Table des matières

Table des figures

I Contexte général

I.1 Introduction

Le développement des moyens informatiques et de calcul permet le stockage (bases de données), le traitement et l'analyse d'ensembles de données très volumineuses . Plus récemment, avec le perfectionnement des logiciels et de leurs interfaces, l'explosion du e-commerce, la popularisation de nouveaux appareils connectés à internet (smartphone, tablettes,..), ainsi que les données produites par les systèmes embarqués (caméras, montres, voitures, etc.), offrent aux utilisateurs le confort et la facilité de l'utilisation, mais génère énormément de données. Chacun d'entre nous participe en enrichissant ses bases de données, celles-ci peuvent contenir plusieurs types, des coordonnées de géo-localisation, des informations sur les personnes et leurs habitudes, des statistiques de l'utilisation, des fichiers de logs, des données des images ou de la parole ,ou encore des données sur l'être humain.

Les données recueillies massivement sont analysées pour l'aide à la décision comme ce fut le cas en marketing quantitatif avec la fouille de données. La redondance de l'information dans cette masse d'informations, fait appel à des méthodes statistiques, pour la bonne estimation de résultat et une meilleure qualité. Les techniques de l'analyse et de la recommandation deviennent de plus en plus demandées, afin de trouver une information pertinente dans un tas de données en un temps raisonnable, chose qui a engendré la naissances de plusieurs méthodes d'apprentissage automatique et de recommandation. Il s'agit de proposer des connaissances nouvelles qui enrichissent les interprétations

L'apprentissage automatique est un domaine qui fait référence au développement, à l'analyse et à l'implémentation de méthodes, qui permettent à une machine de simuler des mécanismes d'apprentissage, et d'évoluer grâce à un processus d'apprentissage, et ainsi d'exécuter des tâches qui sont difficiles, ou impossibles d'exécuter par des moyens algorithmiques plus classiques. Son champ d'application augmente de jour en jour, dès qu'un domaine dispose de données, la question de l'utilisation de celui-ci pour améliorer les algorithmes du domaine se pose systématiquement. Ce dernier a fait objet de nombreux travaux de recherche, dans les modèles d'apprentissages existant. Nous utilisons l'apprentissage discriminatif, qui a pour objectif, d'apprendre et de raisonner dans des contextes génératifs et incertains .

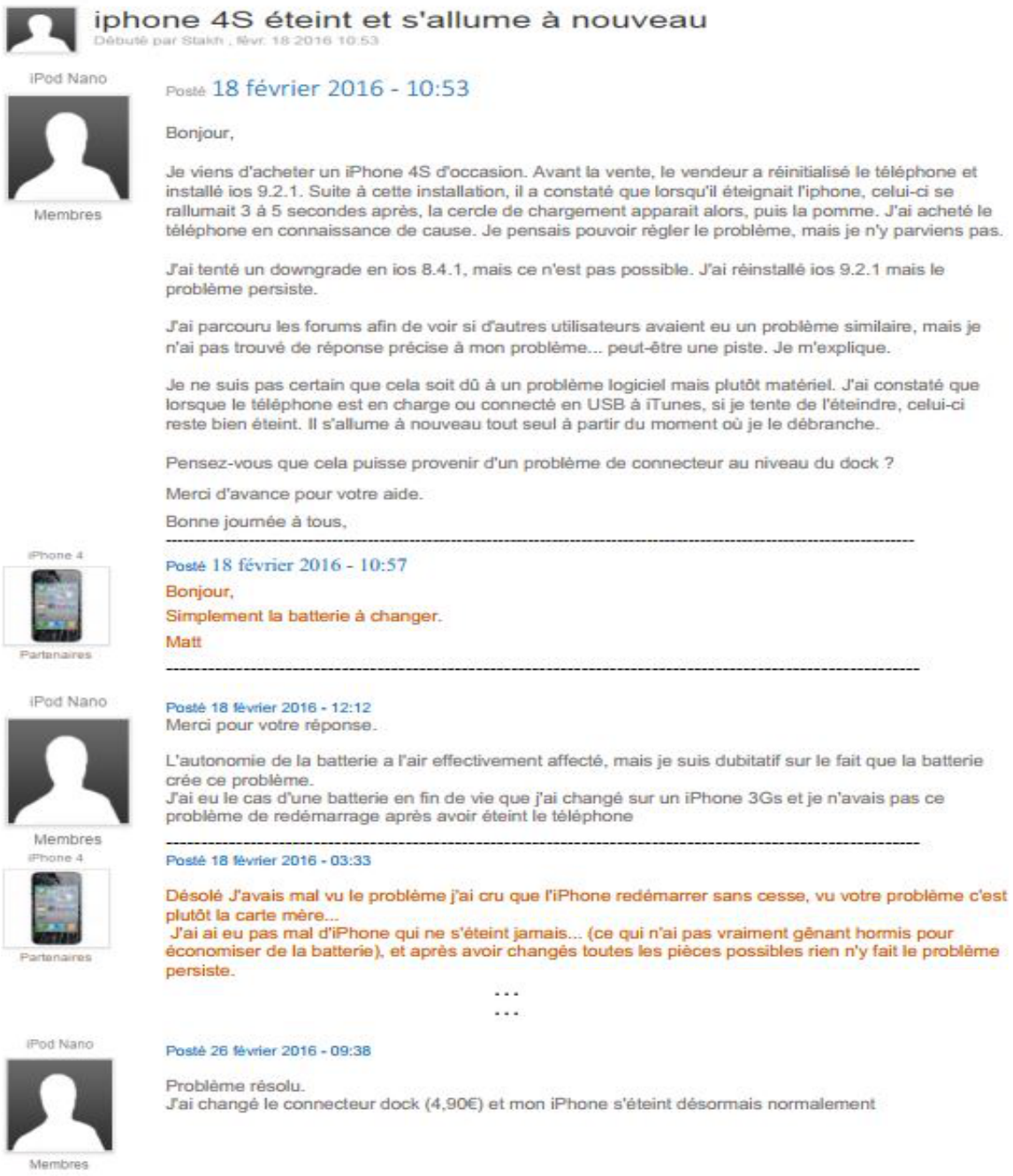
I.2 Présentation du projet

L'objectif initial du projet était d'étudier et d'utiliser des modèles graphiques probabilistes, plus précisément, les modèles relationnels probabilistes pour la recommandation de solutions pertinentes, sur un corpus de documents en se basant sur des connaissances déjà acquises. Nous avons convenu d'utiliser des modèles relationnels probabilistes avec des incertitudes de références qui s'adaptent au contexte de la prédiction de relations entre problèmes et solutions, avec pour objectif de recommander une solution pertinente à un problème remonté sous forme de texte écrit en langage naturel, à partir des informations contenues dans le texte, mais aussi éventuellement des caractéristiques de l'utilisateur.

Exemple d'un post : le post pour un problème donné sur un forum, s'effectue en attachant un titre, et une description détaillée. Parfois, il existe un certain nombre de tags qui permettent de trier les posts de blog en différentes catégories. Les lecteurs peuvent filtrer les posts de blog en utilisant les tags afin de lire des posts sur des sujets spécifiques. Après avoir posté ce dernier, les membres de communauté participent avec des suggestions, lorsque la solution testée par l'auteur réussit, le sujet passe en "résolu"

L'exemple de la figue 1, page 7 est tiré de "forum-iphone.fr" ; un lieu d'échange sur des thèmes de la nouvelle technologie. Le sujet du forum, est un sujet technique traitant un problème de redémarrage d'un smart-phone "iphone", l'exemple illustre la contribution des internautes pour mieux cerner le dysfonctionnement puis proposer des solutions convenables. Le post "iphone 4S s'éteint et s'allume à nouveau" est présenté par son auteur "Stakh", la contribution à la solution est faite par d'autres utilisateurs du forum.

Nous imaginons un cas similaire avec un appareil d'un autre type, dans le cas normal, le scénario le plus probable est de chercher dans les discussions existantes sur le forum, lire tous les posts, tester les solutions jusqu'à trouver par chance la bonne. S'il n'y a aucune réponse satisfaisante, il faut créer un nouveau post et attendre la participation des membres. La question posée est ce que nous pouvons arriver à identifier le problème automatiquement et favoriser une solution parmi d'autres existantes dans notre base ? Autrement est-ce que nous pouvons prédire des solutions, qui vont nous permettre de gagner du temps et de la performance ?



iphone 4S éteint et s'allume à nouveau

Débuté par Stakh , févr. 18 2016 10:53

iPod Nano

Membres

Posté 18 février 2016 - 10:53

Bonjour,

Je viens d'acheter un iPhone 4S d'occasion. Avant la vente, le vendeur a réinitialisé le téléphone et installé ios 9.2.1. Suite à cette installation, il a constaté que lorsqu'il éteignait l'iphone, celui-ci se rallumait 3 à 5 secondes après, la cercle de chargement apparait alors, puis la pomme. J'ai acheté le téléphone en connaissance de cause. Je pensais pouvoir régler le problème, mais je n'y parviens pas.

J'ai tenté un downgrade en ios 8.4.1, mais ce n'est pas possible. J'ai réinstallé ios 9.2.1 mais le problème persiste.

J'ai parcouru les forums afin de voir si d'autres utilisateurs avaient eu un problème similaire, mais je n'ai pas trouvé de réponse précise à mon problème... peut-être une piste. Je m'explique.

Je ne suis pas certain que cela soit dû à un problème logiciel mais plutôt matériel. J'ai constaté que lorsque le téléphone est en charge ou connecté en USB à iTunes, si je tente de l'éteindre, celui-ci reste bien éteint. Il s'allume à nouveau tout seul à partir du moment où je le débranche.

Pensez-vous que cela puisse provenir d'un problème de connecteur au niveau du dock ?

Merci d'avance pour votre aide.

Bonne journée à tous,

---

iPhone 4

Partenaires

Posté 18 février 2016 - 10:57

Bonjour,

Simplement la batterie à changer.

Matt

---

iPod Nano

Membres

Posté 18 février 2016 - 12:12

Merci pour votre réponse.

L'autonomie de la batterie a l'air effectivement affecté, mais je suis dubitatif sur le fait que la batterie crée ce problème.
J'ai eu le cas d'une batterie en fin de vie que j'ai changé sur un iPhone 3Gs et je n'avais pas ce problème de redémarrage après avoir éteint le téléphone

---

iPhone 4

Partenaires

Posté 18 février 2016 - 03:33

Désolé J'avais mal vu le problème j'ai cru que l'iPhone redémarrer sans cesse, vu votre problème c'est plutôt la carte mère...
J'ai ai eu pas mal d'iPhone qui ne s'éteint jamais... (ce qui n'ai pas vraiment gênant hormis pour économiser de la batterie), et après avoir changés toutes les pièces possibles rien n'y fait le problème persiste.

...

...

iPod Nano

Membres

Posté 26 février 2016 - 09:38

Problème résolu.
J'ai changé le connecteur dock (4,90€) et mon iPhone s'éteint désormais normalement

FIGURE 1 – Exemple d'échange questions / solutions entre les membres d'un forum

La section II propose un état de l'art sur les méthodes utilisées dans le cadre du projet. L'objectif initial du projet était de pouvoir prédire des liens entre les différents objets (réclamations et solutions). Ce problème de prédiction de lien à partir de données existantes rentre dans un domaine qui s'appelle le "relational data mining", ou le "statistical relational learning". Les modèles relationnels probabilistes décrits dans la section II.2 sont parmi les modèles les plus pertinents pour traiter à la fois l'information relationnelle (liens existants entre les réclamations et les solutions) et l'information apportée par les "descripteurs" associés à ces entités.

Le corpus fourni par eMinove ne relevant plus d'un problème de recommandation de solutions, nous avons proposé d'utiliser une méthode de "topic modelling", suffisamment générique pour pouvoir être utilisée dans le problème initialement posé, et aussi dans le domaine de la fouille d'opinion dont relevait le corpus proposé. La section II.5 décrit les différentes étapes classiques en fouille de texte pour pouvoir utiliser ces méthodes.

L'intérêt de ce type de méthode pour la fouille d'opinion est aussi brièvement présenté dans la section II.6.

La section III présente ensuite une modélisation du problème initial, avec la formalisation du schéma relationnel décrit dans la section III.1. La section III.2 montre la mise en place d'une "preuve de concept", chaîne de traitement, partant des données brutes fournies, application d'une méthode de topic modelling, et visualisation des résultats, pouvant ensuite être utilisée et déclinée pour différentes analyses. L' installation et l'utilisation des outils nécessaires à cette chaîne de traitement sont respectivement présentés dans les annexes I et II.

II État de l'art

II.1 Introduction

L'objectif de cette partie est de présenter les modèles graphiques probabilistes comme les réseaux bayésiens et les modèles relationnels probabilistes (MRP), dont l'extension appelée MRP avec incertitude de référence peut être utilisée pour prédire les liens entre items. Certaines des données d'entrée étant sous forme de texte, nous présenterons aussi les méthodes simples de pré-traitement de textes, puis les méthodes de modélisation de sujet (topic modelling) classiquement utilisés dans le domaine de la fouille d'opinion.

II.2 Modèles graphiques probabilistes

II.2.1 Réseaux bayésiens

Les réseaux bayésiens (RB) sont des modèles graphiques probabilistes introduit par Judea Pearl [21] en 1982 et constituent une technologie puissante en Intelligence Artificielle, et en apprentissage automatique. Ils s'appliquent sur un ensemble prédéfini de variables aléatoires, dont la relation entre elles est fixée à l'avance. Les RB constituent un langage graphique et une méthodologie, simples et corrects, pour exprimer pratiquement ce de quoi on est certain ou incertain, avec l'utilisation des probabilités jointes, pour décrire l'incertitude de faits, et à partir de ces probabilités jointes, on peut retrouver les probabilités conditionnelles souhaitées, ils reposent sur la formule de Bayes, reliant des probabilités conditionnelles avec des probabilités jointes. Au cours de la dernière décennie ils ont connu un grand succès dans une grande variété d'applications de recherche dans le monde réel et professionnel.

Définissons plus formellement un réseau bayésien : Un réseau bayésien est un graphe orienté sans circuit avec un ensemble X de noeuds et un ensemble E d'arcs orientés. Un noeud contient, 1-le nom d'une variable, 2-une table de probabilités de cette variable en fonction des valeurs de ses parents

> **Définition II.2.1** (Réseau Bayésien)**.** *Un réseau bayésien [21] un tuple* $\mathcal{B} = (\mathcal{G}, \theta)$ *tel que :*
> – $\mathcal{G} = (X, E)$ *graphe dirigé sans circuit dont les sommets sont associés à un ensemble de variables aléatoires* $X = \{X_1, ..., X_n\}$,
> – $\theta = \{P(X_i|Pa(X_i))\}$ *, ensemble des probabilités de chaque nœud* X_i *conditionnellement à l'état de ses parents* $Pa(X_i)$ *dans* $\mathcal{G}$.

Illustrons cette définition à l'aide d'un exemple concret.

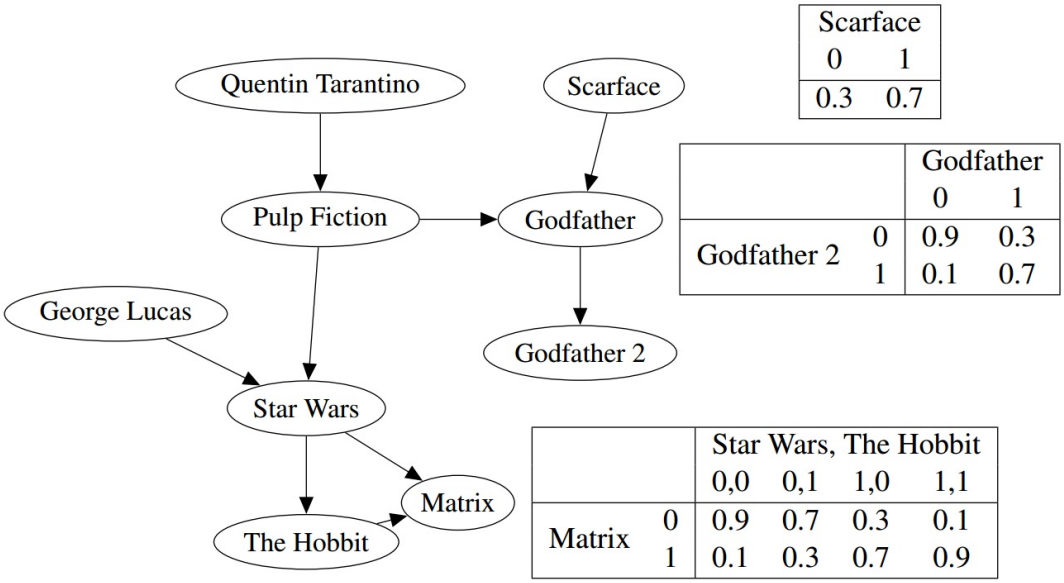


FIGURE 2 – Exemple de structure de réseau bayésien dans un contexte de recommandation de films [5]. Les variables sont ici booléennes et expriment le goût d'une personne particulière pour un film ou un réalisateur. Chaque variable est associée à une distribution de probabilités définie sachant les valeurs de ses variables parentes dans le graphe de structure. Plusieurs distributions sont proposées à titre d'exemple sous la forme de tables de probabilités conditionnelles.

II.2.2 Exemple des réseaux bayésiens

Un exemple de structure de réseau bayésien en figure 2 (de la page 10). Cette structure représente le fonctionnement d'un système de recommandation de films. Le nœud "George Lucas" représente la variable aléatoire qui correspond à la chance que ce réalisateur soit aimé par une personne. La variable "Star Wars" a deux parents et deux enfants. Enfin les valeurs (Matrix = 0 et Star wars = 0.1 donne une probabilité 0.7 pour le poids de l'arc P(Matrix=0 | Starwars=0 et TheHobbit=1) = 0.7, chaque variable est associée à une distribution de probabilités définie, sachant les valeurs de variables parents, les distributions de cet exemples sont proposées sous la forme de tables de probabilités conditionnelles. Nous pouvons par exemple déduire de ce système la recommandation des films qui n'ont pas encore été vus, mais qui ont une grande probabilité d'être appréciés par cette personne.

II.3 Modèles relationnels probabilistes

II.3.1 Introduction

Toute information est intéressante, l'exploitation et l'analyse d'information demande un traitement bien spécial qui dépend du domaine de l'utilisation et de l'individu. Pour cette raison la conception et la création d'un modèle générique permet un raisonnement spécial sur un ensemble de données. L'utilisation d'un modèle relationnel probabiliste permet de faire un traitement plus précis sur les données, ainsi qu'une forte adaptation du modèle sur plusieurs cas.

Au cours de la dernière décennie, Les réseaux bayésiens ont été utilisés avec succès dans une grande variété d'applications de recherche dans le monde réel. Cependant, malgré leur succès, les réseaux bayésiens sont souvent inadéquats pour représenter des domaines grands et complexes. Comme définie précédemment un réseau bayésien pour un domaine donné implique un ensemble prédéfini de variables aléatoires, dont la relation à l'autre est fixée à l'avance.

II.3.2 Langage relationnel

Comme tout sorte de langage, un langage permet de décrire certaines fonctions dans le domaine d'utilisation, dans notre cas le langage relationnel nous permet de définir de manière explicite les types d'objets dans notre domaine, la figure 3 (a) extrait de [8] montre le schéma pour un domaine simple que nous prenons comme exemple, le domaine est une université, qui contient :

$$professeurs, etudiants, cours, inscriptions$$

Les classes dans ce schéma sont : professeur, étudiant, cours et inscription. Un schéma pour un modèle relationnel décrit un ensemble de classes, X ={X1, ..., Xn}. Chaque classe est décrite par un ensemble d'attributs descriptifs.

Exemple
Professeurs a comme attribut descriptif : popularité
Étudiant a comme attribut descriptif : intelligence.

L'ensemble des attributs descriptifs d'une classe X est notée A(X) ou X.A, avec "A" : Attribut et "X" : la classe, et son espace de valeurs est notée V(X.A). Nous supposons ici que les places de valeur sont finies. Par exemple, la classe « étudiant » a le descriptif d'attributs Intelligence et classement. L'espace de valeur pour « Étudient.Intelligence » dans cet exemple est {high, low}.

Les reference slots, permettent à un objet de se référer à un autre objet. Par exemple on peut vouloir qu'un cours puisse avoir une référence à l'instructeur du cours. Un dossier d'enregistrement doit se référer à la fois au cours associé et à l'étudiant qui suit le cours. La façon la plus simple d'atteindre cet effet est la suivante.

Reference Slot : Chaque classe est associée à un ensemble de slot référence. L'ensemble des slot références d'une classe X est noté R(X). Nous utilisons $X.\rho$ pour désigner la référence slot ρ de X. Le schéma spécifie le type de plage d'objet "Range" et "Dom" qui peuvent être référencés. Plus formellement, pour chaque ρ dans X, le domaine $Dom\,[\rho]$ = X et la plage Range $[\rho]$ = Y pour une certaine classe Y dans X.

Nous prenons notre exemple de la figure 3 (a), la classe "cours" a une référence slot ρ = instructeur, avec $Dom\,[\rho]$ = course et $Range\,[\rho]$ = professeur, et la classe "registration" a des références slot ρ = { Cours et étudiants }. Dans la figure 3 (a) les références slot ρ sont soulignées. La même représentation que celle de bases de données relationnelles. Chaque classe correspond à une seule table et chaque attribut correspond à une colonne, la référence slot ρ d'une table X est la clé étrangère référençant la classe Y.

Reference Slot inverse ρ^{-1} : Pour chaque ρ slot de référence, nous pouvons définir un slot inverse ρ^{-1}, qui est interprété en tant que fonction inverse de ρ. Par exemple, nous pouvons définir un slot inverse pour le slot étudiants de l'enregistrement et l'appeler Enregistré-In. A noter que ceci n'est pas une relation une à une, mais renvoie un ensemble d'objets d'inscription. Plus formellement, si $Dom\,[\rho]$ est X et $Range\,[\rho]$ est Y, alors $Dom\,[\rho^{-1}]$ = Y et $Range\,[\rho^{-1}]$ = X .

L'exemple de la figure 3 (a), nous avons la classe Enregistrement à comme référence slot ρ = étudiants, sa référence slot inverse ρ^{-1} = Enregistrement-In (tous les enregistrement de l'étudiant en question) $Dom\,[\rho^{-1}]$ = Etudiant, $Range\,[\rho^{-1}]$ = Enregistrement.

Slot chaîne : L'autre de l'article [8] a définis la notion de slot chaîne. Une slot chaîne $\{\rho_1....\rho_k\}$ à une suite de slot (direct ou inverse) tel que pour tout i, $Range\,[\rho_i] = Dom\left[\rho_i^{-1}\right]$ ce qui nous permet de composer des slot chaînes. Le point fort des slot chaînes est de définir les fonctions des objets à d'autres objets auxquels ils sont indirectement liées.

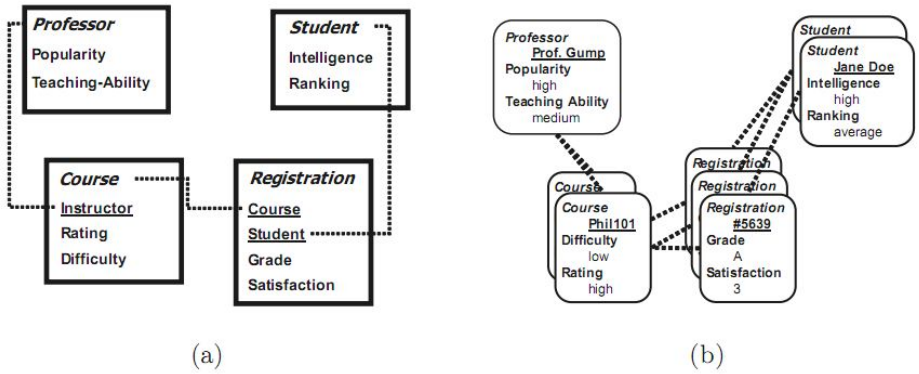


FIGURE 3 – Exemple [8] (a) d'un schéma relationnel pour un simple domaine "université", les attributs soulignés sont des référence slot de la classe et les lignes en pointillés indiquent les types d' objets référencés . (b) d'une instanciation de ce schéma.

Par exemple, Etudiant.Registered-In.Cours.Instructeur peut être utilisé pour désigner les instructeurs d'un étudiant. détaillant cette slot chaîne "Etudiant.Registered-In" tous les enregistrements de l'étudiant x, "Etudiant.Registered-In.Cours" tous les cours où il est enregistré, "Etudiant.Registered-In.Cours.Instructeur" tous les enseignants des cours où il est enregistré. A norter qu'une slot chaîne décrit un ensemble d'objets d'une classe.

II.3.3 Schéma d'instanciation

Un schéma d'instanciation est un schéma qui est composé des interprétations logiques en utilisant le langage relationnel, a fin d'accéder à des compositions dont nous avons besoins. tout simplement c'est un graphe où il y a les relations entre les objets, qui respectent les contraintes, et les valeurs des attributs, Il est défini dans [8] par :

Définition II.3.1 (schéma d'instanciation)**.** *Une instance $\mathcal{I}$ d'un schéma est tout simplement une interprétation logique relationnelle de ce vocabulaire. Il défini pour chaque classe X l'ensemble des objets dans cette classe, pour chaque attribut A(X) une valeur x (dans le domaine approprié), et une valeur y pour chaque $x.\rho$ slot référence, $y \in range\,[\rho]$ qui est un objet dans le type de range approprié . A l'inverse, $y.\rho^{-1} = \{x | x.\rho = y\}$. Pour $x \in X$ .Pour chaque objet x*

Utilisateur			
id	nom	sexe	age
U1	Albert	H	30
U2	Fred	H	40

FILM			
id	genre	classement	année
F1	Action	1	2005
F2	Action	20	1990

Acteur			
id	age	pays	célébrité
A1	20	USA	fort
A2	22	France	moyen

évaluation			
id	utilisateur	film	évaluation
E1	U2	F1	2
E2	U1	F2	4

jouer			
id	film	acteur	rôle
J1	F1	A1	principal
J2	F2	A2	secondaire

FIGURE 4 – Exemple d'instanciation pour un domaine de film contenant deux attributs par table

dans l'instance et chacun de ses attributs A, nous utilisons $\mathcal{I}_x.A$ pour désigner la valeur de x.A dans $\mathcal{I}$.

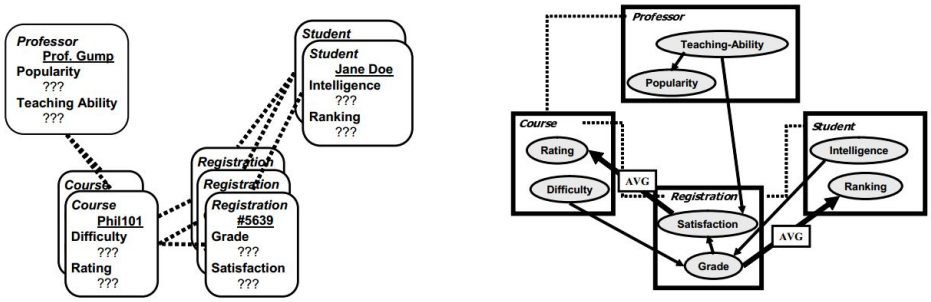
la figure 3(b) montre un exemple du schéma d'instanciation. Dans ce simple cas il y a l'existence d'un seul professeur, deux cours, trois inscriptions, et deux Élèves. Les relations entre eux montrent que le professeur est le responsable de deux cours, et qu'un étudiant («Jane Doe») est inscrit dans une seule inscription ("Phil 101"), tandis que l'autre étudiant est inscrit dans deux inscriptions

Par exemple, la figure 4 est un exemple du schéma d'instanciation. Dans ce domaine de film, nous avons cinq tables : utilisateur, film ,acteur, évaluation , jouer. Nous avons aussi deux attributs par tables, les relations entre eux montrent que l'acteur A1 joue le rôle principal dans le film F1, l'utilisateur U2 évalue le film F1 d'une note de 2/5.

II.3.4 Squelette relationnel

Le *squelette relationnel* est la connaissance partielle sur le jeu de données comme montré sur la figure 5, Cette connaissance partielle consiste en l'énumération des tuples existants, ainsi que les associations existantes entre eux par le biais de contraintes référentielles,

Définition II.3.2. *[8] Un squelette relationnel σr d'un schéma relationnel est une spécification partielle d'un cas de figure d'un schéma. Il spécifie l'ensemble d'objets $\sigma r(Xi)$ pour chaque catégorie et les relations qui détiennent entre les objets. Cependant, il laisse les valeurs des attributs non spécifiées.*



(a) (b)

FIGURE 5 – Exemple [8], (a) de squelette relationnel pour le domaine "université" et (b) de modèle relationnel probabiliste pour le même exemple.

Utilisateur

id	nom	sexe	age
U1	?	?	?
U2	?	?	?

FILM

id	genre	classement	année
F1	?	?	?
F2	?	?	?

Acteur

id	age	pays	célébrité
A1	?	?	?
A2	?	?	?

évaluation

id	utilisateur	film	évaluation
E1	U2	F1	?
E2	U1	F2	?

jouer

id	film	acteur	rôle
J1	F1	A1	?
J2	F2	A2	?

FIGURE 6 – Exemple de squelette relationnel pour un domaine de film contenant deux attribut par table

La figure 5 (a) montre un squelette relationnel pour notre exemple. Le squelette relationnel définit les variables aléatoires dans notre domaine ; nous avons une variable aléatoire pour chaque attribut de chaque objet dans le squelette. Un PRM spécifie alors une distribution de probabilité sur les mondes possibles compatibles avec le squelette relationnel σr.

La figure 6 est un squelette relationnel. Dans ce domaine de film, nous avons cinq tables : utilisateur, film ,acteur, évaluation , jouer. La table jouer comporte les informations des acteurs et des films sans les valeurs des attributs, par exemple nous avons l'acteur A1 joue dans le film F1, mais nous avons pas la valeur de l'attribut rôle, la table Film contient les films sans les valeurs des autres attributs.

II.3.5 Modèles relationnels probabilistes

Un réseau bayésien ne peut pas être utilisé pour traiter des domaines où nous pourrions rencontrer un nombre d'entités faisant varier dans une variété de configurations. Cette limitation de réseaux bayésiens est une conséquence directe du fait qu'ils manquent du concept "d'objet" (ou l'entité de domaine). D'où, ils ne peuvent pas représenter des principes généraux des objets semblables multiples qui peuvent alors être appliqués dans des contextes multiples.

Les modèles relationnels probabilistes (MRP) [8], ou réseaux bayésiens relationnels sont une extension relationnelle des réseaux bayésiens. Une approche adaptée pour la représentation de l'incertain dans un contexte relationnel avec un ajout des notions de l'orienté objet, pour être plus précis c'est un réseau bayésien orienté objet avec un schéma de base de données en entrée. L'utilisation de plusieurs concept de l'orienté objet a permis au MRP de s'adapter au différents cas, un MRP spécifie un modèle pour une distribution de probabilité sur une base de données relationnelle. Le modèle comprend un composant relationnel décrivant le schéma relationnel, et un composant probabiliste décrivant les dépendances probabilistes. Étant donné un ensemble d'objets, un MRP spécifie une distribution de probabilité sur un ensemble d'interprétations impliquant ces objets. Les deux composants de la syntaxe MRP sont :

Le schéma relationnel : c'est une description logique du domaine, il décrit un ensemble de classes, X ={X1, ..., Xn}. Chaque classe est associée à un ensemble d'attributs descriptifs.

Le modèle graphique probabiliste : décrit les dépendances probabilistes dans le domaine.

Dans la phase d'apprentissage, l'entrée contient un schéma relationnel qui spécifie le vocabulaire de base dans le domaine. C'est l'ensemble des classes, des attributs associés aux différentes classes, et les types possibles de relations entre les objets dans les différentes classes. Les données de formation se composent d'une instance entièrement spécifiée de ce schéma sous la forme d'une base de données relationnelle.

Dans la phase d'analyse, une fois que le MRP apprend, il sert comme un outil pour l'analyse exploratoire des données et peut être utilisé pour faire des prédictions et des inférences complexes dans des situations différentes.

[8] définissent les MRP par :

> **Définition II.3.3** (MRP)**.** *Étant donné un schéma de base de données* $\boldsymbol{R} = \langle \mathcal{R}^i \rangle_{i \leqslant m}$*,un MRP* $\mathcal{M} = (\mathcal{S}, \Theta)$ *est composé d'une structure* $\mathcal{S}$ *et d'un ensemble de paramètres* Θ*. La structure est composée d'un ensemble de variables*

aléatoires $\mathcal{V}$ et d'une fonction de parents pa :$\mathcal{V} \rightarrow \mathcal{P}(\Psi \times \Sigma(\boldsymbol{R}) \times \mathcal{V})$ où Ψ est un ensemble de fonctions d'agrégation (contenant la fonction id d'identité) et $\Sigma(R)$est l'ensemble des chaînes de références qu'il est possible de définir sur $\boldsymbol{R}$. L'ensemble $\mathcal{V}$ contient une variable aléatoire $A_{i,j}$ pour chaque attribut $A_j \in att(R^i)$ avec $dom(A_{i,j}) = ran(A_j)$. Finalement nous avons $\Theta = \{\theta_{i,j} = P(A_{i,j} | pa(A_{i,j}))\}$.

Nous pouvons définir dans un MRP deux types d'attributs :

- ☐ L'attribut x.A peut dépendre d'un autre attribut probabiliste x.B : Cette dépendance formelle induit une dépendance correspondante pour des objets individuels, pour tout objet x dans $\sigma r(X)$.

- ☐ L'attribut x.A peut également dépendre des attributs d'objets x.K.B : K est une slot chaîne, pour un objet x individuel, nous avons x.K représente l'ensemble des objets qui sont K-parents de x.

- ☐ L'attribut x.A dépend de la probabilité sur une propriété globale d'un multi-ensemble : Il y a beaucoup de notions naturelles et utiles de l'agrégation d'un ensemble : Son mode (valeur la plus fréquente), sa valeur moyenne (si les valeurs sont numériques), la médiane, maximum ou minimum (si les valeurs sont ordonnées), son cardinal, etc.

II.3.6 Exemple de MRP

Dans la figure 5 (b), les flèches définissent la structure de dépendance. On distingue deux types de parents formels.

1. L'attribut x.A dépendra de la probabilité sur x.B : la popularité d'un professeur dépend de sa capacité d'enseignement.
2. L'attribut x.A dépend des attributs d'objets x.K.B : Dans la figure, le grade d'un étudiant dépend de « registration.Student.Intelligence » et « Registration.Course.Difficulty ». Ou nous pouvons avoir une slot chaîne plus longue comme par exemple la dépendance de la satisfaction des étudiants sur « Registration.Course.Instructor.Teaching-Ability ».
   En outre, nous pouvons avoir une dépendance à l'égard du classement des élèves sur « Student.Registered-In.Grade ».
3. L'attribut x.A dépend de la probabilité sur une propriété globale d'un multi-ensemble : le classement d'un étudiant dépend des notes dans les cours auxquels

il est inscrit. Toutefois, chaque étudiant peut être inscrit à un nombre différent de cours, et nous aurons besoin de la moyenne de ses notes.

II.3.7 Conclusion

Nous avons vu que les réseaux bayésiens ne peuvent pas être utilisés pour traiter des domaines où le nombre d'entités varie dans une variété de configurations, nous avons présenté dans cette partie les concepts généraux nécessaires à la compréhension des modèles relationnels probabiliste. Nous avons ainsi commencé par présenter le langage relationnel et ses différents objets, ainsi que des exemples pour bien assimiler l'approche. Dans la partie suivante, nous allons présenter en détail une autre extension des MRP.

II.4 MRP et incertitude de référence

II.4.1 Introduction

Les modèles relationnels probabilistes (MRP) représentent une connaissance impliquant les différents attributs d'un schéma de base de données. Ceci implique que toute instance doit connaître le nombre de tuples pour chaque relation, et que chaque contrainte d'intégrité référentielle pour tout tuple concerné est certaine. En d'autres termes, cela signifie que les références entre tuples sont connues. Inférer sur les instanciations de tels MRP est alors restreint aux valeurs des attributs descriptifs de ces tuples.

La certitude de références entre tuples n'est pas toujours vérifiée. À titre d'exemple, dans un contexte de recommandation, nous ne connaissons a priori pas toutes les associations susceptibles d'exister entre des utilisateurs et les produits en vente et la tâche de recommandation vise justement à inférer la valeur d'associations inconnues à partir des informations certaines sur les entités et associations du domaine. Dans ce type de contextes où nous souhaitons pouvoir inférer sur les références entre tuples, il nous faut avoir recours à des versions étendues des MRP [7].

II.4.2 Entités et Associations

A fin de faciliter la compréhension dans la suite de ce chapitre, nous allons aborder la description d'un schéma de base de donnée dans le cadre d'une implémentation et une application aux modèles relationnels probabilistes avec incertitude de référence. Nous allons diviser le schéma de base de données en deux sous ensembles de schéma de relations, appelés respectivement types d'entités et types d'associations.

Définition II.4.1 (schéma de base de données)**.** *Soit* $\boldsymbol{R}$ *un schéma de base de données. Nous pouvons diviser l'ensemble de ses schémas de relation en deux sous-ensembles* $\mathcal{R}_e$ *et* $\mathcal{R}_a$*. Un schéma de relation* $\mathcal{R} \in \mathcal{R}_e$ *est appelé type d'entités et ne comporte aucune contrainte de référence, tandis qu'un schéma de relation* $\mathcal{R} \in \mathcal{R}_a$ *est appelé type d'association et comporte au moins une contrainte de référence. Chaque tuple d'une instance de type d'entité (resp.d'association) est appelé entité (resp.association) [5].*

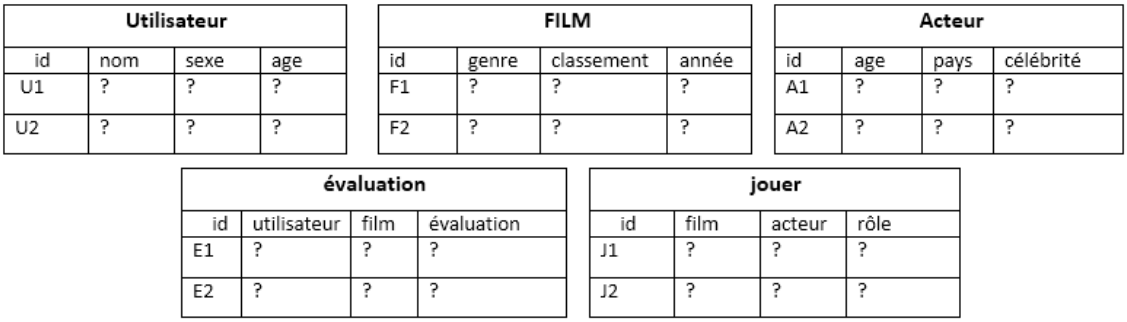
II.4.3 Incertitude de référence

Un jeu de données relationnel est composé de tuples instanciant différents schémas de relation. Dans le cas de MRP orientés attributs, tels que définis précédemment, les modèles appris prennent la forme d'un ensemble de dépendances probabilistes et de distributions de probabilités conditionnelles entre les différents attributs des différents schémas de relation de la base de données. À partir d'un tel modèle, une prédiction sur une nouvelle instance du schéma de base de données considéré se fait obligatoirement en considérant les références entre tuples de cette instance comme connaissance a priori. Les références sont dans ce contexte supposées exactes et complètes, et un MRP permet alors d'inférer les valeurs d'attributs des différents tuples en fonction des valeurs des attributs de ses tuples directement ou indirectement voisins dans le squelette relationnel de l'instance.

Il est possible d'étendre le degré d'incertitude sur les jeux de données en considérant à la fois une incertitude d'attributs, mais aussi une incertitude structurelle entre tuples, c-à-d. sur leurs associations entre eux. Une manière de modéliser cette incertitude de structure est le paradigme d'incertitude de référence. Dans ce contexte, un MRP définit des dépendances probabilistes et des distributions de probabilités conditionnelles entre les différents attributs, mais aussi entre les différentes variables de référence des différents schémas de relation de la base de données. Une prédiction à partir d'un tel modèle se fait sans connaissance préalable des valeurs de références entre tuples, ou sur une connaissance partielle de celles-ci, mais sachant toujours une énumération complète de l'ensemble des associations pour chaque type. Un objectif de prédiction peut à la fois être une valeur d'un attribut de tuple ou une valeur de référence pour une association particulière, dans le dernier cas sous la forme d'un identifiant de tuple du schéma de relation concerné.

II.4.4 Squelette objet

Squelette objet d'une instance décrit ainsi l'ensemble des entités et des associations existantes, sans toutefois connaître pour ces dernières les valeurs de leurs ré-



Utilisateur

id	nom	sexe	age
U1	?	?	?
U2	?	?	?

FILM

id	genre	classement	année
F1	?	?	?
F2	?	?	?

Acteur

id	age	pays	célébrité
A1	?	?	?
A2	?	?	?

évaluation

id	utilisateur	film	évaluation
E1	?	?	?
E2	?	?	?

jouer

id	film	acteur	rôle
J1	?	?	?
J2	?	?	?

FIGURE 7 – Exemple de squelette objet du domaine de film contenant deux attribut par table

férences, Contrairement au squelette relationnel d'une instance décrivant l'ensemble des tuples existant ainsi que les associations entre eux, où les valeurs de toutes les références sont connues, l'information disponible pour inférer dans une instance de schéma de base de données dans un contexte d'incertitude de référence est réduite.

> **Définition II.4.2.** *Le squelette objet d'une instance $\mathcal{I}$, noté $\pi(\mathcal{I})$, est l'union des restrictions des tuples de $\mathcal{I}$ à leurs seuls attributs impliqués dans une contrainte de clé primaire [5].*

un squelette relationnel décrit un hyper graphe dont les nœuds sont les entités (sans valeur pour les attributs) et où il existe un hyper arc pour chaque association reliant les entités qu'elle référence, le squelette objet décrit un ensemble de noeuds non liés entre eux.

La figure 7, page 20, présente le squelette objet d'une instance de domaine UFA, les entités et les associations, par exemple pour l'association évaluation, nous connaissons qu'il existe une liaison entre utilisateur et film, mais nous ne connaissons pas quel film associé à quel utilisateur, il conserve que les identifiants de tuple pour chaque schéma de relation.

II.4.5 Fonctions de partition

Une approche plus générique est définie dans les MRP avec incertitude de référence par l'utilisation de fonctions de partition. Le rôle d'une telle fonction est de fournir une connaissance résumée des tuples d'une relation, en les regroupant dans les mêmes parties d'une partition en fonction de leurs similarités deux à deux. Cela permet de conserver un couplage faible entre le MRP et une instance, tout en limitant la complexité amenée par ces distributions. Spécifier une fonction de partition

peut être fait de différentes façons. La façon la plus triviale peut être de définir un cluster pour chaque valeur dans le produit cartésien des domaines des attributs des tuples considérés, regroupant ceux qui sont exactement identiques à l'égard de cette description.

Un modèle probabiliste est donc défini en spécifiant une distribution sur les parties d'une partition, encodant la probabilité qu'une valeur de référence soit prise dans une partie par rapport aux autres. Une fois le groupe choisi, choisir un tuple spécifique à l'intérieur de celui-ci est réalisé selon un tirage aléatoire suivant une loi de probabilités uniforme restreinte aux individus du groupe.

II.4.6 MRP avec incertitude de référence

Un MRP avec incertitude de référence (MRP-IR) est définie par [5], de la manière suivante :

> **Définition II.4.3.** *Étant donné un schéma de base de données $\boldsymbol{R} = \langle \mathcal{R}^i \rangle_{i \leq m}$, un MRP-IR $\mathcal{M} = (\mathcal{S}, \Phi, \Theta_{\mathcal{S},\Phi})$ est composé d'une structure graphique $\mathcal{S}$, d'un ensemble de fonctions de partition Φ et d'un ensemble de paramètres $\Theta_{\mathcal{S},\Phi}$. La structure graphique est composée d'un ensemble de variables aléatoires $\mathcal{V}$ et d'une fonction de parents pa :$\mathcal{V} \rightarrow \mathcal{P}(\Psi \times \Sigma(\boldsymbol{R}) \times \mathcal{V})$ où Ψ est un ensemble de fonctions d'agrégation est un ensemble de fonctions d'agrégation (contenant la fonction d'identité id) et $\Sigma(\mathcal{R})$ est l'ensemble des chaînes de références qu'il est possible de définir sur $\boldsymbol{R}$.*
> *Pour chaque schéma de relation $\mathcal{R}^i \in \boldsymbol{R}$, nous définissons une variable aléatoire $\mathcal{A}_{i,j}$ pour chaque attribut $\mathcal{A}_j \in att(\mathcal{R}^i)$ avec $dom(A_{i,j}) = ran(A_j)$. De plus, pour chaque type d'association $\mathcal{R}^i_a \in \mathcal{R}_a \subseteq \mathcal{R}$, nous ajoutons une variable aléatoire de référence $\mathcal{R}_{i,n}$ et une variable aléatoire sélecteur $\mathcal{S}_{i,n}$ pour chaque contrainte de référence $r_n \in fk(\mathcal{R}^i_a)$ avec $ran(r_n) = \mathcal{R}^k$.*
> *Pour chaque variable aléatoire sélecteur $\mathcal{S}_{i,n}$, obtenue pour la contrainte de référence $r_n \in fk(\mathcal{R}^i_a)$ avec $ran(r_n) = \mathcal{R}^k$, nous définissons la fonction de partition $\phi_{i,n} : \mathcal{I}(\mathcal{R}^k) \rightarrow \{1, ..., c_{i,n}\}$, où $c_{i,n}$ dénote la cardinalité ou nombre de groupes de la partition, sur les tuples de $\mathcal{R}^k$ pour une instance $\mathcal{I}$. L'identité suivante doit être vérifiée : $dom(\mathcal{S}_{i,n}) = ran(\phi_{i,n}) = \{1, ..., c_{i,n}\}$. La fonction de partition possède également un ensemble d'attributs $\mathcal{A}(\phi_{i,n}) \subseteq att(\mathcal{R}^k)$ qui sont utilisés pour réaliser le partitionnement.*

La fonction pa doit respecter des contraintes pour définir un MRP-IR valide, c.-à-d. elle doit être telle que toute instanciation future du MRP-IR donnera lieu à un RB valide, donc acyclique.

II.4.7 Le projet PILGRIM

La plate-forme PILGRIM (ProbabIListic GRaphIcal Models) en cours de développement regroupe un ensemble de projets développés au sein du Laboratoire d'Informatique de Nantes Atlantique (LINA), interdépendants, extensibles, écrits en C++, ayant pour but de proposer des outils logiciels efficaces permettant de définir, réaliser de l'inférence et apprendre différents modèles graphiques probabilistes. Elle est actuellement orientée vers les modèles dirigés, en particulier les RB et les MRP ainsi que certaines des extensions de ces derniers. La plate-forme est organisée en quatre sous-projets distincts :

— PILGRIM General : propose des fonctionnalités communes à tous les autres projets, telles que la définition et la sérialisation d'un réseau bayésien, l'ac-cès à des jeux de données au format CSV, et certaines mesures comme la KL-divergence

— PILGRIM Structure Learning : propose des algorithmes d'apprentissage de structure pour les RB (p.ex. recherche gloutonne, MMHC) et des implémentations de fonctions de score (p.ex. BDeu, BIC), de tests statistiques (p.ex. information mutuelle), et de mesures pour évaluer la qualité d'une structure d'un RB appris par rapport à la structure connue du modèle d'origine (p.ex. distance structurelle de Hamming)

— PILGRIM Relational : propose de nombreuses fonctionnalités pour la définition, l'inférence et l'apprentissage dans un contexte de MRP et de leurs extensions. C'est ce projet que nous avons étendu et utilisé pour réaliser nos expérimentations durant cette thèse. C'est pourquoi nous nous focalisons dessus dans la suite de ce chapitre.

— PILGRIM Applications : contient un ensemble d'implémentations d'algorithmes dédiés à l'application des différentes briques de PILGRIM pour résoudre des problèmes. Actuellement, ce projet propose des implémentations orientées systèmes de recommandations à base de MRP

PILGRIM sera bientôt diffusée comme logiciel libre sous licence GPL.

II.5 Traitement de données textuelles

II.5.1 Lemmatisation

La lemmatisation est une analyse lexicale qui permet de regrouper les mots d'une même famille ensemble : c'est un regroupement par lemme. Chaque mot à une forme canonique (forme racine) et des formes fléchies (différentes occurrences possibles). Ces dernières sont toutes les déclinaisons qu'une entité peut prendre : verbes à l'infinitif ou conjugué, mots au singulier ou pluriel, déclinaisons masculin ou féminin, etc.

La lemmatisation est très répandue dans les moteurs de recherche. Elle permet d'étudier le contenu éditorial d'une manière plus simple (le nombre de termes est diminué) et d'améliorer la recherche d'informations (une base de données réduite permet des traitements complexes beaucoup plus rapidement).

Exemple : La lemmatisation du verbe 'positionner' (forme canonique) peut se faire par rapport à ce genre d'occurrences : positionnant, positionnait, positionnées, positions, position, positionna.

II.5.2 Stop words

Les stop words ou mots vides, sont des mots courts qui ont plus un rôle syntaxique qu'un sens en eux-même. Ce sont les articles, prépositions etc. Ils ne sont pas toujours pris en compte par les moteurs de recherche, et il peut être utile de les enlever des textes pour en simplifier la représentation.

Exemple mots vides : alors, au, aucuns, aussi, autre, avant, avec, avoir, bon, car, ce, cela, ces, ceux, chaque, ci, comme, comment...

II.5.3 Les n-grammes

la première étape dans un processus de traitement d'un gros corpus au moyen d'un outil statistique est de subdiviser le texte à traiter en plusieurs unités d'information appelées tokens qui sont, traditionnellement, des mots simples, Ce processus de tokenisation pose une question primordiale sur le plan informatique, la manière de repérer un mot,on appelle segmentation (en entités) (tokenization en anglais) le processus permettant de former des entités lexicales à partir d'un flux entrant de mots.

Le choix des n-grammes apporte un autre avantage très important : il permet de contrôler la taille de lexique et de la maintenir à un seuil raisonnable et est très utilisé pour l'analyse sémantique des textes.

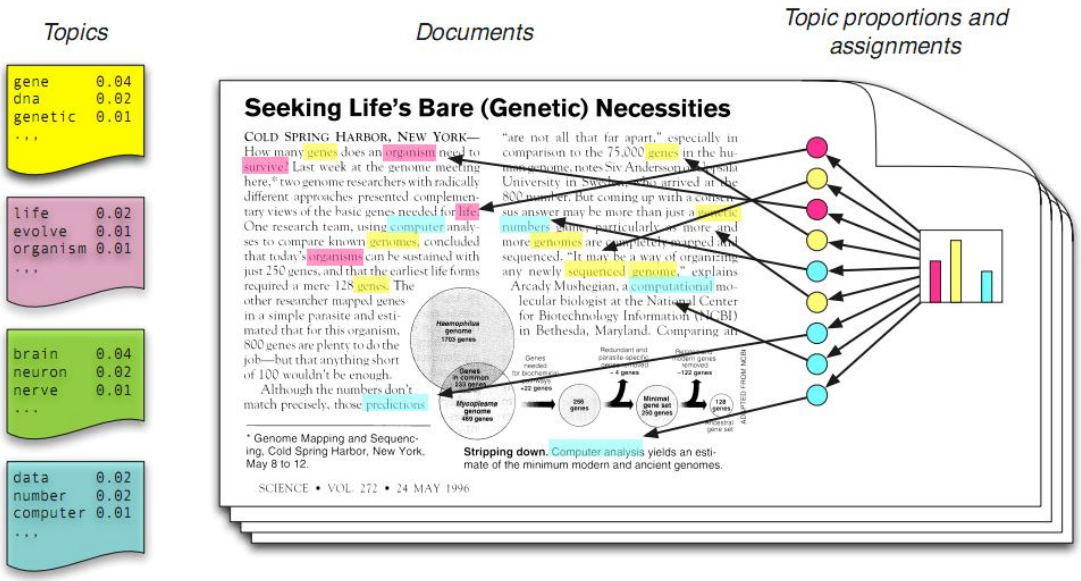


FIGURE 8 – Schéma décrivant LDA [1]. A gauche, on peut voir la structure de chaque topic, donnant une probabilité à chaque mot d'un vocabulaire fixe. Pour un document donné, l'histogramme à droite décrit la distribution de topics dans ce document. Pour chaque mot du document, on choisit d'abord un sujet depuis cette distribution (les bulles), puis on tire un mot depuis le sujet choisi.

II.5.4 Allocation de Dirichlet latente

Le modèle Latent Dirichlet Allocation (LDA) [1] est un modèle probabiliste génératif qui permet de décrire des collections de documents de texte ou d'autres types de données discrètes. LDA fait partie d'une catégorie de modèles appelés "topic models", qui cherchent à découvrir des structures thématiques cachées dans des vastes archives de documents. Ceci permet d'obtenir des méthodes efficaces pour le traitement et l'organisation des documents de ces archives : organisation automatique des documents par sujet, recherche, compréhension et analyse du texte, ou même résumé des textes. Aujourd'hui, ce genre de méthodes s'utilise fréquemment dans le web, par exemple pour analyser des ensembles d'articles d'actualité, les regrouper par sujet, faire de la recommandation d'articles, etc.

Le LDA est un modèle Bayesien hiérarchique à 3 couches (voir la Figure 8 pour une représentation graphique) : chaque document est modélisé par un mélange de topics (thèmes) qui génère ensuite chaque mot du document.

II.6 Fouille d'opinion

Le sujet initial du projet (Recommandation de solutions pertinentes) ayant dévié vers une demande plus liée à de la fouille d'opinion, il nous a paru intéressant de compléter cet état de l'art initial par des pointeurs plus généraux sur le domaine de la fouille d'opinion (Opinion mining, ou Sentiment analysis) montrant que les méthodes de Topic Modelling choisis précédemment sont très largement utilisées dans ce domaine. Le cas des textes courts (souvent le cas pour des verbatims clients) est lui aussi évoqué.

II.6.1 Généralités

La fouille d'opinion est un domaine qui a donné lieu à de très nombreux travaux ces dernières années, et à plusieurs états de l'art synthétiques [20], [15] que ce soit du côté informatique ou du côté sociologie du numérique [2]. Le cas de la fouille d'opinion dans des forums utilisateurs est aussi passé en revue dans [11].

II.6.2 LDA pour la fouille d'opinion

Les méthodes de topic modelling type LDA (et ses variantes) ont largement été utilisées dans le domaine de la fouille d'opinion, que ce soit pour analyser les avis (positifs, négatifs, ...) que pour regrouper les caractéristiques des produits à partir de ces avis. Voici une liste non exhaustive de travaux de ce type, publiés par les chercheurs de référence et/ou dans les grandes conférences du domaine ces 10 dernières années :

— Topic Sentiment Mixture : Modeling Facets and Opinions in Weblogs [16]

— Learning Document-Level Semantic Properties from Free-text Annotations [3]

— Product feature categorization with multilevel latent semantic association [9]

— An Unsupervised Aspect-Sentiment Model for Online Reviews [4]

— Opinion Digger : An Unsupervised Opinion Miner from Unstructured Product Reviews [17]

— Grouping Product Features Using Semi-Supervised Learning with Soft-Constraints [24]

— Aspect and Sentiment Unification Model for Online Review Analysis [13]

— ILDA : Interdependent LDA Model for Learning Latent Aspects and Their Ratings from Online Product Reviews [18]

— Constrained LDA for Grouping Product Features in Opinion Mining [26]

— Exploiting effective features for chinese sentiment classification [27]

— Clustering Product Features for Opinion Mining [25]

— The FLDA Model for Aspect-based Opinion Mining : Addressing the Cold Start Problem [19]

— Dynamic Joint Sentiment-topic Model [10]

— A Joint Model for Topic-sentiment Modeling from Text [6]

II.6.3 Fouille d'opinion dans des textes courts

Le cas spécifique des textes courts a aussi été abordé en fouille d'opinion :

— Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy [12]

— Opinion Target Extraction for Short Comments [23]

— Twitter Opinion Topic Model : Extracting Product Opinions from Tweets by Leveraging Hashtags and Sentiment Lexicon [14]

III Réalisation pratique

Comme évoqué en présentation du projet, section I.2, page 6, à travers les différentes discussions des forums, il existe une grande masse d'information circulant entre les deux entités "Recommandations" et "solutions" des individus. Chaque catégorie comporte un certain nombre de mots et de relations, la prédiction dans ce domaine est d'essayer de trouver un lien entre les nouvelles entités et les entités déjà existantes.

Ce problème de prédiction de lien à partir de données existantes rentre dans un domaine qui s'appelle le "relational data mining", ou le "statistical relational learning". Les modèles relationnels probabilistes sont parmi les modèles les plus pertinents pour traiter à la fois l'information relationnelle (liens existants entre les réclamations et les solutions) et l'information apportée par les "descripteurs" associés à ces entités. Nous disposons d'un problème de recommandation et de prédiction de lien pour lequel nous proposons d'utiliser modèles relationnels probabilistes avec incertitude de référence [5] présentés dans la section II.4. La mise en place d'un tel modèle nécessite une réflexion autour de deux angles décrits dans la figure 9. La première étape concerne la modélisation du problème, et tout d'abord la formalisation du schéma relationnel décrit dans la section III.1. La seconde étape concerne le choix d'une méthode de partitionnement. Le corpus fourni par eMinove ne relevant plus d'un problème de recommandation de solutions, nous avons proposé d'utiliser LDA, méthode suffisamment générique pour pouvoir être utilisée dans le problème initialement posé, et aussi dans le domaine de la fouille d'opinion dont relevait le corpus proposé. La section III.2 montre la mise en place d'une "preuve de concept", chaîne de traitement, partant des données brutes fournies, à une partie visualisation des résultats, pouvant ensuite être utilisée et déclinée pour différentes analyses. L' installation et l'utilisation des outils nécessaires à cette chaîne de traitement sont respectivement présentés dans les annexes I et II.

III.1 Schéma relationnel

L'objectif initial de ce projet était de construire un modèle de recommandation probabiliste s'appuyant sur les données des discutions des différents forum. La Figure 10 présente notre conception du schéma relationnel l'architecture globale du projet, les tables principales de notre schéma relationnel et leurs associations.

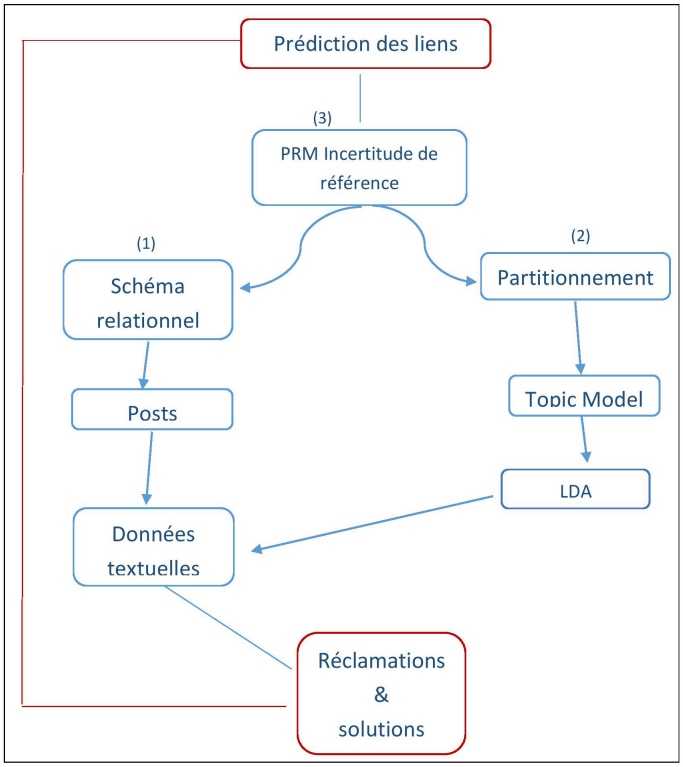


FIGURE 9 – Schéma de principe montrant le plan de la réalisation pratique pour la prédiction de liens entre réclamations et solutions à l'aide de MRP avec incertitude de référence : (1) proposition d'un schéma relationnel et (2) partitionnement de données textuelles (topic modelling) par méthodes de type LDA.

Tables

membre	les différents utilisateurs du forum
post	les réclamations postés par les membres de forum, ils peuvent contenir des tags, et c'est une suite de mots
tags	l'ensemble des tags liées à chaque post
solution	les solutions proposés par les utilisateurs pour les réclamations existantes.

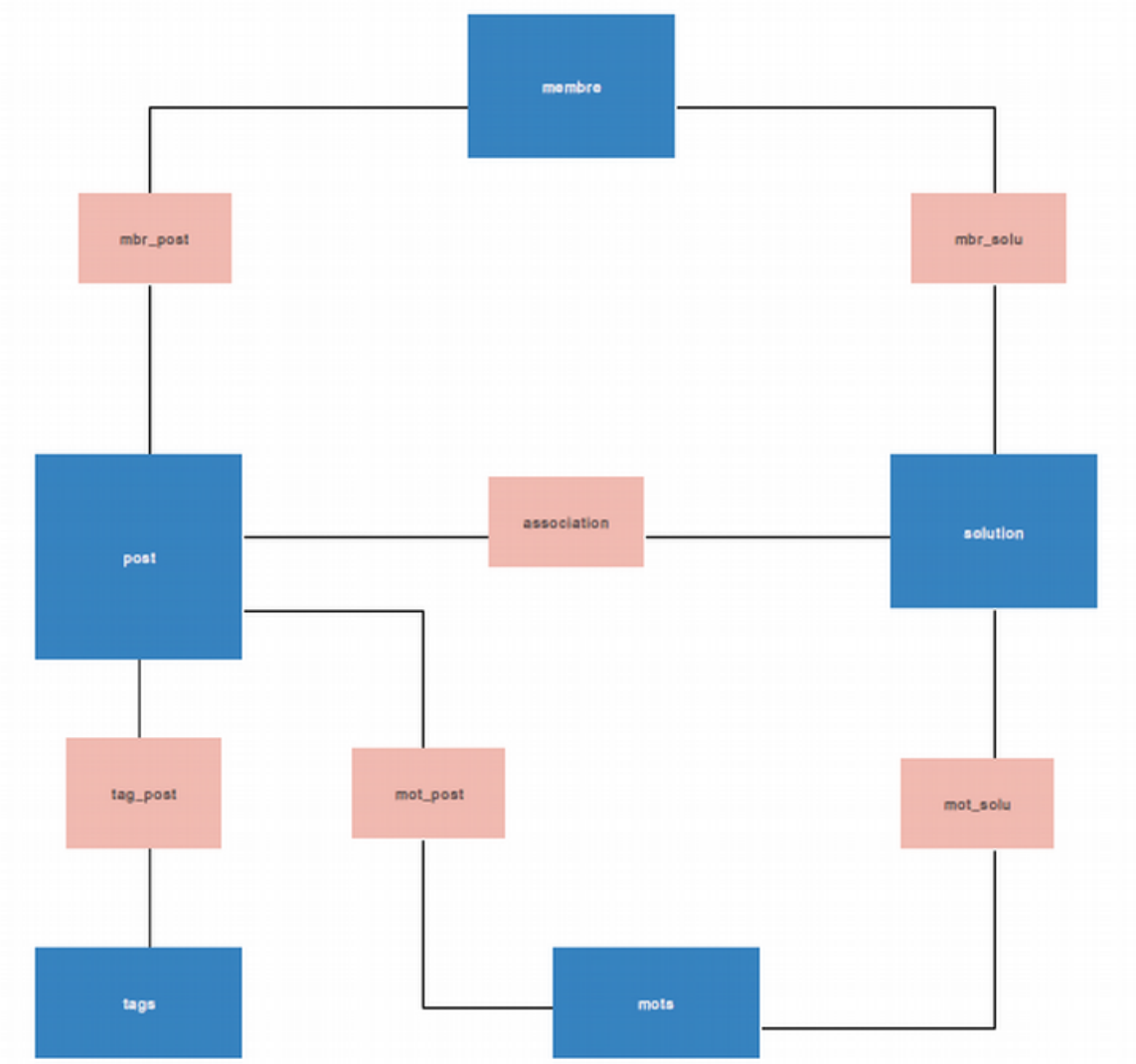


FIGURE 10 – Schéma relationnel général, reliant les tables par des associations, pour le domaine "Réclamations et solutions".

Associations	
mbr-post	association entre les posts et les membres
mbr-solu	association entre les solutions et les membres
association	association entre les posts et les solution
tag-post	association entre les posts et les tags de ce dernier
mot-post	association entre les posts et les mots qu'il contient
mot-solu	association entre les solutions et les mots qui construisent le texte.

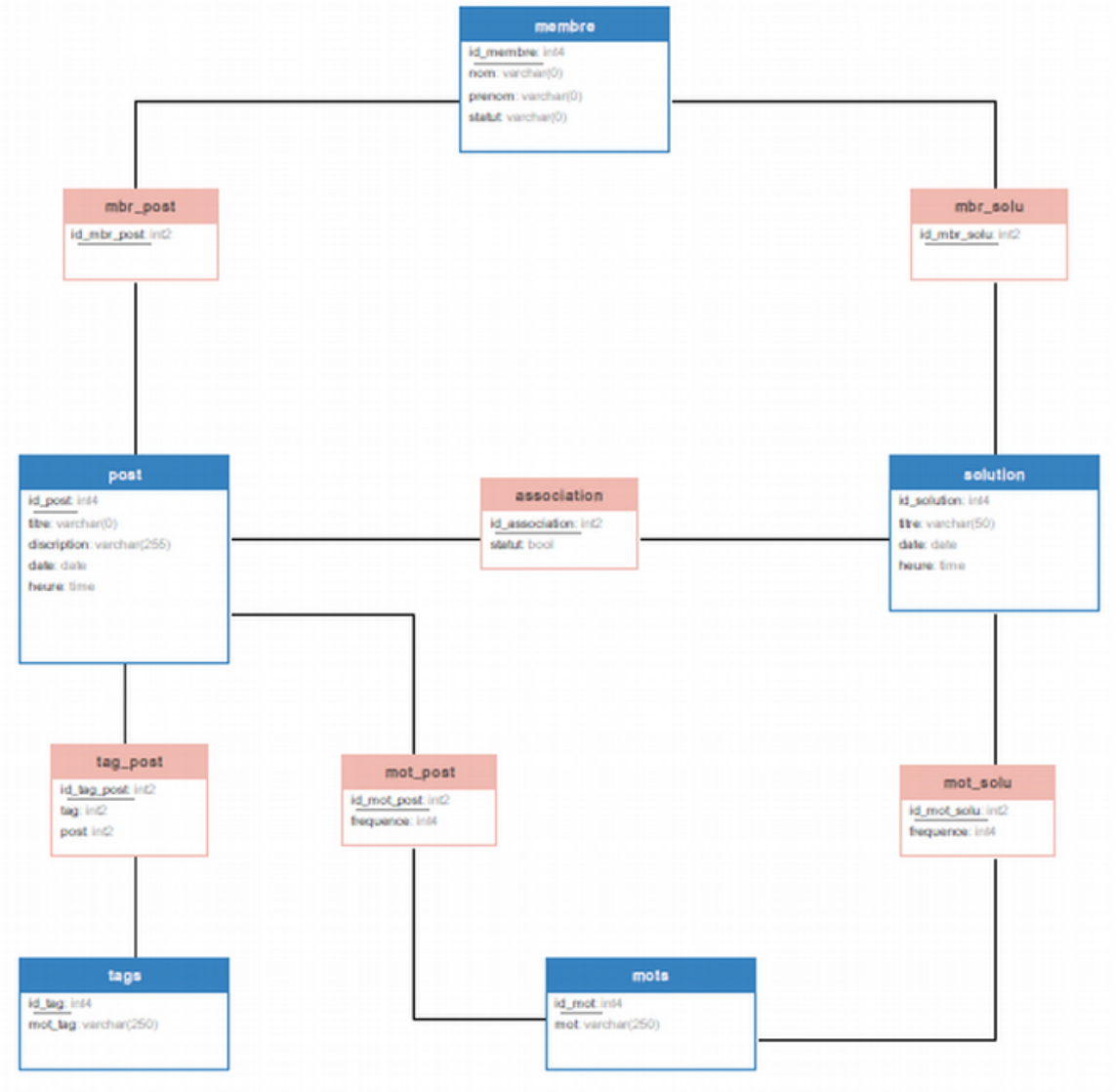


FIGURE 11 – Schéma relationnel précisant les champs de chaque table, pour le domaine "Réclamations et solutions".

Le schéma relationnel de la figure 11, page 30, met en évidence les différentes tables et les attributs de chacune, dans la présentation des tables de la figure 11, nous avons mis dans les relations les identifiants des associations, lors de la génération des bases de données relationnelles les migrations des clés étrangères, figures directement

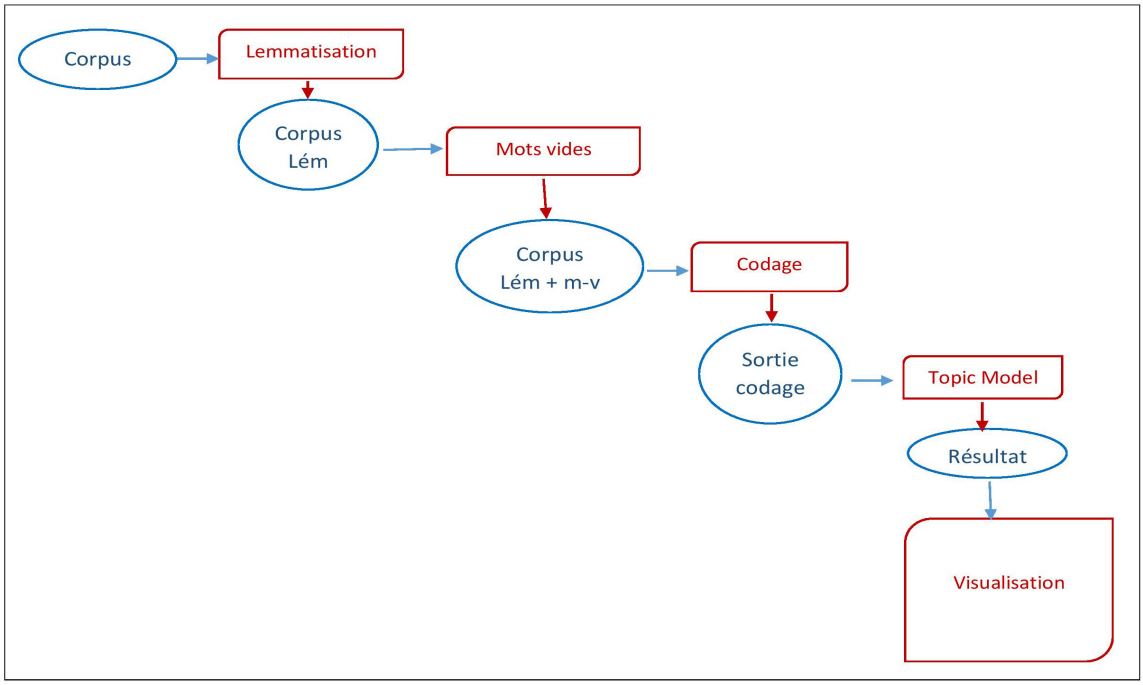


FIGURE 12 – Processus de partitionnement avec les différentes étapes de prétraitement des données textuelles fournies

sur les associations. Pour l'alimentation de la base de données relationnelle, nous utilisons les mots du texte des réclamions ainsi que celui des solutions,

III.2 Partitionnement

III.2.1 Principe général

Nous avons obtenu un corpus composé des Réclamations des différents clients d'une société sous forme de texte, nous avons choisi comme méthode de topic modeling le GibbsLDA++ [1], qui a fait référence de robustesse sur ce type de données. Nous n'avons besoin que des mots significatifs pour obtenir un résultat significatif. Avant d'appliquer le processus de LDA[1], nous avons procédé comme illustré sur le schéma 12. Le processus s'est déroulé en différentes étapes.

Lemmatisation : à l'entrée il y a notre corpus non traité, nous avons utilisé pour

1. http ://gibbslda.sourceforge.net/

la lémmatisation du texte "TreeTagger" [2] une application qui tourne sous windows, la sortie de ce dernier est peut être sous cette forme :

ai	VER :pres	avoir
trouvé	VER :pper	trouver
intéressant	ADJ	intéressant

On peut aussi être sous cette forme la seule différence, au niveau des mots "tokens" originaux au début

VER :pres	avoir
VER :pper	trouver
ADJ	intéressant

Nous avons utilisé la deuxième présentation, le corpus traité passe par un programme que nous avons développé en langage C++. L'entrée du programme un fichier contenant le corpus sous la deuxième forme, la sortie du programme est un fichier contenant le corpus lemmatisé sous la forme suivante :

pouvoir féliciter avoir temps écoute client
avoir trouver intéressant avoir point vue

Suppression des mots vides : nous avons utilisé pour la suppression des mots vides, la librairie lucene [3]en langage JAVA, à l'entrée le fichier contenant le corpus lemmatisé, le traitement se fait sur ce dernier pour enlever les mots qui n'ont pas de sens, en sortie nous avons un corpus lemmatisé sans les mots vides sous cette forme.

pouvoir féliciter temps écoute client
trouver intéressant point vue

Codage : Dans la partie codage, nous avons utilisé deux méthodes de tokenization : au début nous avons définit chaque token comme un seul mot. Comme défini pour l'entrée du modèle LDA, dans chaque ligne du fichier, il y a des mots séparés par des espaces. Le résultat donné est satisfaisant.

2. http ://www.cis.uni-muenchen.de/ schmid/tools/TreeTagger/
3. https ://lucene.apache.org/

trouver intéressant point vue

Après l'analyse des résultats, nous avons compris que la première méthode nous donne un résultat purement statistique qui ne nous permet pas de distinguer entre deux sens opposés : "*intéressant* et *pas intéressant*" ou "*intéressant et non intéressant*" vu que notre corpus passe par des étapes de nettoyage textuel, les formes de négations ne figurent pas après le traitement. Pour pouvoir creuser un peu plus dans les sens des données textuelles, et pouvoir donner un résultat sémantique plus avancé sur l'ensemble des fichiers du corpus, nous avons choisi une deuxième méthode, utiliser des N-grammes, où chaque token peut être défini par un bi-gramme.

LDA : nous avons utilisé dans cette partie le programme GibbsLDA++ [4], avec des modifications dans son code, pour pouvoir obtenir un fichier adéquat à l'entrée de la partie suivante.

Visualisation : La partie visualisation est développée en javascript avec l'utilisation de la librairie D3.JS. Ce dernier point de la chaîne de traitement nous permet de fournir une visualisation des résultats obtenus lors du processus de partitionnement. Nous avons opté pour deux types de visualisation :

Nuages de mots : façon simple de cartographier les mots essentiels, instantané visuel qui offre une forme de représentation synthétique pour faciliter l'analyse du texte.

Texte coloré : pour bien analyser chaque partie de texte,en offrant une interprétation visuelle complémentaire en proposant une représentation de tout le corpus avec une couleur pour chaque "topic".

III.2.2 Premiers exemples de résultats

La chaîne de traitement proposée est générale, est peut être exécutée en faisant varier plusieurs paramètres de LDA (dont le nombre de topics visés). La figure 13 montre un exemple de résultat de la chaine de traitement sur le corpus de verbatims fourni, pour 2 topics, avec des documents décrits par leurs mots (1-grammes).

Une analyse rapide des résultat montre que l'utilisation simple des mots ne permet pas d'illustrer le côté positif ou négatif des avis. Cela peut être amélioré à l'aide de la méthode de tokenisation n-grammes en prenant en compte les bi-grammes (n = 2), pour être capable de créer des tokens comme pas-satistfait, ou très-satisfait.

4. http ://gibbslda.sourceforge.net/

FIGURE 13 – Nuage des mots les plus pertinents pour les 2 topics obtenus par LDA à partir du texte représenté par ses 1-grammes.

TOPIC 0 TOPIC 1 TOPIC 2 TOPIC 3 TOPIC 4

[pouvoir féliciter temps écoute client]
[trouver intéressant point vue]
[présentation concerner évolution métier banque présentation permettre voir métier banque importance croissant permettre prendre conscience digital falloir calculer intérêt agio logique pur dur marge manœuvre présentation permettre comprendre banque besoin mois suite]
[avis particulier]
[sujet intéressant permettre élargir champ thème aborder cadre vie professionnel apport perceptible niveau activité direct enrichissant modifier vision concept]
[vision métier enrichissant]
[donner visibilité stratégie banque rapport passer niveau place bancaire fermeture agence banque société général]
[intéresser vision banque avenir métier]
[pertinent instructif connaître vision banque métier banque enjeu actuel futur perception banque banque mode attendre banque]
[démarche permettre foi fois sortir quotidien aperçu complémentarité métier vue client intéressant]
[démarche bon changement interlocuteur donner vision terrain]
[vision clair avenir banque niveau vulgarisation]
[prise conscience enjeu digital intéressant]
[présentation permettre comprendre enjeu banque vision global dépasser cercle intéressant culture général bancaire]
[important type événement diversifier]
[visite agence piloter innovation travailler modernisation organisation physique agence trouver intéressant moderne banquier directeur agence travailler open réunion salle pause café bureau entretien entretien client petit salon salle accueil rendre entretien rapide convivial moderne tablette écran monde sembler ravir faciliter contact relation personne directeur agence idée ramener banque cloisonner ouvrir client collègue travailler but rechercher programme]

FIGURE 14 – Corpus d'entrée, coloré en fonction des topics les plus associés aux mots du corpus.

La figure 15 nous montre les résultats obtenus par LDA sur les bi-grammes. Nous y voyons bien certains nouveaux concepts apparaître (Développement durable, très intéressant, prendre conscience, mieux comprendre, sujet intéressant, ...), mais le fait de ne traiter que les bi-grammes génère aussi des mots-vides inutiles (ne-être, être-pas, être-très, ...). Cette constatation nous a amené à construire une nouvelle façon de traiter les stop-words avec les bi-grammes, en nous basant sur [22] et de

FIGURE 15 – Nuage des mots les plus pertinents pour les 2 topics obtenus par LDA à partir du texte représenté par ses 2-grammes.

FIGURE 16 – Nuage des mots les plus pertinents pour les 2 topics obtenus par LDA à partir du texte représenté par ses 1-grammes et 2-grammes.

représenter les documents à la fois avec ces un-grammes et bi-grammes.

III.2.3 Autres exemples de résultats

Les données fournies correspondaient à des verbatims recueillis lors de 3 événements. Nous proposons ici un exemple de regroupement en topics sur l'ensemble du corpus, avec 5 topics, dans la figure 17. La figure 18 montre ensuite le regroupement en topics sur chaque événement pris séparément. Une mesure de distance entre topics (distance cosinus) montre très peu de corrélations entre les topics issus de chaque événement.

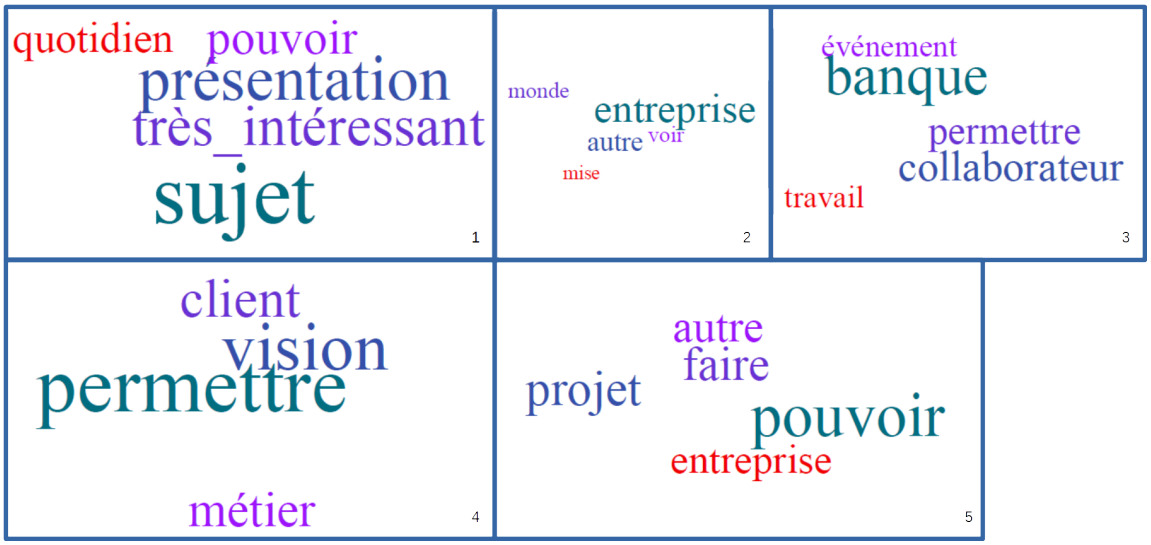


FIGURE 17 – Nuage des mots les plus pertinents pour les 5 topics obtenus par LDA.

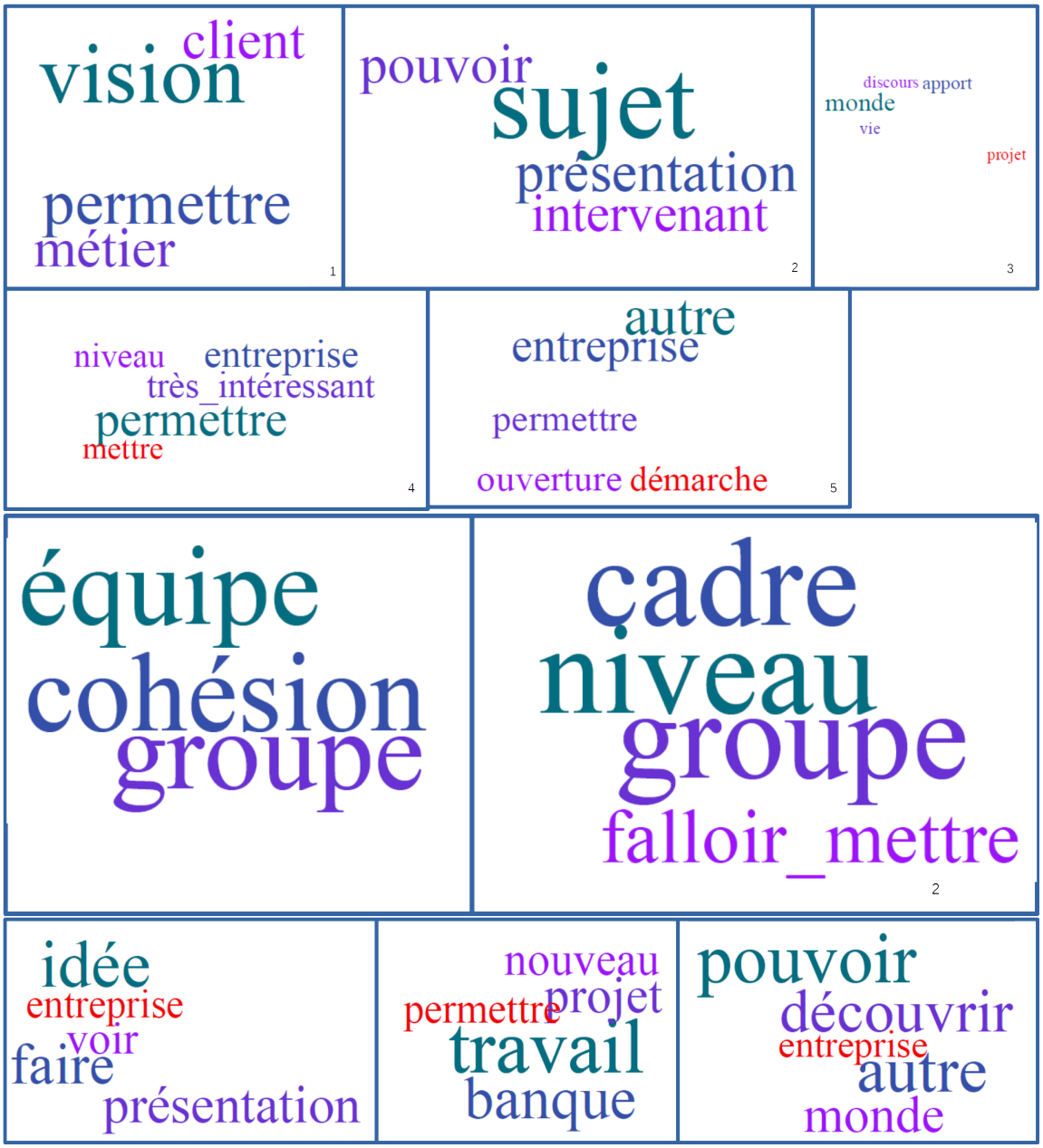


FIGURE 18 – Nuage des mots les plus pertinents pour les topics obtenus par LDA pour chacun des 3 événements

Annexes

I Guide d'installation

Nous avons utilisé un certain nombre de programmes dans la chaine de traitement proposée dans la section III.2. Dans cette annexe, nous allons parler des différents programmes utilisés, et de leur installation.

Ce document est destiné à deux type de profils :

— Profil développeur : celui qui va ajouter ou changer dans le code, avec les sections I.1 etI.2,

— Profil utilisateur : qui installera uniquement ce qui est nécessaire pour que les programmes tournent avec la section I.3.

En attachement le projet contenant tout les fichiers du projet, à télécharger et à décompresser.

I.1 Programmes à installer (développeur)

Tâche	**Programme ou IDE**	**Bibliotèque**
XSL vers CSV	Microsoft EXCEL	
Traitement Corpus	Visual studio express 2010 (c++)	
Lemmatisation	TreeTagger	
Stop-words	Eclipse (java)	Lucene-core-3.6.0.jar (java) Lucene-analyzers-3.6.0.jar (java)
LDA	Cygwin	
	GibbsLDA++	
Visualisation	Sublime text 3 (javascript)	D3.js (javascript)
	Wampserver 5.7.9	

Tâche	Programmes developpés spécifiques
Traitement Corpus	projet "Treetagger_corpus_lda" en C++
Stop-words	projet "Lucene_stop_word" en Java
	"Lucene_stop_word.jar.jar"
Visualisation	projet "Visualisation" en Javascript

Les méthodes citées en haut "Treetagger_corpus_lda" et "Lucene_stop_word, ont été développées en différents langages pour répondre rapidement à la problématique. Il faut les importer et les utiliser pour la suite du traitement, suivant le profil.

Dans la racine "C :\" il faut créer un nouveaux répertoire "Projet_eminove", et placer tous les fichiers dans ce chemin "C :\Projet_eminove\".

Ajouter ensuite les variables d'environnements correspondant aux emplacements convenus de chaque programme, pour le bon fonctionnement.
exemple : C :\Perl64\site\bin ;C :\Perl64\bin ;C :\ProgramData\Oracle\Java\javapath ; C :\Projet_eminove\treetagger\bin ;C :\Projet_eminove\cygwin64\bin ;

I.2 Installation (développeur)

Visual studio express 2010 (c++)

— Téléchargement
`https://www.microsoft.com/fr-fr/download/confirmation.aspx?id=23691`.

— Installation
- Cliquez sur le bouton Télécharger de cette page pour lancer le téléchargement du package .exe.
- Enregistrer les fichiers à télécharger sur votre ordinateur, cliquez sur Enregistrer.
- Exécutez le programme d'installation depuis l'emplacement sur votre ordinateur et suivez les instructions à l'écran.
- Importer la solution "Treetagger_corpus_lda".

TreeTagger
Treetagger fonctionne sous Windows avec le programme perl, il doit être installé sur la machine windows.

— Téléchargement TreeTagger
`http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/data/tree-tagger-windows-3.2.zip`

— Téléchargement interface TreeTagger
`http://www.smo.uhi.ac.uk/~oduibhin/oideasra/interfaces/winttinterface.htm`

— Téléchargement Perl
`http://www.activestate.com/activeperl/`

— Installation Perl
- Cliquez sur le lien ci dessus pour télécharger Perl
- Cliquez sur l'icône "Télécharger Activeperl"
- Exécutez le programme d'installation depuis l'emplacement sur votre ordinateur et suivez les instructions à l'écran.

— Installation TreeTagger
- Cliquez sur le lien TreeTagger pour télécharger "TreeTagger" et "Fichier français de paramètres (UTF-8)"
- Cliquez sur le lien Interface TreeTagger pour télécharger l'interface windows pour TreeTagger, l'interface windows pour le programme de formation TreeTagger et textfile (ttmodels.txt).
- Décompresser les fichiers dans la racine "C :\Projet_eminove\TreeTagger"
- Pour ajouter l'interface graphique, il suffit de placer les deux programmes d'interface (de wintreetagger.exe et wintraintreetagger.exe) dans le sous-répertoire bin, aux côtés des deux fichiers ".exe" à partir de la distribution TreeTagger (tree-tagger.exe et le train-arborescente-tagger.exe). Et placer le textfile (ttmodels.txt) dans le sous-répertoire \lib , aux côtés des fichiers de modèles de langue.
- decompresser le fichier "french-par-linux-3.2-utf8.bin.gz" placer le fichier "french.par" dans le sous-répertoire \lib.
Pour plus d'instructions sur l'installation, voir sur le fichier " INSTALL.txt ".

Eclipse

Eclipse est un environnement de développement (IDE) historiquement destiné au langage Java, Eclipse nécessite une machine virtuelle Java (JRE) pour fonctionner. Mais pour compiler du code Java, un kit de développement (JDK) est indispensable. JRE et JDK sont disponibles sur le site officiel d'Oracle `http://www.oracle.com/technetwork/java/javase/downloads/index.html`.

— Téléchargement Eclipse
`https://eclipse.org/downloads/`

— Téléchargement core lucene
`http://www.java2s.com/Code/Jar/l/Downloadlucenecore360jar.htm`.

— Téléchargement lucene Analyzers
`http://www.java2s.com/Code/Jar/l/Downloadluceneanalyzers360jar.htm`.

— Instructions d'installation
- Cliquez sur les liens pour lancer le téléchargement.
- Exécutez Eclipse le programme d'installation depuis l'emplacement sur votre ordinateur et suivez les instructions à l'écran.
- Pour ajouter les "jar" dans eclipse (projet => properties => java build path => Add External JARs) choisir l'emplacement des deux jars.
- Importer le projet "Lucene_stop_word", qui comporte le programme pour la partie du traitement stop-words.

Cygwin

Cygwin est un ensemble de programmes permettant d'émuler, dans une certaine mesure,un environnement linux sous windows. Il ne nécessite aucun partitionnement ou modification du système windows, c'est une couche supplémentaire qui tourne par dessus.

— Téléchargement
`https://cygwin.com/setup-x86_64.exe`

— Installation
Suivez une de ses vidéos "`https://www.youtube.com/watch?v=TjxEH_tr7e0`", ou `https://www.youtube.com/watch?v=hh-V6el8Oxk` pour une installation visuelle ou suivez les étapes suivantes :
- Cliquez sur le lien pour lancer le téléchargement du package ".exe"
- Exécutez le programme d'installation depuis l'emplacement sur votre ordinateur et suivez les instructions à l'écran.
- Trois possibilités sont offertes : a priori *"Install from Internet"* est celle qu'il vous faut. Ce choix demande à Cygwin de télécharger puis d'installer les fichiers que vous demanderez.
- À l'écran suivant, le "Root Directory" est le point de votre disque dur qui sera, plus tard, la racine (/) de votre système de fichiers cygwin. Le choix par défaut, C :\Projet_eminove\ cygwin ", est recommandé. Il est conseillé de laisser les autres options telles que recommandées, sauf si on sait ce qu'on fait...
- L'écran suivant demande le "Local Package Directory", c'est là qu'il stocke les fichiers compressés des composants qui seront installés.
- Si vous n'utilisez pas de proxy, choisissez ensuite "Direct Connection", puis sélectionnez l'un des serveurs ftp" par exemple : `http://cygwin.mirror.`

constant.com", abritant une copie des fichiers.
- Une liste de composants s'affiche, classée par thème. Développez l'arborescence pour connaître le contenu des thèmes. Seront installés ceux qui ont un numéro de version, tandis que les autres sont ignorés ("skip"). Cliquez sur "skip" pour sélectionner d'autres paquetages à installer. Si un paquetage en nécessite d'autres, ils seront sélectionnés automatiquement (par exemple, gcc-g++ nécessite gcc-core). Parmi les composants qui seront le plus intéressants et qui ne sont pas installés par défaut :
gcc-core, g++, make, vim, openssl, ftp, python, ssh
- Une fois le choix fait, appuyez sur "suivant". Les éventuelles «dépendances» supplémentaires sont signalées, acceptez leur installation et continuez.
- Une fois l'installation effectuée vous disposez d'un terminal texte sous bash, similaire à celui des stations linux.

GibbsLDA ++

GibbsLDA ++ est une implémentation C / C ++ de Latent Dirichlet Allocation (LDA) utilisant la technique de Gibbs Sampling pour l' estimation des paramètres et de l' inférence.

— Téléchargement
http://gibbslda.sourceforge.net/.

— Installation
- Cliquez sur le lien pour lancer le téléchargement du package ".exe".
- Décompresser les fichiers.
- Déplacer le dossier dans la racine de " C :\Projet_eminove\ cygwin\ home".

Wamp Server

WampServer propose aux développeurs Web un outil de déploiement local ou en ligne pour le développement de sites Internet dynamiques. Au sein de l'application, on retrouve Apache HTTP Server en tant que serveur HTTP, PHP pour le langage de script, MySQL pour le système de gestion des bases de données (SGBD) ainsi que l'application Web phpMyAdmin pour la gestion des SGBD MySQL.

— Téléchargement
http://gibbslda.sourceforge.net/.

— Installation
- Cliquez sur le lien pour lancer le téléchargement du package ".exe".
- Exécutez le programme d'installation depuis l'emplacement sur votre ordinateur et suivez les instructions à l'écran.

Sublime text
Sublime Text propose aux développeurs un éditeur de texte qui se démarque des autres par son interface et ses fonctionnalités. L'application supporte la coloration syntaxique selon les langages de programmation utilisés. Sublime Text dispose d'une interface pratique qui comprend un panel avec l'arborescence des dossiers des différentes sources éditées.

— Téléchargement
`https://www.sublimetext.com/3`

— Installation
- Cliquez sur le lien pour lancer le téléchargement du package ".exe".
- Exécutez le programme d'installation depuis l'emplacement sur votre ordinateur et suivez les instructions à l'écran.

D3.js
D3.js est une bibliothèque JavaScript pour manipuler les documents basés sur des données. Nous avons utilisé cette dernière bibliothèque pour développer la visualisation des résultats, il suffit d'avoir tout le projet web "visualisation"

— Installation :
- Décompresser les fichiers.
- Déplacer le dossier dans la racine de " C :\Wamp\www\visualisation".

I.3 Programmes à installer (utilisateur)

Pour cette partie, nous n'installons que les programmes dont nous avons besoin pour l'exécution de la chaine de traitement proposée.

Tâche	**Programme et IDE**
XSL vers CSV	Microsoft EXCEL
Lemmatisation	TreeTagger
LDA	Cygwin
	GibbsLDA++
Visualisation	Wampserver 5.7.9
	Navigateur
Tâche	**Programmes developpés spécifiquement**
Traitement Corpus	"Treetagger_corpus_lda.exe"
Stop-words	"Lucene_stop_word.jar"
Visualisation	projet "Visualisation" en Javascript

Les programmes"Treetagger_corpus_lda.exe" et "Lucene_stop_word.jar, sont des exécutables générés pour reproduire les résultats.

Dans la racine "C :\" créer un nouveau répertoire "Projet_eminove", placer tout les programmes et les fichiers dans ce chemin "C :\Projet_eminove\".

Créer un dossier, le renommer "eMinove", déplacer les deux programmes "Treetagger_corpus_lda.exe", "Lucene_stop_word.jar" avec le fichier contenant la liste des stop words "word_stop.txt" : et le corpus à traiter "corpus_treetagger.txt .

TreeTagger : cf. section précédente.

Cygwin : cf. section précédente.

GibbsLDA ++ : cf. section précédente.

Wamp Server : cf. section précédente.

II Guide d'utilisation

II.1 Utilisateur avancé

Traitement excel

Le format du fichier reçu est ".xlsx", nous utilisons le tableur Excel. Le fichier de sortie contient plusieurs lignes, ou sur chacune, un avis d'un client commençant par un point virgule (;). Afin d'arriver à cette présentation nous suivons les méthodes citées ci-dessous.

exemple :

;Intéressant de voir autre chose. Plus accès sur le Hard.
;Format intéressant d'intégrer dans le café des agilistes des retours d'expériences.

- Copier la colonne des Avis du documents fourni par l'entreprise "Eminove".
- Coller dans un nouveaux classeur excel.
- Sélectionner toute la cellule pour supprimer les retours à la ligne. Appuyer sur "ctrl+h", dans la partie "rechercher" tapez "010" en restant appuyer sur "alt", dans la partie "remplacer par", ajouter le caractère espace.
- Dans une autre colonne, concaténer le contenu de la colonne précédente avec le caractère (;). Dans une nouvelle cellule taper la formule (= ";" & numero-de-la-cellule-a-concatener), par exmeple : (= ";" & B1).
- Glisser pour toute la colonne, les changement prendrons effets sur toute la partie sélectionnée.
- Copier la nouvelle colonne, coller avec "coller spécial", pour garder que les valeurs de la colonne dans un nouveaux fichier.
- Enregistrer le nouveaux fichier en (.csv).
- Ouvrir avec avec un éditeur de texte comme "Notepad++" ou autre, chercher et remplacer le caractère (") par un caractère vide.
- Enregistrer le fichier sous le nom de "corpus.txt". Le contenu de ce dernier doit être exactement sous la forme de l'exemple cité dans l'exemple.

TreeTagger et Traitement Lemmatisation

- Dans le dossier "C :\Projet_emminove\TreeTagger" créer un nouveau dossier "file".
- Déplacer le fichier "corpus.txt" dans le répertoire "C :\Projet_emminove\TreeTagger\file"
- Dans "C :\Projet_emminove\TreeTagger\bin" exécuter "wintreetagger.exe".
- Cliquer sur "Input file", ajouter l'emplacement du fichier "C :\Projet_emminove\TreeTagger

\file\corpus.txt".
- Cliquer sur "Output file", choisir le même emplacement que le fichier d'entrer, nommer le "corpus_treetagger.txt".
- Décocher la case "the token", pour avoir un résultat de cette manière :

PUN	;
NAM	<unknown>
ADV	ne
VER :pres	pouvoir
KON	que
PRO :PER	se
VER :infi	féliciter
PRP	de
VER :infi	avoir
DET :ART	un
NOM	temps
PRP	de
NOM	écoute
PRP	de
DET :POS	notre
NOM	client
PUN	;

- Appuyer sur "Run".

Le traitement se fait correctement, le fichier de sortie se trouve dans le chemin suivant "C :\Projet_emminove\TreeTagger\file\coprus_treetagger.txt". Chaque ligne est délimitée par des séparateurs points virgules (; phrase ;).
- Ouvrer la solution "Treetagger_corpus_lda", déplacer le fichier générer "coprus_treetagger.txt".
-Exécuter La fonction $lemmatisseur\_copus\_treetagger(\text{"}coprus\_treetagger.txt\text{"},\text{";"})$, dans le programme C++. Elle contient deux paramètres : le fichier à traiter ,et le séparateur entre les phrases " ;". En sortie nous avons le fichier nommé "$file\_lemmatiser.txt$" sous cette forme.

ne pouvoir féliciter avoir temps écoute client.
avoir trouver intéressant avoir point vue .

-Pour le Bi-gramme, exécuter La fonction *corpus_bigramme*("*file_lemmatiser.txt*")", dans le programme C++. Elle prend en paramètres : le fichier "*file_lemmatiser.txt*". En sortie nous avons le fichier nommé "*corpus_bigramme.txt*" sous cette forme :

ne-pouvoir pouvoir-féliciter féliciter-avoir avoir-temps temps-écoute écoute-client. avoir-trouver trouver-intéressant intéressant-avoir avoir-point point-vue

Traitement Stop-words

Pour enlever les Stop words du fichier, vous suivez les étapes :
-Déplacer le fichier généré "*file_lemmatiser.txt*" dans la racine du programme java "Lucene_stop_word".
-Exécuter La fonction *StopAnalyzer*("*file_lemmatiser.txt*","*trndocs.dat*","*wordstop.txt*"), dans le programme "Lucene_stop_word". Elle contient trois paramètres : le fichier à traiter "file_lemmatiser.txt", le fichier de sortie "trndocs.dat" et le fichier contenant la liste des stop-words "wordstop.txt".

LDA

Pour utiliser lLDA, il faut passer par "Cygwin" pour le compiler et faire tourner le programme. Suivez les étapes pour la mise en place de ce dernier.
Dans notre modèle LDA nous avons effectué quelques changements au niveaux du code source, plus précisément dans le fichier "C :\Projet_emminove\ cygwin64\ home\GibbsLDA\models\casestudy\model.cpp", remplacer ce dernier , par celui que vous trouverez en attachement pour le bon fonctionnement.
- Déplacer le fichier "*trndocs.dat*", généré dans l'étape précédente "Traitement Stop-words" dans le chemin suivant : "C :\Projet_emminove\ cygwin64\ home\GibbsLDA\models\casestudy"".
- Dans "C :\Projet_emminove\cygwin64", exécuter le fichier "Cygwin.bat".
- Taper "make All" pour la compilation du programme.
- Tapez " Src \ lda -est -alpha 0,1 -beta 0,5 -ntopics 2 -niters 2000 -savestep 200 -twords 20 -dfile models\casestudy\trndocs.dat ".
- Après exécution, beaucoup de fichiers sont généré dans "C :\Projet_emminove\ cygwin64\ home\GibbsLDA\models\casestudy", on s'intéresse particulièrement à deux fichiers :"model-final.twords" et "model-final.tassign".

Visualisation

- Démarrer le serveur virtuel "WampServer.exe".
- Vérifier l'existence du dossier "visualisation", dans la racine du "WampServeur".(voir

le guide d'installation.)
- déplacer "model-final.twords" et "model-final.tassign" dans "C :\wamp64\www\visualisation\doc"
- Ouvrer un navigateur, taper "http ://localhost/visualisation/index.html"

II.2 Utilisateur normal

Pour une utilisation normale,et pour la reproduction de l'expérience, nous avons mis en place deux exécutables pour les programmes développées.

Traitement excel

Le format du fichier reçu est ".xlsx", nous utilisons le tableur Excel. Le fichier de sortie contient plusieurs lignes, ou sur chacune, un avis d'un client commençant par un point virgule (;). A fin d'arriver à cette présentation nous suivant les méthodes citées en dessous.

exemple :

;Intéressant de voir autre chose. Plus accès sur le Hard.
;Format intéressant d'intégrer dans le café des agilistes des retours d'expériences.

- Copier la colonne des Avis du documents fourni par l'entreprise "Eminove".
- Coller dans un nouveaux classeur excel.
- Sélectionner toute la cellule pour supprimer les retours à la ligne. Appuyer sur "ctrl+h", dans la partie "rechercher" tapez "010" en restant appuyer sur "alt", dans la partie "remplacer par", ajouter le caractère espace.
- Dans une autre colonne, concaténer le contenu de la colonne précédente avec le caractère (;). Dans une nouvelle cellule taper la formule (= " ;" & numero-de-la-cellule-a-concatener), par exmeple : (= " ;" & B1).
- Glisser pour toute la colonne, les changement prendrons effets sur toute la partie sélectionnée.
- Copier la nouvelle colonne, coller avec "coller spécial", pour garder que les valeurs de la colonne dans un nouveaux fichier.
- Enregistrer le nouveaux fichier en (.csv).
- Ouvrir avec avec un éditeur de texte comme "Notepad++" ou autre, chercher et remplacer le caractère (") par un caractère vide.
- Enregistrer le fichier sous le nom de "corpus.txt". Le contenu de ce dernier doit être exactement sous la forme de l'exemple cité dans l'exemple.

TreeTagger et Traitement Lemmatisation

- Dans le dossier "C :\Projet_emminove\TreeTagger" créer un nouveau dossier "file".
- Déplacer le fichier "corpus.txt" dans le répertoire "C :\Projet_emminove\TreeTagger\file"
- Dans "C :\Projet_emminove\TreeTagger\bin" exécuter "wintreetagger.exe".
- Cliquer sur "Input file", ajouter l'emplacement du fichier "C :\Projet_emminove\TreeTagger \file\corpus.txt".
- Cliquer sur "Output file", choisir le même emplacement que le fichier d'entrer, nommer le "corpus_treetagger.txt".
- Décocher la case "the token", pour avoir un résultat de cette manière :

PUN	;
NAM	<unknown>
ADV	ne
VER :pres	pouvoir
KON	que
PRO :PER	se
VER :infi	féliciter
PRP	de
VER :infi	avoir
DET :ART	un
NOM	temps
PRP	de
NOM	écoute
PRP	de
DET :POS	notre
NOM	client
PUN	;

- Appuyer sur "Run".

Le traitement se fait correctement, le fichier de sortie se trouve dans le chemin suivant "C :\Projet_emminove\TreeTagger\file\coprus_treetagger.txt". Chaque ligne est délimité par des séparateurs points virgules (; phrase ;).

-Déplacer le fichier générer "coprus_treetagger.txt", dans le dossier nommé "eMinove" accoté des deux programme "Treetagger_corpus_lda.exe" et "Lucene_stop_word.jar".

-Exécuter Le programme "Treetagger_corpus_lda.exe".Il y a un fichier en entrée

"corpus_treetagger.txt", le programme génère un fichier en sortie nommé "*file_lemmatiser.txt*" sous cette forme.

ne pouvoir féliciter avoir temps écoute client.
avoir trouver intéressant avoir point vue .

-Et un autre pour le Bi-gramme en sortie nous avons le fichier nommé "*corpus_bigramme.txt*" sous cette forme :

ne-pouvoir pouvoir-féliciter féliciter-avoir avoir-temps temps-écoute écoute-client.
avoir-trouver trouver-intéressant intéressant-avoir avoir-point point-vue

Traitement Stop-words

- Exécuter le programme "Lucene_stop_word.jar" :
Entrée : "*file_lemmatiser.txt*" : fichier lémmatisé et "*wordstop.txt*" : liste des stop-words
Sortie :"*trndocs.dat*" et le fichier de sortie

LDA

Pour utiliser LDA, il faut passer par "Cygwin" pour le compiler et faire tourner le programme. Suivez les étapes pour la mise en place de ce dernier.
Dans notre modèle LDA nous avons effectué quelques changements au niveau du code source, plus précisément dans le fichier "C :\Projet_emminove\ cygwin64\ home\GibbsLDA\models\casestudy\model.cpp", remplacer ce dernier , par celui que vous trouverez en attachement pour le bon fonctionnement.

- Déplacer le fichier "*trndocs.dat*", généré dans l'étape précédente "Traitement Stop-words" dans le chemin suivant : "C :\Projet_emminove\ cygwin64\ home\GibbsLDA\models\casestudy"".
- Dans "C :\Projet_emminove\cygwin64", exécuter le fichier "Cygwin.bat".
- Taper "make All" pour la compilation du programme.

- Tapez " Src \ lda -est -alpha 0,1 -beta 0,5 -ntopics 2 -niters 2000 -savestep 200 -twords 20 -dfile models\casestudy\trndocs.dat ".
- Après exécution, beaucoup de fichiers sont généré dans "C :\Projet_emminove\ cygwin64\ home\GibbsLDA\models\casestudy", on s'intéresse particulièrement à deux fichiers :"model-final.twords" et "model-final.tassign".

Visualisation

- Démarrer le serveur virtuel "WampServer.exe".
- Vérifier l'existence du dossier "visualisation", dans la racine du "WampServeur".(voir le guide d'installation.)
- déplacer "model-final.twords" et "model-final.tassign" dans "C :\wamp64\www\visualisation\doc"
- Ouvrer un navigateur, taper "http ://localhost/visualisation/index.html"

Bibliographie

[1] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3 :993–1022, 2003.

[2] Dominique Boullier and Audrey Lohard. *Opinion mining et Sentiment analysis : Méthodes et outils*. OpenEdition Press, 2012.

[3] S.R.K. Branavan, Harr Chen, Jacob Eisenstein, and Regina Barzilay. Learning document-level semantic properties from free-text annotations. In *Proceedings of ACL-08 : HLT*, pages 263–271, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[4] Samuel Brody and Noemie Elhadad. An unsupervised aspect-sentiment model for online reviews. In *Human Language Technologies : The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 804–812, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[5] Anthony Coutant. *Modèles Relationnels Probabilistes et Incertitude de Références*. PhD thesis, Université de Nantes, 2015.

[6] Mohamed Dermouche, Leila Kouas, Julien Velcin, and Sabine Loudcher. A joint model for topic-sentiment modeling from text. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, SAC '15, pages 819–824, New York, NY, USA, 2015. ACM.

[7] Lisa Getoor, Nir Friedman, Daphné Koller, and Benjamin Taskar. Apprentissage des modèles probabilistes de structure de lien. *The Journal of machine Research Learning*, 3 :679–707, 2003.

[8] Lise Getoor, Nir Friedman, Daphne Koller, and Avi Pfeffer. Learning probabilistic relational models. In *Relational data mining*, pages 307–335. Springer, 2001.

[9] Honglei Guo, Huijia Zhu, Zhili Guo, XiaoXun Zhang, and Zhong Su. Product feature categorization with multilevel latent semantic association. In *Procee-*

dings of the 18th ACM Conference on Information and Knowledge Management, CIKM '09, pages 1087–1096, New York, NY, USA, 2009. ACM.

[10] Yulan He, Chenghua Lin, Wei Gao, and Kam-Fai Wong. Dynamic joint sentiment-topic model. *ACM Trans. Intell. Syst. Technol.*, 5(1) :6 :1–6 :21, January 2014.

[11] Minqing Hu and Bing Liu. Mining opinion features in customer reviews. In *Proceedings of the 19th National Conference on Artifical Intelligence*, AAAI'04, pages 755–760. AAAI Press, 2004.

[12] Jay J. Jiang and David W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *CoRR*, cmp-lg/9709008, 1997.

[13] Yohan Jo and Alice H. Oh. Aspect and sentiment unification model for online review analysis. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 815–824, New York, NY, USA, 2011. ACM.

[14] Kar Wai Lim and Wray Buntine. Twitter opinion topic model : Extracting product opinions from tweets by leveraging hashtags and sentiment lexicon. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, pages 1319–1328, New York, NY, USA, 2014. ACM.

[15] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. Sentiment analysis algorithms and applications : A survey. *Ain Shams Engineering Journal*, 5(4) :1093 – 1113, 2014.

[16] Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. Topic sentiment mixture : Modeling facets and opinions in weblogs. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 171–180, New York, NY, USA, 2007. ACM.

[17] Samaneh Moghaddam and Martin Ester. Opinion digger : An unsupervised opinion miner from unstructured product reviews. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 1825–1828, New York, NY, USA, 2010. ACM.

[18] Samaneh Moghaddam and Martin Ester. Ilda : Interdependent lda model for learning latent aspects and their ratings from online product reviews. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 665–674, New York, NY, USA, 2011. ACM.

[19] Samaneh Moghaddam and Martin Ester. The flda model for aspect-based opinion mining : Addressing the cold start problem. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 909–918, New York, NY, USA, 2013. ACM.

[20] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2) :1–135, January 2008.

[21] Judea Pearl. Reverend bayes on inference engines : A distributed hierarchical approach. *AAAI - 82 Proceedings*, 1982.

[22] Dheeraj Rajagopal, Daniel Olsher, Erik Cambria, and Kenneth Kwok. Commonsense-based topic modeling. In *Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining*, WISDOM '13, pages 6 :1–6 :8, New York, NY, USA, 2013. ACM.

[23] Lin Shang, Haipeng Wang, Xinyu Dai, and Mengjie Zhang. Opinion target extraction for short comments. In *Proceedings of the 12th Pacific Rim International Conference on Trends in Artificial Intelligence*, PRICAI'12, pages 528–539, Berlin, Heidelberg, 2012. Springer-Verlag.

[24] Zhongwu Zhai, Bing Liu, Hua Xu, and Peifa Jia. Grouping product features using semi-supervised learning with soft-constraints. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 1272–1280, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[25] Zhongwu Zhai, Bing Liu, Hua Xu, and Peifa Jia. Clustering product features for opinion mining. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 347–354, New York, NY, USA, 2011. ACM.

[26] Zhongwu Zhai, Bing Liu, Hua Xu, and Peifa Jia. Constrained lda for grouping product features in opinion mining. In *Proceedings of the 15th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining - Volume Part I*, PAKDD'11, pages 448–459, Berlin, Heidelberg, 2011. Springer-Verlag.

[27] Zhongwu Zhai, Hua Xu, Bada Kang, and Peifa Jia. Exploiting effective features for chinese sentiment classification. *Expert Systems with Applications*, 38(8) :9139 – 9146, 2011.