
– *Les algorithmes enveloppants (wrapper)* : Se servent d'un algorithme d'apprentissage pour évaluer le sous-ensemble d'attributs sélectionnés. L'avantage de ces algorithmes c'est que l'ensemble trouvé est optimal pour le classifieur étudié et ils donnent de meilleurs résultats par rapport aux algorithmes filtrants, leur inconvénient est qu'ils sont moins rapides.
– *Les algorithmes hybrides* : Réunissent l'avantage des algorithmes filtrants en ce qui concerne la rapidité, à celui des algorithmes enveloppants, *i.e.*, meilleurs résultats. En adaptant un algorithme hybride, nous opérons en deux étapes : Durant la première étape, nous utilisons un algorithme filtrant pour réduire la taille de l'ensemble d'attributs. Puis, durant la seconde étape, nous utilisons un algorithme enveloppant pour sélectionner le meilleur sous-ensemble d'attributs à partir de l'ensemble réduit d'attributs.

Notre travail consiste à développer un algorithme hybride de sélection de motifs biologiques pour faire la classification des séquences biologiques. Il se compose de deux étapes : étape de filtrage et une étape enveloppante.

Ce papier est organisé comme suit : dans la deuxième partie nous présenterons l'algorithme d'extraction d'attributs et la méthode de pondération. Nous présenterons notre algorithme dans la troisième section. Dans la quatrième section nous présenterons les expérimentations réalisées sur des familles de protéines, enfin nous clôturons notre article par une conclusion.

2 Extraction et pondération d'attributs

2.1 Extraction d'attributs

Ils existent plusieurs algorithmes pour extraire les attributs utilisés dans la création du tableau de données (Sauwens, 2010). Dans notre travail, nous avons utilisé un algorithme d'extraction d'attributs adoptant une démarche descendante (Mhamdi et al., 2013). Cette dernière, consiste à construire des attributs de taille variable en se basant sur la méthode *n*-grammes. D'une manière générale, hiérarchiquement, on extrait les (*n*-*i*)-grammes à partir des *n*-grammes tant que $n\text{-}i \leq 2$. Nous utilisons le terme descendant car nous commençons par l'extraction des motifs (attributs) de taille *n* (avec *n* donné par l'utilisateur), puis comme deuxième étape, nous extrairons les motifs de taille *n*-1 et nous répétons la procédure jusqu'à *n*=2. Par exemple : si *n*=5 l'algorithme commence à extraire les 5-grammes, les 4-grammes, les 3-grammes et enfin les 2-grammes.

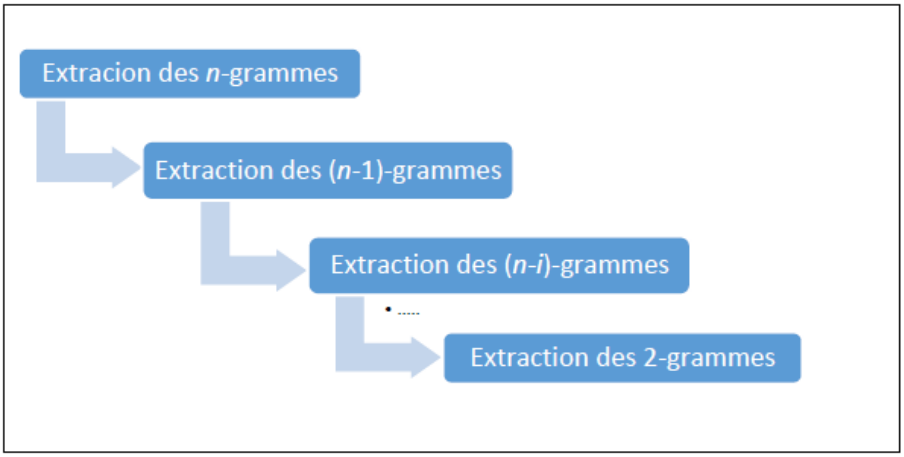


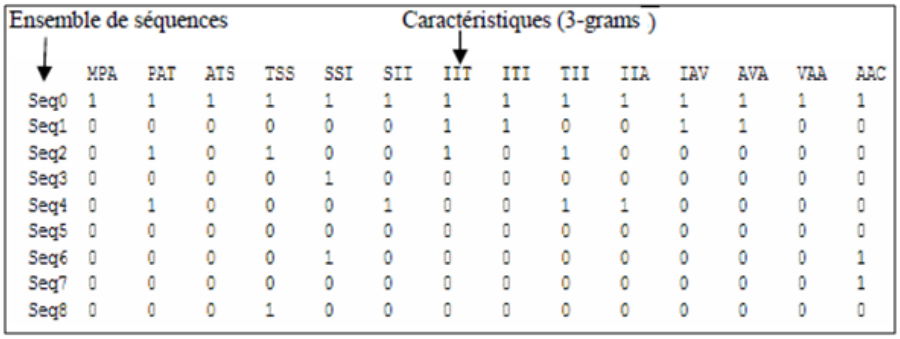
FIG. 4 – *Processus descendant d'extraction d'attributs*

2.2 Pondérations des attributs

Les technique de fouille de données un tableau de données T (Individus X Attributs) pour faire l'apprentissage. Les individus représentent les séquences biologiques et les attributs représentent les *n*-grammes. Chaque case T[*i*][*j*] représente le poids d'un *n*-gramme *j* dans une séquence *i*. Dans la littérature, nous distinguons plusieurs types de pondérations dont nous

pouvons citer la pondération booléenne, la pondération par occurrence, la pondération par fréquence et la pondération TF-IDF (Mhamdi et al., 2006).

Dans le cas de notre travail, nous avons opté pour la pondération booléenne suite à des travaux de recherches réalisés dans (Mhamdi et al., 2004, 2006). Nous avons effectué une étude comparative et nous avons constaté que la méthode de la pondération booléenne donne de meilleurs résultats. D'où notre tableau de données sera un tableau booléen, T[i][j] = 1 si le n-gramme j appartient à la séquence i et T[i][j] = 0 sinon.



	MPA	PAT	ATS	TSS	SSI	SII	IIT	ITI	TII	IIA	IAV	AVA	VAA	AAC
Seq0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Seq1	0	0	0	0	0	0	1	1	0	0	1	1	0	0
Seq2	0	1	0	1	0	0	1	0	1	0	0	0	0	0
Seq3	0	0	0	0	1	0	0	0	0	0	0	0	0	0
Seq4	0	1	0	0	0	1	0	0	1	1	0	0	0	0
Seq5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Seq6	0	0	0	0	1	0	0	0	0	0	0	0	0	1
Seq7	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Seq8	0	0	0	1	0	0	0	0	0	0	0	0	0	0

FIG. 5 – *Tableau booléen d'apprentissage*

3 Algorithme Hybride de Sélection d'attributs

Dans cette section nous présentons notre algorithme hybride de sélection d'attributs. Comme nous avons mentionné précédemment, les algorithmes hybrides fusionnent la rapidité des algorithmes filtrants et la précision des algorithmes enveloppants. Notre algorithme de sélection se compose de deux phases principales : une première phase filtrante et une deuxième phase enveloppante (*wrapper*), dans ce qui suit nous expliquons ces deux phases et nous terminerons par un schéma général de notre approche.

1. *Phase filtrante* : la phase filtrante se caractérise par l'utilisation d'un critère statistique permettant de générer un meilleur sous-ensemble de notre ensemble d'attributs de départ extraits à l'aide de l'algorithme d'extraction.

   La phase filtrante se réalise en deux étapes :

   (a) *Première étape* : nous utilisons le critère de la fréquence d'un attribut dans un fichier de protéines, le choix de la fréquence sert à éliminer les motifs inutiles et de très faible fréquence et de les ignorer par la suite. Cette étape consiste à éliminer les attributs qui ont un pourcentage de présence inférieur ou égal à un taux (soit x%), ou x est une constante. La meilleure valeur de x sera conservée. Nous avons réalisé plusieurs expérimentations sur un ensemble de [2,..,5] grammes où nous avons varié x et les résultats montrent que la meilleure valeur de x minimisant le taux d'erreur est de 20% (voir Tableau 1), la fréquence d'un attribut A est exprimé

par :

$$Freq(A) = \frac{nombre\ d'apparition\ de\ A\ dans\ les\ sequences}{nombre\ total\ de\ sequences} \quad (1)$$

(b) *Deuxième étape* : la sélection avec la fréquence présente quelques inconvénients dont on peut mentionner le fait qu'elle ne tient pas compte des redondances des attributs. En effet, si on duplique plusieurs fois l'attribut le plus pertinent, les premières places seront occupées par ce même attribut. Il est ainsi nécessaire d'introduire une nouvelle contrainte dans notre approche de sélection d'attributs. Pour cela, nous avons essayé de traiter le problème de la redondance des attributs, tout en considérant leurs pertinences. Après un premier filtre par la fréquence les attributs sont soumis à un deuxième filtre : celui de la mesure de corrélation entre les descripteurs (Mhamdi et al., 2006; Sauwens, 2010).

La redondance entre les attributs se fait par une construction d'une matrice de corrélation attribut-attribut. Nous pouvons distinguer une corrélation (redondance) attribut-attribut si la plus longue chaîne commune entre deux attributs (PLCC) existe. Pour garder le meilleur attribut et le plus pertinent, nous utilisons le calcul des indices α et β associés à chaque attribut dans les différentes familles. Pour chaque famille *fi* si $\alpha \leq \alpha_0$ et $\beta \leq \beta_0$ alors l'attribut concerné est considéré comme étant discriminant et minimal sinon il est considéré comme étant ambigu et il faut l'éliminer.

Les indices α et β sont exprimés par :

$$\alpha = \frac{nombre\ des\ chaines\ de\ fi\ dans\ lequels\ X\ apparait}{nombre\ total\ des\ chaines\ de\ fi} * 100 \quad (2)$$

$$\beta = \frac{nombre\ des\ chaines\ \bigcup fi\ dans\ lequels\ X\ apparait}{nombre\ total\ des\ chaines\ de\ \bigcup fi} * 100 \quad (3)$$

A la fin de la deuxième étape, nous trions les attributs restants par la corrélation selon un ordre décroissant de la valeur absolu de α-β avec abs=|α-β|.

2. *Phase enveloppante* : Cette phase consiste à l'utilisation d'un algorithme enveloppant (*wrapper*)pour sélectionner le meilleur sous ensemble d'attributs à partir de l'ensemble réduit d'attributs de la première phase. Nous adoptons l'algorithme d'apprentissage SVM (Cortes et Vapnik, 1995; Cristianini et Shawe-Taylor, 2000; Muller et al., 2001) vu sa puissance de précision et son efficacité.

Nous réalisons une sélection d'attributs, suivant une direction de *recherche en avant* (*best-first search*) (Devijver et Kittler, 1982; Miller, 1990) : on initialise l'ensemble des attributs à sélectionner $\Phi_k(f_1, f_2,\ldots, f_n)$ à l'ensemble vide. Puis, durant chaque itération, on sélectionne l'attribut qui possède la valeur |α-β| la plus élevée, on le supprime de $\Psi_k(f_1,f_2,\ldots,f_n)$, on l'insère dans $\Phi_k(f_1, f_2,\ldots, f_n)$ et on calcule le nouveau taux d'erreur de classification *Emin* obtenu suite à une classification faite par le classifieur SVM. On réitère ce processus jusqu'à ce que $\Psi_k(f_1,f_2,\ldots,f_n)$ soit vide. L'ensemble $\Phi_k(f_1, f_2,\ldots, f_n)$ ainsi obtenu représente l'ensemble des attributs sélectionnés.

Notre algorithme est de complexité O(*nbrmot***cpfreq***cpfreq*) en temps de calcul avec *nbrmot* le nombre d'attributs dans la phase d'extraction et *cpfreq* est le nombre d'attributs après

la phase de sélection par fréquence. La figure suivante résume notre approche de sélection d'attributs :

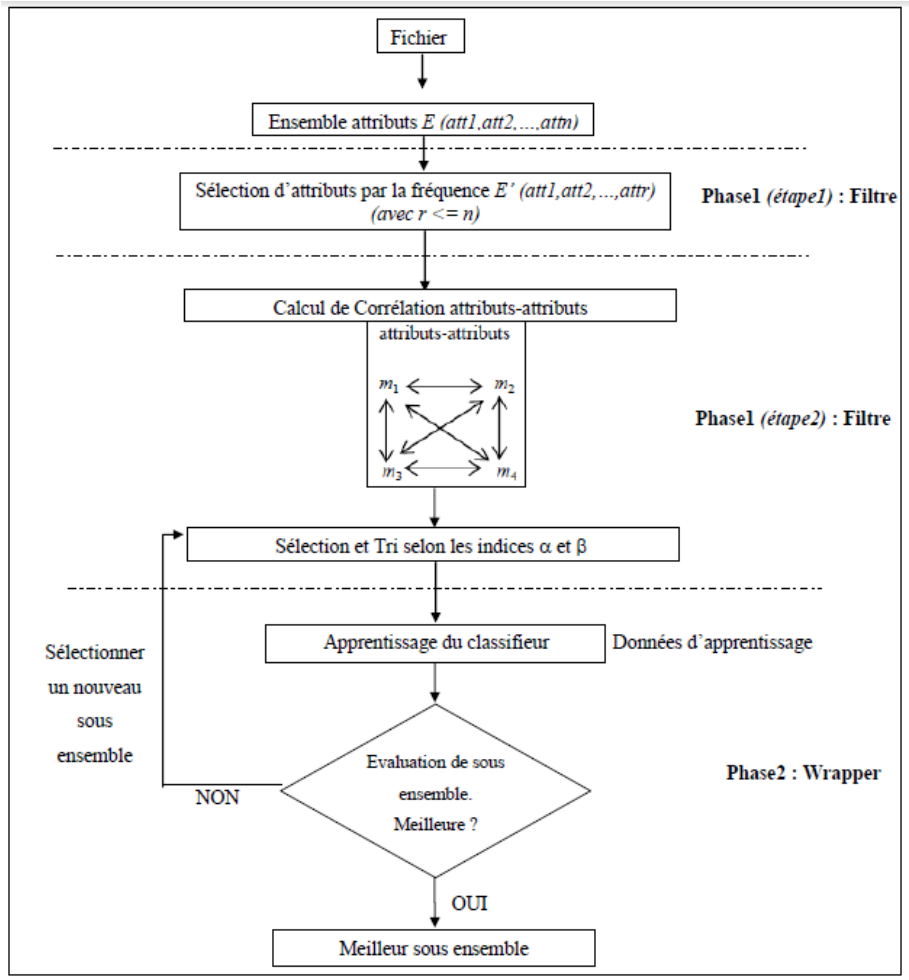


FIG. 6 – *Processus de l'algorithme hybride de sélection*

4 Expérimentations et résultats

Pour évaluer les performances de notre algorithme de sélection d'attributs, nous disposons de 5 familles de protéines de la banque de données SCOP (Murzin et al., 1995), ces familles sont rassemblées deux à deux ainsi nous avons 10 fichiers de données que nous utilisons pour faire nos expérimentations.

4.1 Expérimentation du premier critère de filtrage

Nous expliquons notre choix pour le premier critère de filtrage de notre algorithme celui de la fréquence. En effet, nous avons opté pour la fréquence de 20 % vu les expérimentations faites en variant les valeurs de fréquence. Nous avons varié la fréquence entre trois valeurs : 10%, 20% et 30%.

Familles	Résultats avant sélection		Résultats après sélection par fréquence					
	Nombre d'attributs	Err. SVM	Nombre d'attributs sélectionnés			Taux d'erreur avec SVM		
			10 %	20 %	30 %	10 %	20 %	30 %
F1∪F2	59366	0,0256	2185	647	395	0,0140	0,0116	0,0163
F1∪F3	57546	0,0851	2119	783	437	0,0553	0,0468	0,0250
F1∪F4	61557	0,0300	2140	805	459	0,0293	0,0283	0,0350
F1∪F5	58237	0,0556	2048	663	403	0,0398	0,0426	0,0489
F2∪F3	80774	0,0240	2685	896	494	0,0200	0,0200	0,0220
F2∪F4	84920	0,0172	2738	891	505	0,0172	0,0125	0,0187
F2∪F5	81479	0,0281	2467	796	452	0,0211	0,0211	0,0228
F3∪F4	82571	0,0746	2691	979	479	0,0567	0,0507	0,0537
F3∪F5	79399	0,0475	2409	840	462	0,0413	0,0492	0,0459
F4∪F5	83171	0,0351	2525	863	435	0,0351	0,0324	0,0351

TAB. 1 – *Taux d'erreur de classification avant et après la sélection d'attributs par fréquence*

4.2 Expérimentations de l'algorithme hybride de Sélection

Nous allons présenter les résultats de notre approche hybride de sélection. Comme nous avons expliqué, pour garder l'attribut le plus pertinent et pour éliminer la redondance entre les attributs nous avons construit une matrice de corrélation attribut-attribut et nous avons utilisé les indices α et β associés à chaque attribut dans les différentes familles afin de ne garder que l'attribut discriminant. Les expérimentations ont montré que la meilleure valeur de α et β est de : α= 40 ; β=60 (voir TAB 2).

α	β	Nb de motifs	Taux d'erreur
100	0	0	-
90	10	0	-
80	20	3	0,0817
70	30	26	0,0705
60	40	127	0,0209
50	50	189	0,0525

TAB. 2 – *Taux d'erreur obtenues pour les familles F1∪ F2*

Les tableaux et figures suivantes présentent la comparaison des taux d'erreur obtenus avec les attributs avant sélection, avec les attributs sélectionnés selon leurs fréquences et avec les attributs sélectionnés par notre nouvel algorithme.

k-gramme	avant sélection		étape1		étape2		sélection hybride (*Emin*)	
	nb att.	Err. SVM	nb att.	Err. SVM	nb att.	Err. SVM	nb att.	Err. SVM
2 gr	400	0.0163	371	0.0163				
[2,3] gr	7000	0.0163	637	0.0116				
[2..4]gr	30408	0.0186	644	0.0116				
[2..5]gr	59366	0.0256	647	0.0116	127	0.0209	61	0.0093
[2..6]gr	89976	0.0302	648	0.0116				
[2..7]gr	121635	0.0302	648	0.0116				
[2..8]gr	154079	0.0070	648	0.0116				

TAB. 3 – *application de l'algorithme hybride pour le fichier F1∪F2*

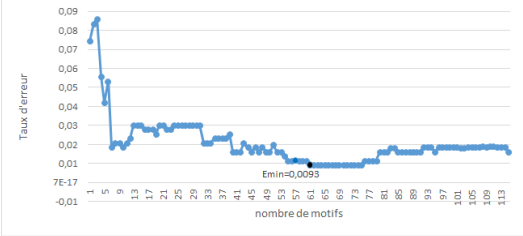


FIG. 7 – *Variations du taux d'erreur en fonction du nombre d'attributs sélectionnés dans le cas de F1∪F2*

k-gramme	avant sélection		étape1		étape2		sélection hybride (*Emin*)	
	nb att.	Err. SVM	nb att.	Err. SVM	nb att.	Err. SVM	nb att.	Err. SVM
2 gr	400	0.0163	384	0.0078				
[2,3] gr	7498	0.0125	792	0.0125				
[2..4]gr	40683	0.0125	855	0.0125				
[2..5]gr	84920	0.0172	891	0.0125	82	0.0125	59	0.0125
[2..6]gr	132629	0.0266	916	0.0141				
[2..7]gr	182541	0.1812	933	0.0156				

TAB. 4 – *application de l'algorithme hybride pour le fichier F2∪F4*

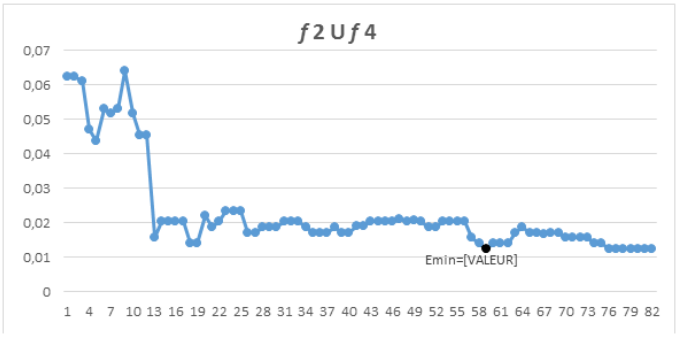


FIG. 8 – *Variations du taux d'erreur en fonction du nombre d'attributs sélectionnés dans le cas de F2∪F4*

Les résultats illustrés dans les tableaux *TAB.3* et *TAB.4* et les figures *FIG.7* et *FIG.8* montrent que notre algorithme réduit le nombre d'attributs d'une façon remarquable. Le nombre d'attributs est passé de quelques milliers à quelques dizaines. Cette réduction s'est représentée positivement sur les performances du classifieur utilisé. De même, les taux d'erreur obtenus confirment que les critères que nous avons adopté montrent leur efficacité et fournissent des résultats optimaux et aussi grace à notre approche adopté pendant l'extraction des attributs.

5 Conclusion

Dans ce papier, nous avons présenté un algorithme hybride de sélection d'attributs. Ces attributs sont utilisés pour faire le classement de protéines. Pour cela nous avons appliqué le processus d'Extraction de Connaissances à partir de Données (ECD).

Nous avons présenté les différentes étapes de notre processus : Extraction d'attributs, pondération d'attributs, sélection d'attributs et classement de protéines.

Les résultats de l'étude expérimentale ont montré l'efficacité de notre algorithme en le comparant avec des algorithmes filtres de sélection d'attributs. Comme perspective à notre travail nous pensons à améliorer nos résultats en comparant notre algorithme avec d'autre algorithme filtre et hybride, tels que l'algorithme Relief (Liu et al., 2002), CFS (Hall, 2000), FCBF (Yu et Liu, 2003).

Références

Cortes, C. et V. Vapnik (1995). Support-vector networks. *Mach. Learn. 20*(3), 273–297.

Cristianini, N. et J. Shawe-Taylor (2000). *An Introduction to Support Vector Machines*. Cambridge, UK: Cambridge University Press.

Devijver, P. et J. Kittler (1982). *Pattern Recognition: A Statistical Approach*. Prentice-Hall.

Fayyad, U., G. Shapiro, et P. Smyth (1996). From data mining to knowledge discovery: A overview. In E. Simoudis, J. Han, et U. Fayyad (Eds.), *Advances in Knowledge Discovery and Data Mining*, pp. 1–34. MIT Press.

Frawley, W. J., G. Piatetsky-Shapiro, et C. J. Matheus (1992). Knowledge discovery in databases: An overview. *AI Mag. 13*(3), 57–70.

Hall, M. A. (2000). Correlation-based feature selection for discrete and numeric class machine learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, San Francisco, CA, USA, pp. 359–366. Morgan Kaufmann Publishers Inc.

Liu, H., H. Motoda, et L. Yu (2002). Feature selection with selective sampling. In *Proceedings of the Nineteenth International Conference on Machine Learning*, ICML '02, San Francisco, CA, USA, pp. 395–402. Morgan Kaufmann Publishers Inc.

Mhamdi, F., M. Elloumi, et R. Rakotomalala (2004). Discriminant descriptors extraction for proteins classification. In *Neuro-Computing and Evolving Intelligence*, Auckland, New Zealand.

Mhamdi, F., M. Elloumi, et R. Rakotomalala (2006). Extraction et sélection des n-grammes pour le classement de protéines. In *Atelier Extraction et gestion de connaissances appliquées aux données biologiques (Bio-EGC 06), Lille*, pp. 25–37.

Mhamdi, F., M. Kchouk, et S. Aouled El Haj Mohamed (2013). Nouveaux Algorithmes dŠExtraction de Motifs et de Pondération pour le Classement de Protéines. In *SeqBio 2013*, Montpellier, France.

Mhamdi, F., R. Rakotomalala, et M. Elloumi (2006). A hierarchical n-grams extraction approach for classification problem. In *SITIS*, Hammamet, Tunisie, pp. 211–222.

Miller, A. (1990). *Subset Selection in Regression*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis.

Muller, K. R., S. Mika, G. Ratsch, K. Tsuda, et B. Scholkopf (2001). An introduction to kernel-based learning algorithms. *Trans. Neur. Netw. 12*(2), 181–201.

Murzin, A. G., S. E. Brenner, T. Hubbard, et C. Chothia (1995). Scop: A structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology 247*(4), 536 – 540.

Piatetsky-Shapiro, G., C. Djeraba, L. Getoor, R. Grossman, R. Feldman, et M. Zaki (2006). What are the grand challenges for data mining?: Kdd-2006 panel report. *SIGKDD Explor. Newsl. 8*(2), 70–77.

Sauwens, C. (2010). *Algorithmes pour lŠExtraction de connaissances à partir de données biologiques*. Thèse de doctorat, Université Tunis elManar.

Yu, L. et H. Liu (2003). Feature selection for high-dimensional data : A fast correlation-based filter solution. In 20^{th} *International Conference on Machine Learning*, pp. 856–863.

Summary

To the classification of proteins we apply the process of Knowledge Discovery from Databases (KDD) . In the preprocessing phase of this process is one attribute extraction and encoding of data to construct the table of data that will be the entry of data mining techniques . These techniques are applied to the classification, consolidation, etc. . For the great mass of biological data , the number of attributes retrieved is huge. This affects the quality of the data table and therefore the results of learning algorithms . Hence the need to go through a phase of selection of attributes to eliminate irrelevant attributes and/or redundant . The elimination of these attributes enhances the performance of learning algorithms error rate and computation time . In this paper, we present a hybrid algorithm for selecting attributes. The experiments on real biological data extracted from biological databases yielded good results.

Index

EGC 2014

Organisateurs

Sponsors