Indexation et recommandation d'informations : vers une qualification précise des items par une approche ontologique, fondée sur une modélisation métier du domaine : application à la recommandation d'articles économiques

David Werner

SPIM

Thèse de Doctorat

école doctorale **sciences pour l'ingénieur et microtechniques**

UNIVERSITÉ DE BOURGOGNE

Indexation et recommandation d'informations : vers une qualification précise des items par une approche ontologique, fondée sur une modélisation métier du domaine

Application à la recommandation d'articles économiques

DAVID WERNER

SPIM

Thèse de Doctorat

école doctorale **sciences pour l'ingénieur et microtechniques**

UNIVERSITÉ DE BOURGOGNE

N° X X X

THÈSE présentée par

DAVID WERNER

pour obtenir le

Grade de Docteur de
l'Université de Bourgogne

Spécialité : **Informatique**

Indexation et recommandation d'informations : vers une qualification précise des items par une approche ontologique, fondée sur une modélisation métier du domaine

Application à la recommandation d'articles économiques

Unité de Recherche :
Équipe Checksem, Le2i, UMR CNRS 6306, Université de Bourgogne

Soutenue publiquement le Juin 2015 devant le Jury composé de :

M. JACKY AKOKA	Rapporteur	Pr., CNAM Paris
MME. CECILIA ZANNI-MERK	Rapporteur	MCF HC HDR, INSA Strasbourg
M. NUNO SILVA	Examinateur	Pr., GECAD Porto
M. CHRISTOPHE CRUZ	Directeur de thèse	MCF HDR, Univ. Bourgogne
MME. AURÉLIE BERTAUX	Encadrante de thèse	MCF, Univ. Bourgogne

REMERCIEMENTS

SOMMAIRE

1

INTRODUCTION

> L'histoire de l'humanité est habituellement décrite en termes d'âge dont les noms reflètent les âges de développement par lesquels elle a transité : l'âge de pierre, l'âge de bronze, l'âge du fer et ainsi de suite pour aboutir à l'âge industriel qui a établi les fondements de notre société moderne. Aujourd'hui, il est généralement admis que nous avons entamé une nouvelle ère, une étape postindustrielle où la capacité d'utiliser l'information est devenue décisive. Non seulement pour la production de biens, mais aussi pour les efforts qui tendent à améliorer la qualité de la vie. Ce nouvel âge est désormais nommé âge de l'information.
>
> Publicité de la firme IBM, 1977.

Nous vivons à l'âge de l'information, cette constatation date d'avant même la naissance du web. Dans cette âge, comprendre, représenter, accéder et utiliser l'information est d'une importance capitale pour l'homme, comme pour la machine chargée de l'assister.

La gestion efficace de grandes quantités d'informations est devenue un défi de plus en plus important pour les systèmes d'information. Tous les jours, de nouvelles sources d'information émergent sur le web, les boutiques en ligne comme Amazon ou Cdiscount se muent en métaboutiques recueillant de plus en plus de références. La production d'articles scientifiques n'a jamais été aussi importante et continue à croître sur les plateformes documentaires. Les sites qui proposent le visionnage de vidéos, comme YouTube ou Dailymotion, voient la quantité de vidéos disponibles s'accroître de centaines d'heures par minute. Les services d'écoute de musique en ligne comme Deezer, Spotify ou GrooveShark référencent chaque jour de nouveaux artistes tout en conservant les anciens dans leur catalogue.

En 2010, Eric Schmidt annonçait lors d'une conférence "*Tous les deux jours, nous produisons autant d'informations que nous en avons générées depuis l'aube de la civilisation jusqu'en 2003*" [**?**].

Un humain peut assez facilement retrouver ce qu'il cherche sur ces plateformes, s'il cherche un article, une vidéo, un artiste précis. Il devient par contre assez difficile, voire impossible, d'avoir une démarche exploratoire pour découvrir de nouveaux contenus. Il est totalement impossible à un humain de connaître toutes les vidéos qui peuvent l'intéresser sur une plateforme comme YouTube, car il lui faudrait tout simplement plusieurs vies pour les visionner et s'en faire une idée.

Début 2014, sur la plateforme de diffusion de musique en ligne Spotify, plus de 4 millions de morceaux n'avaient jamais été écoutés, soient 20% du catalogue disponible [**?**]. La prise de conscience de cette situation donna naissance au site web Forgotify [1] se proposant de les faire découvrir. Cela ne signifie pas qu'ils ne correspondent aux goûts d'aucun

1. http ://forgotify.com/

utilisateur, mais souligne l'incapacité de l'utilisateur seul à découvrir des titres qui ne lui ont pas été conseillés par d'autres utilisateurs, ou par un processus automatisé, dans un espace de possibilités aussi vaste.

Les systèmes de recommandation sont des outils logiciels ayant pour objectif de répondre au problème de la surcharge d'information. Ils ont pour objectif d'assister l'humain afin de lui fournir un accès plus efficace à l'information. Pour cela, ces systèmes doivent être capables de prédire l'intérêt d'un utilisateur pour une information. Ainsi ils doivent pouvoir présélectionner, dans une vaste quantité d'informations, celles qui sont pertinentes pour un utilisateur donné.

Les travaux présentés dans cette thèse portent sur un cas concret et appliqué de recommandation d'articles d'actualités économiques. Le contexte dans lequel ce travail a été réalisé est présenté dans la section suivante.

1.1/ CONTEXTE

Les travaux de thèse ont été réalisés dans le cadre d'un financement CIFRE, fruit d'une collaboration entre la société Actualis Sarl du groupe FirstECO [2] et l'équipe Checksem [3] du laboratoire LE2I [4] (i.e. Laboratoire Électronique, Informatique et Image) de l'Université de Bourgogne [5].

Le laboratoire Electronique, Informatique et Image (LE2I) est un laboratoire de recherche labellisé CNRS et adossé à l'Université de Bourgogne. L'équipe Checksem dirigée par le professeur Christophe Nicolle mène un projet transversal visant à mettre à profit les avancées dans les domaines de l'ingénierie des connaissances et du web sémantique à un large panel de domaines d'applications tels que le tourisme, la recommandation de nouvelles économiques, la criminalistique informatique ou encore la gestion du cycle de vie de patrimoine immobilier. L'objectif de l'équipe Checksem est le développement de systèmes de gestion de connaissances basés sur des ontologies et des règles métiers. Les travaux de recherche de l'équipe proposent des solutions à des verrous liés au volume et à l'hétérogénéité des données à traiter, à la contextualisation des informations, au profilage des utilisateurs et à l'analyse de données complexes.

L'entreprise Actualis Sarl partenaire et financeur de ces travaux est spécialisée dans la production et la distribution de revues de presse. Afin de rester en phase avec les tendances actuelles du marché, le processus de prise de décisions dans le domaine économique nécessite la centralisation et l'apport de grandes quantités d'informations. Pour

2. http ://www.firsteco.fr/
3. http ://checksem.u-bourgogne.fr/www/
4. http ://le2i.cnrs.fr/
5. http ://www.u-bourgogne.fr/

cela, les hommes d'affaires, les entrepreneurs, les vendeurs et responsables commerciaux doivent parfaitement connaître leur environnement. Ils doivent donc maintenir une veille économique constante facilitant l'identification des perspectives d'affaires, permettant de décrocher de nouveaux contrats. A ce titre la revue *First Eco* est un outil de veille externalisé qui fournit quotidiennement des synthèses de l'information économique régionale (e.g. investissements, recrutements, redressements judiciaires, ainsi qu'une sélection d'informations légales et d'appels d'offres). *First Eco* est un ensemble de huit revues distribuées par courriels, sous la forme de fichier PDF ou HTML. Le territoire veillé, la France, est divisé en huit régions, chacune faisant l'objet d'une revue propre. En fonction de la zone géographique souhaitée par l'utilisateur, son abonnement portera sur une à plusieurs revues régionales. Chacune des revues contient un ensemble d'articles traitant de l'actualité économique. Ces articles sont rédigés par les experts de l'entreprise suite à un processus de veille qui est détaillé en annexe **??**. Sur l'ensemble des huit régions, ce sont plusieurs centaines d'articles traitant de l'information économique locale qui sont produits tous les jours par l'entreprise, et ce, pour plusieurs milliers de lecteurs.

L'étude de l'offre de service de l'entreprise a permis de mettre en avant certaines limites qui font l'objet de la section suivante.

1.2/ VERROUS

L'objectif de la collaboration entre l'équipe de recherche et l'entreprise est la mise au point d'un nouveau produit. En effet, l'étude de la revue *First Eco*, produit proposé par l'entreprise éponyme depuis 2001 a permis de mettre en avant trois principales limites :

- La limite **structurelle**. En effet, la revue possède une structure fixe composée d'articles. Les documentalistes rédigent des articles compilés sous la forme d'une page web statique ou d'un document PDF. Avec la revue classique la France est divisée en huit zones. Une revue propre à chacune de ces zones est produite tous les jours. Les lecteurs peuvent veiller une ou plusieurs de ces zones et ainsi recevoir plusieurs revues. Tous les utilisateurs ayant souscrit aux informations d'une région reçoivent exactement les mêmes informations. Pourtant les besoins des lecteurs sont très divers, certains ont des besoins très spécifiques voire uniques. Le lecteur est donc obligé de filtrer la surcharge d'informations qui lui sont fournies par l'entreprise, ce qui s'avère être une tache longue et fastidieuse. Une enquête auprès des clients (cf. annexe **??**) a montré que les lecteurs n'ont que peu de temps à consacrer à cette veille bien qu'elle soit stratégiquement décisive, il arrive qu'ils passent à côté d'informations importantes par manque de temps.
- La limite **sémantique**. Aucune information ou métadonnée sur les revues ou sur

les articles n'est conservée par le système. Les informations ne sont ni indexées, ni qualifiées de façon à faciliter leur exploitation ultérieure tant par la machine que par l'humain.

- La limite **pragmatique**. La revue *First Eco* telle qu'elle a été réalisée ne permet pas d'obtenir de retours implicites ou d'analyses vis-à-vis des utilisateurs. En effet, il n'est pas possible de savoir quel usage est fait de la revue par l'abonné (e.g. la revue est-elle consultée tous les jours, quels sont les articles lus, combien d'articles sont lus, combien de temps l'utilisateur consacre-t-il à la revue, etc ... ?). L'entreprise doit avoir recours à des enquêtes, auxquelles il est difficile de faire répondre les clients. Ceux-ci n'ayant pas ou peu de temps à leur consacrer. De plus, les enquêtes ne permettent pas de donner une vision des cas particuliers de chaque utilisateur. En profilant l'abonné par rapport à son comportement de lecture, il serait alors possible de lui fournir une information plus pertinente et personnalisée.

Afin de proposer un nouveau produit, hautement compétitif, ces limites doivent être comblées. Ce nouveau produit doit permettre de fournir la bonne information à la bonne personne de façon rapide et automatique.

1.3/ CONTRIBUTIONS

Afin de répondre aux différentes limites mises en avant précédemment, la solution envisagée consiste à mettre au point un système de recommandation. Ces systèmes fournissent des items aux utilisateurs en fonction de leurs besoins. Ainsi, dans le cas de l'entreprise Actualis et de sa revue FirstEco, des articles contenant des informations économiques sont fournis aux lecteurs de la revue de façon quotidienne par le système. Celui-ci proposant à chaque utilisateur une liste d'articles organisée en fonction de leurs correspondances avec les attentes de l'utilisateur.

La contribution principale de ce travail est la mise en place d'une architecture complète permettant la recommandation d'articles économiques aux clients de l'entreprise. Ce travail a permis à l'entreprise partenaire, First ECO, de commercialiser un nouveau produit hautement compétitif, *FristECO Pro'fil*. Au cours de ce travail, l'étude des systèmes de recommandation ainsi que des sous-processus et outils qui les composent a mis en lumière trois principaux éléments de l'architecture sur lesquels nous avons porté notre attention.

L'**architecture** mise en place repose sur une **base de connaissances** contenant une modélisation du domaine traité. Cette modélisation prend la forme d'un vocabulaire contrôlé, structuré et formel. Notre première contribution consiste à proposer un modèle pour la base de connaissances du système de recommandation. Celui-ci répond à différentes contraintes. (i) Il permet la description des items à l'aide de facettes de description. Ce

qui a pour avantage de permettre la gestion de domaines complexes tout en conservant des descriptions aisément accessibles pour un humain. Chaque facette est composée d'une ressource terminologique de type, liste, taxonomie ou thésaurus. Ces ressources ont en premier lieu été pensées pour une utilisation humaine, et sont donc facilement accessibles pour des humains. Le modèle permet la réutilisation de ressources terminologiques existantes. (ii) Ce modèle repose sur une ontologie, il est donc formel et manipulable par la machine. Il repose sur les logiques de description ce qui permet l'utilisation de contraintes logiques. (iii) Ce modèle gère chacune des ressources terminologiques selon le principe de l'abstraction conceptuelle, ce qui permet de faciliter la maintenance et l'évolution du vocabulaire qu'il contient. (iv) Lors de l'utilisation de ressources terminologiques simples, comme des listes, le modèle permet facilement d'en augmenter l'expressivité. Il permet par exemple, l'ajout de contraintes ou la réorganisation hiérarchique des termes.

L'**indexation** des articles et des profils c'est-à-dire la description du besoin et de l'offre d'information repose sur le vocabulaire de la base de connaissances. Notre seconde contribution consiste à proposer une approche pour l'automatisation de l'indexation des items sur la base de vocabulaires contrôlés et structurés. Notre approche considère l'indexation comme une tâche d'indexation multi-label (ou multi-label hiérarchique dans le cas de vocabulaires d'indexation organisés de façon hiérarchique). L'indexation multi-label nécessite l'apprentissage d'un modèle prédictif. La base de connaissances de notre système étant une ontologie, nous avons considéré le modèle prédictif comme une connaissance à intégrer à cette base. Celui-ci est donc traduit sous la forme de contraintes logiques intégrables dans l'ontologie. L'indexation est le résultat d'un processus d'inférence logique produit par des raisonneurs. A notre connaissance aucun travail similaire n'a été réalisé. Nous avons démontré la faisabilité de cette approche dont le but est de conserver le modèle de prédiction au plus près de la modélisation du domaine des experts. C'est une première étape vers des systèmes plus évolutifs, à même de gérer efficacement l'évolution du vocabulaire d'indexation et ses répercussions sur le processus d'indexation.

La **recommandation** consiste en la comparaison des descriptions du besoin et de l'offre d'information. Notre troisième contribution consiste à proposer un algorithme de comparaison qui exploite pleinement la description des items. La base de connaissances du système permet la description d'items sur la base de vocabulaires contrôlés et structurés. Ainsi les vocabulaires utilisés peuvent avoir une structure hiérarchique qui permet une description riche des items. Ceux-ci peuvent donc être décrits à l'aide de termes plus ou moins précis. Les algorithmes de comparaisons classiques déduisent directement la pertinence de la précision et ne peuvent donc pas prendre en compte la différence de précision qu'il peut y avoir entre l'expression du besoin de celle de l'offre d'information. Notre algorithme comble un manque de l'état de l'art en ce qui concerne la prise en compte du degré de précision de la description des items.

1.4/ ORGANISATION DU DOCUMENT

Cette section présente une brève vue d'ensemble de chacun des chapitres composant cette thèse.

1.4.1/ CHAPITRE 2 : ÉTAT DE L'ART

Ce chapitre a pour objectif de présenter une étude des systèmes de recommandation ainsi que des sous-processus et outils qui les composent. Il présente des approches et solutions existantes et met en avant leurs avantages et limites quant à l'objectif qui nous importe, c'est-à-dire, adapter la distribution des informations produites par l'entreprise aux besoins des utilisateurs. Nous présentons donc dans un premier temps, les systèmes de recommandation d'articles d'actualité, ou proches de cette problématique. Cette étude démontre l'intérêt des systèmes basés sur le contenu et du cas particulier des systèmes basés sur la sémantique. En effet, les systèmes basés sur la sémantique permettent de conserver une forte adéquation avec l'humain et le savoir-faire métier des experts. Les systèmes basés sur la sémantique utilisent une base de connaissances. Nous introduisons donc à la suite, la notion de vocabulaires contrôlés. Ces vocabulaires, contenus dans la base de connaissances constituent le référentiel à partir duquel la description des items nécessaire au processus de recommandation est réalisée. Enfin, les processus d'apprentissage et de classification provenant du domaine de l'intelligence artificielle sont présentés. En effet, les cas particuliers de classification multi-label et multi-label hiérarchique correspondent à la problématique d'indexation d'items sur la base d'un vocabulaire contrôlé et éventuellement structuré de façon hiérarchique (i.e. type de vocabulaire contenu dans la base de connaissances et correspondant au savoir-faire métier des experts).

1.4.2/ CHAPITRE 3 : MODÉLISATION DE CONNAISSANCES POUR LA DESCRIPTION DIMENSIONNELLE D'ITEMS

Ce chapitre propose un modèle permettant de structurer la base de connaissances d'un système de recommandation sémantique. Ce modèle permet de répondre aux objectifs du système ainsi qu'aux verrous mis en lumière par l'état de l'art. Ainsi, un premier modèle, le modèle unificateur est proposé. Ce modèle permet de gérer l'hétérogénéité des langages documentaires. Un second modèle, le modèle intégrateur permet de proposer une base d'indexation riche, unifiée et contrôlée, en intégrant les vocabulaires unifiés par le modèle unificateur. Ce référentiel d'indexation (i.e. base de connaissances reposant sur le modèle intégrateur) est facilement accessible aux humains tout en étant manipulable par la machine, ce qui facilite leurs interactions nécessaires au fonctionnement du

système.

1.4.3/ CHAPITRE 4 : LES PROCESSUS D'INDEXATION

Ce chapitre présente les processus d'indexation des articles et profils. Les descriptions créées reposent sur un vocabulaire d'indexation contrôlé et structuré. Après présentation des approches manuelles et supervisées, nous proposons une méthode nouvelle d'indexation automatisée. Celle-ci considère l'indexation comme un processus de classification multi-label. Notre approche est novatrice, car elle consiste à intégrer au sein de la base de connaissances gérant le vocabulaire d'indexation du système, un modèle prédictif. Ce modèle prend la forme de contraintes logiques utilisables par des raisonneurs. La base de connaissance est une ontologie reposant sur le modèle intégrateur défini dans le chapitre **??**, ce qui permet l'utilisation de contraintes logiques. L'indexation est ainsi inférée par le raisonneur à partir des connaissances de la base. Le processus d'indexation se fait donc au plus près de la modélisation du domaine faite par les experts, afin de faciliter la supervision. Cette approche est une première étape vers des systèmes à même de gérer efficacement l'évolution du vocabulaire d'indexation et ses répercussions sur le processus d'indexation.

1.4.4/ CHAPITRE 5 : LES PROCESSUS DE RECOMMANDATION

Ce chapitre présente une méthode d'évaluation de la pertinence, *PEnSIVE*. Cette méthode permet de prendre en compte de façon efficace les connaissances du domaine en vue de fournir une recommandation de qualité. Les chapitres précédents ont présenté la base de connaissances du système ainsi que les processus permettant de produire une description des items à l'aide du vocabulaire contenu dans cette base de connaissances. Le modèle intégrateur présenté dans le chapitre **??**, sur lequel repose la base de connaissances permet l'utilisation de vocabulaires riches et structurés. La structuration principalement hiérarchique des vocabulaires permet de prendre en compte la précision du besoin exprimé ou de l'information fournie. *PEnSIVE* est une méthode d'évaluation de la pertinence qui, contrairement à la simple évaluation de la similarité, prend en compte la différence de précision pouvant exister entre l'expression du besoin et celle de l'offre d'information proposée par un document.

1.4.5/ CHAPITRE 6 : IMPLÉMENTATION

Ce chapitre présente l'implémentation et l'architecture de la solution développée ayant fait l'objet des chapitres précédents. Il montre comment cette architecture répond aux limites mises en avant dans l'ancienne revue, *FirstECO* tout en prenant en compte un

certain nombre de nouvelles contraintes afin de donner naissance au nouveau produit de l'entreprise partenaire, *FirstECO Pro'fil*.

1.4.6/ CHAPITRE 7 : CONCLUSION

Ce chapitre présente la conclusion du document. Il synthétise les apports. Il présente les travaux en cours, deux thèses faisant suite aux travaux présentés dans ce document. Il propose de futurs travaux, pouvant se baser eux aussi sur le travail réalisé durant cette thèse et en particulier sur les possibilités offertes par l'architecture mise en place avec l'entreprise.

2

ÉTAT DE L'ART

Ce chapitre présente des approches et solutions existantes et met en avant leurs avantages et limites quant à l'objectif qui nous importe, c'est-à-dire, adapter la distribution des informations produites par l'entreprise aux besoins des utilisateurs. Dans un premier temps nous présentons, les systèmes de recommandation d'articles d'actualité, ou proche de cette problématique. Cette étude démontre l'intérêt des systèmes basés sur le contenu et du cas particulier des systèmes basés sur la sémantique, car ils permettent de conserver une forte adéquation avec l'humain et le savoir-faire métier des experts. Les systèmes basés sur la sémantique utilisent une base de connaissances. Nous introduisons donc à la suite, les notions de vocabulaire contrôlé. Ces vocabulaires contenus dans la base de connaissances constituent le référentiel à partir duquel la description des items nécessaires au processus de recommandation est réalisée. Enfin, les processus d'apprentissage et de classification provenant du domaine de l'intelligence artificielle sont présentés. En effet, les cas particuliers de classification multi-label et multi-label hiérarchique correspondent à la problématique d'indexation (i.e. description) d'items sur la base d'un vocabulaire contrôlé et éventuellement structuré de façon hiérarchique.

Ce chapitre a pour objectif de présenter les approches et solutions existantes en terme de recommandation d'informations, ainsi que leurs limites par rapport à nos besoins. Notre besoin consiste à adapter la distribution des informations produites par l'entreprise (i.e. articles de veille économique) aux besoins des utilisateurs (i.e. lecteurs de la revue produite par l'entreprise). Notre objectif est donc de filtrer l'information disponible chaque jour afin de fournir une revue personnelle pour chaque lecteur. Le principal problème que nous cherchons à résoudre est la surcharge cognitive que peut produire la revue classique pour les clients.

Dans un premier temps nous présentons deux catégories bien connues de systèmes existants dans le cadre de la recommandation d'articles d'actualité ou proche de cette problématique. Dans chacune de ces catégories, nous mettons en avant les avantages et inconvénients des solutions. Nous concluons ce premier état de l'art par une réflexion sur la recommandation de nouvelles de façon générale afin de mettre en perspective ces systèmes avec le cas précis qui nous importe. Dans un second temps, nous introduisons la notion de vocabulaire d'indexation. L'objectif étant de permettant l'indexation d'informations sur la base d'un référentiel commun entre différents acteurs et processus intervenant lors de la recommandation. Dans un troisième temps, nous nous intéressons à l'automatisation de l'indexation à l'aide de processus d'apprentissage et de classification provenant du domaine de l'intelligence artificielle et en particulier, à l'indexation multi-label hiérarchique.

2.1/ SYSTÈMES DE RECOMMANDATION

Les systèmes de recommandation ont pour objectif de suggérer des items aux utilisateurs en accord avec leurs goûts ou leurs besoins. Un item peut être un morceau de musique, une vidéo, ou un bien de consommation matériel comme un livre. Ces systèmes permettent de guider l'utilisateur dans un processus de prise de décision. Quels livres acheter ? Quelles vidéos regarder ? Quelles informations lire ? etc. Contrairement aux systèmes de recherche d'informations, l'utilisateur n'est pas forcément dans une démarche active de recherche. Sur les plateformes de musique, ou d'actualités, l'utilisateur ne sait souvent pas définir ce qu'il cherche. En revanche, quand le système fait des propositions, l'utilisateur sait reconnaître si cela correspond ou non à ce qu'il aime ou à ce dont il a besoin [**?**].

Les systèmes de recommandation sont devenus un domaine de recherche important ces vingt dernières années au point que d'importantes conférences en recherche d'informations comme ACM SIGIR ou en datamining ACM SIGKDD l'ont incorporé dans les problématiques traitées. La conférence ACM RecSys (i.e. Recommender Systems), créée en 2007, s'est rapidement imposée et est une des premières conférences totalement dé-

diées au sujet. La définition et l'amélioration des systèmes de recommandation est donc un domaine de recherche relativement récent par rapport aux systèmes de recherche d'informations qui remontent au début de l'informatique [**?**].

La première idée de recommandation par ordinateur remonte à la création d'un bibliothécaire électronique en 1979, [**?**]. Ce système fut un premier pas vers les systèmes de recommandation. Mais, il est généralement considéré dans la littérature que ces systèmes sont nés dans les années 90, afin de filtrer les informations des groupes de discussions (i.e. newsgroup) [**?**] [**?**] [**?**]. Ensuite, ils se sont diversifiés, afin d'être appliqués à différents domaines. La montée en puissance des plateformes de e-commerce, puis des services de musiques et de vidéos à la demande ont mis en évidence leur nécessité et les a popularisé.

Le principe des premiers systèmes était assez simple. Dans notre vie de tous les jours, nous sommes souvent aidés, guidés par d'autres individus. Nous donnons ou suivons les conseils d'autres personnes, amis, famille, ou personnes de référence comme, par exemple, des critiques de presse. Nous nous basons sur leurs avis et conseils pour choisir quel livre nous allons acheter, ou quel film choisir au cinéma [**?**] [**?**]. Eux-mêmes vont alors se baser sur leurs propres goûts pour nous guider, ce qui est le cas des critiques de presse. Les personnes qui nous connaissent le mieux seront également influencées dans leur processus de conseil par ce qu'elles pensent avoir compris de nos goûts. Les premiers systèmes ont cherché à mimer ce comportement humain et à l'automatiser [**?**]. Ces systèmes sont parfois qualifiés de sociaux, nous les nommerons systèmes de *filtrage collaboratif* dans la suite de ce document.

L'objectif principal des systèmes de recommandation consiste à gérer la surcharge d'informations. Beaucoup de ces systèmes ont été mis en place avec succès ces dix dernières années, notamment sur les plateformes de e-commerce. Ils permettent à un utilisateur de filtrer l'information et donc d'évacuer les informations inutiles. De plus, elles permettent aux entreprises qui les mettent en place de mieux connaître et d'évaluer l'intérêt d'un utilisateur pour un produit (i.e. item) donné [**?**].

Entre 2006 et 2009, l'entreprise Netflix [1], dont la plateforme de vidéos à la demande éponyme est la plus importante au monde a lancé une compétition qui avait pour objectif d'améliorer l'algorithme de recommandation utilisé par l'entreprise sur sa plateforme. Le site de e-commerce Amazon [2] utilise, pour sa part, un système de recommandation afin de personnaliser la boutique en ligne en fonction de l'utilisateur connecté [**?**]. Quotidiennement une grande partie des sites internet les plus importants, utilisés par des millions d'utilisateurs, exploitent une forme de système de recommandation plus ou moins visible par l'utilisateur.

1. https ://www.netflix.com/fr/
2. https ://www.amazon.fr/

La littérature distingue généralement trois types de systèmes de recommandation, les systèmes basés sur le contenu, les systèmes à base sociale dits de filtrage collaboratif et les systèmes hybrides [**?**].

Les systèmes basés sur le contenu recommandent les items en se basant sur une description de chacun d'eux. La façon de créer cette description change en fonction du système, souvent influencée par le type d'items recommandés. Par exemple, IMDB [3] propose de décrire un film à partir de certaines informations comme l'auteur, le genre, les acteurs, etc. Pour les items de type texte, comme des articles scientifiques, une représentation à l'aide de mots-clés contenus dans le texte est souvent utilisée. Les descriptions d'items sont comparées avec les descriptions d'utilisateur, ou profils. Cette comparaison permet d'évaluer si l'item correspond au besoin de l'utilisateur. Le profil est généralement vu comme un item idéal correspondant parfaitement aux besoins ou goûts de l'utilisateur. Les profils utilisateurs sont, soit définis manuellement par l'utilisateur [**?**], soit appris en fonction de son comportement [**?**].

Les systèmes de filtrage collaboratifs ne prennent pas en compte le contenu des items auxquels l'utilisateur a accès, mais se basent sur les opinions et les goûts des utilisateurs pairs pour générer une recommandation. A l'origine, cette méthode se référait à une méthode de filtrage d'informations basé sur les préférences exprimées par des utilisateurs sur des items. Un système de filtrage collaboratif est un système dans lequel les items ne sont pas analysés. Le premier et le plus connu est GroupLens [**?**] qui utilise une approche basée sur le voisinage. Le but est de proposer de nouveaux items à un utilisateur particulier en fonction de ce qu'il a aimé par le passé ainsi que de l'opinion d'autres utilisateurs. C'est une méthode de recommandation très largement implémentée sur les sites de e-commerce car il possèdent une large base d'utilisateurs. L'opinion de l'utilisateur peut être soit donnée de façon explicite par l'utilisateur, soit déduite implicitement de son historique de navigation ou d'achats dans la cas de sites de e-commerce. Le système va tenter de créer des groupes d'utilisateurs similaires, ou de chercher les utilisateurs les plus proches.

Dans cette introduction, nous avons présenté quelques notions et concepts du domaine. Nous avons introduits les deux principales approches que nous détaillerons dans les sections suivantes.

2.1.1/ LES SYSTÈMES BASÉS SUR LE CONTENU

Dans l'objectif d'orienter l'utilisateur au travers d'une vaste quantité d'informations disponibles, les systèmes basés sur le contenu proposent à l'utilisateur des items similaires aux items qu'ils ont aimés par le passé. Les items évalués positivement par l'utilisateur,

3. http ://www.imdb.com/,

de façon explicite ou implicite, sont utilisés par le système afin de comprendre ce qui intéresse l'utilisateur. La représentation des besoins et préférences de chaque utilisateur du système est généralement nommée *profil*. Afin d'effectuer une recommandation, une comparaison a donc lieu entre la description de l'intérêt de l'utilisateur, c'est-à-dire le profil et celles du contenu des items. L'objectif étant de proposer à l'utilisateur de nouveaux items en accord avec ses préférences.

D'après Baeza-Yates et Ribeiro-Neto [**?**], les systèmes de recommandation basés sur le contenu sont à l'intersection des domaines des systèmes de recherche d'informations et de l'intelligence artificielle. D'autres publications mettent en avant la proximité qui existe entre les systèmes de recommandation et les systèmes de recherche d'informations [**?**] [**?**].Les systèmes de recherche d'informations comparent également une représentation du besoin de l'utilisateur à une représentation du contenu des documents. Dans ces systèmes, l'utilisateur est en phase active de recherche, il exprime son besoin ponctuel, sous la forme de ce qui est appelé une *requête*. Le profil, quant à lui, est une représentation de ce qui intéresse l'utilisateur sur le long terme. Cette démarche est souvent passive, c'est le système qui déduit, qui apprend le profil et non l'utilisateur qui le fournit. Le fait que beaucoup de systèmes apprennent le besoin de l'utilisateur en fonction de son comportement montre bien la proximité avec l'intelligence artificielle. En effet, la recommandation peut être vue comme un problème de classification (i.e. labellisation). Une approche basée sur l'apprentissage artificiel peut être utilisée pour générer un modèle prédictif permettant de décider du degré de pertinence d'un item donné pour l'utilisateur (cf. section **??**). Le modèle prédictif est une fonction qui permet de faire le lien entre un ensemble de données d'entrée et un domaine de valeurs réelles ou un ensemble de classes prédéfinies [**?**].

Dans la suite de cette section, nous présentons les approches par vecteurs de mots-clés, puis les approches utilisant des connaissances externes au système. Influencés par les méthodes venues du domaine du web sémantique, les systèmes modernes tendent à prendre en compte la sémantique afin de pallier certains manques des approches par mots-clés.

2.1.1.1/ LES SYSTÈMES BASÉS SUR DES VECTEURS DE MOTS-CLÉS

Dans les systèmes basés sur le contenu, les items sont représentés par un ensemble de descripteurs. Ces descripteurs sont très souvent textuels, et ce, même pour des items qui ne le sont pas. Ainsi une vidéo, une image (e.g. photo) ou un son (e.g. morceau de musique) peuvent être décrits à l'aide de mots-clés. Ces termes sont extraits des commentaires des utilisateurs, d'un texte explicatif associé, des métadonnées (i.e.labels ou tags) associés par l'utilisateur ayant mis l'item à disposition ou par d'autres utilisateurs

ayant eu accès à l'item, etc. Les profils étant généralement définis comme des items idéaux, la même modélisation ainsi que les mêmes descripteurs sont utilisés pour leur représentation. Afin de réaliser la recommandation, une approche vectorielle initialement utilisée en recherche d'informations est généralement appliquée et couplée à une pondération TF-IDF (i.e. Term Frequency – Invers Document Frequency).

Cette section décrit la modélisation d'un item dans un espace vectoriel, la caractérisation des items à partir de documents, la comparaison entre items et pour finir présente un ensemble de travaux associés.

Modélisation d'items

Le modèle vectoriel (i.e. Vector Space Model) est une approche algébrique qui permet la représentation de documents notamment textuels sous la forme de vecteurs. Il a été mis au point dans le cadre du projet SMART [**?**] visant a développer un système de recherche d'informations. Aujourd'hui, il est utilisé dans différents autres domaines tels que le filtrage d'informations.

Les items et profils sont représentés par des vecteurs dans un espace vectoriel. Chaque dimension de l'espace est alors un terme du vocabulaire de description (i.e. d'indexation), c'est-à-dire un mot-clé descripteur extrait du contenu du document. Différents traitements linguistiques sont effectués sur chacun des documents afin d'extraire de leur contenu les mots-clés permettant leur description. Les traitements linguistiques utilisés sont généralement assez simples :

- Tokenisation afin d'identifier les frontières entre les mots, et donc de découper les phrases en mots. Selon le perfectionnement de l'algorithme, il sera capable, ou non de prendre en compte les mots composés.
- Suppression des mots vides, c'est-à-dire, des mots qui n'apportent pas directement du sens (e.g. le, la, les, un, une, des, etc.).
- Racinisation (i.e. stemming), recherche de la racine des mots, suppressions des suffixes et préfixes (i.e. dérivationnels) ainsi que de l'expression du genre ou du nombre, voire du temps (i.e. flexionnels). L'ensemble des mots-clés extraits de tous les documents constitue le vocabulaire de description (i.e. vocabulaire d'indexation).

Dans les premières versions des systèmes utilisant cette approche, les vecteurs étaient binaires et permettaient de mettre en évidence la présence ou non d'un terme du vocabulaire de description dans le document. Afin d'améliorer la description des items, la pondération de chacun des termes pour chacun des documents s'est généralisée. Les vecteurs descripteurs sont composés de poids permettant de définir l'importance de chaque terme du vocabulaire d'indexation pour chacun des documents.

- Soit $V = \{t_1, t_2, \cdots, t_x\}$ l'ensemble des x termes du vocabulaire de description.
- Soit $I = \{i_1, i_2, \cdots, i_M\}$ un ensemble de M items, le corpus de documents.
- Soit $d_y = \langle p_1, p_2, \cdots, p_x \rangle$ le vecteur de pondérations des x termes du vocabulaire de description pour le y-ieme item du corpus.

Caractérisation d'un item

Cette modélisation pose plusieurs questions : (i) Comment choisir les pondérations ? (ii) Comment évaluer la pertinence (i.e. comment comparer les vecteurs) ? Car, la pondération de chacun des termes du vocabulaire d'indexation pour chacun des documents a pour objectif de mettre en évidence l'importance du lien qui existe entre les deux. Le but consiste à déterminer si le terme t_k est un descripteur de qualité pour l'item i_l (i.e. si son utilisation est réellement porteuse de sens). Pour cela, les premières approches se sont basées sur la fréquence du terme dans le document, TF (i.e. Term Frequency).

$$TF(t_k, i_l) = \frac{n_{k,l}}{\sum_z n_{z,l}} \tag{2.1}$$

En 1949, Zipf propose une approche, connue sous le nom de loi de Zipf [**?**]. D'après cette loi empirique, les mots ne sont pas organisés de façon aléatoire dans un texte, mais selon une loi inversement proportionnelle à leur rang. Le rang étant la position du terme dans la liste de tous les termes du texte, organisée selon leur fréquence d'apparition. Cette loi est basée sur l'observation de la fréquence des mots dans un texte. En 1949, en analysant "Ulysse" de James Joyce, Zipf observa que le mot le plus utilisé l'est deux fois plus que le deuxième mot le plus utilisé, trois fois plus que le troisième et ainsi de suite. Le phénomène avait déjà été observé en 1916 par Jean-Baptiste Estoup [**?**], alors sténographe au parlement français. Cela a permis de définir la relation qui existe entre le rang et la fréquence d'un mot dans un texte.

$$TF(t_k, i_l) = \frac{C}{Rang(t_k, i_l)} \tag{2.2}$$

L'équation **??** met en évidence le fait que la fréquence d'un terme dans un document est fonction de son rang dans le tableau des fréquences, et d'une constante C. Cette loi a été généralisée par Benoit Mandelbrot, et appliquée à d'autres domaines comme la prévision des revenus des ménages ou de la taille des villes dans un pays [**?**].

En 1958, Luhn se base sur la loi de Zipf pour proposer sa conjecture [**?**]. L'objectif étant de déterminer *l'informativité* d'un terme pour un document donné. Cette notion permet de déterminer la pertinence descriptive, le niveau d'information fourni par un mot. D'après Luhn, si le rang est trop faible, le descripteur est trop fréquent et n'a donc pas de pouvoir discriminant. S'il est trop élevé, alors le descripteur est peu pertinent, car le mot

est trop rare et donc peu utilisé comme le montre la figure **??**. Cette conjecture permet notamment la diminution de la taille des index (i.e. du vocabulaire d'indexation).

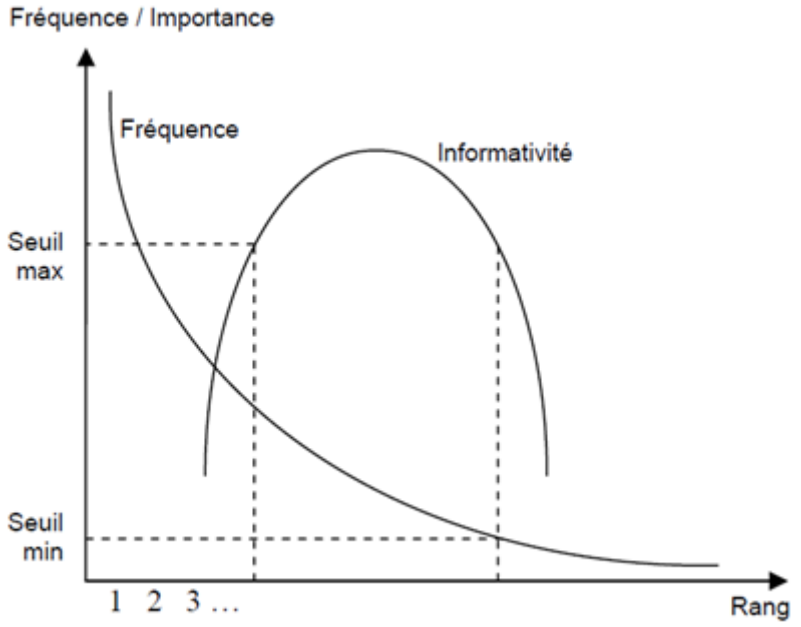


FIGURE 2.1 – Présentation de la conjecture de Luhn [**?**]

Les travaux dans le domaine de la recherche d'informations ont identifié une méthode qui s'est rapidement imposée. Cette méthode est dite TF-IDF (i.e. Term Frequency – Invers Document Frequency). Contrairement aux approches de Zipf et de Luhn qui ne se concentraient que sur des aspects locaux (i.e. propre aux documents auxquels ils appartiennent) dans le but d'évaluer la valeur descriptive des termes, l'approche TF-IDF, prend aussi en compte les aspects globaux. Dans cette approche, le contenu du document n'est pas le seul élément à être pris en compte, le contexte dans lequel il est nécessaire de le décrire l'est aussi (i.e. le corpus de document global) [**?**]. La méthode TF-IDF s'applique dans les cas où il s'agit de créer une description des documents appartenant à un corpus de documents fixes. L'idée est, de ne pas seulement prendre en compte les informations qui décrivent le document (i.e. termes plus ou moins fréquents). Mais de prendre en compte aussi les informations qui permettent de distinguer le document des autres. Par exemple, le terme "usine", ne sera probablement pas pertinent pour qualifier un document s'il ne s'agit d'indexer que des documents traitant du domaine industriel. Par contre dans le contexte plus général d'un corpus traitant de sport, de cinéma, de musique, de politique, d'économie, etc. ce terme gagne en pertinence.

En ce qui concerne les pondérations en fonction de l'*aspect local*, plusieurs approches peuvent être utilisées. Par exemple, une approche binaire consiste à pondérer à 1 le terme s'il est présent dans le document et à 0 s'il ne l'est pas. Dans la plupart des systèmes récents, le calcul de la fréquence du terme est utilisé (cf. équation **??**).

Concernant l'*aspect global*, la pondération permet de prendre en compte la représentativité globale d'un terme, en donnant un poids relativement important au terme peu fréquent dans le corpus. En effet, les termes utilisés dans beaucoup de documents sont moins discriminants. Pour cela, le calcul IDF (i.e. Inverse Document Frequency) illustré par l'équation **??**, a été proposé.

$$IDF\,(t_k) = \log \frac{M}{m_k} \tag{2.3}$$

Dans l'équation **??**, M désigne le nombre de documents du corpus et m_k le nombre de documents contenant le terme t_k. La pondération d'un terme pour un document donné à l'aide de la méthode TF-IDF est donc illustrée par l'équation **??**. Cette formule prend en compte la pondération locale TF (cf. équation **??**) et la pondération globale IDF (cf. équation **??**).

$$TF - IDF\,(t_k, i_l) = \frac{n_{k,l}}{\sum_z n_{z,l}} \cdot \log \frac{M}{m_k} \tag{2.4}$$

Nous avons présenté la modélisation des documents sous la forme de vecteurs. Les profils peuvent eux aussi être représentés par des vecteurs pondérés. Dans le cas d'un profil, la pondération a pour objectif de déterminer l'importance de chaque terme pour l'utilisateur. Ces profils sont, soit donnés explicitement par l'utilisateur, soit appris à partir de la liste des documents préférés, détectés par un vote explicite de l'utilisateur ou déduits de son comportement de façon implicite.

Comparaison d'items

Nous avons présenté comment, pour chaque document et profil, une modélisation vectorielle peut être créée. Nous présentons maintenant comment comparer les vecteurs afin d'évaluer la pertinence d'un document pour un utilisateur donné.

Le modèle vectoriel a été introduit en recherche d'informations afin de permettre un classement des résultats par pertinence. Le profil utilisateur est souvent défini comme un document idéal, à savoir le document correspondant totalement aux attentes de l'utilisateur. D'après Luhn [**?**] , "*Plus deux représentations de documents partagent de termes, plus il y a de probabilité qu'ils représentent la même information*". Il est donc nécessaire, pour évaluer la pertinence d'un document, de mesurer sa similarité avec le document idéal (i.e. profil) défini pour un utilisateur. Les documents ainsi que les requêtes étant représentés par des vecteurs, il est possible de réaliser des mesures de similarité à l'aide de différentes techniques de comparaison de vecteurs. La plus utilisée est la similarité cosinus (cf équation **??**) utilisée dans le projet SMART [**?**]. D'autres mesures peuvent être utilisées, telles que la similarité Jaccard [**?**] ou euclidienne pour ne citer que les plus connues.

$$Sim\left(i_h, i_j\right) = \frac{\sum_{n=1}^{x} p_{n,j} \cdot p_{n,h}}{\sqrt{\sum_{n=1}^{x} p_{n,j}^2} \cdot \sqrt{\sum_{n=1}^{x} p_{n,h}^2}} \tag{2.5}$$

L'équation **??** présente la formule de la similarité cosinus pour la comparaison de deux items i_j et i_h appartenant au corpus I. Dans cette formule $p_{(n,j)}$ désigne la pondération du n-ieme terme du vecteur de description de l'item i_j.

Travaux associés

Les systèmes basés sur des vecteurs de mots-clés sont très utilisés, et cela dans de multiples domaines. Nous nous intéressons ici au domaine spécifique de la recommandation de nouvelles (i.e. actualités, news). Les systèmes basés sur le contenu ayant une approche à l'aide de vecteurs de mots-clés les plus connus pour la recommandation de nouvelles sont YourNews [**?**], NewT [**?**], NewsDude [**?**], DailyLearner [**?**]et NewsJunkie [**?**].

YourNews est un système de filtrage d'articles d'actualité dont l'objectif est de donner confiance à l'utilisateur. Pour cela, le système permet à l'utilisateur de voir et de modifier son profil. Ce système est un système classique fonctionnant à l'aide de vecteurs pondérés par la méthode TF-IDF et comparés à l'aide de la similarité cosinus. Comme NewT (i.e. News Tailor), ce système utilise non pas un mais plusieurs profils pour chaque utilisateur. L'objectif est de permettre la gestion du sujet des articles recommandés. Les articles proposés par les systèmes sont divisés en flux en fonction du sujet. Les sujets sont prédéfinis et fixes. Un agent de recommandation est indépendamment utilisé pour chaque flux.

YourNews gère pour chaque flux d'articles, un profil de court terme et un de long terme. Ainsi, comme le système distingue 8 sujets d'informations différents, il utilise 16 profils pour chaque utilisateur. Les systèmes NewsDude et DailyLearner distinguent eux aussi des profils de long terme et de court terme. Ces trois systèmes utilisent une approche TF-IDF à partir d'un certain nombre d'articles récemment lus et appréciés par l'utilisateur (e.g. les 20 derniers pour YourNews) afin de construire un modèle utilisateur de court terme. En revanche, alors que YourNews utilise la même méthode pour les profils de long terme, NewsDude et DailyLearner utilisent eux des classifieurs bayésiens naïfs. Cette distinction entre des besoins de court terme et celui de long terme est spécifique aux systèmes de recommandation d'articles d'actualités.

Le système NewT permet à l'utilisateur de fournir un retour d'information (i.e. feedback) positif ou négatif explicite sur les articles proposés par le système. Ce sont ces votes positifs et négatifs qui sont utilisés par le système lors de la création des profils. Le NewsDude permet aussi à l'utilisateur de fournir des retours explicites, notamment : 1. "*interesting*" un équivalent de "*j'aime*", 2. "*more about this*" il veut plus d'informations sur ce sujet précis, 3. "*not interesting*" il "*n'aime pas*", 4. "*I already know this*" il ne le lit pas

parce que le contenu est "*déjà connu*" de l'utilisateur.

Le système NewsDude permet aussi à l'utilisateur de demander au système une explication, c'est-à-dire demander au système pourquoi l'article lui a été recommandé. Cela s'inscrit comme pour YourNews dans une démarche de transparence du système vis-à-vis de l'utilisateur dans le but de lui donner confiance. NewsDude ne permet pas à l'utilisateur d'accéder à son profil. YourNews a montré que ce genre d'approche permet un gain de confiance de l'utilisateur dans le système. Par contre, il montre aussi que cela a comme effet une baisse de la qualité de la recommandation fournie par le système. Plus généralement, les autres systèmes utilisent le comportement des utilisateurs sur la plateforme dans l'intention de déduire les goûts ou les besoins de façon implicite (i.e. implicite feedback). Par exemple, pour Google News, un clic sur un article est un vote positif.

En plus des termes ou mots-clés, certains systèmes tentent de prendre en compte d'autres descripteurs pour la représentation des articles et des profils. Par exemple, NewT, prend en compte les auteurs et les sources, alors que NewsJunkie ajoute aux mots-clés des entités nommées (personnes, lieux, organisations, etc.). NewsJunkie associe les articles à des sujets prédéfinis, comme NewsDude. De façon générale, un des critères de description d'articles souvent pris en compte lors de la recommandation d'articles d'actualité en plus des vecteurs de mots clés, est la temporalité. NewT, YourNews, NewsDude et NewsJunkie prennent en compte la nouveauté de l'information lors du processus de recommandation.

Cette section a présenté les principaux systèmes développés depuis les années 90. Nous nous sommes concentrés sur les systèmes spécifiques à la recommandation de nouvelles. Ce que nous constatons, c'est que la plupart de ces systèmes produisent de bons résultats, avec un historique du comportement utilisateur suffisant.

L'automatisation de la recommandation met le logiciel à la place de l'humain pour effectuer la tâche de filtrage de l'information [**?**], ce qui pose des difficultés notamment terme de gestion du langage naturel. La recommandation peut être vue comme un problème d'intelligence artificiel. En effet, la plupart des systèmes utilisent l'historique de navigation de l'utilisateur comme base d'apprentissage. Cette approche d'apprentissage vise à qualifier la pertinence d'un document pour un utilisateur donné à l'aide d'un modèle prédictif (cf. section **??**).

Les systèmes présentés tentent d'ajouter de nouvelles dimensions (i.e. facettes) à la description des items et profils. Ces facettes concernent généralement la prise en compte du sujet. Cela souligne en partie les manques que peuvent engendrer une modélisation entièrement basée sur des vecteurs de mots clés. (i) Il est difficile de donner de la dimension aux descriptions ce qui limite leur expressivité. (ii) Les approches par mots-clés posent des problèmes inhérents à l'ambigüité de la langue. La sémantique des mots, les rela-

tions qui peuvent exister entre eux ne sont pas prises en compte. Ainsi, l'évaluation de la pertinence d'un document pour un utilisateur donné est indirectement liée à des comparaisons de chaînes de caractères. Dans le langage naturel, une idée ou un concept peut être représenté par différents termes (i.e. synonymie). Cela induit des erreurs du fait qu'un article peut correspondre au besoin de l'utilisateur, mais contenir dans sa description des synonymes qui ne vont donc pas être détectés comme correspondant au profil. De même qu'un terme peut désigner différentes notions (i.e. polysémie). Cela cause d'autres erreurs, comme des fausses détections d'articles pertinents. De façon plus complexe, les mêmes concepts et idées peuvent être portés par des chaînes de caractères différentes, mais qui sont des variantes d'une même chaîne de départ. Ces variantes sont nommées flexions et dérivations. Un des derniers cas est celui des notions proches, mais non similaires. Les approches par mot-clé ne peuvent pas directement appréhender le fait que "*Martin Parr*" est un "*photographe*" ou qu'un "*pare-brise*" est une partie d'une "*voiture*" (i.e. méronymie).

2.1.1.2/ LES SYSTÈMES BASÉS SUR LA SÉMANTIQUE

Les systèmes basés sur la sémantique représentent un cas particulier des systèmes basés sur le contenu [**?**]. Ils ont pour objectif l'utilisation des technologies du web sémantique pour pallier certaines limites des systèmes basés sur le contenu classiques. Les technologies du web sémantique permettent une représentation des informations, des données et métadonnées améliorées. Les langages et structures de données proposés permettent une représentation sémantiquement plus riche [**?**]. L'expression "*web sémantique*" a été proposée par Tim Berners-Lee de la façon suivante : "*The first step is putting data on the Web in a form that machines can naturally understand, or converting it to that form. This creates what I call a Semantic Web – a web of data that can be processed directly or indirectly by machines.* " en page 191 du livre "Weaving the Web" [**?**]. Le web sémantique est généralement présenté comme une évolution ou une extension du web. Il a pour objectif d'améliorer encore l'échange d'informations et la collaboration homme / machine [**?**]. Pour cela, des métadonnées sémantiques sont associées aux documents ce qui implique une séparation entre la structure du document et son contenu, ainsi que la nécessité d'agents logiciels capables d'exploiter ces nouvelles ressources [**?**] [**?**].

Les systèmes basés sur la sémantique permettent une prise en compte des connaissances externes au processus de recommandation. Ils pallient certains manques dans la représentation de l'information, notamment dans les systèmes par mots clés [**?**]. Ces connaissances sont généralement représentées à l'aide de taxonomies, thésaurus ou d'ontologies formelles. Les langages développés dans le contexte d'étude du web sémantique et proposés par le W3C, permettent de décrire ces connaissances. La façon la plus expressive de représenter ces connaissances est l'utilisation d'ontologies. Elles ont pour

origine les travaux d'Aristote sur la représentation du monde et d'une façon plus générale l'étude des propriétés de tout ce qui est. Originaires de la philosophie, elles ont inspiré en intelligence artificielle des travaux visant à définir une formalisation de la connaissance capable de faire le lien entre l'homme et la machine. En informatique, une ontologie peut être définie comme une conceptualisation d'un domaine compréhensible par l'humain, mais utilisable par la machine, composée d'entités, d'attributs, de relations et d'axiomes [**?**]. Les systèmes récents de recommandation basés sur le contenu exploitent cette approche sémantique. Pour cela, on peut distinguer deux types de connaissances prises en compte.

1. Les connaissances propres au domaine concerné par le système de recommandation.
2. les connaissances plus générales, celles de la langue avec des bases de données lexicales (i.e. réseaux sémantiques) comme WordNet [**?**].

Utiliser des connaissances extérieures à un système afin d'en améliorer les performances est une problématique qui a été mise en avant dans le cadre de la recherche d'informations pour la première fois par Voorhees [**?**]. L'objectif était d'utiliser des connaissances lexicales pour améliorer la description du besoin de l'utilisateur dans un système de recherche d'informations. Pour cela, une méthode appelée "*expansion de requête*" a été proposée.

L'*expansion de requêtes* consiste à ajouter des mots, aux mots-clés déjà fournis par l'utilisateur voulant effectuer une recherche. Les mots clés sont ajoutés aux mots fournis par l'utilisateur dans le vecteur de requête, en se basant sur les connaissances lexicales de la base de connaissances. Selon les systèmes, des méthodes plus ou moins complexes sont mises en œuvre. Toutefois, les termes ajoutés sont principalement les synonymes et/ou les méronymes des termes fournis par l'utilisateur, généralement pondérés de façon à avoir moins d'importance que les termes de la requête initiale. L'hypothèse est qu'en ajoutant des termes sémantiquement proches à la description du besoin de l'utilisateur, l'expansion de requête doit permettre de résoudre en partie les problèmes de vocabulaire dépareillé entre la requête et le document. Dans l'outil proposé par Voorhees, seules quelques relations parmi toutes celles proposées dans WordNet sont utilisées pour trouver les termes en relation avec les mots-clés proposés par l'utilisateur afin de les ajouter à la requête. Les résultats de l'expérience montrent que peu de différences apparaissent si la requête est relativement longue et complète (en ce qui concerne la description de ce qui est cherché) et si le concept permettant l'expansion est choisi manuellement. Les requêtes courtes quant à elles, gagnent réellement en efficacité avec cette méthode si les concepts sont choisis manuellement. Toutefois, les performances sont grandement dégradées quand le choix des concepts utilisés pour l'expansion est fait

de façon automatique et non manuelle. Cette technique permet l'amélioration du *rappel*, il est donc normal que les requêtes courtes en bénéficient plus. Les requêtes longues permettent déjà une bonne recherche, car elles donnent plus de détails. Ces travaux ont été poursuivis et complétés notamment par Gonzalo [**?**]. L'annexe **??** introduit les méthodes d'évaluation des systèmes de recherche ou de recommandation d'informations et définit les notions de précision et de rappel.

Pour la recommandation de news, les systèmes basés sur la sémantique les plus connus sont Athena [**?**], Quickstep [**?**], Foxtrot [**?**], News@hand [**?**] et RelatingRSSNews [**?**]. Le système [**?**] n'est pas réellement un système de recommandation de nouvelles, mais pourrait assez aisément être adapté à cet effet. Il a pour but d'évaluer les relations sémantiques qui existent entre des articles fournis par différents sites proposant des nouvelles. L'objectif est d'améliorer l'accès à l'information notamment venant de plusieurs sources. Pour cela, le système ne compare pas des articles et des profils comme les systèmes de recommandation basés sur le contenu, mais les articles entre eux. Les profils étant souvent considérés comme un item idéal, l'approche proposée ici est donc proche de ce qui peut être fait en recommandation basée sur le contenu. Comme dans les systèmes Google news [**?**], News Dude [**?**] et YourNews [**?**], le système RelatingRSSNews [**?**] prend en compte plusieurs sources d'informations. Cela augmente la probabilité que plusieurs articles traitent de la même information. Afin de gérer cela, différentes approches ont été mises en place selon les systèmes. News Dude permet de prendre en compte le fait que l'information puisse être déjà acquise par l'utilisateur aussi bien par le système que par des sources extérieures au système. Pour ce faire, il permet à l'utilisateur d'exprimer explicitement "*I already know this*" afin que le profil ne soit pas à tort affecté négativement par son comportement de lecture. [**?**] propose un système dont l'objectif final serait de fusionner tous les articles traitant du même sujet, afin de ne pas avoir plusieurs fois la même information. L'idée serait de prendre en compte le fait que certains articles peuvent se compléter, d'autres se contredire. YourNews [**?**] ne propose pas de synthétiser toutes les informations dans un seul et même article, mais simplement de fournir toutes les informations afin que l'utilisateur ait une vision globale, de tous les points de vue qu'il peut y avoir sur le même évènement, sur le même sujet, sur la même information de base. Une même information peut être traitée avec des regards différents, par exemple les news émanant de sites russes ne traitent pas de la même façon les évènements de février 2014 en Ukraine que les sites européens ou américains. Les systèmes basés sur le contenu comme News Dude [**?**] et YourNews [**?**], précédemment présentés, ne permettent pas, comme le fait [**?**] à l'aide de WordNet, de savoir si deux articles traitent du même sujet, mais sont en contradiction, ou alors traitent exactement de la même chose avec les mêmes informations ou encore traitent de la même chose et sont complémentaires. l'utilisation de WordNet permet de détecter les termes ayant des sens proches ou contraires dans les articles comparés. NewsJunkie [**?**] se rap-

proche des systèmes sémantiques, car il distingue des entités nommées (i.e. personnes, lieux, organisations) afin de classer les articles selon 12 sujets prédéfinis. News Dude [**?**], se rapproche également des systèmes sémantiques, car il classe les articles selon 6 "*channels*", en utilisant une forme de connaissances externes. Un ensemble de mots caractéristiques de chacun des domaines a préalablement été sélectionné manuellement (i.e. vocabulaire d'indexation contrôlé). Ces traitements sont des approches de classification de texte beaucoup moins complexes que ce qui est proposé par [**?**]. En effet, [**?**], propose une façon d'étendre les vecteurs de description des articles à l'aide des connaissances contenues dans la base de connaissances WordNet. Cette méthode est analogue à l'*expansion de requêtes*. Elle pourrait être qualifiée d'*expansion d'items*. [**?**], montre que la prise en compte des connaissances externes via WordNet permet une amélioration de l'efficacité du système dans le contexte d'une comparaison complexe d'articles par rapport aux approches classiques basées sur des vecteurs de mots clés TF-IDF. Athena [**?**] est un système utilisant, également des connaissances lexicales de WordNet. Cette base de connaissances est indirectement utilisée pour la recommandation d'articles. En effet, la recommandation est sémantiquement basée sur une ontologie de domaine, mais WordNet est utilisé afin de faire le lien entre les articles et les classes de l'ontologie du domaine. Le processus de classification d'articles est basé sur la plateforme GATE [**?**]. La base de connaissances du système a été créée par des experts. L'ontologie est utilisée pour prendre en compte les concepts en relation avec les concepts qui apparaissent dans les articles. L'objectif est de faire correspondre au mieux le profil de l'utilisateur à ses intérêts. Par exemple, l'auteur propose d'ajouter à la description du profil d'un utilisateur intéressé par Google, les concepts de la base de connaissances Eric Shmidt qui a la relation est_CEO avec ce concept et d'ajouter Yahoo ! qui a la relation est_concurrent. Cette méthode est encore une fois analogue à l'approche proposée par Voorhees. Nous la nommerons "*expansion de profils*". Comme pour l'approche proposée par [**?**], l'évaluation proposée par [**?**] compare l'approche sémantique à l'approche classique TF-IDF afin d'en montrer les avantages (i.e. "l'*expansion de profils*" améliore les résultats, elle est impossible sans la réutilisation de connaissances externes). Quickstep [**?**], est un système qui comme Athena [**?**] et News@Hand [**?**] utilise des connaissances de domaine afin d'améliorer la recommandation. L'ontologie utilisée pour Quickstep est basée sur DMOZ et contient 27 classes. News@Hand utilise lui plusieurs ontologies en fonction du domaine traité. Au total, le système prend en compte 17 ontologies pour des sujets aussi variés que le sport, la science, la politique ou la religion. Comme Athéna, Quickstep utilise un classifieur afin de faire le lien entre les articles et les concepts de la base de connaissances. Les articles sont représentés comme des vecteurs TF (cf. équation **??**) et la méthode des K plus proches voisins [**?**] est utilisée afin de réaliser la classification sur la base d'un corpus d'entrainement indexé à la main. Cela permet de fournir comme pour Athena [**?**] une représentation des articles basés sur des vecteurs de sujets à partir

des vecteurs de termes pondérés. Un des avantages en plus d'un passage des termes aux concepts, est de manipuler des descriptions plus courtes. Un vecteur qui représentait au départ des milliers de termes, et donc de dimensions, ne contiendra que quelques sujets. Les profils sont créés en fonction de l'analyse du comportement de l'utilisateur et prennent, eux aussi, la forme de vecteurs de sujets pondérés en fonction de l'intérêt de l'utilisateur.

Foxtrot [**?**] est dérivé de Quickstep [**?**]. Il fournit à l'utilisateur une interface qui lui permet notamment de consulter son profil. La représentation des informations par l'intermédiaire d'une ontologie permet un affichage plus simple et compréhensible pour l'utilisateur qu'un vecteur de mots-clés pondérés. Cela permet de donner à l'utilisateur une plus grande clarté pour lui inspirer confiance.

L'utilisation de connaissances linguistiques lexicales est mise en avant par l'utilisation de WordNet permettant de prendre en compte le caractère dépareillé du vocabulaire entre différents articles pouvant pourtant traiter de la même information, ou au moins des sujets proches. Une approche qui ne prendrait en compte que de simples mots-clés verrait ses performances affectées négativement. Mais cette connaissance générale n'est pas suffisante pour adapter un processus de recommandation à un domaine précis. C'est pourquoi, beaucoup de systèmes utilisent une base de connaissances correspondant au(x) domaine(s) spécifique(s) traité(s) par le système de recommandation. C'est notamment le cas de Quickstep, Athena et News@Hand. L'utilisation de ces bases de connaissances permet une contextualisation du système dans son domaine, offrant un gain de performance par rapport aux approches. La granularité et la complexité de la description sémantique prise en compte varient selon les systèmes. Par exemple, WordNet contient des centaines de milliers de synsets (unités de sens) ainsi que plusieurs types de relations entre eux, alors que l'ontologie utilisée par Quickstep [**?**], ne contient que 27 concepts organisés hiérarchiquement via un seul type de relation "*is_a*".

Le passage des vecteurs de mots-clés à des vecteurs de concepts permet de limiter le nombre de dimensions des vecteurs de description, ce qui limite les calculs et permet une recommandation plus rapide [**?**]. C'est important en ce sens, que les systèmes de recommandation de news tendent à devenir temps réel. Le passage de vecteurs de mots clés à des vecteurs de concepts permet aussi de supprimer lors de la recommandation de documents textuels les effets dus à la différence de taille des articles [**?**] [**?**].

2.1.2/ LES SYSTÈMES DE FILTRAGE COLLABORATIFS

Les systèmes de recommandation sociaux (i.e. basés sur du filtrage collaboratif) tentent de guider l'utilisateur en lui recommandant des items que d'autres utilisateurs, possédant des goûts similaires, ont aimés par le passé. C'est pourquoi Schafer [**?**] qualifie les sys-

tèmes collaboratifs de systèmes à corrélation "*people-to-people*". En effet, ces systèmes comparent les historiques des utilisateurs afin d'évaluer leur similarité. Ces systèmes sont extrêmement populaires, car ils ne nécessitent pas, contrairement aux systèmes basés sur le contenu, l'analyse des items à recommander. C'est d'autant plus intéressant lorsque les items sont complexes à analyser, comme c'est le cas, des images ou des vidéos. Cela permet aussi de recommander au sein d'un même système des items de nature totalement différente (e.g. toute la variété de produits que l'on peut trouver dans une boutique en ligne comme Amazon.com).

Les systèmes de filtrage collaboratifs ne se basent pas uniquement sur les évaluations d'items fournis explicitement ou implicitement par l'utilisateur. Ils se basent sur les évaluations fournies par tous les utilisateurs du système. Ces derniers sont modélisés par la liste des évaluations qu'ils ont données, explicitement ou implicitement à certains items. Ces évaluations peuvent prendre la forme de notes exprimées lors de l'évaluation explicite d'un item, ou de notes attribuées en fonction d'un comportement. Par exemple, la note peut évoluer afin de refléter le nombre de visites de l'utilisateur sur la page d'un item et donc son intérêt pour celui-ci. D'autres comportements, comme l'achat, peuvent peser plus encore sur la note. L'intérêt de l'utilisateur pour l'item acheté étant pleinement mis en évidence par l'acte d'achat. Les systèmes de filtrage collaboratifs sont dépendants de leurs utilisateurs. Leur fonctionnement part du principe que ces derniers ont pour objectif d'avoir de bonnes recommandations et donc vont produire des informations qui leur seront utiles et le seront également aux autres utilisateurs. Evidemment, afin de prendre en compte le fait que ce paradigme de départ est naïf, certains systèmes tentent de détecter les utilisateurs dont l'objectif est de dégrader le système afin d'en annuler les effets [**?**] et de favoriser la recommandation de certains items au détriment d'autres. Les enjeux économiques, de ce type de manœuvres peuvent être très importants sur une importante plateforme de e-commerce telle qu'Amazon.

Les systèmes collaboratifs sont généralement regroupés dans deux catégories, les systèmes basés sur le voisinage (i.e. basés sur la mémoire) et les systèmes basés sur des modèles prédictifs. Les systèmes basés sur le voisinage proposent une prédiction de l'évaluation des utilisateurs en fonction de leurs évaluations passées. Le cas typique consiste à considérer que le degré de correspondance d'un utilisateur U avec un item I peut être prédit comme étant, la moyenne des évaluations de cet item I par tous les autres utilisateurs pondérée en fonction de la similarité des goûts de ces utilisateurs avec ceux de l'utilisateur U. Les systèmes basés sur des modèles prédictifs tentent de concevoir des modèles utilisateurs à l'aide de leurs évaluations passées et utilisent ces modèles afin de prédire l'évaluation des utilisateurs sur des items qui leurs sont inconnus. Ces systèmes utilisent des algorithmes d'apprentissage de modèles, par exemple des modèles probabilistes, comme les modèles bayésiens.

La première implémentation de Google news [**?**] combine les deux approches. Ce système était l'un des rares avec GroupLens [**?**] à recommander des nouvelles en se basant sur une approche entièrement collaborative. La nature des informations proposées par Google News, bien que proche, est différente de celle proposée par GroupeLens. GroupeLens permet de recommander des articles dans le cadre de groupes de discussions, alors que Google News recommande des articles d'actualité provenant de la quasi-totalité des sites d'information (e.g. Le Monde, New York Time, Der Spiegel, El Mondo, The Guardian, etc).

Dans le cadre de la recommandation de documents textuels, l'avantage des approches basées sur un filtrage collaboratif est qu'en s'affranchissant du contenu, ils s'affranchissent de la langue. Par exemple, le système Google News a été mis en place aisément dans de nombreux pays. Toutefois, le principal inconvénient des systèmes collaboratifs concerne la difficulté à gérer de nouveaux items, car tant qu'ils n'ont pas été notés par un nombre suffisant d'utilisateurs, le système est limité dans sa capacité à les recommander. Ils sont par conséquent rarement utilisés dans un contexte où la base d'items à recommander évolue aussi rapidement que dans le cas de la recommandation d'articles d'actualités (i.e news). Pour être recommandé, un item doit être évalué par suffisamment d'utilisateurs. Les articles de presse ne possèdent une forte valeur que durant un faible laps de temps après leur production. Il faut donc qu'assez rapidement, suffisamment d'utilisateurs aient fourni une évaluation afin de permettre à l'information d'être proposée à d'autres utilisateurs. Peu de systèmes ont autant d'utilisateurs et conservent autant d'historiques que Google, ce qui à permis a Google News de compenser les désavantages des systèmes collaboratifs.

GroupLens et SCENE [**?**] sont des systèmes basés sur le voisinage utilisateur. C'est-à-dire que l'intérêt d'un item pour un utilisateur est évalué en fonction de l'intérêt exprimé par les autres utilisateurs sur ce même item. Une mesure de similarité est utilisée pour comparer les utilisateurs. L'estimation de l'intérêt d'un utilisateur est calculée comme la moyenne des évaluations des K utilisateurs les plus similaires. Evidemment, seuls les utilisateurs ayant évalué l'item peuvent être pris en compte. SCENE [**?**] n'est pas un système purement collaboratif, c'est un système de recommandation de news hybride, qui utilise à la fois une approche basée sur le contenu et une approche collaborative. La comparaison du profil utilisateur et de la description de l'item est couplée avec une approche semblable à GroupLens où le voisinage de l'utilisateur est pris en compte. Ainsi, dans SCENE, les utilisateurs sont comparés à l'aide d'une similarité Jaccard [**?**] entre les vecteurs de notes qu'ils ont attribuées. Les notes sont binaires, soit l'article a été lu et la note est 1, soit il ne l'a pas été et la note est 0. Ainsi, les utilisateurs sont comparés en fonction de leur comportement de lecture. Plus ils ont d'articles en commun, plus ils sont similaires. Tout comme Google News, SCENE prend en compte la problématique du passage à l'échelle et du traitement en temps réel de la recommandation. En effet, le nombre

d'articles d'actualité produits chaque jour par les différentes sources d'information, édition web des journaux et magazines autant que les nouveaux supports Pure Player (i.e. tout en ligne), est très important. Le nombre d'utilisateurs susceptibles de vouloir accéder à l'information par l'intermédiaire d'internet est chaque jour plus important et concerne potentiellement des millions de personnes. Ces systèmes sont capables de répondre à leurs attentes de façon efficace et rapide. De même, Google News et SCENE, intègrent dans leur démarche des spécificités dues au type d'item recommandé. Contrairement aux systèmes de recommandation des boutiques d'e-commerce où le catalogue de produits est relativement fixe, la recommandation de nouvelles utilise une base d'items dynamique. Plus un article est ancien, plus il perd en intérêt pour l'utilisateur. De plus, tous les jours, des milliers de nouveaux items sont à recommander.

Nous avons présenté les systèmes ainsi que les notions directement liées ou proches de la problématique de la recommandation d'articles d'actualité. Nous avons introduit des problématiques ainsi que les solutions existantes. Dans la section suivante, nous mettons ces solutions en perspective par rapport à notre problématique (cf. chapitre **??**).

2.1.3/ SYNTHÈSE À PROPOS DE LA RECOMMANDATION D'ACTUALITÉS

Dans cet état de l'art, nous avons distingué les deux principaux types de systèmes de recommandation, c'est-à-dire, les systèmes basés sur le contenu et les systèmes collaboratifs. Nous avons présenté différentes implémentations de ces systèmes dans un contexte proche de celui de la recommandation d'articles d'actualité. L'analyse de ces systèmes nous a permis de mettre en balance les avantages et inconvénients de chacune de ces approches, que nous détaillons ci-dessous.

Les avantages des systèmes basés sur le contenu sont les suivants :

- Indépendance des utilisateurs : Les systèmes basés sur le contenu ne se basent que sur le comportement de l'utilisateur afin de le profiler. Alors que les systèmes collaboratifs utilisent les historiques de tous les utilisateurs, et voient leurs performances impactées par le nombre d'utilisateurs actifs (i.e. un nombre minimum est nécessaire au bon fonctionnement).
- Unicité des goûts : Les systèmes basés sur le contenu sont capables de recommander des items à des utilisateurs possédant des goûts uniques. Cela découle directement du point précédent. Contrairement aux systèmes collaboratifs, seul le comportement de l'utilisateur influence son profil. Il n'est donc pas nécessaire de trouver des utilisateurs semblables afin de pouvoir recommander des items à un utilisateur.
- Démarrage à froid (i.e. cold start) : Les systèmes basés sur le contenu permettent

de recommander des items avant qu'ils n'aient été vus ou évalués par d'autres utilisateurs. Ainsi, contrairement aux systèmes collaboratifs, un nouvel item peut être recommandé

- Transparence : Le processus dont découle la recommandation d'un item est plus facilement compréhensible par un utilisateur dans le cas d'un système basé sur le contenu que dans celui d'un système collaboratif où la recommandation est influencée par le comportement d'autres utilisateurs inconnus. Les systèmes, basés sur la sémantique, sont encore plus simples à appréhender par l'utilisateur.

Les inconvénients des systèmes basés sur le contenu sont les suivants :

- Complexité des items : (i) Il est très complexe et parfois impossible de créer une représentation automatique de tous les types d'items. La vidéo et le son sont particulièrement difficiles à gérer. Certaines descriptions doivent donc être réalisées à la main, ce qui est chronophage. (ii) La façon de représenter l'item va influencer les performances du système. Si la représentation de l'item ne permet pas de capturer les éléments de description pertinents pour l'utilisateur, la recommandation ne sera pas de bonne qualité . (iii) Il existe des items intéressants pour l'utilisateur, que les systèmes basés sur le contenu ne permettent pas de recommander, parce qu'ils ne ressemblent pas à ce l'utilisateur a apprécié par le passé. Afin de pallier ce manque, certains systèmes mettent en place des mécanismes visant à ajouter du bruit dans les réponses en recommandant aux utilisateurs des items hors de leurs besoins connus. Cela permet des découvertes fortuites (i.e. sérendipité). Les systèmes basés sur un filtrage collaboratif n'ont pas besoin d'avoir recours à ce type de méthodes. (iv) Les items de natures différentes ont souvent besoin d'un type de description différente. Ils ne sont généralement pas recommandables sur une même plateforme ce qui est pourtant le cas avec les systèmes collaboratifs.

- Sur-spécialisation : Beaucoup de systèmes basés sur le contenu utilisent l'historique du comportement de l'utilisateur afin de modéliser son profil. Au fur et à mesure de l'utilisation de l'outil, le profil devient plus précis. Le risque est une description trop précise du besoin empêchant de trouver par un heureux hasard une information intéressante (i.e. sérendipité) qui, ne correspondrait pas au besoin de l'utilisateur, tel qu'il est connu par le système.

- Démarrage à froid des utilisateurs : Beaucoup de systèmes basés sur le contenu utilisent l'historique du comportement de l'utilisateur (uniquement sur le système) pour modéliser son profil. Or, pour un nouvel utilisateur, il n'existe pas d'historique. Il est donc impossible de lui recommander des items, sauf en lui attribuant un profil par défaut.

Les avantages des systèmes de filtrage collaboratifs sont les suivants :

- Indépendant du contenu : Contrairement aux systèmes basés sur le contenu, la recommandation est totalement indépendante du contenu et même du type d'items. Ainsi, toutes sortes d'items peuvent être recommandés sans nécessiter de processus complexe d'analyse. De même, ces systèmes sont capables de recommander, à un utilisateur, des items totalement différents des items qu'il a apprécié par le passé.
- Évolution positive : Ces systèmes évoluent généralement positivement. Plus le système est utilisé, plus les historiques des utilisateurs sont importants et plus il gagne en performance. Cette évolution du système se fait sans pour autant générer de surspécialisation. Ainsi, contrairement aux systèmes basés sur le contenu, la capacité à recommander des items différents de ceux appréciés par le passé est conservée.

Les inconvénients des systèmes de filtrage collaboratifs sont les suivants :

- Démarrage à froid des items (i.e. cold start / first raster problem) : Un système collaboratif ne peut pas, contrairement à un système basé sur le contenu, recommander aisément de nouveaux items. Avant d'être capable d'en fournir une recommandation, le système doit attendre que suffisamment d'utilisateurs aient évalué implicitement ou explicitement l'item.
- Gestion des cas rares : Un système collaboratif ne peut pas, contrairement à un système basé sur le contenu, recommander des items à des utilisateurs ayant des goûts uniques. De même, ces systèmes fonctionnent mal dans des contextes produisant peu d'historiques ou des plateformes possédant un nombre trop faible d'utilisateurs.

Cet état de l'art a permis de mettre en avant les spécificités du traitement des articles d'actualités qui vont nous orienter vers certains choix en terme d'approche, en fonction des avantages et des inconvénients de chacune, développée ci-dessus. En effet, les articles d'actualité, ont des spécificités que les systèmes doivent prendre en compte afin d'être efficaces.

Les nouvelles sont gérées sous forme de flux. Les informations que contiennent les articles n'ont de pertinence que de l'immédiateté [**?**]. Plus une information est ancienne, plus elle perd en importance. Sa popularité, ainsi que sa pertinence, évoluent au cours du temps [**?**] [**?**]. Des sites d'information comme LeMonde.fr ou TheGuardian.uk proposent un flux de plusieurs dizaines d'articles par jours librement accessibles. Ces articles sont envoyés en temps réel au cours de la journée. L'ensemble des items à recommander

évolue donc continuellement [**?**]. Du fait de l'évolution rapide des items à recommander, les méthodes de filtrage collaboratives ne sont pas directement utilisables [**?**] [**?**] [**?**].

L'approche filtrage collaboratifs fonctionne bien mieux avec un contenu relativement statique [**?**] [**?**]. Elle demande de plus un historique important [**?**]. Les importantes plateformes de e-commerce comme Amazon.com ont recours à ce type de systèmes, car ils ont beaucoup d'utilisateurs et d'historiques de comportement. De plus, ils ont besoin de pouvoir recommander des items de natures totalement différentes.

Group Lens [**?**] [**?**] ainsi que la première version de Google News [**?**] étaient des systèmes totalement collaboratifs, ils n'utilisaient aucune analyse du contenu. Compte tenu de la spécificité des articles d'actualité et des problèmes qu'engendraient une solution totalement collaborative, Google a modifié son approche afin de créer un système hybride [**?**].

Comme nous l'avons présenté en introduction, notre système doit compter quelques milliers d'utilisateurs, et gérer des flux d'articles nouveaux tous les jours. Cela semble écarter les systèmes collaboratifs, en raison de notre faible nombre d'utilisateurs et de la dynamique de renouvellement des items. De même, notre système doit être capable de recommander de l'information à un utilisateur ayant un besoin unique. Nous nous orientons donc vers un système basé sur le contenu. Le système doit prendre en compte, dans son fonctionnement, le savoir-faire de l'entreprise qui est de savoir décrire les informations ainsi que de comprendre le besoin des utilisateurs. Les relations entre les différents processus et acteurs de la plateforme doivent être facilitées le plus possible. Par conséquent, nos premiers travaux se sont concentrés sur la mise en place d'un système basé sur la sémantique, capable de répondre à l'ensemble de ces besoins. C'est pourquoi nous nous intéressons dans la section suivante aux vocabulaires d'indexation et à la formalisation d'un domaine, nécessaire à la prise en compte de connaissances sémantiques par notre système.

2.2/ RÉFÉRENTIEL D'INDEXATION

Les systèmes de recommandation sont des systèmes automatisés qui visent à assister l'humain dans la tâche de gestion de l'information. Les systèmes basés sur le contenu, ainsi que ceux basés sur la sémantique, fonctionnent en deux étapes : (i) indexation et (ii) comparaison [**?**].

Dans ce type de système l'étape d'indexation est un préalable obligatoire à la recommandation. Afin de permettre la comparaison automatisée entre les items, il est nécessaire de permettre à la machine de les comprendre et de les manipuler. La machine n'est pas seule à intervenir dans la tâche de recommandation. Différents acteurs humains peuvent

intervenir, en particulier l'utilisateur ou l'expert. Ainsi, proposer une description des items qui soit compréhensible également aux humains doit permettre de faciliter les interactions et l'utilisation ainsi que d'augmenter la confiance envers le système. Il existe différentes façons de décrire le contenu d'un item, les principaux systèmes de recommandation basés sur le contenu, non sémantiques, utilisent des vecteurs de mots clés. Or, (i) les vecteurs de mots clés se heurtent aux différents problèmes que pose l'utilisation non contrôlée du langage naturel. La difficulté principale est la prise en compte par la machine du sens et des relations qui peuvent exister entre les termes. (ii) Les vecteurs ne permettent pas de prendre en compte les différentes dimensions descriptives des items et (iii) les vecteurs ne sont pas simples à appréhender pour un humain. C'est pourquoi nous nous intéressons dans cette section aux méthodes basées sur des vocabulaires contrôlés, aux représentations formelles permettant de faire le lien entre l'humain et la machine et aux méthodes permettant de distinguer les différentes dimensions descriptives d'un item, les facettes.

La notion de facette a été introduite en science de l'information par R. Ranganathan en 1924 [**?**]. Du fait du sens même de ce terme en vocabulaire courant, la notion est restée ambiguë [**?**]. Elle est utilisée et définie de façon diverse, c'est pourquoi nous en imposons la définition suivante. Les facettes sont les dimensions descriptives d'un item. La description d'items a pour objectif de faciliter leurs identifications, distinctions et recherches dans un ensemble d'items. La notion de facettes est un outil particulièrement intéressant lors de la mise en place de systèmes de filtrages ou de recherche d'information [**?**], c'est un mode de fonctionnement familier pour les utilisateurs. Elles permettent de lutter contre la rigidité de la classification taxonomique (type CDD [**?**], CDU [**?**]) et le chaos des indexes non structurés. Cette notion développée par des experts de la gestion documentaire permet une représentation des connaissances fidèle à la richesse du monde réel et pratique [**?**].

Le choix et la pertinence d'une facette va dépendre de la complexité, de la nature des items et du contexte d'application. Il est possible d'imaginer pour un item donné, une infinité de facettes descriptives. Par exemple, sur une plateforme de e-commerce, un ordinateur va être décrit à l'aide de plusieurs facettes : type de microprocesseur, type de mémoire, quantité de mémoire seront parmi les critères les plus pertinents. La couleur de la coque, ainsi que la matière qui la compose sont des facettes imaginables, mais non pertinentes dans le contexte d'application. Dans un autre contexte, comme la gestion du recyclage d'ordinateurs, la facette composition de la coque serait pertinente.

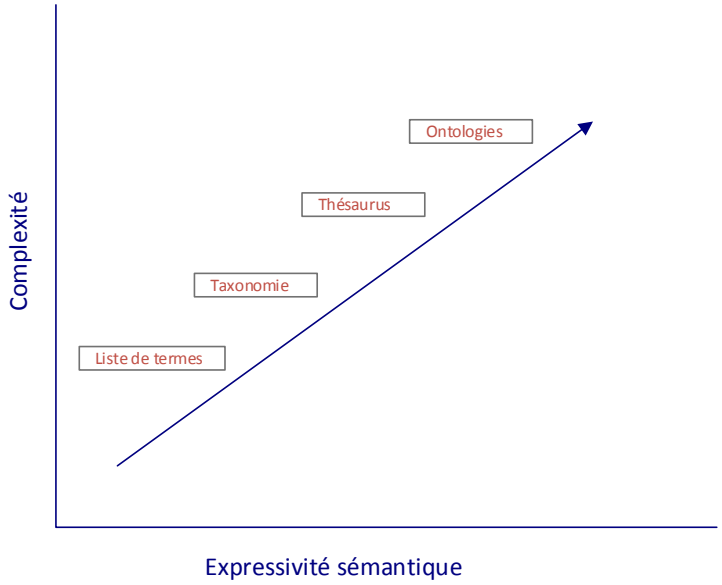


FIGURE 2.2 – Expressivité et complexité des différents types de ressources terminologiques et ontologiques [**?**]

Nous présentons ci-dessous un tour d'horizon de différents outils existants permettant la gestion du vocabulaire nécessaire à la description d'items. Ces outils sont plus ou moins complexes et expressifs sémantiquement parlant comme le montre la figure **??**.

Les informations contenues dans un item de type texte ne sont pas facilement compréhensibles pour la machine. Même l'humain, manipulant bien plus aisément le langage naturel que la machine, doit au préalable lire le texte du document afin de savoir de quoi il traite.

L'utilisation d'une représentation du contenu pour chacun des items permet à la machine autant qu'à l'humain de comprendre leur contenu sans pour autant avoir à les lire. Ce qui est un gain de temps pour l'utilisateur qui cherche une information. A cette fin, les systèmes de gestion documentaire ainsi que les bibliothèques ont développé des langages documentaires.

Selon l'AFNOR, un langage documentaire est un " *procédé conventionnel de représentation des informations d'un document sous une forme condensée et normalisée. Langage artificiel, constitué de représentations de notions et de relations entre ces notions et destiné, dans un système documentaire, à formaliser les données contenues dans les documents et dans les demandes des utilisateurs*" [**?**].

Comme le montre la définition de l'AFNOR, les langages documentaires s'intéressent particulièrement à la description du contenu des documents. C'est une approche de des-

criptions dite substantielle des documents, contrairement aux approches dites administratives, complémentaires, qui permettent la gestion d'informations annexes (e.g. le format, l'auteur, l'éditeur, etc).

Les sous-sections suivantes détaillent les solutions de gestion des vocabulaires d'indexation. Dans un premier temps, nous nous intéressons aux outils utilisés par les documentalistes, dans les bibliothèques ou plateformes de gestion documentaire. Dans un second temps, nous présentons cette problématique telle qu'elle a été gérée plus récemment à l'échelle du web par des informaticiens afin des créer des systèmes manipulables par la machine ou accessibles à des utilisateurs non experts.

2.2.1/ Approches orientées gestion documentaire

Les bibliothèques ainsi que les plateformes de gestion documentaire font reposer l'indexation des documents sur des langages documentaires. Ces langages sont composés de termes, c'est-à-dire des mots ou des groupes de mots. Il peuvent être plus ou moins structurés, par exemple les listes plates sont des ensembles de termes non structurés. Dans cette section nous traitons des langages structurés, parmi lesquels nous considérons particulièrement les taxonomies et les thésaurus qui seront décrits ci-dessous.

2.2.1.1/ Classification, taxonomie et nomenclatures

Originellement, les taxonomies sont des outils de classification définis par la systématique, branche de la biologie ayant pour objectif le classement du vivant. Le premier exemple est l'arbre d'Aristote. Les Taxonomies ont donc principalement été utilisées en biologie, mais se sont imposées dans d'autres sciences et en particulier dans les sciences de l'information. Les approches modernes que sont la classification décimale de M. Dewey 1876 (i.e. CDD) [**?**] puis la classification décimale universelle de P. Otlet et H. La Fontaine 1905 (i.e. CDU) [**?**]. Elles ont pour objectif de classer l'ensemble du savoir humain selon une division hiérarchique. Pour Otlet et La Fontaine cela passait par la réalisation de fiches descriptives définissant un catalogue de l'ensemble des publications depuis l'invention de l'imprimerie classées avec leur classification décimale universelle.

- **Classe 0 :** - Sciences et connaissance. Organisation. Informatique. Information. Documentation. Bibliothéconomie. Institutions. Publications
  - **00 :** - Prolégomènes. Fondements de la connaissance et de la culture
  - **01 :** - Bibliographie(s). Catalogues
  - **02 :** - Bibliothéconomie
  - **030 :** - Ouvrages généraux de référence (en tant que sujet)
  - **04 :** - Informatique

FIGURE 2.3 – Exemple extrait de la classification décimale universelle [**?**]

Dans ces deux approches encore largement utilisées aujourd'hui dans les bibliothèques, les classes sont codées par une succession de chiffres comme l'illustre la figure **??**. Les taxonomies sont des structurations hiérarchiques de vocabulaires. Elles sont donc composées de nœuds et de feuilles comme le montre la figure **??**. Le nœud racine définissant le cas échéant le terme le plus général.

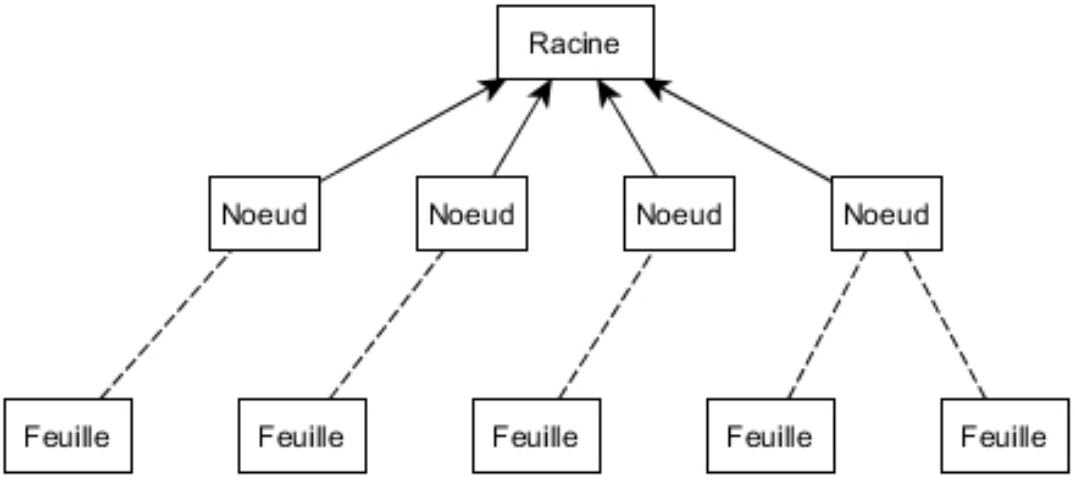


FIGURE 2.4 – Structure d'une taxonomie

La hiérarchie consiste généralement en une relation entre des termes précis et des termes plus larges. Deux types de relations sont principalement pris en compte dans la hiérarchie :

- La relation *partie-tout* : dite, relation de *composition*. Elle fait le lien entre deux termes, en spécifiant que l'un est un sous élément de l'autre.
- La relation *générique-spécifique* : dite, de propriété ou de genre à espèce. Elle fait le lien entre deux termes, en spécifiant que l'un est plus général que l'autre.

Il peut y avoir une assez grande variété de nuances sémantiques dans ce qui est considéré comme relation hiérarchique entre termes. Les définitions et points de vue sur ce qu'est une taxonomie sont divers et parfois contradictoires. A l'origine le terme avait pour définition le regroupement d'objets ayant des caractéristiques similaires. La définition a aujourd'hui évolué pour en faire un système classificatoire exclusivement hiérarchique. Nous considérons ici une taxonomie comme un langage d'indexation contrôlé organisé formellement de façon à expliciter des relations exclusivement hiérarchiques a priori entre les termes composant ce vocabulaire.

Les nomenclatures sont, tout comme les classifications, des taxonomies, ou taxinomies, dans le sens ou ce sont des systèmes de classement mono hiérarchiques. A la différence des classifications comme la CDD ou la CDU, les nomenclatures permettent à travers différents niveaux, de couvrir l'ensemble des notions présentes dans un domaine spécifique de la connaissance. Les classifications ne se limitent pas à un domaine, mais tentent de couvrir l'ensemble du savoir humain. Les nomenclatures sont donc constituée d'un langage contrôlé normalisé (i.e ensemble de termes) afin de référencer de façon univoque les notions contenu dans des documents (ou des requêtes d'utilisateurs). Ce

- **Personnalité :** le concept principal du document
- **Matière :** une substance ou une propriété
- **Énergie :** l'opération ou action subie par l'objet
- **Espace :** localisation géographique
- **Temps :** localisation chronologique et temporelle.

FIGURE 2.5 – Les 5 facettes définies par Ranganathan, S. R.

vocabulaire est complété par un ensemble de règles d'utilisation permettant de classer les informations de façon cohérente. Tout comme les classification elles ont un rôle de référentiel. Elles permettent d'organiser les connaissances, d'indexer des documents. Contrairement aux classifications CDU ou CDD, elles sont spécifiques à un domaine ou à une discipline. Elles peuvent aussi être spécifiques à un contexte métier et permettre au sein d'une entreprise de faciliter la communication. Dans ce cadre elles constituent un langage référentiel commun à différentes personnes, services et systèmes de l'entreprise en adéquation non seulement avec le domaine, mais avec le point de vue métier de l'entreprise sur le domaine.

Les schémas de classification de type CDD et CDU ne sont pas sans poser de problèmes. En biologie, les taxonomies tentent d'indexer l'ensemble du vivant, en gestion documentaire les classifications ont pour objectif d'indexer l'ensemble des connaissances humaines. Leur élaboration, leur maintien ainsi que leur utilisation est complexe. Et ce, dans le cas de domaines d'indexation restreints définis par des nomenclatures.

Prenons un exemple : un documentaliste souhaite classer un document traitant du sujet des pneumatiques. La classification propose pneumatique à deux endroits.

Pneumatique - caoutchouc - pétrochimie - chimie
Pneumatique - véhicule roulant - véhicule - transport
Où le documentaliste doit-il indexer le document ? Dans ce cas le classement va dépendre du point de vue de l'expert, de sa vision du monde. S. R. Ranganathan est l'un des premiers à tenter d'apporter une réponse à la rigidité des schémas de classifications taxonomiques (en particulier CDD). En 1933, il introduit le premier schéma de classification [**?**] à prendre en compte la notion de facettes. Il y définit alors 5 facettes, illustrées par la figure **??**, ordonnées dans un ordre sensé correspondre à l'ordre naturel de la pensée. Dans cette classification, le monde est divisé en une centaine de classes principales découpées selon les trois premières facettes (personnalité, matière, énergie) puis organisé sous la forme de hiérarchies. Temps et espace sont communs à toutes les classes. La *colon classification* est ainsi une composition de micro classification.

L'idée d'organiser le vocabulaire d'indexation sous la forme de facettes a bien des avan-

tages. Cette approche est très utilisée aujourd'hui dans le domaine du e-commerce. Elle permet de faciliter le filtrage, de se concentrer sur les caractéristiques importantes, essentielles ou persistantes de certains des objets. Elle permet aussi de prendre en compte plus facilement la complexité des documents à indexer. Alors que dans une classification, l'ajout de nouvelles notions peut être très complexe, ici la création d'une facette n'impacte pas le reste du vocabulaire. Toutefois, si l'ajout de la nouvelle notion ne se fait pas par l'ajout d'une facette mais par la modification d'une facette existante les difficultés de maintien de la cohérence sont les mêmes que dans le cas d'une classification, à ceci près que le vocabulaire d'une facette est généralement plus limité. Lorsque le système a été inventé l'utilisation de facettes posait comme problème principal (bien que répondant a la rigidité des mono hiérarchies) le positionnement physique du document au sein de la bibliothèque. Un document ne devrait être placé qu'à un seul endroit ce qui est compliqué avec un système multi-facettes, cela nécessite la mise en place de mécanismes de type pointeurs. Aujourd'hui, avec la gestion informatisée des documents cela ne constitue plus un problème.

2.2.1.2/ THÉSAURUS

Les thésaurus nés dans les années 50, sont d'après la norme ISO 2788 1986, le "*vocabulaire d'un langage d'indexation contrôlé organisé formellement de façon à expliciter les relations a priori entre les notions (par exemple relation générique - spécifique)*". D'après cette même norme, un langage d'indexation est un "*ensemble contrôlé de termes choisis dans une langue naturelle et utilisés pour représenter sous forme condensée, le contenu des documents".*

Ainsi les termes contenus dans un thésaurus servent à décrire un document. Ils sont principalement utilisés pour des tâches d'indexation manuelles de documents par des documentalistes, car ils sont connus pour présenter des avantages dans la réalisation de cette tache [**?**]. En effet, ils permettent de présenter l'ensemble du vocabulaire d'un domaine et les relations principales de façon organisée et facilement compréhensible par l'homme. Ils permettent aussi lors de la présence de synonymes de forcer l'utilisation d'un seul terme. Enfin, ils permettent la gestion de différents degrés de précision lors d'une recherche de documents. Une partie de ces termes servent directement à la description : les descripteurs. Une autre partie sert à orienter l'utilisateur vers le bon descripteur : les non-descripteurs.

D'après l'AFNOR, un descripteur est un "*terme ou groupe de mots retenus dans un thésaurus et choisis parmi un ensemble de termes équivalents pour représenter sans ambiguïté une notion apparaissant dans un document ou dans une demande de recherche documentaire*". AFNOR 1987

Les thésaurus intègrent une taxonomie, c'est à dire des relations hiérarchiques entre les termes du vocabulaire qui les composent. En plus de ces relations hiérarchiques deux autres types de relations sémantiques sont prises en compte, les relations d'équivalence (ou de synonymie) et les relations associatives (ou de voisinage).

- Les relations hiérarchiques permettent de gérer les différents niveaux de précision du vocabulaire utilisé. Elles ont été détaillées dans la section précédente **??**. Dans un thésaurus les relations d'un terme générique vers un terme spécifique sont notées TS (i.e. NT, Narrower Term, en anglais) et les relations d'un terme spécifique vers un terme générique TG (i.e. BT, Broader Term, en anglais). Par exemple *mémoire vive* TG *mémoire* et *mémoire* TS *mémoire vive* signifient que *mémoire vive* est un terme plus spécifique que *mémoire*.
- Les relations d'association entre termes permettent de faire le lien entre des termes qui ne font pas forcement parti du même thésaurus. La sémantique associée à cette relation est assez vague. Elle se rapproche des notions de graphes sémantiques, ou réseaux sémantiques entre termes. Nous la traduisons par "voir également" ou "est en relation avec". Cette relation symétrique est notée TA, pour Terme Associé (i.e. RT, Related Term, en anglais). Par exemple, *carte graphique* TA *écran*.
- Les relations d'équivalence permettent de gérer les termes associés à une notion. Ainsi une notion est définie par un terme "principal" (le descripteur), mais d'autres termes "synonymes" peuvent exister et faire référence à la même notion. Ils seront gérés en tant que non-descripteurs. La relation du non-descripteur vers le descripteur est notée EM, pour Employer (i.e. USE, en anglais) et celle du non-descripteur vers le descripteur EP, pour Employer Pour (i.e. UF, Used For, en anglais). Par exemple, *CPU* EM *microprocesseur*, et la relation inverse *microprocesseur* EP *CPU*.

Parfois, il existe une dernière relation : la note. La note ayant pour objectif de lever toute ambiguïté polysémique. Cette relation est notée NA, pour Note d'Application (i.e. SN, Scope Note, en anglais). Par exemple, *mémoire* NA *informatique*, signifie que le terme mémoire est ici utilisé avec la signification qui lui est donnée dans le domaine de l'informatique et non dans celui de la médecine par exemple.

Comme dans les classifications présentées à la section **??** précédente, la nature de la relation hiérarchique n'est pas forcement une stricte subsomption (ISO 2788 :1986). Afin d'uniformiser la représentation et l'utilisation des thésaurus de multiples normes ont été proposées (ANSI/NISO 739.19 :1993, ISO 2788 :1986).

Marché monétaire

- **UF :** marché monétaire international
- **RT :** marché financier
- **RT :** zone monétaire
- **NT1 :** monnaie
  - **NT2 :** monnaie fiduciaire
    - **UF :** billet de banque
  - **NT2 :** monnaie nationale
  - **NT2 :** monnaie scripturale
    - **NT3 :** chèque
    - **NT3 :** monnaie électronique
      - **UF :** monétique
      - **UF :** paiement électronique
      - **UF :** carte bancaire
      - **UF :** carte de crédit
      - **RT :** bancatique

FIGURE 2.6 – Extrait du thésaurus Eurovoc, normes ISO 2788 :1986 et ISO 5964 :1985

La figure **??**, illustre l'exemple d'un extrait du thésaurus Eurovoc[4]. Dans cet exemple, "*marché monétaire international*" est un synonyme de "*marché monétaire*". En cas d'indexation le second est préféré, en cas de recherche le premier mène au second. Le sigle NT, utilisé pour spécifier la relation d'un terme général vers un terme spécifique est suivi d'un numéro permettant de savoir à quelle profondeur de la hiérarchie se trouve le terme. Ainsi, "*monnaie électronique*" est le terme le plus profond dans cet extrait de la hiérarchie. Il a une profondeur de 3. Il est plus précis que le terme "*monnaie scripturale*" qui a une profondeur de 2 et qui est lui même plus précis que le terme "*monnaie*". Le terme "*monnaie*" est lui plus spécifique que le terme "*marché monétaire*". Le terme "*marché monétaire*" renvoie également aux termes "*marché financier*" et "*zone monétaire*".

Les normes récentes permettent la distinction entre les notions de concepts et de termes. Un concept pouvant être compris comme étant la signification d'un terme. Plusieurs termes pouvant avoir la même signification. Le concept est, une représentation générale et abstraite de la réalité d'un objet, d'une situation ou d'un phénomène. Le concept se distingue donc aussi bien de la chose représentée par ce concept, que du mot, du groupe de mots, du terme, de la notion, ou de l'énoncé verbal, qui est le signifiant de ce concept.

Bien que cette distinction soit ancienne aussi bien en philosophie qu'en linguistique (triangle sémiotique), sa prise en compte explicite lors de l'élaboration de thésaurus est le fruit de travaux récents remontant à 2000 et au groupe de travail rédigeant la norme BS 8723. La figure **??** illustre cette distinction dans un extrait du schéma UML modélisant un thésaurus selon la norme [**?**]. Avec la définition du langage SKOS[5], et la parution récente de nouvelles normes pour les langages documentaires ISO 25964, en continuité du travail mené sur la norme BS 8723, un travail important est en cours afin de rendre ces langages documentaires initialement orientés vers l'humain, plus facilement manipulable par la machine.

4. http ://eurovoc.europa.eu/drupal/ ?q=fr
5. http ://www.w3.org/TR/swbp-skos-core-guide/

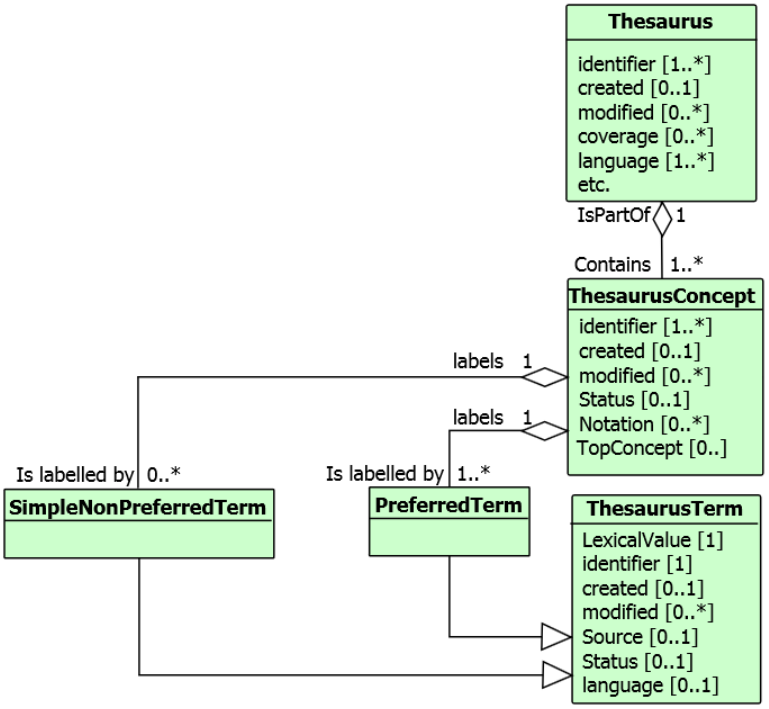


FIGURE 2.7 – Extrait du modèle UML fruit des travaux sur la norme BS8723 [**?**]

Comme pour les classifications, il existe des thésaurus à facettes. Dans ces thésaurus, les termes sont organisés selon plusieurs hiérarchies, chacune représentant une facette du domaine traité.

Les outils que nous venons de présenter n'ont pas été pensés pour une utilisation par la machine ou par des utilisateurs non expérimentés, pourtant l'utilisation de vocabulaires d'indexation est nécessaire à grande échelle afin de faciliter l'échange d'informations sur le web. Nous présentons donc, ci-dessous des approches pensées pour une utilisation dans le contexte du web.

2.2.2/ APPROCHES ORIENTÉES WEB

Les outils précédemment présentés sont des vocabulaires contrôlés structurés (i.e. langages documentaire) facilitant la gestion documentaire. Ils sont principalement utilisés dans des centres de gestion documentaire ou des bibliothèques. Ils sont dits, contrôlés, car l'ensemble des termes ainsi que les relations entre ces termes sont définis à priori. Même si la gestion a été automatisée et informatisée, ces outils ont été pensés au départ pour une utilisation humaine. Avec l'arrivée dans les années quatre-vingt-dix du web, la problématique consistant à trouver les documents correspondants à un besoin c'est étendue à la gestion de tous types de ressources accessibles sur le réseau via un navigateur. D'après Berners-Lee, une ressource est tout ce qui peut être identifié par une URI sur le

web [**?**]. Une URI (i.e. Uniform Resource Identifier, soit identifiant uniforme de ressource) est une chaîne de caractères suivant une structure syntaxique contrainte définie par la norme RFC 3986 [6]. Afin de faciliter la gestion des ressources sur le web, des métadonnées y sont ajoutées. Elles permettent de faciliter les échanges et d'assurer l'accessibilité aux ressources. D'après le dictionnaire LAROUSSE, une métadonnée est une "Donnée servant à caractériser une autre donnée, physique ou numérique". D'après le dictionnaire de l'Académie Française, une donnée est une "Représentation d'une information sous une forme conventionnelle adaptée à son exploitation".

Dans les approches orientées web, nous distinguons (i) les folksonomies, fruit du besoin de dynamiser la gestion de l'information et laissant la main à l'utilisateur et non plus à l'expert et (ii) les ontologies, fruit du besoin de réduire l'écart entre l'homme et la machine, afin de permettre à la machine d'assister l'homme plus efficacement.

2.2.2.1/ FOLKSONOMIE

Le web dit 2.0 amenant plus d'interactions entre les utilisateurs, ils ont été mis à contribution afin de qualifier certaines ressources. Ainsi, des plateformes de gestion d'images et de photographies comme *Flickr* [7], mais aussi de gestion de favoris comme *delicious* [8] ont permis à leurs utilisateurs de qualifier les ressources proposées par ces plateformes à l'aide de tags. Les tags sont des métadonnées (aussi appelé meta-tags) permettant de décrire le contenu d'une ressource (e.g. image, site web, etc.).

Le terme de folksonomie a été créé par Thomas Vander Wal en 2004 [9]. Une folksonomie est une liste de termes définis par des non-spécialistes. Contrairement aux vocabulaires contrôlés définis par des spécialistes que nous avons présentés précédemment. Les folksonomies sont donc des vocabulaires libres. Elles peuvent être propres à une personne ou à un groupe de personnes (i.g. comme sur les sites *delicious*, ou *Flick*). La définition d'une folksonomie consiste généralement à regrouper les tags (i.e. mots-clés) fournis par les utilisateurs d'un service afin de qualifier une ressource (item, documents, photos, articles, vidéo ...). La définition de Thomas Vander Wal est disponible : http://vanderwal.net/folksonomy.html. De multiples relations sémantiques existent entre ces termes, mais elles ne sont pas prises en compte. Le même document peut être qualifié par deux utilisateurs à l'aide de termes totalement différents.

L'évolution du web et la naissance de la notion de web sémantique ont favorisé l'émergence d'outils de structuration, les ontologies, ainsi que de normes pour la gestion des

6. http ://www.ietf.org/rfc/rfc3986.txt
7. https ://www.flickr.com/
8. https ://delicious.com/
9. http ://vanderwal.net/folksonomy.html

métadonnées, les langages RDF [10] (i.e. Resource Description Framework), RDFS [11] (i.e. RDF Schema) et OWL [12] (i.e. Web Ontology Language). Ils permettent de faciliter l'utilisation des métadonnées par les moteurs de recherche, et donc l'indexation de ressources.

2.2.2.2/ ONTOLOGIES

Bien que la notion d'ontologie prenne racine en philosophie, où les ontologies sont définies comme une partie de la métaphysique qui s'intéresse à l'organisation de la nature et de la réalité, dans leur utilisation moderne les ontologies sont très proches des thésaurus que nous avons présentés précédemment **??**. Ainsi d'après Dziri :

> " Le thésaurus permet, tout comme les ontologies, la modélisation des domaines de connaissances à des fins heuristiques, mais il reste plus attaché à la production manuelle pour des descriptions documentaires qu'à la conception de modèles informatiques. " [**?**]

Malgré leur proximité, les thésaurus tendent encore à être réalisés par des humains pour des humains alors que les ontologies permettent de faire le lien entre l'humain et la machine. De plus, le niveau de formalisme et d'expressivité sémantique est plus élevé pour les ontologies (cf. figure **??**). Les ontologies sont en informatique, une nouvelle approche de modélisation de connaissances. Contrairement aux thésaurus et aux classifications que nous avons présenté précédemment, elles s'inscrivent dans un contexte d'ingénierie informatique. Même si les ontologies formelles s'inscrivent dans la suite de travaux sur la représentation des connaissances, ce sont les travaux théorisant le web sémantique qui les ont popularisées.

En informatique, différentes définitions sont présentes dans la littérature.

La première définition de Neches, et al 1991 [**?**] :

> "*An ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary* [13]"

La plus référencée est celle de Gruber, 1993 [**?**] :

> "*An ontology is an explicit specification of a conceptualization* [14]"

10. http ://www.w3.org/TR/2014/REC-rdf-schema-20140225/
11. http ://www.w3.org/TR/rdf-schema/
12. http ://www.w3.org/TR/owl2-overview/
13. Une ontologie définit les termes et les relations de base du vocabulaire d'un domaine ainsi que les règles qui permettent de combiner les termes et les relations afin de pouvoir étendre le vocabulaire
14. Une ontologie est une spécification explicite d'une conceptualisation

Dans sa thèse, Borst [**?**] reprend et complète la définition donnée par Gruber :

"*An ontology is a formal specification of a shared conceptualization*[15]"

Nous retenons de la définition de Neches :
Une ontologie permet la gestion d'un vocabulaire ainsi que des relations qui existent entre les termes de ce vocabulaire.
Le vocabulaire géré par une ontologie ainsi que sa structuration permettent de représenter un domaine.
Nous retienons de Gruber :
Les ontologies sont explicites.
Les ontologies sont une conceptualisation, c'est-à-dire un modèle abstrait.
Ce que Borst complète de la façon suivante :
Les ontologies sont formelles, c'est-à-dire, interprétables par la machine.
Les ontologies sont partagées, c'est-à-dire potentiellement acceptées par une communauté d'utilisateurs et/ou de processus.
Une ontologie permet donc de structurer les connaissances d'un domaine et le vocabulaire qui permet de s'y référer. Pour cela des concepts, relations, axiomes et instances sont utilisés. On peut considérer une ontologie comme une taxonomie à laquelle des relations, contraintes et règles ont été ajoutées. Elles ne visent pas exclusivement l'indexation de documents [**?**], mais aussi la gestion des connaissances d'un domaine, afin d'y spécifier explicitement une conceptualisation qui pourra par la suite être partagée.

Les ontologies formelles ne visent pas directement une exploitation par l'humain, mais principalement par des programmes informatiques. Ainsi, elles sont essentielles dans le cadre du web sémantique pour la mise en relation et la recherche d'informations. C'est pour cela qu'elles sont représentées à l'aide de langages formels, standardisés par le W3C[16], comme OWL (contrairement aux classifications et thésaurus). Cette formalisation poussée permet l'utilisation de raisonneurs. Les raisonneurs permettent de générer de nouvelles connaissances à partir de connaissances existantes.

2.2.3/ SYNTHÈSE À PROPOS DES VOCABULAIRES D'INDEXATION

Cette sous-section a présenté différentes structures permettant la gestion et la représentation de connaissances. Nous y constatons une grande hétérogénéité.

Ainsi, il existe différents *types de ressources* terminologiques, les listes, classifications (i.e. CDD, CDU), taxonomies, nomenclatures, thésaurus, folksonomies et ontologies, que nous présentons ci-dessus.

15. Une ontologie est une spécification formelle d'une conceptualisation partagée
16. http ://www.w3.org/

En fonction du type de ressources, il existe différentes *syntaxes*. Les symboles utilisés afin de présenter une relation de hiérarchie ne sont pas les mêmes pour une classification de type CDD (i.e. basée sur des nombres décimaux) que pour un thésaurus. Les thésaurus basés sur les normes ISO 2799 :1986 et ISO 5964 :1985 utilisent les symboles NT et BT pour exprimer ce type de relations.

Certaines ressources sont donc normalisées à l'aide de *normes* nationales ou internationales. Pour un type de ressources donné, il peut exister un nombre important de normes. Par exemple dans le cas des thesaurus, il existe les normes : ISO 2786 :1986 (thésaurus monolingue), ISO 5964 :1985 (thésaurus multilingue), les normes françaises NF Z47-100 :1981 (thésaurus monolingue), NF Z47-101 :1990 (thésaurus multilingue), la norme étasunienne ANSI/NISO Z39.19-2005 et la norme britannique BS 8723 :2005, ainsi que toutes leurs précédentes versions. Certaines de ces normes sont en cours de remplacement par les normes ISO 25964-1 et ISO 25964-2.

Les ressources disponibles et réutilisables peuvent donc avoir été créées sur la base de normes diverses, de même elles sont disponibles sous des *formats de persistance* divers. Certaines sous la forme de fichiers PDF (e.g. thesaurus Delphes), XLS ou base de données DBF (e.g. Nomenclature NAF), OWL ou RDF (i.e. ontologies formelles), XML (e.g. MeSH) ou encore site web HTML (e.g. Eurovoc).

Les *objectifs d'utilisation* ainsi que le degré d'*expressivité* et de *complexité* de la ressource terminologique vont influencer le format de persistance. Ainsi, pour une utilisation par un humain, une version HTML ou PDF est préférable. Alors, que pour une intégration à un système informatique les formats base de données DBF, ou ontologies RDFS/OWL sont plus adaptés, car *formels*. De même, dans le cas de vocabulaires riches, et complexes (i.e. intégrant une grande quantité de termes et de *relations sémantiques* entre ces termes), les formats de type texte, comme les fichiers PDF sont mal adaptés à l'exploration de ce vocabulaire.

Enfin, si l'objectif du vocabulaire est une description précise du domaine (i.e. forte *granularité* et *couverture* du domaine) ou un simple fractionnement de celui-ci en quelques catégories, le type de vocabulaire choisi ne sera pas le même. Dans le premier cas, une ontologie ou un thesaurus sont plus adaptés, alors que dans le second une liste de termes sans prise en compte des relations sémantiques entre eux, peut être suffisante.

Dans le contexte d'un système de recommandation sémantique, différents acteurs sont amenés à intervenir. Des acteurs humains, qui vont par exemple alimenter le système, ou des processus, qui vont comparer le besoin des utilisateurs aux informations disponibles. L'utilisation d'un référentiel commun aux différents acteurs, qu'ils soient humains ou logiciels, permet de faciliter les interactions dans l'objectif d'une recommandation de qualité. Cela tend à imposer l'utilisation d'ontologies. En effet, leur caractère formel les rends facilement manipulable par la machine. Bien qu'elles soient aussi manipulables par

l'humain, leur forte expressivité sémantique et leur formalisme peut facilement en rendre l'utilisation complexe. Les humains préfèrent des structures plus simples avec un nombre limité de relations entre les termes (e.g. taxonomies). Les systèmes de recommandation sémantiques existant fonctionnent pour la plupart sur la base d'ontologies légères, qui ne sont bien souvent que de simple taxonomies [**?**] [**?**].

Dans ce contexte, l'adaptation du système de recommandation à un domaine précis, passe par l'adaptation du référentiel d'indexation commun, c'est-à-dire, du vocabulaire contrôlé d'indexation. La complexité de la modélisation de domaines vastes et/ou complexes à l'aide de vocabulaires organisés sous la forme de monohiérarchie a débouché sur l'invention de systèmes d'indexation à facettes. Ces systèmes sont aujourd'hui largement popularisés, notamment dans le domaine du e-commerce.

Sur la base de ces faits, un référentiel d'indexation propre au système de recommandation est l'objet du chapitre **??**. Ce référentiel prend la forme d'une ontologie afin d'être manipulable aisément par la machine. La forte expressivité sémantique de celle-ci est utilisée pour créer un *modèle intégrateur* capable d'intégrer des vocabulaires d'indexation plus simples et plus facilement manipulable pour l'humain (i.e. listes, taxonomies, thésaurus). Ces vocabulaires seront préalablement unifiés par un *modèle unificateur* permettant de faciliter le processus d'intégration au modèle intégrateur. Chacun des vocabulaires unifiés et intégrés à la base de connaissances du système permet la définition d'une facette du domaine traité. Cela permet de rendre l'utilisation de cette ontologie plus aisée pour l'humain tout en permettant la modélisation de domaines complexes.

L'indexation de documents est une tâche longue pour un humain, même quand celle-ci est facilitée par l'utilisation de vocabulaires contrôlés et structurés. Les approches venant du domaine de l'intelligence artificiel peuvent être utilisées afin d'apprendre à la machine à réaliser cette tâche de façon automatique. L'automatisation est l'objet de la section suivante.

2.3/ AUTOMATISATION DE L'INDEXATION

Dans cette section, nous introduisons des définitions fondamentales nécessaires pour la modélisation des systèmes de recommandation. La création d'un système de recommandation est un effort multidisciplinaire, qui réunit des experts de divers domaines tels que l'intelligence artificielle, les technologies de l'information, la fouille de données, les statistiques et systèmes d'aide à la décision [**?**].

La fouille de données (i.e. data mining) est largement utilisée au sein des systèmes de recommandation. Cette méthode d'exploitation de données permet à un système de recommandation d'apprendre à réaliser de manière optimale une certaine tâche à l'aide

d'exemples, de données ou d'expériences passées [**?**]. Dans cette section nous nous intéressons dans un premier temps à l'apprentissage automatique (i.e. machine learning) puis à la classification, processus par lequel il est possible de réaliser l'indexation d'informations nécessaire à certains systèmes de recommandation. Pour finir, nous présentons un tour d'horizon des approches de classification automatique incluant l'utilisation d'ontologies.

La fouille de données est une étape particulière du processus de découverte de connaissances utiles à partir de données. Elle consiste à utiliser des algorithmes spécifiques pour l'extraction de modèles de données [**?**]. La terminologie du domaine de la fouille de données est synthétisée dans la figure **??** [**?**]. Cette terminologie sépare les paradigmes de la fouille de données en deux principaux types [**?**] [**?**] [**?**] :

- **Découverte** : le système trouve de nouvelles règles et des modèles de manière autonome.
- **Vérification** : le système vérifie les hypothèses des utilisateurs.

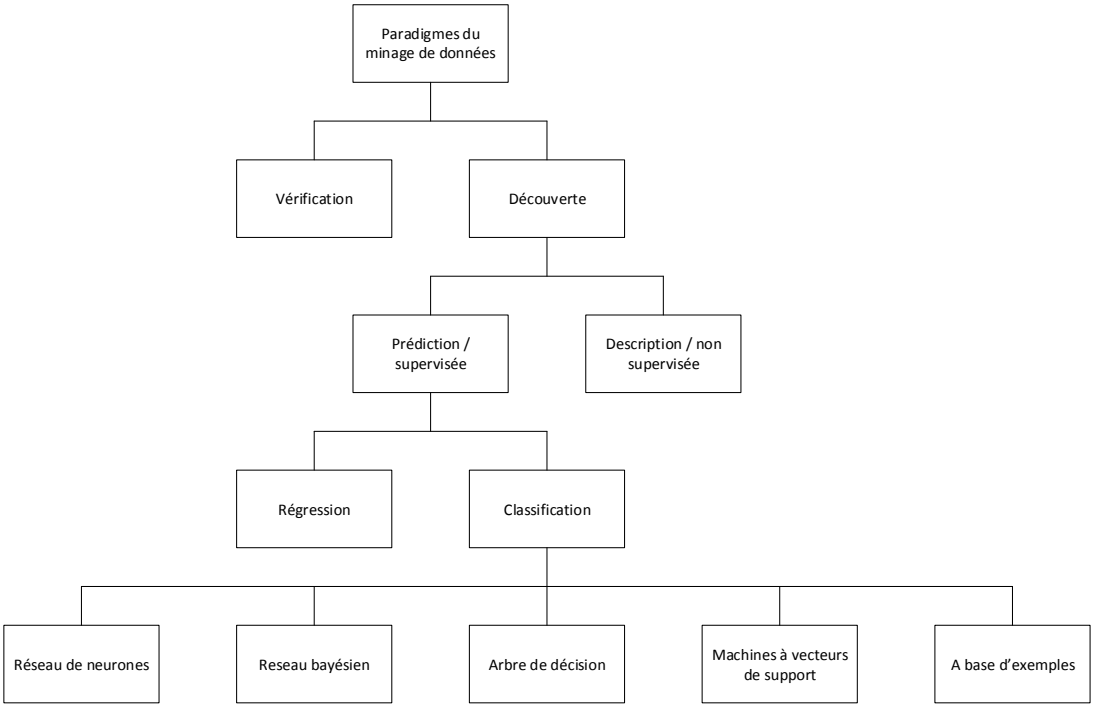


FIGURE 2.8 – Terminologie du domaine de la fouille de données

Il est communément admis dans la littérature que les méthodes d'exploration de données de type **découverte** peuvent être divisée en deux catégories, les méthodes d'apprentissage supervisées et non supervisés [**?**] [**?**] [**?**] [**?**].

- Les méthodes d'**apprentissage supervisé** sont des méthodes qui tentent de découvrir les relations entre les attributs d'entrée, parfois appelées *variables indépendantes*, et des attributs cibles, parfois appelés *variables dépendantes* [**?**]. La relation découverte est représentée dans une structure dénommée, modèle [**?**].

- Les méthodes d'**apprentissage non supervisé** sont des méthodes qui ont pour objectif de diviser un ensemble d'instances en sous-groupes homogènes. Les instances les plus similaires sont associées au sein d'un même groupe alors que les plus différentes se retrouvent dans des groupes différents [**?**].

Il existe dans la littérature une autre terminologie qui distingue les méthodes de fouille de données de type découverte orientée **prévision** de celles orientées **description** [**?**] [**?**]. La **prévision** correspond totalement à l'**apprentissage supervisé**, alors que la **description** inclut l'**apprentissage non supervisé**.

2.3.1/ APPRENTISSAGE SUPERVISÉ

L'objectif des méthodes supervisées d'apprentissage automatique est de construire un modèle concis de la distribution des labels de classes par la découverte des relations entre les attributs d'entrée et les attributs cibles. Pour cela des instances exemples sont utilisées à des fins d'apprentissage [**?**] [**?**]. Le classifieur résultant est utilisé pour attribuer des labels de classes aux instances test. En apprentissage supervisé, les données d'apprentissage comportent à la fois les données d'entrée de l'algorithme et le résultat attendu [**?**]. Deux approches principales sont distinguées dans la littérature : la classification et la régression [**?**].

- **La classification** consiste en la découverte supervisée du modèle prédictif qui a pour objectif de prévoir l'appartenance à des classes en fonction de certaines contraintes. Le modèle créé peut ensuite être utilisé pour classifier les données nouvellement disponibles [**?**] [**?**]. Dans la littérature, les techniques de classification sont communément divisées en cinq groupes [**?**] [**?**] (Figure **??**) : réseaux de neurones (i.e. Neural networks) [**?**], réseaux bayésiens (i.e. Baysian networks) [**?**], arbres de décision (i.e. Decision trees) [**?**], machines à vecteurs de support (i.e. support vector machines) [**?**], à base d'exemples (i.e. Instance based) [**?**].

- **La régression** consiste en la découverte du modèle prédictif qui permet de faire le lien entre un ensemble de données d'entrée et un domaine de valeurs réelles. Contrairement aux processus de classification qui ne fait le lien qu'avec un ensemble de classes prédéfinies. Par exemple, une régression peut consister à prédire la demande pour un produit donné en fonction de ses caractéristiques. Pour

cela des algorithmes de type régression linéaire ou d'un arbre de régression peuvent être utilisés. [**?**].

2.3.2/ APPRENTISSAGE NON SUPERVISÉ

Les méthodes d'apprentissage non supervisées essaient d'organiser convenablement les éléments en formant des groupes d'éléments semblables sur la base de propriétés uniquement statistiques [**?**] [**?**]. Ces méthodes sont basées sur des algorithmes de partitionnement (i.e. Partitioning Algorithms) [**?**] [**?**], de regroupement basé sur la densité (i.e. Density-based clustering) [**?**] [**?**], algorithmes de grille basés sur la densité (i.e. Grid Density Based Algorithms) [**?**], de regroupement message-passage (i.e. Message-passing clustering) [**?**], de regroupement hiérarchique (i.e.Hierarchical Clustering) [**?**] [**?**]. Dans ce type d'approches, il n'y a pas de classes prédéfinies.

2.3.3/ CLASSIFICATION

Les méthodes de classification supervisée sont communément utilisées dans les systèmes de recommandation [**?**] [**?**] [**?**] [**?**]. Nous nous intéressons à ces méthodes afin d'automatiser la tâche d'indexation nécessaire aux systèmes basés sur le contenu. Les termes "*label*" et "*étiquette*" utilisés dans ce chapitre, sont équivalents.

Il est possible de diviser les problématiques de classification en quatre types principaux [**?**] [**?**] [**?**] [**?**] (cf. figure **??**) :

- La classification **simple étiquette** (i.e. single label classification), fait le lien entre un item et un label de classes provenant d'un ensemble prédéfini de labels disjoint dont le nombre est supérieur à 1. $|L| > 1$, avec L l'ensemble des étiquettes possibles.
- La classification **binaire** (i.e. binary classification), consiste à classifier les items selon deux labels de classes s'excluant mutuellement, $|L| = 2$. Par exemple oui, ou non.
- La classification **multi-étiquette** (i.e. multi-label classification), consiste à classifier les items avec une ou plusieurs étiquettes de classes. Un sous ensemble Y de l'ensemble des étiquettes L est associé à un item, $|L| \geq |Y|$ et $Y \subseteq L$ [**?**].
- La classification **multi-classes** (i.e. multi-class classification), se distingue de la classification multi-label (i.e. Multi-étiquettes) par le fait que chaque instance n'est associée qu'avec un seul élément de L et non un sous ensemble.

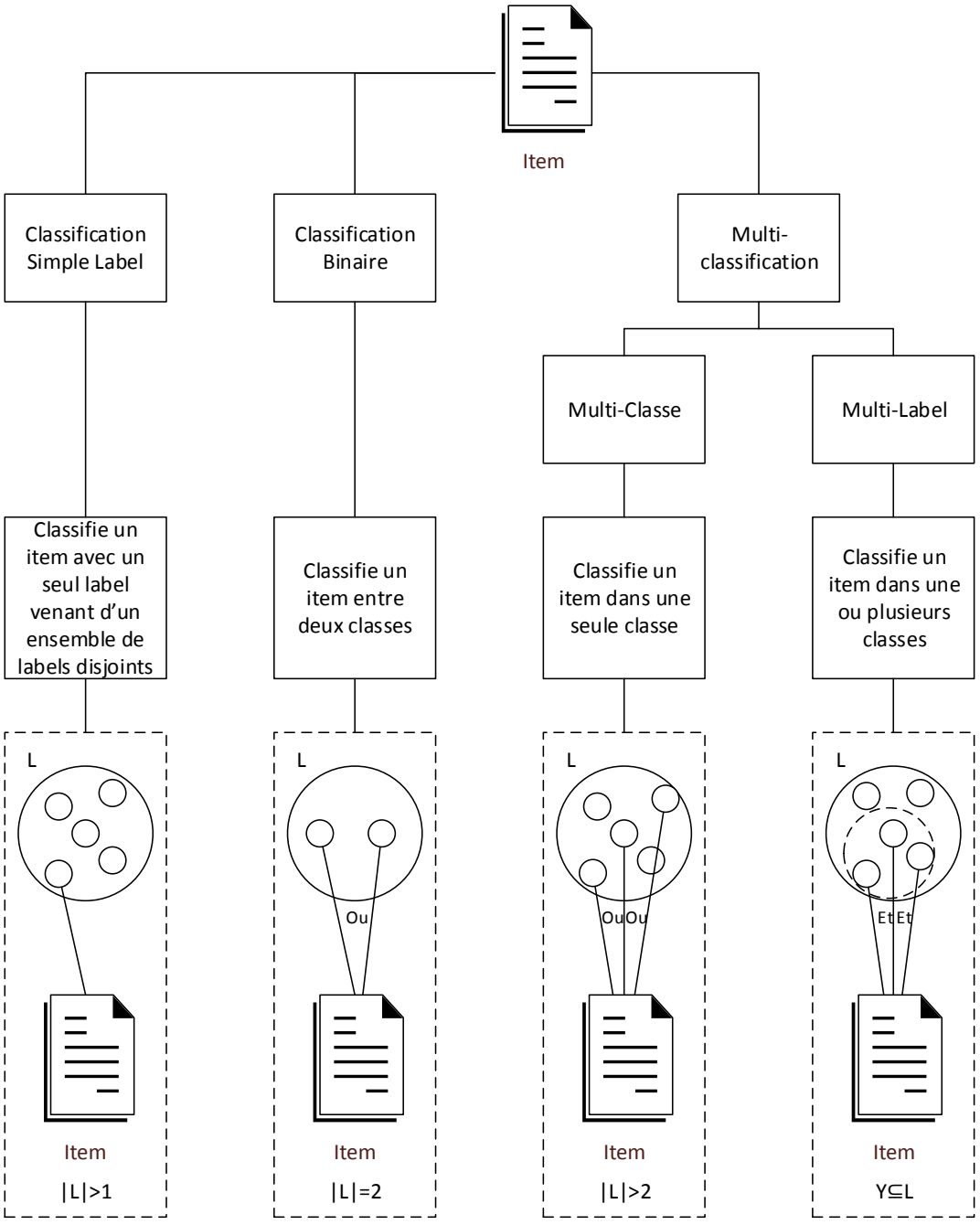


FIGURE 2.9 – Illustration des types de classification

Nous décrivons ci-dessous la classification multi-label suivie de la classification hiérarchique. Puis nous nous intéressons à la combinaison de ces deux problématiques, c'est-à-dire, la classification multi-label hiérarchique (i.e. HMC, Hierarchical Multi-label Classification). Enfin, des travaux de classification dans le contexte de l'utilisation d'ontologies sont présentés.

2.3.3.1/ APPRENTISSAGE MULTI-LABEL

Les méthodes multi-label permettent de classer les items avec au moins un label de classe. Un sous ensemble Y de labels est choisi à partir de l'ensemble de tous les labels possibles L. Il est par exemple souvent nécessaire dans le contexte de la classification de documents textuels d'avoir recours à des approches de classification multi-label permettant d'associer plus d'un label de classe à un document [**?**]. Le domaine de l'apprentissage multi-label se compose de trois tâches principales [**?**] :

- La classification **multi-étiquette** (i.e. multi-label classification, MLC) a pour objectif l'apprentissage d'un modèle qui permet de fournir en sortie une division de l'ensemble des labels disponible L en deux sous ensembles, les labels pertinent et les non pertinents par rapport à un item (i.e. instance) donné en entrée.

- Le **classement d'étiquettes** (i.e. label ranking, LR) a pour objectif l'apprentissage d'un modèle permettant de fournir en sortie les labels triés en fonction de leur pertinence pour une instance (i.e. item) donnée en entrée.

- Le **classement multi-étiquettes** (i.e. multi-label ranking, MLR) est une combinaison du LR et du MLC qui a pour objectif d'ordonner les labels pertinents par ordre de pertinence et de les distinguer des non pertinents. Seules les étiquettes pertinentes sont triées.

En apprentissage multi-étiquettes deux types de méthodes peuvent être distingués : (1) les *méthodes de transformation* du problème et (2) les *méthodes d'adaptation* d'algorithme [**?**] [**?**] [**?**] [**?**]. Nous développons ces méthodes ci-dessous.

Méthode de transformation de problème : Les méthodes de transformation de problèmes sont indépendantes des algorithmes. Elles transforment la tâche d'apprentissage en une ou plusieurs tâches de classification simple label. Il existe de nombreuses méthodes de transformation de problèmes et parmi elles, les plus populaires sont [**?**] [**?**] [**?**] :

- Les transformations par **blocs d'étiquettes** (i.e. Label Power Set) consistent à traduire les problèmes d'apprentissage multi-étiquettes en un seul problème simple label.
- Les transformations par **pertinence binaire** (i.e. binary relevance) consiste à traduire les problèmes d'apprentissage multi-étiquettes en de multiples problèmes simple label.

Le tableau **??** présente un exemple de classification multi-label dans lequel quatre items peuvent être associés à un ou plusieurs des quatre labels de classes.

item	label1	label2	label3	label4
item1	x		x	
item2				x
item3		x	x	
item4	x			x

TABLE 2.1 – Exemple de transformation de problèmes multi-labels

Le résultat de l'application d'une approche par blocs de labels avec le problème présenté dans le tableau **??** est présenté dans le tableau **??**. Les labels label1&3, label2&3, label1&4 ont été créés, l'étiquette label4 est la seule à restée non groupée.

item	label1&3	label4	label2&3	label1&4
item1	x			
item2		x		
item3			x	
item4				x

TABLE 2.2 – Résultat pour l'approche blocs de labels

Le résultat de l'application d'une approche par pertinence binaire avec le problème présenté dans le tableau **??** est présenté dans les tableaux **??** et **??**. Chacun des tableaux représente un problème de classification binaire pour une des étiquettes.

item	label1	¬ label1
item1	x	
item2		x
item3		x
item4	x	

item	label2	¬ label2
item1		x
item2		x
item3	x	
item4		x

TABLE 2.3 – Résultat pour l'approche pertinence binaire - partie 1

item	label3	¬ label3
item1	x	
item2		x
item3	x	
item4		x

item	label4	¬ label4
item1		x
item2	x	
item3		x
item4	x	

TABLE 2.4 – Résultat pour l'approche pertinence binaire - partie 2

Méthode d'adaptation d'algorithmes : Les méthodes basées sur des adaptations d'algorithmes étendent des algorithmes d'apprentissage existant pour gérer le cas des données multi-labellisées. Nous pouvons citer les algorithmes suivants : C4.5, AdaBoost.MH, AdaBoost.MR et Perceptron multiclasse multilabel (i.e. MMP, Multiclasse Multilabel Perceptron), pour ne citer que les principaux [**?**] [**?**].

2.3.3.2/ CLASSIFICATION HIÉRARCHIQUE

Dans le cas de la classification hiérarchique, contrairement aux cas précédents les labels de classes, ne sont pas considérés comme indépendants. Il y a un nombre important de cas de classifications pour lesquelles il existe des labels de classes prédéfinis qui peuvent être divisés en sous-classes, ou regroupés en super-classes [**?**] [**?**] [**?**]. Les labels de ces classes désignent parfois des notions complexes et sémantiquement dépendantes entre elles. C'est pourquoi nous présentons dans la section précédente **??** des structures pour la modélisation des connaissances, permettant de gérer ces relations.

L'organisation hiérarchique des labels de classe se distingue en deux cas : les arbres hiérarchiques et les graphes orientés acycliques (i.e. Directed Acyclic Graph, DAG), dans lesquels une classe peut avoir plusieurs parents (cf. figure **??**).

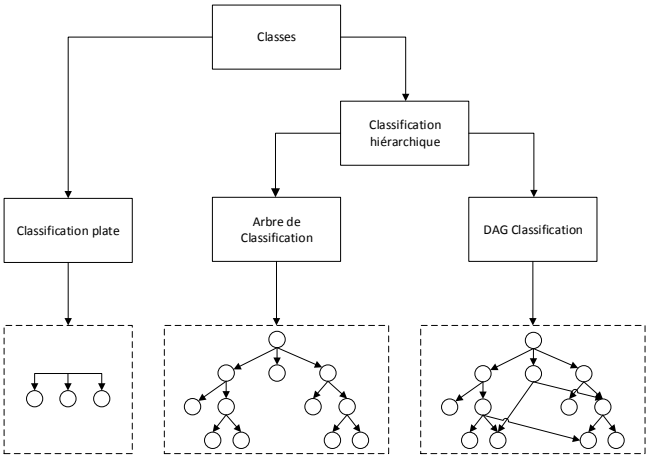


FIGURE 2.10 – Exemples de classification plate et hiérarchiques

2.3.3.3/ CLASSIFICATION HIÉRARCHIQUE MULTI-LABELS

La classification hiérarchique multi-labels (i.e. Hierarchical Multi-Label Classification, HMC) est la combinaison de la classification hiérarchique et de la classification multi-labels [**?**] [**?**] [**?**]. Ainsi, en HMC les items peuvent être classés simultanément avec plusieurs labels de classe se situant à différents niveaux dans différentes branches de la hiérarchie des labels comme cela est illustré par la figure **??**.

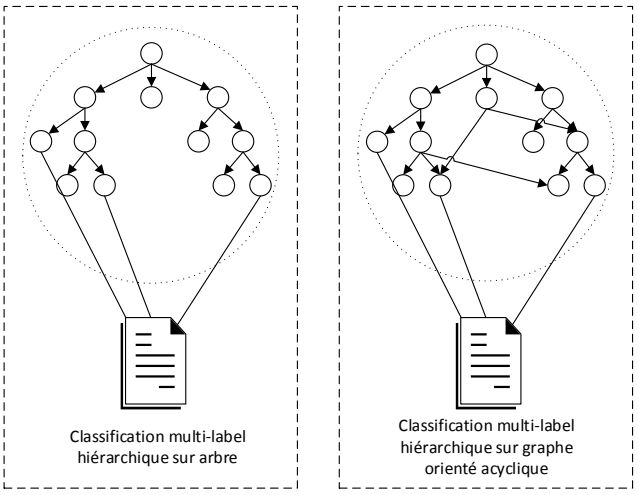


FIGURE 2.11 – Exemple de classification sur arbre et DAG

Les travaux de Santos [**?**] ainsi que de Cerri [**?**] distinguent deux principales approches afin de résoudre le problème de la classification multi-labels hiérarchique :

- L'approche hiérarchique **locale** produit une série d'algorithmes de classification en fonction de la hiérarchie descendante. L'apprentissage de chacun des classifieurs se fait sur la base des informations locales de chacun des labels de classes de la hiérarchie (i.e. les nœuds en relation directe) [**?**].

- L'approche hiérarchique **globale** utilise un unique classifieur qui prend en compte l'intégralité de la hiérarchie de labels [**?**] [**?**]. Il n'est pas possible d'utiliser directement les algorithmes conventionnels. Les travaux les plus courants consistent donc à adapter des algorithmes existants à ce cas particulier [**?**] [**?**].

2.3.4/ ONTOLOGIE ET CLASSIFICATION

Nous avons défini les ontologies comme étant la spécification formelle et explicite d'une conceptualisation partagée [**?**] [**?**] dans la section **??**. Elles sont utilisées afin d'organiser les notions d'un domaine et permettent à différents acteurs et processus d'interagir sans ambiguïtés sur la base de connaissances communes. Elles fixent arbitrairement une interprétation commune de ces connaissances. Leur modélisation formelle permet la création de contraintes logiques ce qui permet à partir des connaissances contenues dans l'ontologie et de logiciels appelées raisonneurs de déduire de nouvelles connaissances. L'étude des travaux [**?**] [**?**] [**?**] a permis d'identifier deux cas d'utilisation d'ontologies en classification :

- **Capturer et représenter** à l'aide d'ontologies, les règles de classification et le domaine de connaissances.

- **Raisonner** sur une ontologie. Au-delà de la capture et la représentation de connaissances, les moteurs d'inférence utilisent les ontologies afin de les aider dans le processus de classification.

Nous présentons dans la section suivante des travaux qui utilisent des ontologies durant leur processus de classification.

2.3.4.1/ CAPTURE ET REPRÉSENTATION DE CONNAISSANCES

Les ontologies peuvent être utilisées pour enrichir leurs vecteurs de description des items à classifier, à partir des connaissances qu'elles contiennent [**?**] [**?**] [**?**]. C'est à partir de ces vecteurs de description que la classification est réalisée.

Dans [**?**] les auteurs proposent l'utilisation des ontologies pour assister la classification semi-supervisée. Les ontologies sont utilisées afin de fournir les mots attendus dans les documents correspondants à une classe particulière.

Dans [**?**] les auteurs proposent un framework pour la représentation des sources de connaissances (i.e. documents, pages web). Les sources de connaissances sont sémantiquement représentées à l'aide de vecteurs sémantiques étendus dans un espace vectoriel (i.e. vector space model, VSM) à l'aide des connaissances de l'ontologie.

Dans [**?**] les auteurs présentent une méthode pour améliorer la classification de documents médicaux à l'aide de l'ontologie de domaine (MeSH - Medical Subject Headings). La méthode est composée de deux étapes :

- Première étape : faire le lien entre les termes et les concepts. Cette stratégie permet d'apparier et de désambiguïser les vecteurs de représentation des documents. Tous les mots en lien avec un même concept, par exemple appendicitis (1) et appendiceal (2) sont liés au concept appendicitis, la fréquence des deux termes est associé à la fréquence du concept.

- Seconde étape : le vecteur est étendu par l'ajout des hyperonymes des concepts présents (information présente dans l'ontologie de domaine).

L'expansion de vecteurs utilisée dans les travaux présentés précédemment n'est pas une méthode nouvelle. Elle a été présentée dans un contexte de recherche d'information dans la section **??** précédente.

2.3.4.2/ RAISONNEMENT LOGIQUE

Dans [**?**] les auteurs présentent une méthode de classification de documents qui utilise le raisonnement sur la base d'une ontologie ainsi que des mesures de similarité. Cette méthode se déroule de la façon suivante :

Une représentation des documents et catégories est créée au sein du système. Les documents sont représentés par des vecteurs de termes pondérés. Ces termes sont extraits des documents par un processus semi-automatique assisté par un expert. Les catégories dans lesquelles les documents doivent être indexées, sont représentées par des ontologies. Ces ontologies sont sélectionnées par un expert sur une plateforme d'ontologies librement disponibles. Le choix est effectué en fonction de la correspondance avec la catégorie qu'elles doivent représenter. Des raisonneurs sont utilisés afin de déduire les concepts les plus spécifiques de chacune des ontologies représentant les catégories. Chaque catégorie est alors représentée au sein du système comme un vecteur de termes désignant les concepts les plus spécifiques la composant. La classification consiste à assigner une valeur booléenne (i.e. vrai ou faux) à chaque paire document / catégorie. Le calcul de la similarité sémantique entre un document et une catégorie est effectué à l'aide de la distance Google normalisée [**?**]. Dans cette solution, le raisonnement sur ontologie n'est utilisé qu'indirectement pour la classification. Il est utilisé lors d'une tâche de prétraitement visant à créer les vecteurs de représentations des catégories. La classification n'est pas le résultat du raisonnement réalisés par les raisonneurs. De plus cette méthode nécessite que chaque terme du vocabulaire d'indexation soit décrit par un ontologie, ce qui peut être complexe pour des notions très précises. Dans l'exemple, les auteurs utilisent un nombre de catégories limité a 15 termes très généraux, tels que "sport", "science" ou "jeune".

Nous n'avons pas trouvé de travaux utilisant le processus de raisonnement sur ontologie de façon plus poussée lors de la tâche de classification d'informations.

2.3.5/ SYNTHÈSE À PROPOS DE L'AUTOMATISATION DE L'INDEXATION

L'indexation automatique de documents sur la base de nos vocabulaires structurés peut être réalisée à l'aide d'une approche de classification multi-labels hiérarchique. Dans le contexte d'une indexation sur la base d'un référentiel d'indexation de type ontologie, aucune approche à notre connaissance ne semble utiliser pleinement les capacités des ontologies. Les ontologies peuvent à la fois capturer et structurer le vocabulaire du domaine servant à l'indexation et intégré à l'aide de contraintes logiques. Un modèle prédictif de classification peut être utilisé par des raisonneurs afin d'automatiser l'indexation.

2.4/ DISCUSSION

L'étude des systèmes de recommandation a permis de mettre en lumière le type de système le plus adapté afin de répondre aux limites structurelles et pragmatiques (cf. chapitre **??**) de la revue *FirstECO*. En effet, ce type de systèmes permet de fournir à chaque lecteur une revue personnalisée en adéquation avec son besoin d'information. De plus le besoin des utilisateurs est analysé par le système. La nouvelle revue, tout en répondant aux limites de la revue d'origine et en intégrant de nouvelles contraintes, prend en compte la situation de départ, héritée de la revue classique.

- (i) La base d'utilisateurs est limitée à quelques milliers d'utilisateurs, lesquels peuvent avoir des besoins précis, rares, voire uniques.
- (ii) La base d'items à recommander est constamment en évolution. Des centaines de nouveaux items sont à recommander tous les jours. La valeur des items décroît rapidement après leur ajout dans la base des items à recommander.

Ainsi, un système basé sur le contenu est plus adapté qu'un système de filtrage collaboratif, afin de prendre en compte ce postulat de départ.

La variante basée sur la sémantique des systèmes basés sur le contenu permet de répondre aux limites pragmatique et sémantique (cf. chapitre **??**), par la gestion de métadonnées sur les articles publiés, ainsi qu'aux contraintes de rapidité et de simplicité. En effet, l'utilisation d'un vocabulaire contrôlé commun à tous les acteurs utilisant le système doit,permettre une utilisation aisée et rapide due à une ambiguïté du vocabulaire d'indexation limitée. De plus, la comparaison des vecteurs représentant l'offre et le besoin d'information, est plus rapide, car les vecteurs sont de dimension plus faible que dans le cas de vecteurs de terme de type TF-IDF. Les experts de l'entreprise sont des documentalistes. Elles sont déjà au fait des problématiques d'indexation à l'aide de vocabulaires contrôlés. L'utilisation d'un référentiel d'indexation basé sur un vocabulaire contrôlé correspond donc à leur savoir-faire.

Dans le cadre de la conception d'un système de recommandation basé sur la sémantique, l'étude présentée ci-dessus nous a amenés à faire des propositions sur les trois principales composantes d'un tel système. En effet, le fonctionnement d'un système de recommandation basé sur la sémantique nécessite (i) une base de connaissances. La base est utilisée afin de contrôler le vocabulaire d'indexation décrivant le domaine dans lequel s'inscrit le système de recommandation. Un tel système nécessite donc (ii) un processus d'indexation, permettant la qualification du besoin des utilisateurs, ainsi que du contenu des documents à recommander. Pour finir, un tel système nécessite (iii) un processus de comparaison, afin d'évaluer la pertinence d'un item par rapport à un profil, sur la base de leur description utilisant le vocabulaire d'indexation.

Nous avons présenté les différents types de vocabulaires pouvant être utilisés pour l'indexation des documents nécessaire au fonctionnement d'un système de recommandation basé sur la sémantique. Les types de vocabulaires présentent une forte hétérogénéité (e.g. normes, formats de persistance, objectifs d'utilisation, degrés d'expressivité, etc.) entre eux (cf. sous section **??**). Les ontologies répondent à la contrainte d'hétérogénéité. De plus, la recommandation doit être automatique. Il ne s'agit pas de faire faire une revue personnalisée manuellement pour chaque lecteur par les documentalistes de l'entreprise. L'automatisation de cette tâche, dans le cas de systèmes basés sur la sémantique, nécessite de permettre à la machine de comprendre le besoin des utilisateurs ainsi que le contenu des articles. Le caractère formel des ontologies, en fait un outil de gestion du vocabulaire d'indexation permettant de répondre à ce besoin. Elle reste néanmoins difficilement accessible à l'humain qui leur préfère les langages documentaires classiques, moins complexes (e.g. taxonomie, thésaurus). Les systèmes présentés, utilisent partiellement, mais sans l'introduire, la notion de facettes. Par exemple, certains systèmes qualifient les documents qu'ils souhaitent recommander à l'aide de vecteurs de termes, complétés par une seconde dimension descriptive : une date, ou l'appartenance à une catégorie de sujets (e.g. sport, politique). Cette seconde dimension influence parfois la recommandation. Nous proposons donc, dans notre solution, un modèle intégrateur, ontologique, permettant de définir un vocabulaire d'indexation sur la base de facettes. Les facettes étant composées de langages documentaires classiques, facilement manipulables par l'humain. L'ensemble étant formel et donc manipulable par la machine. Cette solution permet la définition d'un référentiel d'indexation, commun aux humains et à la machine respectant les spécificités de chacun. L'utilisation de facettes facilite la définition de domaines complexes, de plus l'utilisation de facettes est aujourd'hui habituelle pour les humains, ce qui rend notre solution facile à appréhender pour les utilisateurs. Les vocabulaires utilisés dans les systèmes présentés étant souvent limités à une petite dizaine de catégories. Nous proposons donc une solution, capable d'utiliser cette description riche du domaine, lors de l'indexation ainsi que lors de la recommandation.

Notre solution, consiste à permettre la description du besoin des utilisateurs à l'aide des vocabulaires structurés. Ces vocabulaires sont contenus dans une ontologie, la base de connaissances du système. Cette ontologie constitue, un vocabulaire d'indexation riche et structuré. La tâche d'indexation, même pour des experts peut-être longue et complexe. L'étude des approches permettant d'automatiser l'indexation de documents a mis en avant l'absence de solution exploitant totalement les possibilités offertes par les ontologies. En effet, elles permettent la définition de contraintes logiques, qui, utilisées par des raisonneurs, fournissent des déductions. C'est-à-dire qu'elles déduisent de l'information à partir de l'information contenue dans l'ontologie. Nous proposons donc une approche visant à automatiser l'indexation de documents, par l'intégration dans l'ontologie gérant le vocabulaire d'indexation, de contraintes logiques formant un modèle prédictif. Le modèle

étant appris à l'aide d'indexations manuelles de documents par les experts.

Une fois les documents indexés sur la base du vocabulaire d'indexation, riche et structuré contenu dans la base de connaissances, le système doit être capable de comparer les descriptions du besoin des utilisateurs et celles du contenu des articles. C'est cette comparaison qui permet de générer la recommandation, et donc, la revue personnalisée. Les algorithmes de recommandation basée sur de la comparaison de vecteurs dans un espace vectoriel mélangent les notions de similarité et de pertinence. Ils peinent ainsi à exploiter la richesse en terme de précision permise par une indexation sur la base de vocabulaires contrôlés et structurés. Ces algorithmes sont incapables en l'état de détecter la différence de précision qui peut exister entre l'expression du besoin et celle de l'offre. Nous proposons donc une méthode de comparaison adaptant au sein du modèle vectoriel les algorithmes classiques de comparaison en les adaptant à la richesse de notre vocabulaire d'indexation.

3

MODÉLISATION DES CONNAISSANCES POUR LA DESCRIPTION DIMENSIONNELLE D'ITEMS

Ce chapitre propose un modèle permettant de structurer la base de connaissances d'un système de recommandation sémantique. Ce modèle permet de répondre aux objectifs du système ainsi qu'aux verrous mis en lumière par l'état de l'art. Ainsi, un premier modèle, le modèle unificateur est proposé. Ce modèle permet de gérer l'hétérogénéité des langages documentaires. Un second modèle, le modèle intégrateur permet de proposer une base d'indexation riche, unifiée et contrôlée, en intégrant les vocabulaires unifiés par le modèle unificateur. Ce référentiel d'indexation (i.e. base de connaissances reposant sur le modèle intégrateur) est facilement accessible aux humains tout en étant manipulable par la machine, ce qui facilite leurs interactions nécessaires au fonctionnement du système.

Le chapitre précédent présente différents types de ressources terminologiques permettant l'indexation de documents. L'étude a souligné leur hétérogénéité, ainsi que les avantages et inconvénients de chacun. Le chapitre présent s'intéresse à la prise en compte des particularités de chaque type de vocabulaire ainsi qu'à la prise en charge de leur hétérogénéité. L'objectif étant de proposer un modèle de base de connaissances permettant de faciliter la description de domaines complexes, adapté à la description d'items.

Ce chapitre se focalise dans sa première partie sur la caractérisation des ressources terminologiques. La première section présente dans un premier temps les aspects hétérogènes (cf. sous section **??**), puis dans un second temps les similarités structurelles des langages documentaires, à partir desquels un modèle unificateur sera proposé.

La deuxième section présente la formalisation d'un modèle unificateur, et d'un modèle intégrateur pour la manipulation des ressources terminologiques. Le modèle unificateur transitoire a pour objectif de faciliter la manipulation de tout type de ressource terminologique, ce qui simplifie leur intégration dans notre modèle intégrateur. Le modèle intégrateur proposé a pour objectif d'accueillir les différentes ressources terminologiques dans une syntaxe, une structure et une sémantique uniques désambiguïsées et formelles afin de définir les facettes qui seront nécessaires lors de l'indexation des items.

La troisième section met l'accent sur l'indexation des items à l'aide de facettes. Cette approche permet d'offrir une description sémantique évoluée, exploitable par le processus de recommandation tout en restant en adéquation avec la modélisation du domaine de métier correspondant.

3.1/ RESSOURCES TERMINOLOGIQUES

Les ressources terminologiques sont des vocabulaires utilisés dans le cadre de la qualification de documents. Lorsqu'ils sont définis par des experts du domaine, ils sont dit *contrôlés*. Ces vocabulaires sont constitués d'un ensemble de termes formant une terminologie. Selon les cas, le vocabulaire peut être organisé de différentes façons, voire, ne pas être organisé du tout. Nous distinguons ici les *termes* et les *mots*, les termes étant constitués d'un ou de plusieurs mots. Dans le cadre de la gestion documentaire, les vocabulaires contrôlés sont nommés *langages documentaires*. Il existe aussi des vocabulaires non contrôlés utilisés pour l'indexation de documents, c'est le cas des *folksonomies*.

Ci-dessous nous illustrons les caractéristiques principales des ressources terminologiques et leur hétérogénéité. Puis, nous présentons leurs points communs.

3.1.1/ CARACTÉRISTIQUES HÉTÉROGÈNES DES RESSOURCES TERMINOLOGIQUES

Il existe une très forte hétérogénéité concernant les ressources terminologiques (cf. sous section **??**). Ci-dessous une liste non exhaustive des formes d'hétérogénéité problématiques dans notre cas :

- Différents types de ressources
- Différentes syntaxes
- Différentes normes
- Différents formats de persistance
- Différents objectifs d'utilisation
- Différents degrés d'expressivité
- Différents degrés de complexité
- Différents degrés de formalisation
- Différentes relations sémantiques
- Différents niveaux de granularité et/ou de couverture d'un domaine

Nous nous concentrons ici sur les aspects de cette hétérogénéité problématiques lors de l'usage de ces ressources, par des humains ou des processus, à des fins d'indexation (i.e. lorsque les vocabulaires sont utilisés aussi bien par l'humain que la machine, pour décrire des items).

Les hétérogénéités structurelles, syntaxiques et sémantiques sont liées à l'existence de différents types de ressources terminologiques prenant en compte différentes relations sémantiques entre les termes qui les composent. Ces différents types répondent à différentes normes ou standards définissant le format de représentation de l'information (i.e. sa syntaxe), ainsi que son organisation (i.e. sa structure) et son interprétation (i.e. sémantique associée à sa syntaxe).

Par exemple, la représentation d'un thésaurus illustre sa structuration à l'aide de symboles, d'une syntaxe et d'une sémantique associée aux symboles utilisés par cette syntaxe. Ces symboles permettent de définir les relations existantes entre les termes. Dans un thésaurus les relations hiérarchiques sont exprimées par les symboles NT (i.e. Narrower Term) et BT (i.e. Broader Term). Tandis que dans une classification décimale les relations hiérarchiques sont déduites de la succession de chiffres associés à chaque terme

(e.g. 311 - Science statistique. Théorie et méthode de la statistique). L'hétérogénéité dans la façon d'exprimer une même information entre deux différents types de ressources se matérialise par une syntaxe ainsi qu'une sémantique associée à des symboles qui sont différents.

La syntaxe, les symboles ainsi que la sémantique associée sont fixés par des normes ou des conventions. Eurovoc est un vocabulaire contrôlé couvrant l'ensemble des activités de l'Union Européenne en respectant les normes ISO 2788 :1986 et ISO 5964 :1985. D'autres vocabulaires en revanche, tel MeSH utilisé pour l'indexation d'articles médicaux ne respecte aucune norme. Cette hétérogénéité de normes ou de standards induit directement une hétérogénéité de syntaxe, de structure et de sémantique.

Les ressources terminologiques utilisent des formats de persistance de données qui se caractérisent là aussi par leur hétérogénéité. Par exemple le thésaurus Delphes dans sa version de 2004 était disponible au format PDF, c'est-à-dire au format texte comme les premières versions de la CDD (Classification Décimale de Dewey) publiées sous la forme de livres. Bien que ces vocabulaires étaient principalement destinés à une utilisation par des humains manipulant aisément le format texte, l'utilisation et la compréhension de l'organisation de vocabulaires composés de milliers de termes semble fort peu aisée dans un format texte. C'est pourquoi d'autres formats plus appropriés sont parfois utilisés. La Nomenclature des Activités Françaises NAF, proposée par l'INSEE est disponible au format XLS (fichier tableur Excel) ou DBF (format base de données). D'autres vocabulaires comme MeSH sont disponibles au format XML. Les vocabulaires les plus récents, comme Eurovoc tendent à prendre en compte les avancées proposées par les travaux du domaine du web sémantique et proposent leurs ressources au format RDF. L'existence de différents formats de représentation implique une hétérogénéité de syntaxe.

Nous constatons aussi sur la base de cette hétérogénéité de formats de représentation que bien que les thésaurus aient été en premier lieu pensés pour être utilisés par des humains, certains sont disponibles aux formats RDF, comme Eurovoc. Ce qui permet d'en faciliter la manipulation par la machine. Nous constatons une hétérogénéité de cibles, et d'objectifs pour ces vocabulaires. Certains visent une utilisation par des humains, d'autres par des machines, d'autres encore, les deux ce qui induit une hétérogénéité syntaxique et sémantique.

Bien que les facteurs d'hétérogénéités soient nombreux, les langages documentaires (i.e. listes plates, classifications, taxonomies, nomenclatures, thésaurus) reposent sur une structure similaire. Chacun étant un enrichissement du précédent, ce que nous présentons à la section suivante.

3.1.2/ POINTS COMMUNS DES PRINCIPAUX TYPES DE RESSOURCES TERMINOLOGIQUES

Nous distinguons trois types de structures concernant les ressources terminologiques à destination des humains : les listes plates, les taxonomies (i.e. classifications et nomenclatures) et les thésaurus.

Les *listes plates* de termes : la création de langages documentaires contrôlés par un/des experts suit un certain nombre de règles. L'objectif est (i) de déterminer une terminologie assez vaste et précise pour couvrir un domaine avec le niveau de granularité souhaité, (ii) de prendre en compte l'indépendance sémantique de chacun des termes les uns par rapport aux autres afin d'éviter au maximum qu'ils ne recouvrent des notions communes. La liste de termes contrôlée est donc un ensemble de termes respectant ces règles sans relations sémantiques exprimées explicitement entre eux. Dans le cas de langages non contrôlés, déterminés par les utilisateurs comme les folksonomies (i.e. liste de termes non contrôlée), ces règles n'existent pas (ou du moins ne peuvent être respectées). Ainsi pour une même notion, de multiples synonymes, méronymes [1] et/ou holonymes [2] peuvent exister sans que la nature des relations qui existent entre ces termes ne soient exprimées explicitement.

Les *taxonomies*, *classifications* et *nomenclatures* : ces langages documentaires reposent sur des listes de termes. Ce sont vocabulaires contrôlés dans lesquels les termes sont organisés de façon hiérarchique.

Les *thésaurus* : ce sont des langages documentaires contrôlés et structurés. La terminologie y est organisée de façon hiérarchique comme dans une taxonomie. Sur la base de cette taxonomie des relations de synonymies sont définies de façon à permettre la prise en compte d'un terme principal (i.e terme à utiliser de préférence) pour une notion donnée ainsi que des relations associatives.

Comme le présente la figure **??**, les différents types de vocabulaires contrôlés sont donc une combinaison de termes avec ou sans relations hiérarchiques, associatives ou synonymiques. Dans cette figure, les relations associatives y sont représentées par des flèches en pointillés, les relations hiérarchiques par des flèches classiques, les termes principaux (i.e. termes choisis parmi leur groupe de synonymes pour représenter une notion) sont représentés par des ronds blancs, les termes synonymes sont représentés par des ronds pleins. Les relations de synonymie entre termes principaux et les termes synonymes sont représentées par des traits noirs.

1. Relations sémantiques de la partie à son tout.
2. Relations sémantiques d'un tout à sa partie.

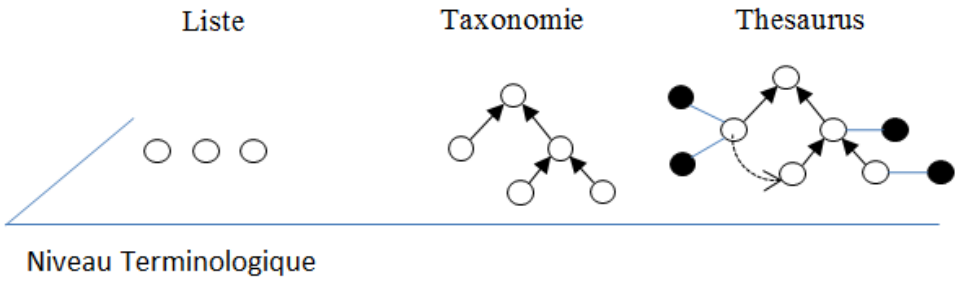


FIGURE 3.1 – Présentation des différents types de ressources et de leur structuration

3.2/ MODÈLES "*unificateur*" ET "*intégrateur*"

La section précédente a mis en avant les différentes formes d'hétérogénéités existantes entre les types de ressources terminologiques tout en soulignant leurs similitudes structurelles. Néanmoins, l'hétérogénéité des ressources terminologiques reste un problème lors de l'utilisation simultanée de plusieurs ressources pour la description d'un item. Bien que ces ressources soient hétérogènes, nous pouvons mettre en avant l'existence d'une structure sous-jacente commune, basée sur une combinaison de termes et de relations.

Nous présentons dans cette section, un modèle mathématique unificateur et formel basé sur des ensembles, pour la manipulation des ressources terminologiques en vue de leur intégration dans une base de connaissances. Ensuite nous présentons le modèle intégrateur de la base de connaissances, permettant la prise en compte de ressources terminologiques pour l'indexation d'items.

L'objectif du modèle unificateur est de traduire en fonction de leur type les ressources terminologiques disponibles de façon à ce qu'elles soient intégrables dans le modèle intégrateur. De cette manière, nous supprimons l'hétérogénéité syntaxique et structurelle. Le modèle intégrateur s'attache a résoudre l'hétérogénéité sémantique.

3.2.1/ MODÈLE "*unificateur*"

Le modèle unificateur proposé est un langage artificiel, basé sur des mathématiques ensemblistes qui possèdent une sémantique formelle. Ce modèle permet la description de l'ensemble des types de ressources terminologiques précédemment présentés.

Par conséquent cette section présente une modélisation mathématique pour chaque type de ressources distinguées dans notre état de l'art afin de proposer une modélisation unique correspondante pour n'importe quel type de ressource terminologique. Nous distinguons trois types : les listes, les taxonomies (i.e. classification, nomenclatures) et les thésaurus.

Définition mathématique des listes plates

$$l \in L = \{ t \in T \mid l \, R_L \, t \}$$

$$R_L \subseteq L \times T \subseteq \mathcal{P}(T)$$

Soit L l'ensemble des listes plates de termes. Soit l une liste plate composée d'un ensemble de termes t appartenant à l'ensemble de tous les termes T. Soit R_L la relation entre les termes t et la liste l.

Définition mathématique des taxonomies

$$t_x \in T_x = \{ t \in T \mid t_x \, R_x \, t \, , \leq_{T_x} \}$$

$$R_x \subseteq T_x \times T \subseteq \mathcal{P}(T)$$

Soit t_x une taxonomie appartenant à l'ensemble des taxonomies T_x. La taxonomie t_x est composée d'un ensemble de termes t appartenant à l'ensemble de tous les termes T organisés selon une hiérarchie $\leq_{T_x}$. Soit R_x la relation entre les termes t du sous ensemble de termes et la taxonomie T_x. Soit $\leq_{T_x}$ une relation d'ordre partiel entre ces termes telle que :

$$t_1 \leq_{T_x} t_1 \quad \text{(réflexivité)}$$

$$t_1 \leq_{T_x} t_2, \, t_2 \leq_{T_x} t_3 \Rightarrow t_1 \leq_{T_x} t_3 \quad \text{(transitivité)}$$

$$t_1 \leq_{T_x} t_2, \, t_2 \leq_{T_x} t_1 \Rightarrow t_1 = t_2 \quad \text{(antisymétrie)}$$

Exemple : Si n_1 = Automobile et n_2 = Véhicule, alors $n_1 \leq_{T_x} n_2$ signifie que le terme Automobile spécialise le terme Véhicule.

Définition mathématique des thésaurus

$$n \in N = \langle \, t_p \in T \, , \{ \, t_s \in T \mid t_p \, R_s \, t_s \, \} \, \rangle$$

$$R_s \subseteq T \times T$$

Soit n un nœud appartenant à l'ensemble des nœuds N. Un nœud est un couple composé d'un terme principal t_p appartenant a l'ensemble des termes T ainsi que d'un ensemble de termes synonymes t_s appartenant à l'ensemble de termes T. Soit R_s la relation de synonymie entre un terme principal t_p et un terme synonyme t_s.

$$t_h \in T_h = \langle N_h = \{ n \in N \mid n \, R_h \, t_h, \leq_{T_h} \}, \, g \in G \mid V_g \subseteq N_h \rangle$$

$$A \subseteq V \times V \subseteq N \times N$$

$$R_h \subseteq N \times T_h$$

Soit t_h un thésaurus appartenant à l'ensemble des thésaurus T_h. Un thésaurus t_h est un couple composé d'un ensemble de nœuds n appartenant à l'ensemble de tous les nœuds N, organisés selon une hiérarchie $\leq_{T_h}$ ainsi que d'un graphe g appartenant à l'ensemble des graphes G. Soit R_h la relation qui lie l'ensemble des les nœuds n au thésaurus t_h. Soit $g \in G$ un graphe orienté défini par $g = (V_g, A_g)$ tel que $V_g \subseteq N_h$ et A_g est l'ensemble des relations non hiérarchiques entre certains nœuds du thésaurus. Nous rappelons la définition d'un graphe orienté : $G = (V, A)$, avec V un ensemble de nœuds et A un ensemble d'arcs. Chaque $a \in A_g$ est un couple de nœuds (v_i, v_j) avec $v_i, v_j \in V_g$ orienté de v_i vers v_j. Soit $\leq_{T_h}$ une relation d'ordre partiel telle que :

$$n_1 \leq_{T_h} n_1 \quad \text{(réflexivité)}$$

$$n_1 \leq_{T_h} n_2,\ n_2 \leq_{T_x} n_3 \Rightarrow n_1 \leq_{T_h} n_3 \quad \text{(transitivité)}$$

$$n_1 \leq_{T_h} n_2,\ n_2 \leq_{T_h} n_1 \Rightarrow n_1 = n_2 \quad \text{(antisymétrie)}$$

Définition mathématique du modèle unifié des ressource terminologiques Afin de n'utiliser qu'une seule modélisation nous définissons une ressource terminologique de la façon suivante :

1) La définition mathématique que nous avons formulée d'un thésaurus est suffisamment expressive pour permettre la gestion des thésaurus comme des vocabulaires plus simples.

$$r_t \in R_T = \langle N_h = \{\, n \in N \mid n\, R_h\, t_h,\ \leq_{T_h} \},\ g \in G \mid V_g \subseteq N_h \rangle$$

2) Si g est un graphe vide $V_g = \varnothing,\ A_g = \varnothing$ et que chacun des nœuds n contient une liste de termes synonymes vide, alors, la ressource est une taxonomie.

3) Si g est un graphe vide $V_g = \varnothing,\ A_g = \varnothing$, que chacun des nœuds n contient une liste de termes synonymes vide et qu'aucun ordre partiel $\leq_{T_h}$ n'est défini entre ces nœuds, alors, la ressource est une simple liste.

3.2.2/ MODÈLE "*intégrateur*"

Nous venons de présenter le modèle mathématique unificateur permettant la manipulation des différents types des ressources terminologiques. Ce modèle est nécessaire

pour le processus d'intégration des ressources terminologiques dans notre modèle intégrateur. Le modèle intégrateur permettant l'intégration des ressources terminologiques unifiées sera exploitable par le processus d'indexation des documents. Il structure de cette manière notre base de connaissances.

Les connaissances contenues dans la base de connaissances sont les suivantes : (i) description du domaine traité définie par des facettes, (ii) terminologies associées aux facettes et (iii) items indexés sur la base de ce vocabulaire. La sous-section suivante présente la notion de facettes liées aux ressources terminologiques qui seront exploitées dans une seconde section décrivant le modèle intégrateur.

3.2.2.1/ LES FACETTES DES RESSOURCES CONCEPTUELLES

Une facette constitue une dimension descriptive d'un item. Cette description se matérialise par l'usage d'une ressource terminologique telle qu'une taxonomie. Le contexte de description d'un item dépend du domaine d'application pour lequel cette description est nécessaire. Toutefois, il est difficile de prendre en compte l'ensemble des connaissances d'un domaine nécessaire à une application avec une seule ressource terminologie [**?**] [**?**]. Raganathan [**?**] a défini la notion de facette, afin de répondre aux problèmes liés à la rigidité des classifications taxonomiques de type "Classification Décimal de Dewey" et "Classification Décimal Universel". Cette difficulté à couvrir l'ensemble d'un domaine, nommée rigidité, n'est pas propre aux taxonomies. L'ensemble des types de ressources terminologiques utilise un nombre limité de relations sémantiques avec lesquelles il est difficile, voir presque impossible, de couvrir l'ensemble des dimensions descriptives d'un item. C'est pourquoi nous proposons l'utilisation d'un ensemble de ressources terminologiques, chacune couvrant une dimension descriptive. Nous proposons donc l'utilisation de plusieurs facettes pour la description d'un item. La définition des facettes ainsi que la structuration et la définition des terminologies qui sont associées dépendent de la vision métier sur le domaine traité. Par conséquent, les facettes résolvent les problèmes liés de manière générale à la rigidité des ressources terminologiques et permettent une prise en compte de la vision métier. L'ensemble des connaissances métiers du domaine modélisé à l'aide de facettes constitue la fondation de notre base de connaissances nécessaire au processus d'indexation des items. Toutefois, il est à noter que certaines ressources terminologiques intègrent dans leur définition la notion de facette afin de décrire différentes dimensions d'un domaine, c'est le cas de thésaurus à facettes.

Une des propriétés des facettes est d'être manipulable intuitivement par les humains. Leurs modélisations à l'aide du modèle intégrateur, formel permettent de les rendre manipulables tout aussi aisément par la machine, c'est-à-dire aux processus automatiques comme un processus de recommandation. Il est nécessaire de préciser que ces deux as-

pects sont importants pour le système décrit dans ce mémoire, car celui-ci sera utilisé à la fois par des processus informatiques et par des utilisateurs allant du simple utilisateur à l'expert. Les facettes sont très populaires sur les sites de e-commerce, car chacune des facettes permet d'appréhender une des dimensions décrivant les items du site. C'est pour cela que les utilisateurs sont aujourd'hui familiers avec leur utilisation [**?**].

Par conséquent, nous proposons un modèle intégrateur utilisant la notion de facettes prenant la forme de ressources terminologiques.Tout comme le modèle unificateur, l'ensemble des ressources terminologiques intégrées à notre modèle intégrateur sont unifiées dans leur structure, leur syntaxe et leur sémantique. Ce modèle repose sur un langage artificiel, permettant la définition d'une ontologie dont la sémantique formelle repose sur la logique de description. Afin de prendre en compte les dernières avancées concernant les langages documentaires (normes ISO 25964 et BS 8723), nous distinguons la notion de concept et celle de terme généralement confondues dans les ressources existantes. Ainsi lors de l'intégration de ressources terminologiques au modèle intégrateur, une abstraction conceptuelle est réalisée. Les termes et les concepts auxquels ils renvoient sont distingués (cf. sous section **??**). L'objectif étant lors de la modélisation de distinguer le concept, c'est a dire la signification d'un terme, du ou des termes pouvant y faire référence.

La section suivante propose une formalisation de notre modèle suivi d'une explication et d'un exemple concernant l'intégration de ressources terminologiques au modèle. Puis nous finissons en présentant l'utilisation de ce modèle pour l'indexation d'items.

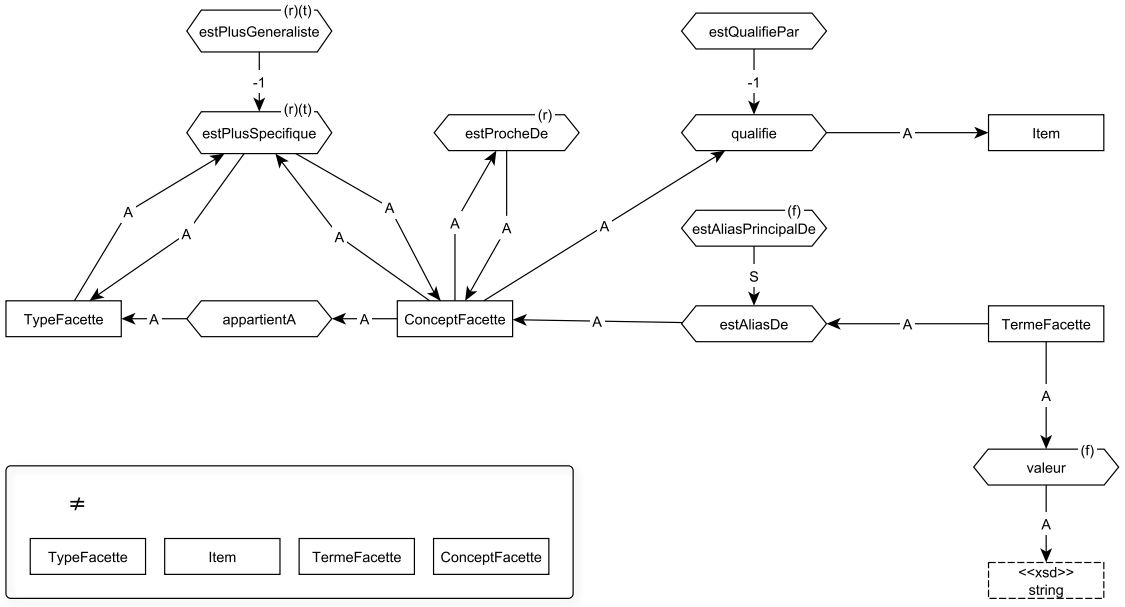


FIGURE 3.2 – Présentation du schéma de l'ontologie au format G-OWL

3.2.2.2/ FORMALISATION DU MODÈLE INTÉGRATEUR

Le modèle intégrateur illustré par la figure **??**, prend la forme d'une ontologie formelle. Dans cette figure (f) signifie fonctionnelle, (t) signifie transitive, (r) signifie réflexive et $\neq$ signifie disjonction entre les concepts. Plus d'informations sur G-OWL dans la publication suivante : [**?**]. Nous la modélisons à l'aide d'un méta-modèle inspiré du meta-modèle de Karlsruhe (cf. annexe **??**) utilisé notamment par Ehrig [**?**] et étendu aux bases de connaissances d'expressivité logique $\mathcal{SHOIN}(\mathcal{D})$ par Pittet [**?**]. Une ontologie est un ensemble d'axiomes [**?**] qui peut être définie comme une structure mathématique. Nous définissons une structure [**?**] comme un n-uplet $\mathcal{S} = (\Omega, \Sigma, \Phi, E)$:

- Soit Ω un ensemble appelé l'ensemble sous-jacent de $\mathcal{S}$.
- Soit Σ une collection d'axiomes de signature $\{\sigma_i : i \in I_1\}$ ou $\sigma_i \subseteq \Omega^{m_i}$ pour $m_i \geq 1$.
- Soit Φ une collection d'axiomes de fonction $\{\varphi_i : i \in I_0\}$ ou $\varphi_i : \Omega^{n_i} \rightarrow \Omega$ pour $n_i \geq 1$.
- Soit E une collection d'éléments distincts $\{\varepsilon_i : i \in I_2\} \subseteq \Omega$.

Les ensembles I_0 I_1 et I_2 peuvent être vides. n_i et m_i sont les arités respectives de φ_i et σ_i.

Notre schéma $\mathcal{S}_{integration}$:

Définition de $\Omega_{integration} = \{(sC, \leq_C), (sT, \leq_T), (sR, \leq_R), (sA, \leq_A), sI, sV, sK_R, sK_A\}$:

- sC = {TopConcept, TypeFacette, ConceptFacette, Item, TermeFacette},
- $\leq_C$ = {(TopConcept, TypeFacette), (TopConcept, TypeFacette), (TopConcept, ConceptFacette), (TopConcept, Item), (TopConcept, TermeFacette)},

sC et $\leq_C$ définissent respectivement l'ensemble des concepts et l'ensemble de définition des relations de subsomption entre ces concepts. C'est-à-dire leur organisation selon une hiérarchie de subsomption. Le **TopConcept** est le concept le plus général, celui qui subsume tous les autres. Nous avons défini deux concepts particulièrement important pour la modélisation des facettes. Le **ConceptFacette** et le **TermeFacette** permettant la distinction entre concepts et termes lors de l'intégration des ressources terminologiques.

- sT = {TopDataType, xsd :string},
- $\leq_C$ = {(TopDataType, xsd :string)},

sT et $\leq_C$ définissent respectivement l'ensemble des types de données, ainsi que l'ensemble de définition des relations de subsomption entre ces types de donnée. Le **TopDataType** est le type de donnée le plus général, celui qui subsume tous les autres. Nous

avons besoin ici du type **xsd :string** [3] afin de gérer les chaînes de caractères correspondant à la valeur des termes des ressources terminologiques.

- sR = {TopRelation, estPlusSpecifiqueQue, estPlusGeneralisteQue, appartientA, estAliasDe, estAliasPrincipalDe, qualifie, estQualifiePar},
- $\leq_R$ = {(TopRelation, estPlusSpecifiqueQue), (TopRelation, estPlusGeneralisteQue), (TopRelation, appartientA), (TopRelation, estAliasDe),(TopRelation, estAliasPrincipalDe), (TopRelation, qualifie), (TopRelation, estQualifiePar), (estAliasDe, estAliasPrincipalDe)},

sR et $\leq_R$ définissent respectivement l'ensemble des rôles et l'ensemble de définition des relations de subsomption entre ces rôles. Le rôle **topRelation** est le rôle le plus général, celui qui subsume tous les autres. Les relations **estPlusSpecifiqueQue** et **estPlusGeneralisteQue** permettent la prise en compte des relations hiérarchiques provenant des ressources terminologiques, notamment des taxonomies et des thésaurus. Les relations **estAliasDe** et **estAliasPrincipalDe** permettent la prise en compte des relations de synonymie provenant des ressources terminologiques, notamment des thésaurus. La relation **estAliasPrincipalDe** permet la gestion du terme principal (i.e terme utilisé de préférence) alors que la relation **estAliasDe** permet la gestion des non-termes (i.e. termes non principaux). Les relations **qualifie** et **estQualifiePar** permettent elles comme nous le verrons ci-dessous la qualification des articles.

- sA = {TopAttribute, possedeValeur},
- $\leq_A$ = {(TopAttribute, possedeValeur)},

sA et $\leq_A$ définissent respectivement l'ensemble des attributs et l'ensemble de définition des relations de subsomption entre ces attributs. L'attribut **topAttribute** est l'attribut le plus général, celui qui subsume tous les autres. L'attribut **possedeValeur** permet comme nous le voyons ci-dessous de donner une valeur au terme.

- $sI = \emptyset$,
- $sV = \emptyset$,

Nous ne présentons ici que le schéma de l'ontologie, c'est-à-dire, l'ensemble des éléments permettant de définir la TBox de la base de connaissances. Les ensembles d'instanciation sI et sV correspondant à la ABox sont donc vides car le modèle n'est pas, pour le moment, peuplé par des instances.

3. xsd est le préfixe de l'espace de nom http ://www.w3.org/2001/XMLSchema

- sk_R = {Functional, Inverse Functional, Transitive},

L'ensemble sk_R permet de définir l'ensemble des caractéristiques de rôle. C'est-à-dire l'ensemble des contraintes logiques qui peuvent être appliquées à une relation définie dans l'ensemble des rôles sR.

- sK_A = {Functional},

L'ensemble sk_A permet de définir l'ensemble des caractéristiques d'attributs. C'est-à-dire l'ensemble des contraintes logiques qui peuvent être appliquées à un attribut défini dans l'ensemble des attributs sA.

Définition de $\Sigma_{integration} = \{\sigma_R, \sigma_A\}$:

- σ_R = {(estPlusSpecifiqueQue, (TypeFacette, TypeFacette)), (estPlusSpecifiqueQue, (ConceptFacette, ConceptFacette)), (estPlusGeneralisteQue, (TypeFacette, TypeFacette)), (estPlusGeneralisteQue, (ConceptFacette, ConceptFacette)), (qualifie, (ConceptFacette, Item)), (estQualifiePar, (Item, ConceptFacette)), (estAliasDe, (TermeFacette, ConceptFacette)), (estAliasPrincipalDe, (TermeFacette, ConceptFacette)), (appartientA, (ConceptFacette, TypeFacette))},

- σ_A = {(possedeValeur,(TermeFacette, xsd :string))},

σ_R et σ_A définissent respectivement la signature des rôles définis dans l'ensemble des rôles sR et la signature des attributs définis dans l'ensemble des attributs sA. La signature consiste en la définition du **range** et du **domain** qui permettent de définir des contraintes sur les relations et les attributs. Dans le cas de relations, ou rôles, ils permettent de définir pour une relation donnée, quels sont les concepts dont les instances peuvent instancier cette relation (i.e. le domaine), et quels sont les concepts dont les instances peuvent être cibles de l'instanciation de cette relation (i.e. le range). Ces relations sont orientées. Dans le cas d'attributs, ils permettent pour un attribut donné de définir quels sont les concepts dont les instances peuvent utiliser cet attribut et quels sont les types de données de l'attribut.

Notre modèle permet à une instance du concept **TermeFacette** d'avoir un attribut **vpossedeValeur** qui a pour type une chaîne de caractères. Cet attribut permet au modèle de gérer la valeur des termes.

Les relations **estPlusSpecifiqueQue** et **estPlusGeneralisteQue** permettent la prise en compte par les facettes des relations de hiérarchie existantes entre les termes provenant de ressources terminologiques de type taxonomie ou thésaurus. Elles s'appliquent

ici notamment aux **ConceptFacette**, car notre modèle permet une abstraction conceptuelle. L'organisation hiérarchique s'applique sur les notions (i.e. **ConceptsFacette**) auxquelles font référence les termes (i.e. **TermeFacette**) et non plus sur les termes principaux comme c'est le cas dans la plupart des thésaurus, ou directement sur les termes comme c'est le cas avec des ressources terminologiques de type taxonomie.

La relation **estQualifiePar** permet la description des items. C'est son instanciation qui permet l'indexation. Elle s'applique entre une instance d'**Item** et une instance de **ConceptFacette**, elle a donc lieu au niveau conceptuel et non au niveau terminologique comme c'est généralement le cas lors de l'utilisation de ressources terminologiques (i.e. langages documentaires) pour l'indexation de documents.

Les relations **estAliasDe** et **estAliasPrincipalDe** permettent la prise en compte des relations de synonymie, dans notre modèle. Elle permettent de faire le lien entre le niveau terminologique et le niveau conceptuel. Toutes les instances du concept **TermeFacette** en relation avec une même instance de **ConceptFacette** étant des synonymes.

Définition de $\Phi_{integration}$ = $\{iC, iT, iR, iA, K_R, K_A, \varepsilon_C, \varepsilon_R, \varepsilon_A, \varepsilon_I, \delta_C, \delta_I, -_C, -_R, maxCard_R, minCard_R, \sqcap_C, \sqcup_C, \sqcup_I, \sqcup_V, \rho_{\exists R}, \rho_{\forall R}, \rho_R, \rho_{\exists_A}, \rho_{\forall_A}, \rho_A\}$

- $iC = \emptyset$,
- $iT = \emptyset$,
- $iR = \emptyset$,
- $iA = \emptyset$,

iC, iT, iR et iA sont les ensembles d'instanciation respectivement de concepts, types de données, rôles et attributs. Ils sont vides ici, car nous définissons dans cette section notre modèle, c'est-à-dire le schéma ontologique de la base de connaissances ou TBox.

- K_R = {(estPlusSpecifiqueQue, Transitive), (estPlusGeneralisteQue, Transitive), (estAliasPrincipal, Functional)},
- K_A = {(possedeValeur, Functional)},

K_R et K_A sont les ensembles de caractérisation respectivement des rôles et des attributs. K_R permet de définir les contraintes logiques s'appliquant sur les rôles de l'ensemble de rôles s_R en fonction de contraintes de rôles définies dans l'ensemble de contraintes de rôles sK_R. K_A permet de définir les contraintes logiques s'appliquant sur les attributs de l'ensemble d'attributs s_A en fonction des contraintes d'attributs définies dans l'ensemble des contraintes d'attributs sK_A.

Les relations **estPlusSpecifiqueQue** et **estPlusGeneralisteQue** sont formellement définies comme transitives. La relation **estAliasPrincipalDe** est défini comme formel ce qui permet de contraindre la relation. Ainsi un **ConceptFacette** ne peut avoir qu'un et un seul **TermeFacette** défini comme étant son terme principal (i.e. le terme préféré).

L'attribut **vpossedeValeur** est lui aussi défini comme fonctionnel, ainsi un **TermeFacette** ne peut avoir qu'une et une seule chaîne de caractères définissant la valeur du terme.

- $\varepsilon_C = \emptyset$,
- $\varepsilon_R = \emptyset$,
- $\varepsilon_A = \emptyset$,
- $\varepsilon_I = \emptyset$,

ε_C, ε_R, ε_A et ε_I permettent de définir respectivement l'ensemble des concepts équivalents, rôles équivalents, attributs équivalents et instances équivalentes. L'ensemble des instances équivalentes ε_I est vide, car nous ne nous intéressons pas ici à la ABox, nous définissions un modèle (i.e. un schéma d'ontologie). Les autres ensembles sont vides aussi car notre modèle ne contient aucun concepts, rôles ou attributs équivalents.

- $-_C = \emptyset$,
- $-_R$ = {estPlusGenerique, estPlusSpecifique), (estQualifiePar, qualifie)},

$-_C$ et $-_R$ permettent respectivement de définir l'ensemble des concepts complémentaires deux à deux ainsi que l'ensemble des relations inverses. Notre modèle ne présente aucun concept comme étant le complément d'un autre concept. Notre modèle définit, par contre, que les relations **estPlusGenerique** et **estPlusSpecifique** sont des relations inverses, de même les relations **qualifie** et **estQualifiePar** sont elles aussi des relations inverses.

- δ_C = {TypeFacette, Item, TermeFacette, ConceptFacette},
- $\delta_I = \emptyset$,

δ_C et δ_I définissent respectivement les ensembles de disjonction de concepts ainsi que de différentiation d'instances. Nous n'avons pas d'instances ici car nous définissons le modèle, l'ensemble δ_I est donc vide. Nous distinguons par contre les concepts **TypeFacette**, **Item**, **TermeFacette** et **ConceptFacette** comme définissant des ensembles disjoints d'instances.

- $maxCard_R = \emptyset$,

- $minCard_R = \emptyset$,

$maxCard_R$ et $minCard_R$ sont des ensemble permettant de définir des contraintes de cardinalité sur les rôles.

- $\sqcap_C = \emptyset$,
- $\sqcup_C = \emptyset$,
- $\sqcup_I = \emptyset$,
- $\sqcup_V = \emptyset$,

$\sqcap_C$, $\sqcup_C$, $\sqcup_I$ et $\sqcup_V$ permettent de définir des ensembles respectivement, d'intersection de concepts, d'union de concepts, d'énumération d'instances et d'énumération de valeurs de données.

- $\rho_{\exists R} = \emptyset$,
- $\rho_{\forall R} = \emptyset$,
- $\rho_R = \emptyset$,
- $\rho_{\exists_A} = \emptyset$,
- $\rho_{\forall_A} = \emptyset$,
- $\rho_A = \emptyset$,

$\rho_{\exists R}$, $\rho_{\forall R}$, ρ_R, $\rho_{\exists_A}$, $\rho_{\forall_A}$ et ρ_A permettent de définir des ensembles de restrictions sur les rôles et les attributs. Ces ensembles correspondent respectivement aux restrictions existentielles, universelles et de valeurs de rôles ainsi que d'attributs.

Définition de l'ensemble des éléments distincts de $E_{integration}$:

- **TopConcept** Concept spécial, subsumant tous les concepts,
- **BottomConcept** Concept spécial, subsumé par tous les concepts,
- **TopAttribute** Attribut spécial, subsumant tous les attributs,
- **BottomAttribute** Attribut spécial, subsumé par tous les attributs,
- **TopRole** Rôle spécial, subsumant tous les rôles,
- **BottomRole** Rôle spécial, subsumé par tous les rôles,

- **TopDataType** Type de données spécial, subsumant tous les types de données,
- **BottomDataType** Type de donnés spécial, subsumé par tous les type de donnés,

$E_{integration}$ = {TopConcept, BottomConcept, TopAttribute, BottomAttribute, TopRole, BottomRole, TopDataType, BottomDataType},

Nous venons de présenter les modèles unificateur et intégrateur, la section suivante illustre leur utilisation.

3.3/ PROCESSUS D'INTÉGRATION

Les modèles unificateur et intégrateur permettent respectivement, la manipulation unifiée des ressources terminologiques et leurs utilisations en tant que facettes de description pour l'indexation d'items. Nous abordons ci-dessous l'intégration des ressources terminologiques au modèle intégrateur par la présentation du processus, illustré par différents algorithmes, ainsi qu'un exemple de résultat.

3.3.1/ PROCESSUS D'ALIMENTATION DU MODÈLE INTÉGRATEUR

Le modèle intégrateur étant une ontologie formelle, le processus d'intégration peut être qualifié de processus de peuplement de l'ontologie. Ainsi, nous présentons ce processus via la création d'une facette à l'aide d'un algorithme, présenté dans les figures **??** et **??** dont le résultat sera illustré par un exemple. Ce processus est composé de trois phases : (i) peuplement de l'ontologie par la création de la facette ainsi que du niveau conceptuel du vocabulaire associé, (ii) peuplement de l'ontologie par la création du niveau terminologique de la facette et sa mise en relation avec le niveau conceptuel, (iii) peuplement de l'ontologie par l'organisation des relations au sein du niveau conceptuel de la facette.

```
1 Entrée :
2 Soit RT une ressource terminologique, Tel Que RT = {T_x, g ∈ G |v ∈ V_g, v ∈ T_h, v ∈ N}.
3 Soit NA le nom de la facette correspondant à ce nouveau vocabulaire.
4 Soit S la base de connaissances, telle que S = (Ω, Σ, Φ, E)
```

L'algorithme prend en entrée la ressource terminologique à intégrer, le nom de la facette qui va être créée à l'aide de cette ressource, ainsi que la base de connaissances dans laquelle cette facette sera créée.

```
1 sI = sI.add(NA);
2 iC = iC.add(("TypeFacette", NA)); // création de la facette
3 Parcourir les nœuds n de RT
4     termePrefere = n.getTermePref(); // récupération du terme principal
5     sI = sI.add(termePrefere + "_c";
6     iC = iC.add(("ConceptFacette", termePrefere + "_c")); // création d'une instance de ConceptFacette
      avec le suffixe "_c" permettant de distinguer l'instance de ConceptFacette et celle
      de TermeFacette pour le terme principal.
7     iR = iR.add(("appartientA", (NA, termePrefere + "_c"))); // mise en relation des instances du
      niveau conceptuel avec la facette à laquelle elles appartiennent
8 Fin Parcourir
```

Phase (i) : Peuplement de l'ontologie par la création de la facette, ainsi que du niveau conceptuel du vocabulaire associé.
Cette partie de l'algorithme présente la création de la facette ainsi que du niveau conceptuel du vocabulaire associé à celle-ci. C'est sur ce niveau conceptuel que le niveau terminologique viendra se greffer. Pour faciliter la compréhension c'est le nom du terme principal auquel la chaîne de caractères "_c" est concaténée qui est utilisée afin de nommer le concept représentant un ensemble de synonymes.

```
1  Parcourir les nœuds n de RT
2      termePrefere = n.getTermePref(); // récupération du terme principal
3      listeSynonymes = n.getListSyno(); // récupération de la liste des termes synonymes
4      sI = sI.add(termePrefere); // ajout du terme principal à l'ensemble des instances
5      iC = iC.add(("TermeFacette", termePrefere)); // déclaration du terme principal en tant
       qu'instance de TermeFacette
6      iR = iR.add(("hasMainAlias", (termePrefere, termePrefere + "_c"))); // création d'une relation
       hasMainAlias entre l'instance de termePéféré de ConceptFacette et celle de
       TermeFacette
7      sV = sV.add(termePrefere);
8      iT = iT.add((xsd : string, termePrefere));
9      iA = iA.add((valeur, (termePrefere, termePrefere))); // instantiation de la valeur de l'attribut
       valeur de chaque instance de TermeFacette alias principal
10     Parcourir les synonymes termeSynonyme de listeSynonymes
11         sI = sI.add(termeSynonyme); // ajout du terme synonyme a l'ensemble des instances
12         iC = iC.add(("TermeFacette", termeSynonyme)); // déclaration du synonyme en tant
13         qu'instance de TermeFacette
14         iR = iR.add((hasAlias, (termeSynonyme, termePrefere + "_c"))); // création d'une relation
15         hasAlias entre l'instance de termePéféré de ConceptFacette et l'instance
16         de termeSynonyme de TermeFacette
17         sV = sV.add(termeSynonyme);
18         iT = iT.add((xsd : string, termeSynonyme));
19         iA = iA.add((valeur, (termeSynonyme, termeSynonyme))); // instantiation de la valeur de
20         l'attribut valeur de chaque instance de TermeFacette alias non-principal
21     Fin Parcourir
22 Fin Parcourir
```

Phase (ii) : Peuplement de l'ontologie par la création du niveau terminologique de la facette et sa mise en relation avec le niveau conceptuel.
Cette partie de l'algorithme présente la création du niveau terminologique et sa mise en relation avec le niveau conceptuel.

FIGURE 3.3 – Algorithme de création d'une facette dans la base de connaissances à partir d'une ressource terminologique - partie 1

```
Parcourir les noeuds n de RT
    listePeres = n.getPeres(); // récupération de la liste des noeuds pères
    Parcourir les noeuds pères npere de listePeres
        iR = iR.add(("estPlusGeneralisteQue", (npere.getTermePref() + "_c", n.getTermePref() + "_c"))); // création
        des relations hiérarchiques au niveau conceptuel, c'est à dire entre les
        instances de ConceptsFacette
        iR = iR.add(("estPlusSpecifiqueQue", (n.getTermePref() + "_c", npere.getTermePref() + "_c"))); // cette
        relation n'est pas créée mais déduite d'un raisonnement, estPlusSpecifiqueQue
        étant la relation inverse de estPlusGeneralisteQue
    Fin Parcourir
Fin Parcourir
```

Phase (iii) : Peuplement de l'ontologie par l'organisation des relations au sein du niveau conceptuel de la facette.
Cette partie de l'algorithme présente la création des relations hiérarchiques au niveau conceptuel du vocabulaire définissant une facette.

```
Parcourir les noeuds n de RT
    listeProches = n.getProches(); // récupération de la liste des noeuds proches
    Parcourir les noeuds proche nproche de listeProches
        iR = iR.add((estProcheDe, (n.getTermePref() + "_c", nproche.getTermePref() + "_c"))); // instantiation
        des relations estProcheDe  au niveau conceptuel c'est à dire entre les instances
        de ConceptFacette
    Fin Parcourir
Fin Parcourir
```

Cette partie de l'algorithme présente la création des relations d'association au niveau conceptuel du vocabulaire définissant une facette.

FIGURE 3.4 – Algorithme de création d'une facette dans la base de connaissances à partir d'une ressource terminologique - partie 2

3.3.2/ EXEMPLE D'INTÉGRATION, ALIMENTATION DU MODÈLE INTÉGRATEUR

Nous proposons ici l'exemple du peuplement de notre base de connaissances par un extrait du thésaurus Eurovoc illustré par la figure **??**. Eurovoc est un thésaurus multilingue respectant les normes ISO 2788 1986 ; ISO 5964 1985. Il couvre l'ensemble du vocabulaire nécessaire aux activités de l'Union Européenne. L'exemple présent est un sous-ensemble des termes de cette terminologie ainsi qu'une illustration des relations suivantes :

- Relations hiérarchiques entre termes BT (i.e. broader term) et NT (i.e. narrower term).
- Relations de synonymie entre les termes (i.e. termes principaux, termes choisis de préférence) et les non-termes (i.e. termes non principaux) USE et UF (i.e. used for et use).
- Relations associatives RT (i.e. related term).

- **:** Marché monétaire
  - **UF :** marché monétaire international
  - **RT :** marché financier
  - **NT1 :** monnaie

FIGURE 3.5 – Court extrait du thésaurus Eurovoc

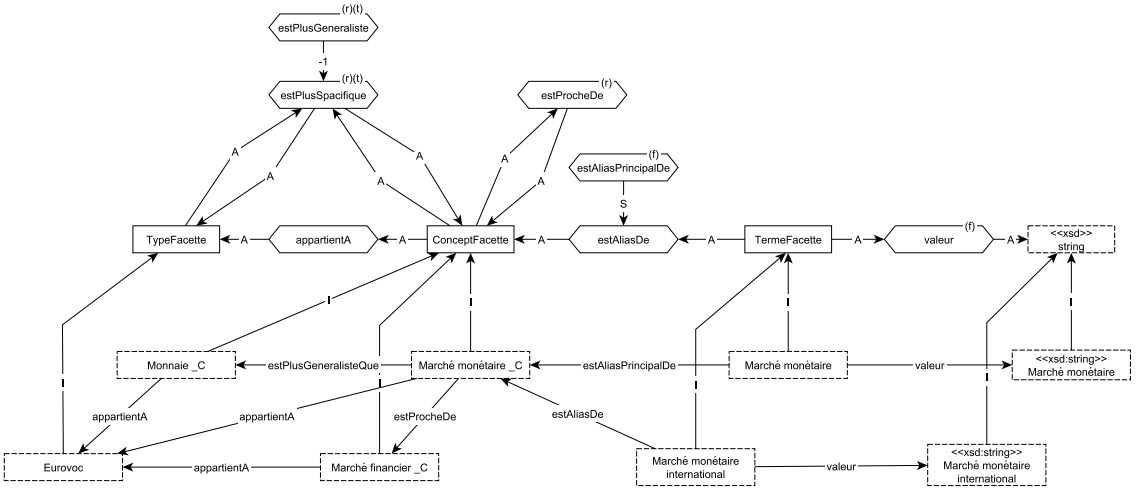


FIGURE 3.6 – Exemple de sous ensemble du thésaurus Eurovoc au format G-OWL

Nous présentons ci-dessous les résultats du peuplement d'une base de base de connaissances structurée comme le propose le modèle intégrateur, par un extrait du thésaurus Eurovoc (cf. figure **??**) à l'aide de la modélisation de Karlsruhe. Une représentation graphique de ce même résultat utilisant la modélisation G-OWL est proposée par la figure **??**. Ces deux modélisations permettent d'illustrer le résultat produit par notre algorithme (cf. figures **??** et **??**) lors du peuplement de la base de connaissances avec un court exemple (cf. figure **??**) de ressource terminologique. Un exemple plus important rendant la compréhension plus complexe sans pour autant apporter plus d'informations.

sI = {Monnaie_C, Marché_monétaire_C, Marché_monétaire, Marché_monétaire_international, Marché_financier_C, Eurovoc},

Ajout des instances à l'ensemble des instances.

sV = {"Marché monétaire", "Marché monétaire international" },

Ajout des valeurs de données à l'ensemble des valeurs de données.

iR = {(estPlusGeneralisteQue,(Marché_monétaire_C, Monnaie_C)), (estPlusSpecifiqueQue, (Monnaie_C, Marché_monétaire_C)), (estProcheDe, (Marché_monétaire_-

C, Marché_financier_C)), (estAliasDe, (Marché_monétaire_international, Marché_monétaire_C)), (estAliasPrincipalDe, (Marché_monétaire, Marché_monétaire_C)), (appartientA, (Marché_monétaire_C, Eurovoc)), (appartientA, (monnaie_C, Eurovoc)), (appartientA, (Marché_financier_C, Eurovoc))},

Instantiation des relations entre les instances.

iC = {(ConceptFacette, Marché_monétaire_C), (ConceptFacette, Marché_financier_C),(ConceptFacette, Monnaie_C), (TermeFacette, Marché_monétaire), (TermeFacette, Marché_monétaire_international), (TypeFacette, Eurovoc)},

Instantiation des concepts (mise en relation des concepts existants avec les instances de l'ensemble d'instances).

iT = {(xsd :string, "Marché monétaire"), (xsd :string, "Marché monétaire international")},

Instantiation des types de données (i.e. mise en relation des valeurs de données avec les types de données).

iA = {(valeur, (Marché_monétaire, "Marché monétaire")), (valeur, (Marché_monétaire_international,"Marché monétaire international"))},

Nous venons de présenter notre modèle intégrateur, permettant l'utilisation de ressources terminologiques existantes pour la description d'items ainsi que le processus permettant l'intégration de ces ressources terminologiques. Afin de clarifier au mieux le processus, celui-ci est illustré par un exemple. Ce processus peut être réalisé de façon manuelle ou automatique. L'automatisation nécessite le développement d'un programme adapté à la ressource visée et permettant sa manipulation sur la base du modèle unificateur.

L'apport principal de ce modèle intégrateur est l'intégration de différentes ressources terminologiques au système. Cette intégration permet la description des items sur la base de facettes constituées de vocabulaires existants, et notamment de vocabulaires contrôlés et structurés.

L'unification des ressources terminologies sous une structuration basée sur un modèle, une syntaxe et une sémantique commune permet d'en faciliter la manipulation, la compréhension et l'utilisation.

Le niveau d'expressivité maximal d'une ressource terminologique étant la prise en compte de toutes les relations proposées par le modèle. Ce modèle permet la prise en compte de l'évolution des vocabulaires qu'il structure. En effet, il peut être utile lors de l'utilisation d'une taxonomie, d'ajouter la prise en charge des synonymes, afin de faciliter la tâche d'indexation, que ce soit par un humain ou une machine.

La prise en compte de l'évolution des ressources terminologiques intégrées est facilitée par l'abstraction conceptuelle réalisée lors de leur intégration. Il est ainsi possible de travailler sur la conceptualisation d'une ressource, de la faire évoluer sans impacter la

terminologie comme l'illustre la figure **??**. Dans cet exemple, une folksonomie est intégrée comme vocabulaire de description d'items. La création de ce type de vocabulaires est un processus non contrôlé, ainsi les termes proposés peuvent ne pas être indépendants bien qu'aucune relation sémantique entre eux ne soit prise en compte. Afin d'améliorer l'indexation des items, un travail de rationalisation du vocabulaire d'indexation peut avoir lieu, de façon manuelle ou automatique. Les relations sémantiques existantes entre les concepts associés aux termes peuvent être renseignées a posteriori. Dans notre exemple deux termes renvoient à un seul et même concept, les concepts sont donc fusionnés de même que deux concepts ont une relation de type hiérarchique, l'un traitant d'une notion plus générale que l'autre.

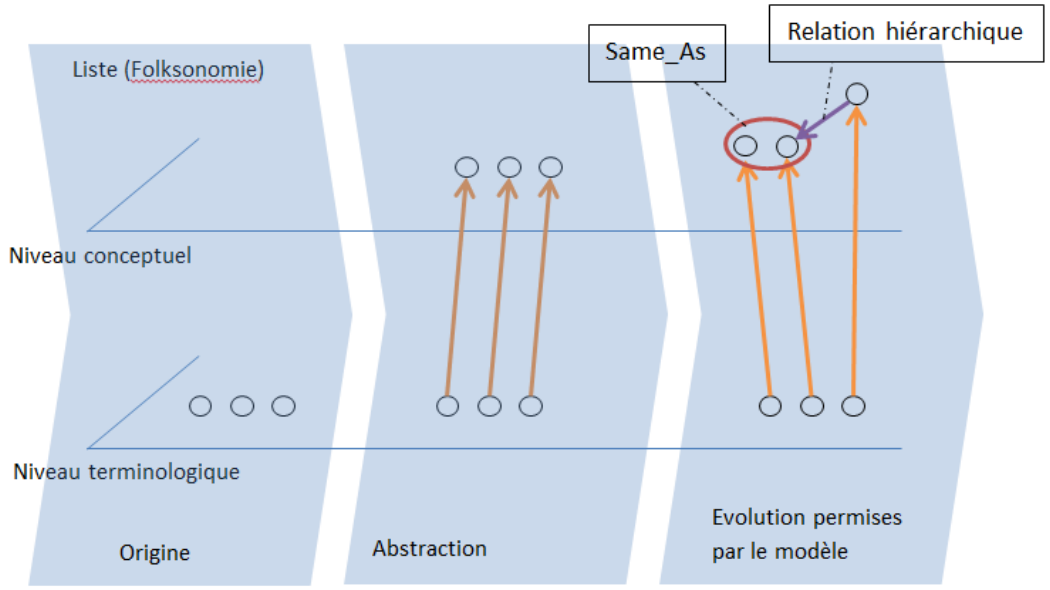


FIGURE 3.7 – Présentation du passage d'une liste de termes dans notre modèle, et des évolutions possibles

Ce modèle repose sur une ontologie, dont la sémantique formelle est basée sur les logiques de description. L'intégration à notre modèle offre aux ressources la prise en compte de certaines contraintes et propriétés. Ainsi les relations hiérarchiques entre ConceptFacettes sont définies comme transitives dans le cas de ressources du type taxonomies ou thésaurus. Les relations permettant de définir le synonyme qui doit être utilisé de préférence afin de référencer un concept, est défini par une relation fonctionnelle. En effet, il ne peut y avoir qu'un terme principal. Les relations inverses des relations existantes sont définies de façon formelle comme des rôles inverses ce qui facilite leur utilisation. Ces différentes propriétés et contraintes logiques peuvent être utilisées par des processus d'inférence.

Ce modèle reposant sur une ontologie formelle, il peut être adapté aux besoins d'une

application. Les relations et concepts qui y sont définis, peuvent être spécialisés comme dans le cas du thésaurus géographique des nations unies, Multilingual Thesaurus of Nations [4] sur lequel un travail de spécialisation des relations a été réalisé. Pour ce faire, des relations ontologiques ont été utilisées. Ainsi, les relations générique/spécifique ont été spécialisées en relation "a pour capitale" et "est un pays de" alors que les relations associatives ont elles été spécialisées en relations "a pour voisin" et "est membre de". Notre modèle permet intrinsèquement la réalisation de ce type d'évolution.

3.4/ INDEXATION DES ITEMS

La base de connaissances du système repose sur une ontologie formelle : le modèle intégrateur. Cette base de connaissances est composée de modules comme l'illustre la figure **??**. Chacun de ces modules distinguent différents types de connaissances. La base de connaissances repose sur le modèle intégrateur. Il est donc utilisé comme une ontologie de haut niveau à partir duquel la base de connaissances peut être enrichie en fonction du domaine d'application.

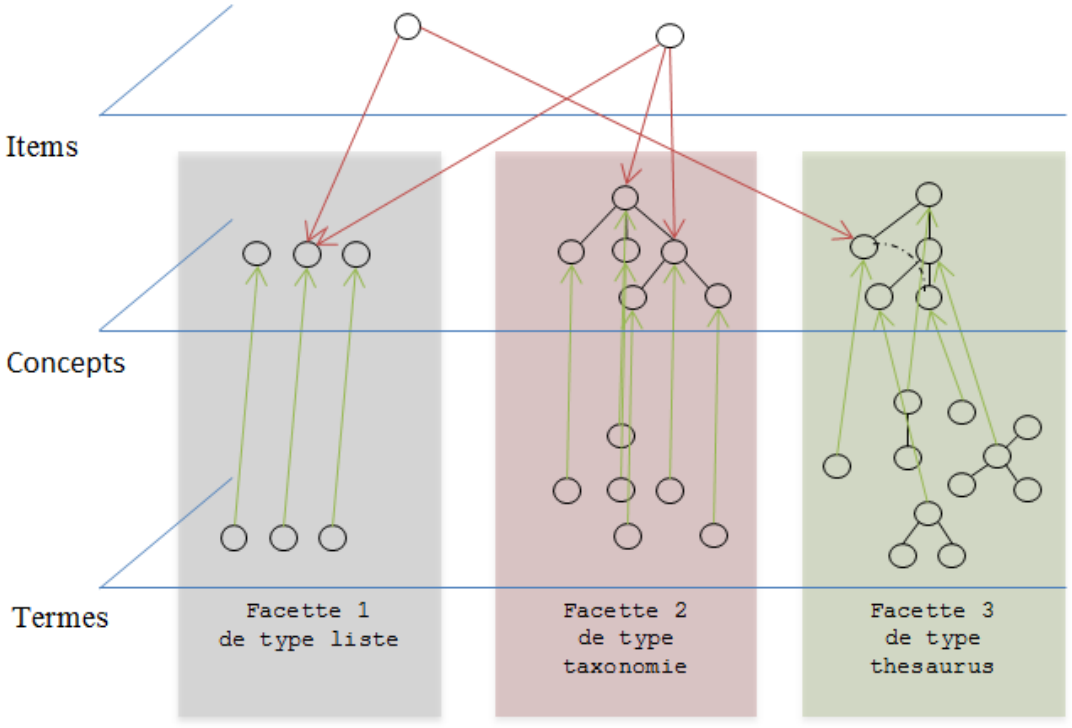


FIGURE 3.8 – Indexation de deux items

Les différents modules liés à l'ontologie de haut niveau, nommé module de référence,

4. http://goo.gl/yMk1dP ; Consulté le 20 septembre 2013

permettent (i) la gestion des concepts généraux sous lesquels viennent se positionner les concepts plus spécifiques de chaque sous-module, (ii) la gestion de concepts transcendants différents types de connaissances. La base de connaissances comporte ainsi les modules suivants :

1. Un module contenant les connaissances générales, c'est-à-dire les connaissances associées aux facettes de description qui ne dépendent pas d'une vision métier sur un domaine précis. Cette partie de la connaissance n'a généralement pas à être adaptée, elle contient les connaissances pour la gestion de l'espace de façon administrative ou du temps afin d'indexer les items en fonction de la géolocalisation et de la temporalité.

2. Un module contenant les connaissances spécifiques au domaine, c'est-à-dire les connaissances associées aux facettes de description qui dépendent du domaine d'application ainsi que de la vision métier sur le domaine.

   Ces deux modules contiennent le niveau conceptuel des ressources terminologiques provenant du processus d'intégration.

3. Un module contenant les connaissances nécessaires au système, c'est-à-dire l'indexation des items en tant que tels, et éventuellement des spécialisations des concepts ou relations nécessaires à la description des items en fonction des besoins de l'application.

4. Un module contenant les connaissances lexicales. Ces connaissances correspondent au niveau terminologique des facettes définies par les ressources intégrées.

La figure **??** illustre un exemple d'indexation sur des ressources terminologiques intégrées à notre modèle. Elle illustre le fait que l'indexation des items se fait au niveau conceptuel (cf. flèches rouges) et non au niveau terminologique, comme cela est généralement le cas. Cette approche, est avantageuse car les termes et langues utilisés pour désigner une notion peuvent évoluer indépendamment de la structuration conceptuelle.

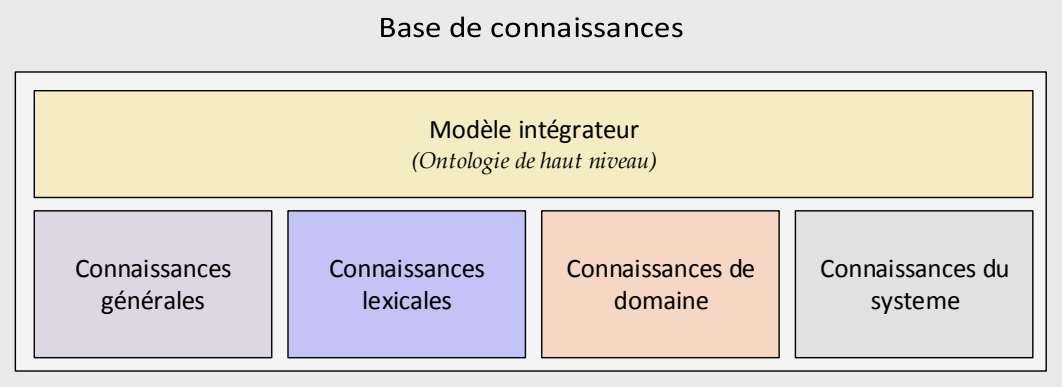


FIGURE 3.9 – Présentation des modules de notre base de connaissances

La figure **??** illustre également un exemple soulignant les différents modules présentés dans la figure **??**. Le module de connaissances générales n'y est pas illustré afin de simplifier l'exemple. Ce module est similaire au module de connaissances de domaine mis à part la présence par défaut des connaissances générales. Cet exemple illustre aussi la possibilité de spécialiser les concepts présents dans l'ontologie de haut niveau afin d'adapter le modèle à un domaine d'application précis. Dans notre exemple, le concept **ConceptFacette** est spécialisé en **Eurovoc_CF** et le concept **item** est spécialisé en **Loi**, car le contexte d'application est l'indexation des lois de l'Union Européenne sur la base du vocabulaire Eurovoc.

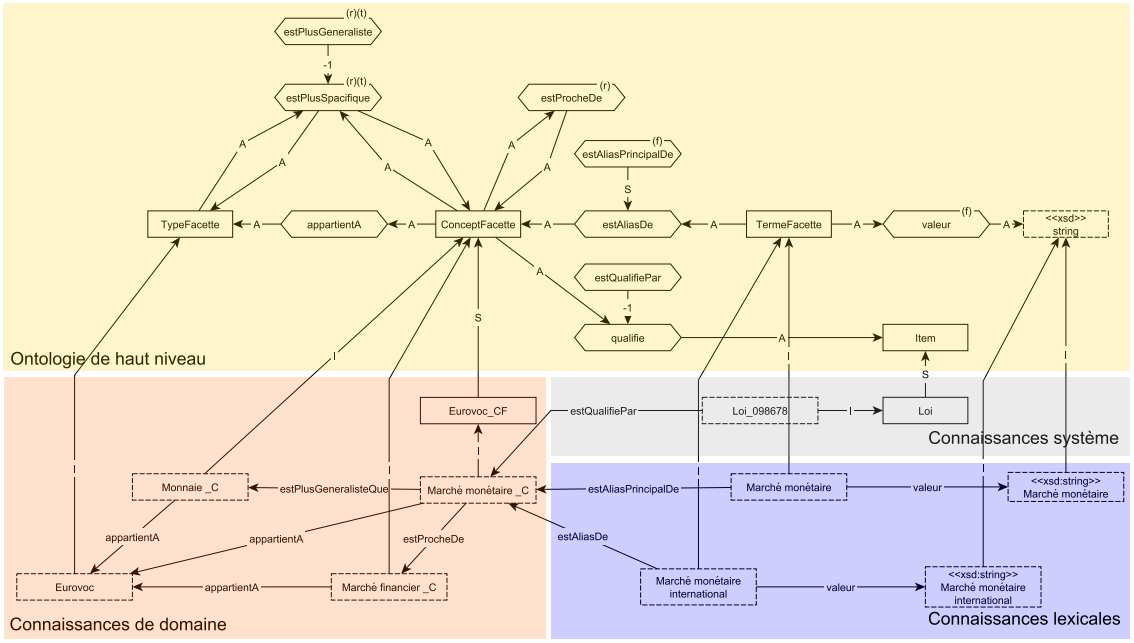


FIGURE 3.10 – Exemple d'indexation avec une représentation graphique G-OWL

3.5/ CONCLUSION

Ce chapitre présente les différents modèles pour la représentation des connaissances nécessaires à la recommandation d'items. Ce point est central au système, car il permet de gérer les descriptions d'items de façon à ce que celles-ci soient manipulables par la machine et par l'humain. Tous deux intervenant dans un système automatique de recommandation.

Face à l'hétérogénéité des ressources et de leurs types, nous avons présenté un modèle mathématique unificateur permettant de manipuler les ressources lors de la phase d'intégration de ressources dans notre modèle intégrateur. Celui-ci permettant la description des items en fonction de facettes. L'intégration des ressources terminologiques à la base de connaissances en respectant le modèle défini possède divers avantages :
(1) Il permet, et c'est là l'objectif principal, la description d'items à l'aide de différentes facettes, chacune associée à une terminologie. Ces descriptions sont aisément accessibles pour une compréhension humaine, car elles sont organisées sous la forme de facettes associées à des ressources terminologiques (i.e. listes plates, taxonomies, thésaurus) pensées en premier lieu pour une utilisation humaine. Les facettes sont une façon simple de modéliser un domaine même complexe. Leur utilisation est aisée pour des humains qui y sont habitués, notamment sur les plateformes de e-commerce.

Les descriptions d'items sont de plus directement manipulables par la machine, car

basées sur un modèle formel unifiant tous les types de ressources. L'utilisation de facettes permet de gérer la complexité et la richesse de la modélisation d'un domaine tout en conservant une modélisation compréhensible et manipulable pour l'humain.
(2) Notre modèle étant une ontologie formelle basée sur les logiques de description, des contraintes et propriétés logiques telles que les rôles inverses, les relations transitives et les relations formelles ont pu être ajoutées aux ressources. Ces contraintes et propriétés sont exploitables par des processus d'inférence.
(3) Lors de l'intégration de ressources conceptuelles au modèle, nous réalisons une abstraction conceptuelle. Le niveau conceptuel et le niveau terminologique sont distingués contrairement aux ressources terminologiques qui ne font pas cette distinction. L'indexation des items a lieu sur le plan conceptuel, indépendamment de l'évolution de la terminologie permettant l'adaptation à une nouvelle langue par exemple. De même qu'une évolution de la conceptualisation peut être réalisée sans impacter la terminologie.
(4) Notre modèle permet d'augmenter le niveau d'expressivité d'une ressource. Par exemple, il permet d'organiser hiérarchiquement les éléments d'une liste afin d'en faire une taxonomie.

La base de connaissances résultant du modèle proposé contient ainsi les connaissances suivantes :

- La modélisation du domaine d'application sous la forme de facettes et le vocabulaire associé.
- La description de chacun des items sur la base de ces facettes.

Le processus permettant l'indexation des items (i.e. la description de chacun des items sur la base des facettes contenues dans la base de connaissances) est l'objet du chapitre suivant.

4

LES PROCESSUS D'INDEXATION

Ce chapitre présente les processus d'indexation des articles et profils. Les descriptions créées reposent sur un vocabulaire d'indexation contrôlé et structuré. Après une présentation des approches manuelles et supervisées, nous proposons une méthode nouvelle d'indexation automatisée. Celle-ci considère l'indexation comme un processus de classification multi-label. Notre approche est novatrice, car elle consiste à intégrer au sein de la base de connaissances gérant le vocabulaire d'indexation du système, un modèle prédictif. Ce modèle prend la forme de contraintes logiques utilisables par des raisonneurs. La base de connaissances est une ontologie reposant sur le modèle intégrateur défini dans le chapitre **??**, ce qui permet l'utilisation de contraintes logiques. L'indexation est ainsi inférée par le raisonneur à partir des connaissances de la base. Le processus d'indexation se fait donc au plus près de la modélisation du domaine faite par les experts, afin de faciliter la supervision. Cette approche est une première étape vers des systèmes à même de gérer efficacement l'évolution du vocabulaire d'indexation et ses répercussions sur le processus d'indexation.

Le chapitre précédent présente la modélisation des connaissances du domaine à l'aide de facettes dans l'objectif de créer une base de connaissances pour l'indexation d'items. Ces facettes sont constituées de ressources terminologiques préexistantes ou créées par des experts de façon ad'hoc. Nous présentons dans ce chapitre la façon dont ces connaissances sont utilisées afin de décrire le contenu des documents ainsi que les besoins des utilisateurs dans le cadre d'un système de recommandation basé sur la sémantique. Cette tâche est nommée, *indexation*. L'indexation consiste à associer à chaque item un ensemble de termes permettant de le décrire. Un item représente ici, soit un document textuel, doit un profil utilisateur.

La mise en relation des items avec une sélection de termes permettant leur description est une tâche centrale dans un système de recommandation basé sur le contenu. Les termes sélectionnés proviennent du vocabulaire des facettes. L'indexation doit impérativement être réalisée afin de permettre par la suite la comparaison des profils et des documents à recommander. Ainsi, la qualité de l'indexation va directement influencer la qualité de la recommandation.

La qualité de l'indexation dépend de deux facteurs :
(i) La *qualité de l'index*, c'est à dire, la couverture, la richesse et la structure du vocabulaire d'indexation disponible.
(ii) La *qualité des associations entre les termes de l'index et les items*, c'est à dire limiter au maximum l'indexation d'items avec des termes qui ne les décrivent pas, ou qui les décrivent mal et réciproquement ne pas omettre des termes qui permettraient de compléter leur description.

La qualité de l'index est assurée lors de sa conception par son adéquation avec le point de vue des experts du domaine et sa structure par facettes qui le rend facile à appréhender, à adapter et à utiliser. Le modèle intégrateur, utilisé pour la conception de la base de connaissances d'un système de recommandation basé sur la sémantique, permet d'assurer cette qualité (cf. chapitre **??**).

Afin de conserver une qualité optimale des associations entre termes et items, l'indexation peut être réalisée manuellement par des experts. Dans le cas d'une automatisation, le résultat produit doit être contrôlé par des experts. L'indexation manuelle est certes qualitative, mais elle est aussi consommatrice de temps, c'est pourquoi nous proposons une approche permettant l'automatisation de l'indexation des items. Cette approche automatisée préserve l'adéquation avec le point de vue des experts ce qui facilite par la suite la supervision.

Nous commençons par la présentation de l'indexation telle qu'elle peut être réalisée grâce à notre modèle intégrateur. Les méthodes manuelles et semi-automatiques supervisées,utilisées pour l'indexation des articles et profils sont ensuite détaillées. Enfin, nous présentons une méthode unifiée pour l'automatisation de cette tâche. Celle-ci se base,

par apprentissage supervisé, sur le travail d'indexation précédemment effectué manuellement ou semi automatiquement par les experts.

4.1/ INDEXATION À L'AIDE DU MODÈLE INTÉGRATEUR

Le processus d'indexation consiste à associer à chaque item un ensemble de termes permettant de le décrire. Des instances représentant les items, ainsi que les termes permettant leur description, sont gérés par notre modèle intégrateur. Ainsi, dans notre base de connaissances l'indexation prend la forme d'une instanciation de la relation *estQualifiePar* entre des instances du concept *Item*, ou une de ses spécialisations et des instances du concept *ConceptFacette*. Ainsi un item peut être vu comme un vecteur de d'instances de *ConceptFacette* :

$$\vec{i} = \langle t_1, t_2, ..., t_m \rangle \quad t_m \in T$$

Soit $\vec{i}$ un item et t_x une instance du concept ConceptFacette utilisée pour la description de l'item et appartenant à l'ensemble de tous les termes T utilisables pour la description d'un item.

Dans le chapitre précédent, l'exemple présenté par la figure **??** illustre l'indexation d'une loi européenne, Loi_098678 sur la facette de description définie par le vocabulaire Eurovoc. La loi, Loi_098678 y est qualifiée par le ConceptFacette Marché_monétaire_c appartenant à la facette Eurovoc. Marché_monétaire_c fait référence au terme principal "Marché monétaire" ainsi qu'au terme synonyme "Marché monétaire international". Comme l'illustre l'exemple, l'indexation est réalisée au niveau conceptuel du vocabulaire des facettes. Cet exemple signifie que pour la facette de description Eurovoc la loi, Loi_098678 est décrite comme traitant du marché monétaire. Dans la section suivante, les processus d'indexation manuelles et semi-automatiques sont présentés.

4.2/ INDEXATION MANUELLE ET SEMI-AUTOMATIQUE

Cette section présente les méthodes manuelles et semi-automatiques utilisées afin de réaliser l'indexation des items. L'indexation de documents textuels dépend des informations dont ils traitent. Deux types d'informations peuvent être distinguées : (i) les informations exprimées explicitement, par exemple des lieux ou des personnes et (ii) les informations compréhensibles implicitement, par exemple les secteurs économiques concernés.

Les informations implicites sont plus complexes à extraire de façon automatique que les informations explicites. Elles nécessitent une analyse et une compréhension plus pro-

fonde du texte. Il est souvent nécessaire de recouper plusieurs informations indices avant de les extraire. Ainsi, nous présentons ci-dessous ; premièrement le fonctionnement de l'indexation manuelle utilisée pour l'indexation des items, notamment en ce qui concerne les facettes qui ne peuvent être associées aux documents textuels que de façon implicite ; puis dans un second temps, le fonctionnement semi-automatique utilisé pour l'indexation des documents en ce qui concerne les facettes exprimées de façon explicites dans le texte.

4.2.1/ INDEXATION MANUELLE

Deux cas d'indexation manuelle peuvent être distingués. (i) La création des *profils* peut être réalisée par des experts durant des échanges avec les utilisateurs. Le dialogue facilitant la compréhension par l'expert du besoin de l'utilisateur. L'indexation est ensuite réalisée manuellement par l'expert, qui renseigne les informations dans le système à l'aide d'une application dédiée. L'indexation des profils utilisateurs est donc réalisée par les experts de façon manuelle, et ce pour toutes les facettes descriptives du profil. (ii) Les items de type *document textuel* peuvent à la suite de leur lecture, ou de leur rédaction par un expert, être indexés de façon manuelle. Cela peut notamment être pertinent afin de conserver une qualité d'indexation forte dans le cas d'informations exprimées de façon implicite dans le document. Ainsi, nous présentons ci-dessous une approche manuelle d'indexation de documents textuels, pour les facettes implicites. Pour les autres facettes (i.e. exprimées de façon explicite), l'indexation sur les facettes explicites de façon semi-automatique est présentée dans la section suivante.

4.2.1.1/ INDEXATION MANUELLE DES PROFILS

Dans le contexte d'un système de recommandation basé sur la sémantique, l'indexation des profils peut être réalisée à la main. Cela permet le démarrage du système. En effet, un des principaux problèmes des systèmes de recommandation basés sur le contenu est le *démarrage à froid utilisateur* (cf. section **??**). Cela signifie que le système, n'ayant pas d'information sur le besoin d'un utilisateur (e.g. pas d'historique des activités, pas de profil qualifié), n'est pas en mesure de lui proposer les items correspondants à son besoin.

La création d'un profil manuel au sein d'un système de recommandation sémantique nécessite une interface d'indexation. Dans une base de connaissances reposant sur le modèle intégrateur, chaque item est décrit en fonction de dimensions descriptives pertinentes pour le cas d'application. Ces dimensions descriptives sont nommées facettes. Chacune d'elle contient un vocabulaire qui lui est propre. Le vocabulaire des facettes peut être, ou non, structuré sous la forme d'une hiérarchie. Les facettes nécessitant un

vocabulaire riche, c'est-à-dire, contenant une grande quantité de termes, ont d'avantage tendance à être organisées sous la forme de nomenclatures ou de thésaurus (cf. chapitre **??**) ce qui facilite leur utilisation par des humains. Les vocabulaires contenant moins de termes se limitent généralement à des listes plates.

Les vocabulaires structurés de façon hiérarchique, peuvent être vastes et donc difficiles à présenter. Ainsi lors de la génération de l'interface d'indexation, une version simplifiée du vocabulaire d'indexation peut être utilisée. Elle consiste, pour les vocabulaires contrôlés organisés de façon hiérarchique à n'afficher dans l'interface d'indexation que les niveaux les plus hauts de la hiérarchie des termes de ces vocabulaires. Avec le modèle intégrateur cela consiste pour une facette donnée, à ne présenter que les instances de ConceptFacette les plus générales. De même, si pour un concept, différents termes synonymes sont disponibles, seul le terme principal sera utilisé pour l'affichage. La figure **??** illustre l'extraction d'une vue sur le vocabulaire global d'indexation contenue dans la base de connaissances pour la création d'une interface d'indexation de profils. Cette extraction est automatique, le degré de simplification (i.e. la profondeur maximale de l'arbre affiché) est défini préalablement. La base de connaissances étant une ontologie OWL, elle peut aisément être interrogée à l'aide du langage SPARQL [1]. Ce langage permet de limiter la profondeur de la recherche lors de l'exploration de relations transitives. Il est aisé à partir de ce langage d'interrogation, de définir une requête permettant au système de ne récupérer qu'une partie limitée du vocabulaire des facettes contenues dans la base de connaissances. Et ainsi, de générer les vues.

1. http ://www.w3.org/TR/rdf-sparql-query/

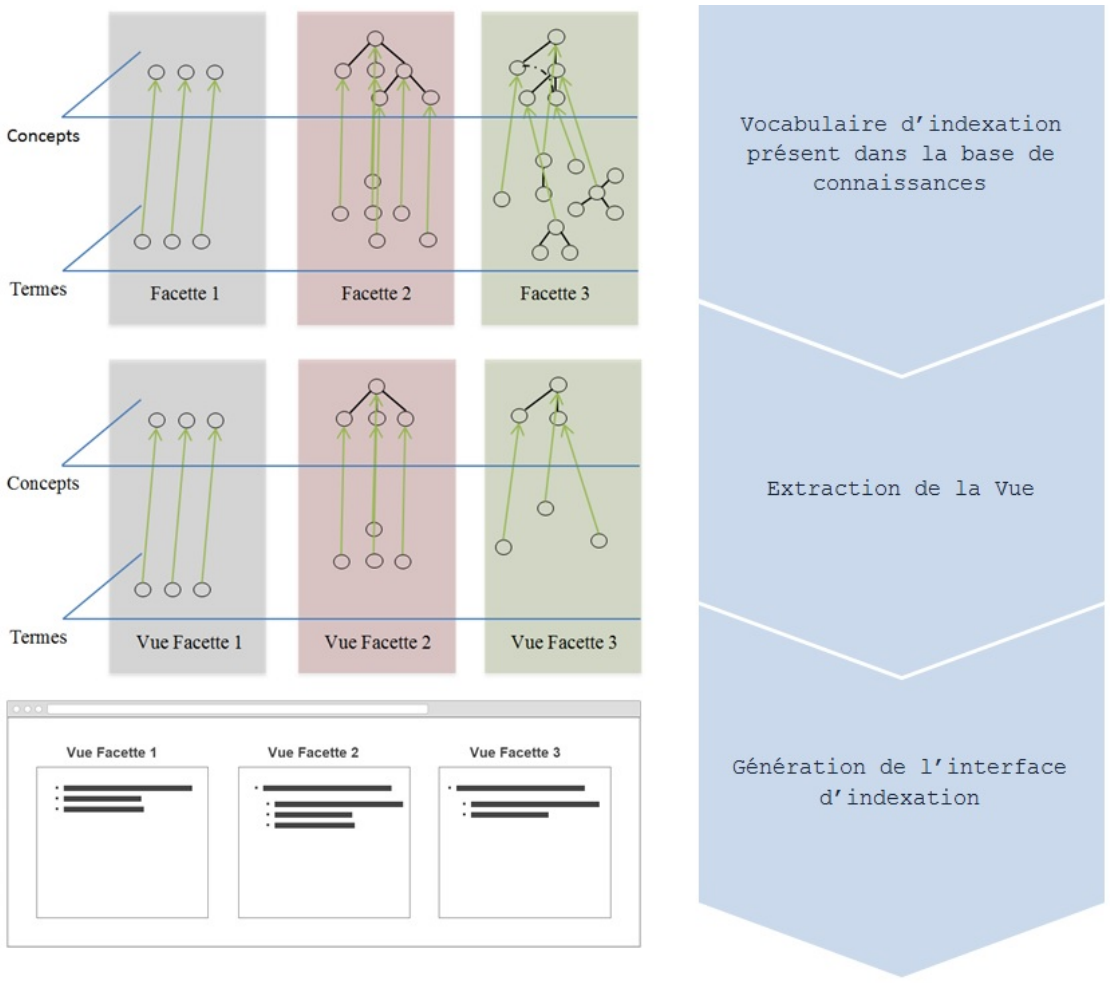


FIGURE 4.1 – Extraction d'une vue sur un vocabulaire d'indexation pour la création d'une interface de profilage

Le processus d'indexation manuel des utilisateurs est le suivant :

1. Discussions avec l'utilisateur afin de comprendre son besoin.
2. Utilisation de l'interface d'indexation afin de renseigner le profil de l'utilisateur.
3. Validation et peuplement de l'ontologie.

Le profil peut être créé manuellement par l'utilisateur, nous faisons ici le choix de le faire faire par un expert. Avec cette approche, nous ne laissons pas à l'utilisateur le droit de modifier lui-même son profil, car des études ont montré que cela pouvait avoir des conséquences négatives sur la qualité de celui-ci et donc de la recommandation proposée à l'utilisateur [**?**].

(1) Les experts réalisant le profilage suivent un processus guidant les échanges avec les utilisateurs. Ce processus est fonction du cas d'application (i.e. du domaine pour lequel le système est adapté). Une interface est proposée aux experts afin qu'ils puissent prendre des notes durant l'entretien.

(2) Une interface d'indexation propose aux experts de sélectionner les termes permettant de décrire le profil d'un utilisateur. Cette interface est générée automatiquement à partir du vocabulaire de la base de connaissances. Dans cette interface, les experts ont accès aux notes prises durant l'entretien avec l'utilisateur.

Le vocabulaire disponible pour l'indexation des profils ne contient pas tout le vocabulaire disponible dans la base de connaissances, mais une vue de taille réduite ne contenant pas les termes trop précis. L'objectif est de réaliser la tâche d'indexation des profils rapidement afin d'amorcer le système. Une compréhension fine du besoin des utilisateurs est consommatrice de temps, aussi bien pour l'utilisateur que pour l'expert. De plus l'utilisateur ne sait parfois pas lui-même exprimer précisément son besoin, c'est pourquoi débuter avec un profil relativement général, permet de lui fournir des informations relativement variées sans toutefois trop s'éloigner du besoin exprimé. Il est possible, afin de préciser le profil de l'utilisateur, d'étudier son comportement sur le système. Car même si l'utilisateur ne sait pas exprimer explicitement son besoin, il sait reconnaître une information qui l'intéresse quand elle lui est proposée [**?**]. Afin de simplifier le travail des experts, un système de recherche assisté est proposé en plus du vocabulaire affiché. L'expert peut ainsi rechercher le terme le plus adéquat en fonction du vocabulaire qu'il souhaiterait utiliser. Au cours de la frappe, les termes les plus proches sont dynamiquement proposés sur la base des informations de la base de connaissances. Les termes sont recherchés dans l'ensemble des termes d'indexation disponibles, au-delà de la vue présentée dans l'interface. La recherche porte donc sur tous les termes qu'ils soient principaux ou synonymes, quels que soit leurs niveaux de précision. Afin de faciliter la compréhension des termes, et donc de simplifier le choix du terme adéquat, l'environnement de chacun des

termes est affiché. Pour un terme appartenant à une facette organisée sous la forme d'un thésaurus, diverses informations sur l'environnement peuvent exister. Ainsi, sa position dans la hiérarchie des termes, les termes plus généraux ou plus spécifiques auxquels il renvoie, ou encore les synonymes et notes explicatives du terme, sont affichées afin de guider l'expert.

(3) Une fois les termes sélectionnés dans l'interface et validés, le système reporte automatiquement les informations dans la base de connaissances. L'ontologie est donc peuplée avec les nouvelles informations. Les termes sélectionnés par l'expert sont des instances de *TermeFacette*. Des relations *estQualifiePar* sont instanciées entre l'instance du concept *Item* correspondant à l'utilisateur dont le profil est en cours de création et les instances du concept *ConceptFacette* auxquelles se réfèrent les termes sélectionnés par l'expert.

4.2.1.2/ INDEXATION MANUELLE DES DOCUMENTS

L'indexation manuelle des documents pour les facettes exprimées de façon implicite dans le document peut être réalisée par les experts selon le processus ci-dessous :

1. Création ou prise de connaissance du document par l'expert.

2. Utilisation de l'interface d'indexation afin d'indexer le document.

3. Validation et peuplement de l'ontologie.

(1) Selon le cas d'application, deux cas sont possibles : (i) l'expert qui crée le document à recommander l'indexe après sa création, (ii) le document est préexistant et un expert doit en prendre connaissance avant de pouvoir l'indexer.

(2) Contrairement à l'indexation des profils, l'indexation des documents doit être la plus précise possible. Afin de faciliter le travail des experts, l'indexation des documents pour les facettes implicites s'effectue principalement à l'aide du système de recherche assisté présenté dans la section précédente.

(3) Comme pour l'indexation des profils, une fois les termes sélectionnés dans l'interface, validés, le système reporte automatiquement les informations dans la base de connaissances. L'objectif est une indexation la plus précise possible. Si lors de la recherche, l'expert sélectionne un terme très précis, l'indexation dans la base de connaissances se fera avec le terme sélectionné. Ce n'est pas le cas avec les profils, car la vue est constitué de termes plus généraux.

4.2.2/ INDEXATION SEMI-AUTOMATIQUE SUPERVISÉE

L'indexation des documents peut être réalisée semi-automatiquement. Comme nous l'avons vu dans la section précédente, une partie de l'indexation peut être réalisée de façon manuelle, car elle correspond aux facettes dites explicites. L'indexation sur ces facettes peut plus facilement être automatisée. L'autre partie, concerne les facettes dites implicites est plus complexe à automatiser. L'automatisation peut être supervisée afin de garantir la qualité de celle-ci. Cette tâche peut être réalisée par les experts selon le processus ci-dessous :

1. Création ou prise de connaissance du document par l'expert.
2. Analyse automatique du document et extraction des termes descripteurs explicites.
3. Utilisation de l'interface d'indexation afin de vérifier et éventuellement corriger ou compléter l'indexation automatique proposée (i.e. supervision).
4. Validation et peuplement de l'ontologie.

(1) Selon le contexte d'application, deux cas sont possibles. (i) L'expert qui crée le document à recommander, l'indexe après sa création ou (ii) le document est préexistant et un expert doit en prendre connaissance avant de pouvoir l'indexer.

(2) Une chaîne de traitement est appliquée sur les documents. Dans le cas de documents textuels la chaîne de traitement suivante permet d'extraire des informations exprimées explicitement :

- Découpage du texte en phrases (i.e. *sentence splitters*).
- Découpage des phrases en mots (i.e. *tokenizer*).
- Analyse morpho-syntaxique des phrases (i.e. *part of speech tagger*).
- Recherche de termes présents dans un dictionnaire (i.e. *gazetteer*). Le dictionnaire peut être généré à partir du vocabulaire des facettes explicites de la base de connaissances du système de recommandation.
- Reconnaissance de motifs (i.e. détection d'informations à partir de *patrons lexico-syntaxiques*, motifs prédéfinis du contexte d'apparition de certaines informations).

En fonction du cas d'application, ces outils doivent être paramétrés à la main. Des outils comme, GATE [**?**] permettent de mettre en place rapidement des chaînes de traitement afin d'extraire des informations explicites.

(3) Les résultats de l'analyse sont présentés dans l'interface d'indexation et vérifiés par les experts. Si cela est nécessaire, ils peuvent être corrigés et / ou complétés. La correction consiste à supprimer de la description du document les termes proposés à tort. Le complètement consiste à ajouter à l'aide du système de recherche assisté (présenté dans les sections précédentes) les termes manquant à la description.

(4) Comme dans les sections précédentes, une fois les termes sélectionnés dans l'interface et validés, le système reporte automatiquement les informations dans la base de connaissances. Comme il s'agit ici d'indexation de documents, l'indexation se fait là aussi au plus précis. L'indexation dans la base de connaissances se fera donc avec le terme sélectionné. Ce n'est pas le cas pour les profils, car les termes doivent appartenir a la vue, et peuvent par conséquent être plus généraux.

4.3/ AUTOMATISATION DU PROCESSUS D'INDEXATION

Les approches d'indexation présentées précédemment sont qualitatives, mais se révèlent longues et fastidieuses pour les experts. Dans notre cas d'application, pour des documents de type articles économiques faisant entre 10 et 15 lignes, le temps d'indexation d'un document textuel est entre 2 et 5 minutes. Aussi l'automatisation de cette tâche tout en conservant un fort niveau de qualité est importante afin de permettre un gain de temps. Cette section s'intéresse à l'automatisation complète du processus d'indexation. Nous proposons ici la mise en place d'une approche d'indexation aussi bien sur les facettes exprimées explicitement qu'implicitement dans un document textuel. Toutefois, l'approche d'automatisation proposée conserve une adéquation forte avec la modélisation du domaine, c'est-à-dire les facettes et la structuration du vocabulaire qui leur est associées, contrairement aux approches existantes (cf. chapitre **??**).

Un document peut être associé à plusieurs termes descripteurs lors de l'indexation. Ainsi la mise en relation automatique des documents avec les termes descripteurs, peut être assimilée à une tâche de classification multi-label. Chaque terme est assimilable à une étiquette (i.e. un label). Lorsque le vocabulaire est structuré hiérarchiquement, c'est alors un processus de classification multi-label hiérarchique (cf. section **??**).

La base de connaissances repose sur le modèle intégrateur, ce modèle permet l'utilisation de vocabulaires structurés de façon hiérarchique. L'indexation de documents sur la base d'un vocabulaire contrôlé organisé hiérarchiquement est assimilable à une tâche de classification multi-label hiérarchique.

Des algorithmes d'apprentissage supervisé sont utilisés afin d'apprendre le modèle prédictif avec lequel l'ontologie est enrichie. Ce modèle prédictif correspond à un ensemble de contraintes logiques permettant la classification multi-label des documents par un pro-

cessus d'inférence. Le modèle prédictif est appris à l'aide des indexations précédemment réalisées de façon manuelle ou semi-automatique.

L'objectif de notre approche est d'inclure les connaissances nécessaires au processus d'indexation dans la base de connaissances. Nous débutons une démarche visant à tester la faisabilité de l'approche. A terme, notre objectif est de faciliter la prise en charge de l'évolution du vocabulaire d'indexation. L'évolution, nécessite l'adaptation du processus d'indexation. En intégrant les vocabulaires et le modèle prédictif conjointement dans une ontologie servant de base de connaissances, nous souhaitons prendre en compte l'évolution du vocabulaire ainsi que du modèle prédictif comme des processus du cycle de vie de l'ontologie. La modification du vocabulaire entraînant automatiquement un certain nombre de conséquences sur le modèle, sans nécessiter un réapprentissage complet de celui-ci. Des travaux récents [**?**] ouvrent des perspectives en ce sens.

Etant donné la fiabilité des classifieurs existants [**?**], principalement dans les cas de classification multi-label hiérarchique [**?**] [**?**] [**?**], et l'importance de l'indexation pour la qualité de la recommandation finale, l'indexation d'un document doit donc être supervisée et éventuellement corrigée ou enrichie par un expert. Notre approche ne prend en compte les résultats des algorithmes qu'à des fins de proposition, que les experts doivent accepter ou modifier.

La réduction du fossé entre la classification automatique et la connaissance des experts doit permettre de faciliter le travail de supervision et de correction afin de conserver une indexation des documents de qualité.

Nous proposons donc une approche en 7 étapes :

1. Indexation manuelle (cf. section **??**).
2. Extraction des indices contenus dans les documents.
3. Apprentissage du modèle prédictif.
4. Enrichissement de l'ontologie par l'ajout des règles logiques.
5. Classification multi-label automatique.
6. Supervision, correction, complément de la classification proposée.
7. Validation, peuplement et mise à jour de l'ontologie.

Etape 1 : L'indexation manuelle des items est réalisée avec les méthodes présentées dans la section **??**. Le travail manuel déjà réalisé par les experts est mis à profit durant l'étape 3, lors de l'apprentissage du modèle prédictif.

Etape 2 : Le processus d'extraction d'indices (i.e. features) peut aller de la simple extraction de termes à l'aide de pondération TF-IDF [**?**] aux processus plus complexes basés sur des outils de traitement du langage naturel ou d'extraction d'informations [**?**] [**?**] [**?**]. De même, concernant les outils de traitement du langage naturel, des traitements plus ou moins complexes peuvent être appliqués lors de l'analyse de documents (e.g. tokenizer, gazetteers, part of speech tagger, lexico-syntaxic / lexico-semantic patterns, shallow / deep parsing) [**?**]. L'objectif est ici d'extraire les informations permettant par la suite la prise de décision d'indexation.

Etape 3 : Divers algorithmes peuvent être utilisés [**?**] [**?**] [**?**] afin de générer le modèle prédictif. L'algorithme utilisé doit produire un modèle traduisible sous la forme de contraintes logiques car nous exploitons à la fois l'ontologie et le moteur d'inférence associé. C'est le cas par exemple des arbres de décision [**?**]. Le processus d'apprentissage automatique a deux objectifs principaux : prédire et décrire. La prédiction concerne l'utilisation de ressources précédemment indexées afin de prédire la bonne classification dans un cas inconnu. La description concerne la recherche de modèles interprétables par des humains permettant de décrire l'indexation qui a été effectuée.

Etape 4 : Le modèle prédictif doit permettre l'interaction avec l'expert. Ainsi, il est nécessaire de choisir un modèle dont l'expert puisse suivre le raisonnement et qui soit aussi capable d'intégrer les informations de celui-ci [**?**]. Cela facilite le travail de supervision, correction, complément et améliore la confiance de l'expert envers le système [**?**].

Afin de gérer cette contrainte nous proposons l'utilisation d'ontologies. En effet, (i) les ontologies sont capables de gérer les ressources terminologiques, les règles de classification, les descripteurs de documents et leurs relations et (ii) "Les ontologies sont la meilleure réponse à la demande de systèmes intelligents capable de fonctionner au plus proche du niveau conceptuel humaine [2] " [**?**]. Les ontologies sont un outil de représentation des connaissances permettant de combler le fossé entre les machines et les humains.

Ainsi la contrainte principale est la possibilité de traduire le modèle fourni par l'algorithme choisi en règles logiques représentables dans l'ontologie. Nous souhaitons ainsi qu'il soit transformable en une série d'expressions prenant la forme suivante :

$$Automobile \leftarrow \{Feature, voiture\} \sqcap \{Feature, contructeurs\}$$

Cette expression se traduit de la façon suivante : Si le document contient les indices "voi-

2. Ontologies are the best answer to the demand for intelligent systems that operate closer to the human conceptual level

ture" et "constructeur", alors, l'instance représentant ce document dans l'ontologie, appartient au concept "Automobile". Le document est donc automatiquement indexé comme traitant d'"Automobile".

Etape 5 : Le processus de classification multi-label automatisé proposé fonctionne à l'aide d'un processus d'inférence. L'inférence est réalisée en logique de description au niveau terminologique ainsi qu'au niveau assertionnel en tenant compte des individus de la base de connaissances. Les raisonneurs sont des programmes informatiques qui, à partir des connaissances de l'ontologie, permettent de déduire de nouvelles connaissances. Il est donc nécessaire d'enrichir l'ontologie avec des contraintes logiques permettant l'évaluation des labels à associer à chaque item. Car, ces contraintes sont utilisées par les raisonneurs pour produire le résultat souhaité. La création des contraintes est l'objet de l'étape précédente (cf. étape 4). L'ontologie est donc en partie utilisée en tant que système de règles logiques sur lesquelles un raisonneur effectue des raisonnements afin d'évaluer les labels à associer à chaque document. Cet ensemble de règles logiques constitue un modèle prédictif.

Le processus d'inférence fonctionne selon deux phases, la phase de hiérarchisation et la phase de réalisation. La **phase de hiérarchisation** consiste en l'analyse des relations entre les classes de l'ontologie. Cette phase peut fournir deux types de résultats. Premièrement la découverte de relations hiérarchiques entre les classes et leur réorganisation de la plus générique à la plus spécifique. Deuxièmement la découverte de classes équivalentes. Dans notre cas d'application, cela signifie que lorsqu'un document est annoté avec une classe subsumée le document est aussi associé aux classes subsumantes. De même que, lorsqu'un document est annoté avec une classe correspondant à une étiquette ayant des classes équivalentes, il est aussi associé aux classes équivalentes. La **phase de réalisation** consiste à trouver toutes les classes les plus spécifiques auxquelles peuvent appartenir les individus. Elle est réalisée par le moteur d'inférence. Les instances de documents peuvent donc bien être associées à de multiples étiquettes (i.e. représentées dans l'ontologie par des classes) organisées ou non de façon hiérarchique en fonction de la façon dont les experts ont modélisé le domaine. En cas d'appartenance à plusieurs classes de la même branche de la taxonomie, seules les plus spécifiques seront conservées.

Etape 6 : Les résultats des raisonnements effectués à l'étape précédente sont présentés et expliqués à l'expert. La plupart des raisonneurs sont capables de détailler les raisons pour lesquelles un résultat de raisonnement est obtenu. Pour chaque résultat si le processus automatisé a fourni une réponse erronée, alors les règles impliquées sont présentées et sont éventuellement corrigées par l'expert. Si des termes d'indexation sont

manquants, ils peuvent être ajoutés par l'expert à l'aide du moteur de recherche assisté.

Etape 7 : Comme dans les sections précédentes, une fois que les termes sélectionnés dans l'interface sont validés, le système reporte automatiquement les informations dans la base de connaissances. Comme il s'agit ici d'indexation de documents, l'indexation se fait là aussi au plus précis. Contrairement aux approches précédentes le peuplement peut aussi consister à mettre à jour certaines règles logiques corrigées par l'expert.

La succession d'étapes 1, 2, 3 et 4 : permet la création du modèle prédictif à partir des données déjà indexées. Elle est présentée dans la figure **??**. Le processus d'apprentissage du modèle n'est exécuté qu'une seule fois. Le modèle prédictif évolue par la suite en suivant les corrections apportées lors de la supervision par les experts.

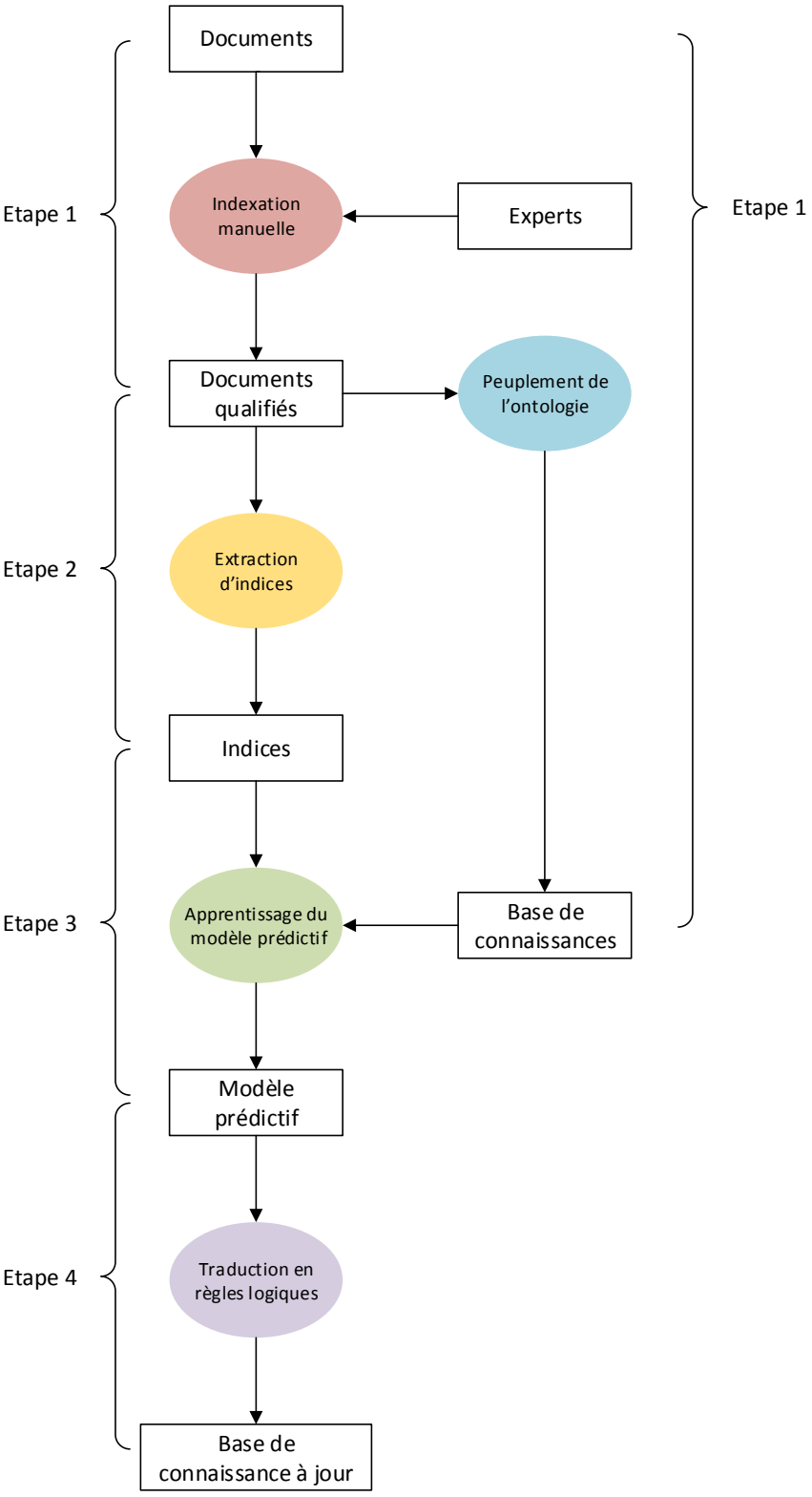


FIGURE 4.2 – Processus d'apprentissage du modèle prédictif

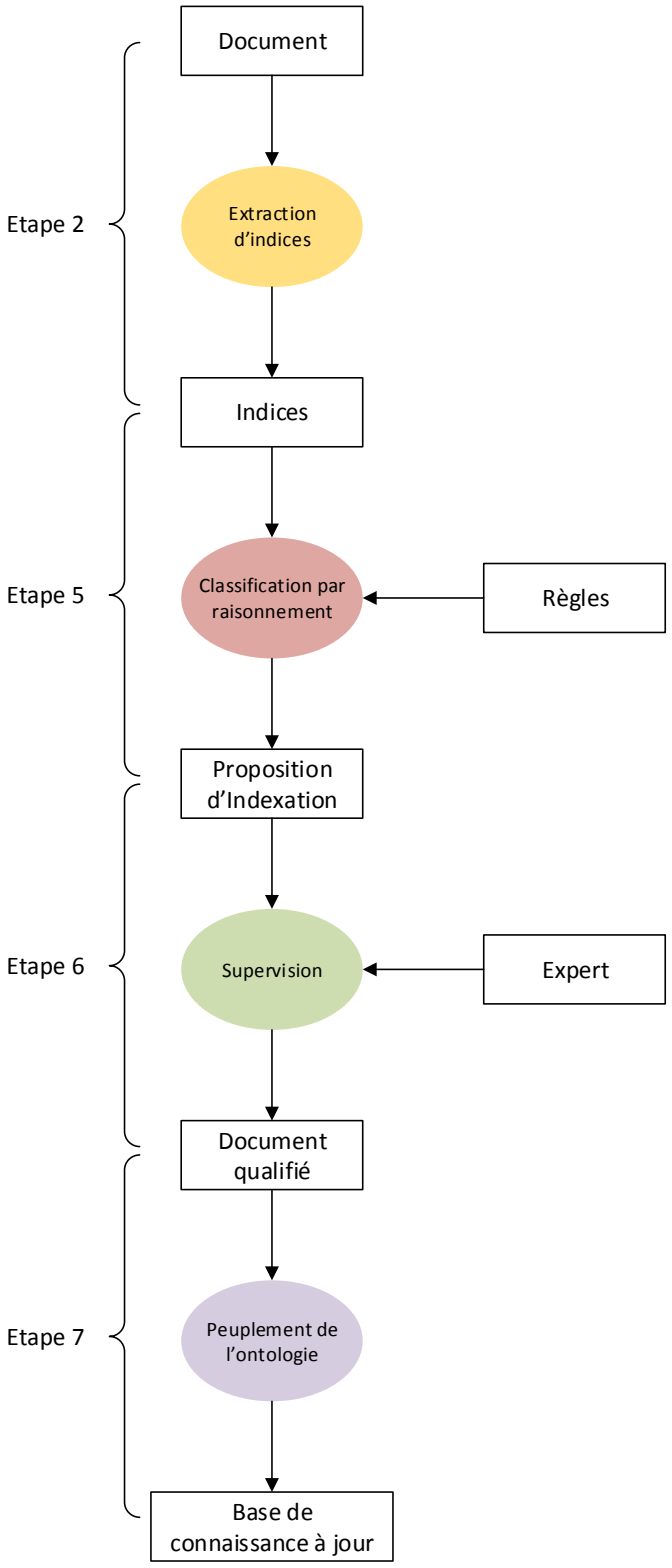


FIGURE 4.3 – Processus d'indexation automatique supervisée de documents

La succession d'étapes 2, 5, 6 et 7 : est exécutée à chaque ajout d'un document au système afin de permettre son indexation. La figure **??** illustre ce processus d'indexation automatique supervisée de documents. Avec cette approche de supervision, le modèle de prédiction s'améliore au fur et à mesure de l'indexation de documents. Le nombre d'erreurs doit progressivement se réduire, facilitant l'indexation des documents.

4.4/ IMPLÉMENTATION DU PROCESSUS AUTOMATIQUE D'INDEXATION

Cette section s'intéresse à l'implémentation de l'approche appliquée à un cas d'étude concret. Afin de confronter notre approche à un travail existant, nous utilisons ici le jeu de données del.icio.us (cf. tableau **??**) disponibles sur le site du projet Mulan [3] et déjà utilisé dans certains travaux sur la classification multi-label [**?**]. Ce jeu de données a été extrait du site web de social bookmarking (i.e. marque-page social, partage de signets) du premier avril 2007. Dans ce jeu de données, des sites web sont associés à des indices textuels ainsi qu'à des termes d'indexation (i.e. labels). Les indices ont été extraits du contenu des sites (i.e. features). Ce jeu de données contient deux sous jeux. Le premier, contenant 12920 exemples de documents indexés, peut être utilisée afin d'apprendre un modèle prédictif de classification. Le second, en contenant 3185, peut être utilisé afin d'évaluer les performances du modèle appris à l'aide du premier. La cardinalité représente le nombre moyen de labels (i.e. étiquettes) par exemple. La densité est une version normalisée de la cardinalité, elle est calculée en divisant la cardinalité par le nombre de labels [**?**].

Dataset	**Exemples**		**Attributs**		**Etiquettes**		
	Apprentissage	Test	Numerique	Nominal	Nombre	Cardinalité	Densité
delicious	12920	3185	0	500	983	19.020	0.019

TABLE 4.1 – Le jeu de données delicious en quelques chiffres.

Avec ce jeu de données, les étapes 1 et 2 concernant l'indexation manuelle ainsi que l'extraction des indices (i.e. features) contenus dans les documents ne sont pas nécessaires. Les features ainsi que les labels sont déjà associés aux documents. Le jeu de données est divisé en deux. Une première partie est dédiée à l'étape 3, c'est-à-dire l'apprentissage du modèle prédictif alors que la seconde est dédié à l'évaluation.

Afin de réaliser la classification multi-label à partir de ce jeu de données, il est nécessaire dans un premier temps de concevoir une ontologie permettant la gestion des connaissances contenues dans celui-ci. Pour cela une ontologie noyau suffisamment générique

3. http ://mulan.sourceforge.net/

pour pouvoir s'adapter à différents domaines a été définie. Cette ontologie doit aussi permettre la gestion des règles logiques utilisées par les raisonneurs pour effectuer l'indexation des documents lors de l'étape 5. Ainsi, dans un second temps, il est nécessaire de définir une méthode pour l'apprentissage d'un modèle prédictif qui puisse se traduire par la création de règles logiques dans l'ontologie. Ceci correspond à l'étape 4. Pour cela nous proposons la création d'un modèle prédictif composé de règles construites sur la base de la présence de termes indices dans les documents. La présence de certains termes ou d'une combinaison de termes indices doit permettre de déduire les labels à associer au document. L'étape 5 est réalisée à l'aide de raisonneurs (i.e. HermiT [4], FacT++ [5], Pellet [6]). Les étapes 6 et 7 de supervision (i.e. correction et complétion) et validation par des experts ne sont pas nécessaires ici. La composition du jeu de données va permettre l'évaluation directe de l'approche de classification automatique.

La première sous-section présente l'ontologie noyau et la création du modèle prédictif. La seconde concerne l'évaluation de l'implémentation réalisée à partir du jeu de données del.icio.us. En fin, la dernière sous-section est une analyse de l'implémentation réalisée.

4.4.1/ ONTOLOGIE ET MODÈLE PRÉDICTIF

Cette sous section présente la création du modèle prédictif et son fonctionnement dans l'ontologie. L'étape 3 de notre approche est composée de deux sous étapes : la vectorisation et la résolution. l'ensemble permettant la création de règles logiques directement utilisables dans l'ontologie, c'est-à-dire, l'étape 4.

La vectorisation consiste à associer à chaque document de l'ensemble d'apprentissage (i.e. documents déjà associés à un ensemble d'étiquettes) un ensemble de termes indices, puis à créer une matrice des fréquences des termes indices en fonction des étiquettes. La résolution se base sur la matrice précédente afin de définir des règles capables de choisir si un document doit être associé à une étiquette sur la base des termes indices qu'il contient. Deux seuils de fréquence sont définis, α et β. Les mots-clés dont la fréquence est supérieure au seuil α sont considérés comme des indices fiables. La présence d'un seul de ces mots est considérée comme suffisante pour que le document soit associé à l'étiquette. Le seuil de fréquence inférieur est β. Dans ce cas, nous avons besoin d'une combinaison de β-termes (i.e. termes dont la fréquence est supérieure à β) pour prendre la décision d'associer un document avec l'étiquette.

4. http ://hermit-reasoner.com/
5. http ://owl.man.ac.uk/factplusplus/
6. http ://complexible.com/

4.4.1.1/ DÉFINITIONS

Ce paragraphe introduit une série de définitions nécessaires à la compréhension des paragraphes suivants.

- Définition 1 : W_j est un terme appartenant à l'ensemble des termes indices extraits des documents à indexer, autrement appelé *feature*.

- Définition 2 : Tax_i est une classe d'indexation, appartenant à une taxonomie de classes, autrement appelée Label.

- Définition 3 : TF_{ij} est la fréquence en pourcentage d'occurrence du terme W_j dans les documents appartenant à la classe Tax_i (i.e. annoté avec le label de la classe Tax_i).

- Définition 4 : α et β sont des seuils.

- Définition 5 : l'ensemble des termes α contient les termes $\omega_\alpha^{Tax_i}$, c'est-à-dire l'ensemble des termes W_j dont la fréquence est supérieure au seuil α pour la classe Tax_i de la taxonomie.

$$\omega_\alpha^{Tax_i} = \left\{ \bigcup_j \{W_j\} \mid TF_{ij} > \alpha \right\}$$

- Définition 6 : L'ensemble des termes β contient les termes $\omega_\beta^{Tax_i}$, c'est-à-dire l'ensemble des termes W_j ayant une fréquence TF_{ij} supérieure ou égale au seuil β et inférieure ou égale au seuil α.

$$\omega_\beta^{Tax_i} = \left\{ \bigcup_j \{W_j\} \mid \beta \leq TF_{ij} \leq \alpha \right\}$$

- Définition 7 : $\overrightarrow{d}$ est un vecteur binaire représentant la présence ou l'absence d'un terme indice W_j dans le document.

$$\overrightarrow{d} = (a_1, ..., a_m) \mid m \text{ est le nombre d'éléments } W_j \text{ et } a_i \in \{0, 1\}$$

- Définition 8 : L'ensemble des termes α, noté $D_\alpha^{Tax_i}$ d'un document $\overrightarrow{d}$ est l'ensemble des termes W_j ayant un TF_{ij} supérieur au seuil α pour la classe Tax_i de la taxonomie.

$$D_\alpha^{Tax_i} = \left\{ \bigcup_j \{W_j\} \mid TF_{ij} > \alpha \ \ et \ a_i \neq 0 \right\}$$

- Définition 9 : L'ensemble des termes β, noté $D_\beta^{Tax_i}$ d'un document $\overrightarrow{d}$ est l'ensemble des

termes W_j ayant un TF_{ij} supérieur ou égal β et inférieur ou égal au seuil α pour la classe Tax_i de la taxonomie.

$$D_{\beta}^{Tax_i} = \left\{ \bigcup_j \{W_j\} \mid \beta \leq TF_{ij} \leq \alpha \;\; et \; a_i \neq 0 \right\}$$

- Definition 10 : La somme des cardinalités des ensembles alpha et beta est définie de la façon suivante :

$$Sum(\alpha) = \sum_{i=1}^{n} |\omega_{\alpha}^{Tax_i}|$$

$$Sum(\beta) = \sum_{i=1}^{n} |\omega_{\beta}^{Tax_i}|$$

Avec n le nombre total d'étiquettes.

- Définition 11 : La moyenne des cardinalités des ensembles de termes α est définie de la façon suivante :

$$Avg(\alpha) = \lceil \frac{Sum(\alpha)}{n} \rceil \tag{4.1}$$

Avec n le nombre total d'étiquettes.

- Définition 12 : La moyenne des cardinalités des ensembles de termes β est définie de la façon suivante :

$$Avg(\beta) = \lceil \frac{Sum(\beta)}{n} \rceil \tag{4.2}$$

Avec n le nombre total d'étiquettes.

4.4.1.2/ L'ONTOLOGIE NOYAU

Nous avons besoin de définir dans le noyau de l'ontologie d'indexation, trois concepts principaux ainsi qu'une relation.

Ces concepts sont : **Feature** afin de définir les termes indices qui apparaissent dans le document (W_j), **Item** de manière à définir les documents à associer aux étiquettes et **Label** en vue de définir les étiquettes qui peuvent être associées aux documents. Un seul rôle est nécessaire, **hasFeature**, il permet de faire le lien entre les documents (i.e. instance de la classe Items) et les indices (i.e. instances de la classe Feature) qu'ils

contiennent et représenté par le vecteur $\overrightarrow{d}$.

- $Item \sqsubseteq \top$
- $Feature \sqsubseteq \top$
- $Label \sqsubseteq \top$
- $Label \equiv \exists hasFeature.Feature$
- $Tax_1 \sqsubseteq Label$
- $Tax_2 \sqsubseteq Label$
- ...
- $Tax_n \sqsubseteq Label$

Pour chaque document, une instance du concept Item est créé ainsi qu'un ensemble de relations avec les instances de Feature correspondant aux mots-clés indices ayant une valeur non nulle dans le vecteur : $hasFeature(d, W_j)|a_i \neq 0$.

L'objectif est ici d'indexer automatiquement des sites web à partir des données du jeu de données delicious. Les sites web sont un type d'item, nous créons donc la classe Website, comme étant une spécialisation de la classe Item. Chaque document du jeu de données devient une instance de la classe Website. Chaque mot-clé indice devient une instance de la classe Feature. La nouvelle instance de Website est mise en relation avec les instances de Feature via la relation hasFeature ou une de ses spécialisations.

Cette version est une version simplifiée de l'ontologie. Elle permet de faciliter le travail des raisonneurs en évitant par exemple l'utilisation de composition de rôles. L'annexe **??** présente une version sémantiquement correcte dans le sens ou une instance d'Item n'est pas définie comme étant une instance de Feature afin de permettre l'indexation.

4.4.1.3/ APPRENTISSAGE DES RÈGLES ET PEUPLEMENT DE L'ONTOLOGIE

Un document est défini à l'aide d'un vecteur $\overrightarrow{d}$ de termes indices (i.e feature) tel que défini par la définition 7.

Pour chaque étiquette, dans le cas d'un ensemble de mots-clés indices α non nuls :

$$Tax_i \equiv \exists hasFeatures \cdot \left(\bigcup_{j=1}^{n} Feature(W_j);\ W_j \in \omega_\alpha^{Tax_i} \right)$$

Un document est annoté avec l'étiquette Tax_i s'il contient au moins un descripteur mot-clé indice (i.e. feature) W_j de l'ensemble des α $\omega_\alpha^{Tax_i}$. Nous nommons ce type de règles, règle de type-A.

Exemple de règle de type-A :

$$Tax\_Dev \equiv \exists\, hasFeature\ some\ \{\ java,\ ruby,\ php\ \}$$

Tax_Dev est une spécialisation de la classe Label, permettant la gestion dans l'ontologie de l'étiquette Tax_Dev. Un item, c'est-à-dire une instance d'Item, ou dans notre cas de sa spécialisation WebSite, sera classé dans la classe Tax_Dev s'il possède au moins un des feature, $java$, $ruby$ ou php.

Pour chaque étiquette, dans le cas d'un ensemble de mots-clés indices β non nuls :

$$Tax_i \equiv \geq \delta\ hasFeatures \cdot \left(\bigcup_{j=1}^{n} Feature(W_j);\ W_j \in \omega_\beta^{Tax_i} \right)$$

Avec $W_j \in \omega_\beta^{Tax_i}$ et $\delta = \lceil |\omega_\beta^{Tax_i}| \times p \rceil$. Un document est annoté avec le label Tax_i s'il contient au moins un pourcentage p du nombre de descripteurs de l'ensemble des β, c'est-à-dire au moins δ descripteurs. Par exemple si l'ensemble des β contient 10 descripteurs ($|\omega_\beta^{Tax_i}| = 10$) et que p vaut 20% alors $\delta = 2$. Nous nommons ce type de règles, règle de type-B.

Exemple de règle de type-B :

$$Tax\_Dev \equiv \exists\, hasFeature\ min\ 3\ \{\ apache,\ cdata,\ linux,\ eclipse,\ mac,\ java,\ ruby,\ php,\ sudo,$$

$$ubuntu,\ python,\ rails,\ mediawiki,\ font,\ md5,\ ctype,\ article,\ msdn,\ os,\ plugin,\ j2ee\ \}$$

Tax_Dev est une spécialisation de la classe Label, permettant la gestion dans l'ontologie de l'étiquette Tax_Dev. Un item, c'est-à-dire une instance d'Item, ou dans notre cas de sa spécialisation WebSite, sera classé dans la classe Tax_Dev s'il possède au moins 3 termes de la liste suivante $apache$, $cdata$, $linux$, $eclipse$, mac, $cobol$, $c++$, c, $sudo$, $ubuntu$, $python$, $rails$, $mediawiki$, $font$, $md5$, $ctype$, $article$, $msdn$, os, $plugin$, js, $nodejs$, $j2ee$.

Pour chaque étiquette, dans le cas d'ensemble de mots-clés indices α et β non nuls ; les deux règles précédentes sont appliquées. Durant l'étape 4, deux classes d'équivalence sont alors définies sur les classes correspondant aux étiquettes dont les ensembles d'α et de β sont tous deux non nuls.

Les deux différents types de règles permettant la création d'un modèle prédictif pour la classification et donc l'indexation de documents, reposent sur des seuils de fréquence. Il est donc nécessaire de définir les valeurs de ces deux seuils, α et β. Pour cela nous

proposons deux approches. (i) Les seuils sont calculés en fonction du nombre moyen d'éléments que doivent contenir respectivement les ensembles. Pour cela nous définissons les formules **??** et **??**. (ii) Les seuils sont propres à chaque étiquette et sont définis en fonction du nombre d'éléments exacts que doivent contenir respectivement les ensembles. Le choix de la taille moyenne ou de la taille exacte des ensembles d'α et de β a deux conséquences lors de la création des règles. Dans le cas de la taille moyenne, il est possible que certains ensembles se retrouvent avec un nombre très élevé d'éléments. Du fait de la complexité polynomiale cela peut rendre les calculs difficiles. D'autres ensembles peuvent se retrouver sans éléments, ce qui ne permet pas la création d'une règle et donc, ne permet d'associer l'étiquette avec des documents. Dans le cas de la taille fixe, il est possible, pour s'assurer la présence du bon nombre de termes indices dans une règle, que des termes non significatifs soient utilisés. Leur fréquence trop faible en faisant de mauvais indices, la règle créée risque d'avoir pour conséquence l'appariement de documents avec de mauvaises étiquettes.

L'algorithme suivant détermine les valeurs α et β pour une valeur objectif donnée. Cette valeur est fixée au début de l'algorithme et correspond au nombre maximum de termes indices que les règles logiques doivent contenir :

```
1
2  ε = 0.25
3  α = 0.5
4  Tant Que ( Objective != Avg(α))
5       Si Objective > Avg(α) Alors α = α − ε
6       Si Objective < Avg(α) Alors α = α + ε
7       ε = ε/2
8  Fin Tant Que
9  Result.Objective → α
10
11 ε = 0.25
12 β = 0.5
13 While ( Objective != Avg(β))
14      Si Objective > Avg(β) Alors β = β − ε
15      Si Objective < Avg(β) Alors β = β + ε
16      ε = ε/2
17 Fin Tant Que
18 Result.Objective → β
19
20 Result.Objective : (α, β)
```

FIGURE 4.4 – Algorithme permettant de déterminer les valeurs des seuils α et β

L'algorithme **??** détermine les valeurs optimales de α et de β pour un objectif donné. α et β sont des seuils de fréquences. Cet algorithme permet de fixer les seuils α et β afin d'atteindre l'objectif donné. L'objectif consiste à produire des règles ayant en moyenne *Objective* éléments. Une valeur pour α et β est fixée au départ. Tant que l'algorithme n'a pas atteint son objectif, les valeurs de α ou de β sont modifiées. La fréquence du seuil augmente d'une valeur ε si le nombre moyen d'éléments est trop important ou diminue dans le cas contraire. A chaque itération la valeur ε diminue afin d'affiner le seuil de fréquence. Pour nos premières évaluations nous avons choisi une moyenne de 10 termes

par règles. La valeur objectif optimal est choisie par expérimentation.

4.4.2/ EVALUATION

Dans cette section, nous présentons une évaluation de l'approche à l'aide du jeu de données *delicious*. Notre approche est applicable à des vocabulaires hiérarchiques ou non hiérarchiques. Dans le jeu de données *delicious* les labels ne sont pas organisés hiérarchiquement.

Notre modèle prédictif est composé de règles α pour les ensembles alpha et β pour les ensembles beta, comme cela est présenté dans la section précédente. L'ontologie est enrichie à l'aide des règles du modèle (étape 4) et peuplée avec les exemples du jeu de données dédiés à l'évaluation. Afin d'évaluer notre approche, différents raisonneurs (i.e. HermiT, FacT++, Pellet cf. annexe **??**) ont été utilisés. Ils permettent la réalisation de la tâche de classification multi-label (étape 5) et donc l'indexation des documents. Les résultats sont par la suite comparés aux résultats idéaux fournis dans le jeu de données, afin de calculer la précision et le rappel. La qualité de la classification n'est pas le seul aspect pris en compte dans l'évaluation. La problématique principale étant la faisabilité de la solution en terme de temps de calcul. Durant l'évaluation, différents raisonneurs ont été utilisés sur différents matériels comme le montre le tableau **??**.

Ensemble d'α et règles de type A	**FaCT++**	**HermiT**	**Pellet**
Core i7 4Go DDR3 Xeon E3 24Go DDR3	**50 s** -	n.e.m. [7] 8 h	n.e.m. 18 h
Ensemble de β et règles de type B	**FaCT++**	**HermiT**	**Pellet**
Core i7 4Go DDR3 Xeon E3 24Go DDR3 Xeon E5 128Go DDR3	n.e.m. n.e.m. **2 h** [9] / out	n.e.m. out [8] out	n.e.m. out out
Ensemble d'α et de β et règles de type A et B	**FaCT++**	**HermiT**	**Pellet**
Core i7 4Go DDR3 Xeon E3 24Go DDR3 Xeon E5 128Go DDR3	n.e.m. n.e.m. **2 h**[9] / out	n.e.m. out out	n.e.m. out out

TABLE 4.2 – Evaluation du temps de calcul des raisonneurs sur des ontologies contenant le modèle prédictif constitué de règles de types A et/ou B.

7. Pas assez de mémoire, Not Enough Memory
8. Temps de calcul supérieur à 3 jours
9. Seule la hiérarchie des labels est inférée (i.e. phase de hiérarchisation). L'ontologie n'a pas été peuplée

Le tableau **??** présente les résultats de l'évaluation. Nous avons lors de ce test utilisé les valeurs suivantes : $objectif = 10$, $p = 20\%$. Les règles de type B sont beaucoup plus consommatrices en temps ainsi qu'en mémoire. Nous n'avons qu'un seul résultat à présenter, avec le raisonneur FacT++ et notre machine la plus performante. Et ce résultat ne prend en compte que la hiérarchisation et non la réalisation. L'évaluation a eu lieu sur une ontologie enrichie par le modèle, mais non peuplée avec les instances de documents. Il faut donc deux heures pour produire une réorganisation hiérarchique des concepts correspondant aux étiquettes utilisées pour annoter les documents.

Le tableau **??** compare la qualité de la classification multi-label réalisée par notre approche avec deux approches proposées dans la littérature. Ces deux approches utilisent le même jeu de données. Cette évaluation de la qualité se base sur l'évaluation de la précision, du rappel ainsi que de la F1-Mesure. Plus d'informations sur ces mesures d'évaluation sont disponibles dans l'annexe **??**. Les résultats de la classification multi-label avec des règles de type B ne sont pas évalués dans le tableau **??** à cause de l'absence de résultats avec cette approche. Les résultats présentés montrent une F-mesure faible. Ces résultats sont meilleurs que l'approche BR (i.e. binary relevance) largement utilisée pour la classification multil-label [**?**] mais moins bon que l'approche HOMER [**?**]. Il est à noter que les trois approches montrent des résultats faibles avec ce jeu de données.

Evaluation	**Precision**	**Recall**	**F1-Mesure**
Eval 1 : Ensemble α avg 10 et règle de type A	30%	6%	10%
Eval 2 : Ensemble α = 10 et règle de type A	1,2%	70%	2,4%
HOMER [**?**]	-	-	14-25%
BR [**?**]	-	-	8,1%

TABLE 4.3 – Comparaison des approches utilisant le même jeu de données

Une seconde évaluation suit cette première évaluation. Pour chaque label l'ensemble des α est défini comme l'ensemble des 10 descripteurs les plus fréquents. La seconde évaluation n'utilise que l'ensemble des α. Les règles de type A et B ne sont pas modifiées. Ainsi pour qu'un document corresponde à une règle de type A, il doit contenir au moins un élément de l'ensemble des α et pour qu'il corresponde à une règle de type B il doit en contenir au moins deux. Les résultats de cette seconde évaluation sont présentés dans les tableaux **??** et **??**. Dans cette évaluation nous utilisons uniquement le raisonneur le plus performant et la machine la plus rapide.

L'ensemble des résultats présentés sont discutés dans la section suivante.

avec les documents, ils n'ont donc pas été associés avec leurs étiquettes (i.e. phase de réalisation)

Règles	Temps	Mémoire
Ensemble α = 10 et règles de type A	**2h26**	**17Go**
Ensemble α = 10 et règles de type B	n.e.m.	$>$ 120Go

TABLE 4.4 – Tests de réalisation avec FacT++ et Xeon E5 128Go DDR3

4.4.3/ ANALYSE ET DISCUSSION

Suite aux différentes évaluations, nous constatons que le type de règles influence énormément le temps de calcul ainsi que la charge en mémoire nécessaire au raisonnement. L'ontologie est peuplée d'un millier de classes et de règles d'équivalence de classe de type A ou B ayant pour but de classer une dizaine de millier d'instances. Lorsque le raisonnement a lieu avec des règles de type B, les temps de calcul et la charge en mémoire sont tellement important qu'aucun traitement n'est arrivé à son terme lors de nos évaluations. Ainsi, la qualité de l'indexation n'a pu être évaluée qu'avec les règles de type A intrinsèquement moins qualitatives que celles de type B.

Notre façon de créer les règles dans la première évaluation, c'est-à-dire avec une moyenne de 10 termes indices (i.e. feature) a pour conséquence la présence de classes d'étiquettes (i.e. labels) n'ayant pas de règles d'équivalence. Cela est dû au fait que leur ensemble α est vide. Pour 983 classes, seules 427 ont une règle d'équivalence, ce qui a pour conséquence le fait que 556 étiquettes ne peuvent être attribuées à des documents. Avec notre approche, une grande partie de ces classent devraient se retrouver avec des règles de type B sur leur ensemble β. Seuls 43% des classes peuvent donc être utilisées afin de décrire des documents. Le rappel s'en trouve donc fortement impacté.

Les règles de type A, sont naïves. Ces règles considèrent que la présence d'un seul terme indice, ayant une fréquence élevée pour le label, car supérieure au seuil α, est suffisant pour annoter un document avec ce label. Les résultats montrent que dans une grande partie des cas, les termes utilisés dans la règle ne sont pas assez significatifs pour permettre, par leur seule présence, de déduire que le document appartient bien à la classe. La présence d'un seul terme n'est donc pas suffisante ce qui impacte négativement la précision.

Dans le deuxième test, le rappel est bien meilleur, car il y a bien une règle par étiquette (i.e. classe spécialisée de la classe Label). Mais les classes qui n'avaient pas de règles dans la première évaluation, se retrouvent avec une règle basée sur les 10 termes les plus fréquents. Dont, la présence d'un seul suffit à satisfaire la règle. Les termes choisis ne sont pas assez significatifs, c'est pour cela que ces classes n'avaient pas de règles de type A, mais des règles de type B lors de la première évaluation. De plus comme nous

nous contentons de la présence d'un seul des 10 termes pour accepter le classement, la précision est très faible, et le rappel très important, car la règle est très aisée à satisfaire.

D'après les deux évaluations, l'utilisation de règles de type B devrait permettre un gain en précision, en permettant une prise de décision basée sur la présence d'un nombre minimum d'indices. L'utilisation d'une méthode permettant de définir les termes descripteurs indices des règles doivent permettre à chaque classe de posséder au moins une règle afin d'impacter positivement le rappel.

Le raisonnement direct avec des règles de type B sur l'ensemble de l'ontologie a un coût très élevé en temps ainsi qu'en mémoire, des solutions permettant de ne raisonner que sur une sous partie de l'ontologie doivent être envisagées.

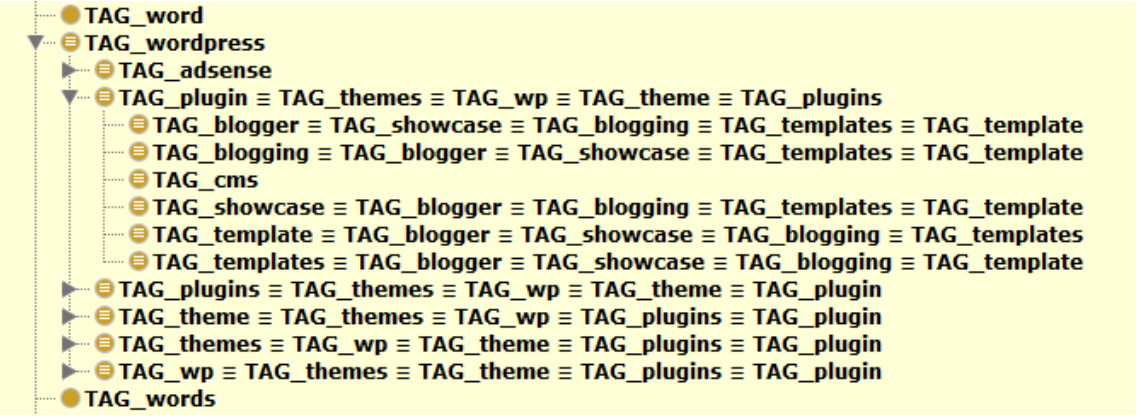


FIGURE 4.5 – Réorganisation des classes correspondantes aux étiquettes lors de la phase de hiérarchisation de l'étape 5

Il est à noter que la phase de réalisation (étape 5), montre des résultats intéressants comme l'illustre la figure **??**. Cette figure présente une partie de la hiérarchie des étiquettes (i.e. spécialisation de la classe Label) inférée durant la phase de hiérarchisation de l'étape 5. Le raisonnement permettant de déduire la hiérarchie est produit à partir des contraintes logiques (i.e. règles A et B) appliquées à ces classes. Cette étape a, par exemple, permis la détection de la proximité sémantique entre les étiquettes "blogueur", "wp" (acronyme pour wordpress), "blogs", "cms" et "wordpress". Bien que cette observation est potentiellement pertinente, une analyse plus approfondie et plus fine doit être effectuée afin de mieux comprendre les avantages, les inconvénients et les applications potentielles.

4.5/ CONCLUSION

Ce chapitre présente le fonctionnement de l'indexation dans notre système de recommandation ainsi qu'une approche pour l'automatisation de l'indexation des documents

textuels. L'objectif est de permettre un gain de temps sur la tâche d'indexation qui est essentielle au bon fonctionnement du système tout en conservant une indexation qualitative et cohérente avec le travail qui aurait été fourni par des experts. En effet, le système de recommandation repose sur une base de connaissances. Cette base de connaissances est une ontologie formelle, ce qui permet de la rendre facilement manipulable par la machine. De plus, l'expressivité de l'ontologie permet une modélisation du domaine riche et en adéquation avec la vision métier des experts, ce qui en facilite la manipulation par les experts.

La démarche proposée vise à la réalisation de l'indexation par une approche de classification multi-label. La classification multi-label de documents textuels est une tâche complexe pour une machine. Une automatisation totale n'était pas envisageable tout en conservant la qualité de l'indexation qui aurait pu être réalisée par un humain. Ainsi, une supervision humaine permettant correction et complétion est nécessaire. Notre proposition consiste à utiliser une ontologie en vue de conserver la modélisation des experts tout en permettant l'intégration d'un modèle prédictif formel. Ce modèle reposant sur des contraintes logiques permet l'utilisation de raisonneurs. La tâche de classification multi-label et donc d'indexation, est le résultat de l'inférence produite par un raisonneur à l'aide des connaissances contenues dans l'ontologie.

Nous proposons dans ce chapitre une approche nouvelle d'indexation par raisonnement ontologique. A notre connaissance aucun travail similaire n'a été réalisé (cf. chapitre **??**. Nous démontrons la faisabilité de cette approche à l'aide d'un prototype et nous comparons cette approche à des approches existantes. Notre approche n'est pas directement utilisable industriellement telle qu'elle a été implémentée. Le jeu de données utilisé pour évaluer l'approche ne comporte pas un vocabulaire d'indexation organisé sur la base de facettes. Ce qui est le cas, du modèle intégrateur proposé dans le chapitre **??** afin de créer une base de connaissances pour un système de recommandation basé sur la sémantique. L'existence ou non de facettes n'influence pas le processus automatisé d'indexation proposé car tous les termes, qu'ils appartiennent ou non au même vocabulaire, se voient affectés un certain nombre de règles (i.e. contraintes logiques) à partir desquelles l'appartenance ou non d'un document à la classe définissant le terme est inférée. Néanmoins, cette évaluation montre que le raisonnement sur ontologie est une tâche coûteuse en mémoire ainsi qu'en temps et que les le type de contraintes logiques nécessaires à une automatisation plus qualitative n'était pas utilisable en l'état. Divers solutions sont envisagées et doivent être étudiées. Des tests ont été effectués afin de réaliser les raisonnements sur des sous-ontologies. Ainsi pour chaque items à indexer, seules les informations nécessaires sont extraites de l'ontologie afin de créer une sous ontologie sur laquelle le raisonnement est réalisé. D'autres approches sont possibles et en cours d'étude comme par exemple l'utilisation de chaînage arrière (i.e. backward chaining).

Dans un contexte d'utilisation au sein d'un système de recommandation sémantique, notre approche, bien qu'évaluée sur un cas d'indexation de documents, peut être adaptée à l'indexation de profils. En effet, il est possible d'extraire des termes indices correspondant à un profil, à partir des documents que l'utilisateur a aimé. Pour cela une analyse de l'historique de l'utilisateur et de son comportement sur le système est nécessaire, afin de détecter les articles aimés. De même, les notes prises par les experts lors du profilage contiennent des termes indices qu'il est possible d'utiliser directement à des fins d'indexation.

La phase de hiérarchisation réalisée par les raisonneurs permet de réorganiser les relations entre les classes sur la base du modèle prédictif appris. L'organisation, notamment hiérarchique des classes entre elles, qu'elle soit préexistante (i.e. modélisation du domaine créé par les experts) ou déduite de la phase de hiérarchisation est directement prise en compte lors de la résolution et influence donc directement la classification. Il s'agit donc d'une classification multi-label hiérarchique. A l'inverse, dans notre implémentation la hiérarchie des classes n'est pas prise en compte lors de la création du modèle prédictif. Cette connaissance de la base de connaissances devrait être utilisée à cette étape du processus afin d'améliorer la qualité du modèle.

A plus long terme, notre approche doit permettre de limiter au maximum les interventions humaines tout en améliorant le système à chaque intervention. L'incapacité lors de cette implémentation à faire fonctionner les règles de type B ne nous a pas permis de poursuivre l'évaluation des tâches de supervision et donc d'étudier l'amélioration du modèle prédictif au fur et à mesure de son utilisation. Toutefois, ce point fait l'objet d'une nouvelle thèse succédant aux travaux présentés ci-dessus. Cette thèse en cours montre la complexité de la problématique ainsi que la possibilité de trouver des solutions, car l'approche est pertinente.

5

LES PROCESSUS DE RECOMMANDATION

Ce chapitre propose une méthode d'évaluation de la pertinence, *PEnSIVE*. Cette méthode permet de prendre en compte de façon efficace les connaissances du domaine afin de fournir une recommandation de qualité. Les chapitres précédents présentent la base de connaissances du système ainsi que les processus permettant de produire une description des items à l'aide du vocabulaire de cette base de connaissances. L'objectif étant de fournir une description du besoin des utilisateurs ainsi que du contenu des documents. Il s'agit, dans ce chapitre, de comparer ces descriptions afin d'évaluer la pertinence d'un document au regard du besoin d'un utilisateur.

Les deux précédents chapitres présentent la base de connaissances de domaine d'un système de recommandation contenant un vocabulaire d'indexation ainsi que les processus permettant de produire des descriptions des items (i.e. profils utilisateur, documents), sur la base de ce vocabulaire d'indexation. L'objet de ce chapitre est la comparaison entre les besoins des utilisateurs, et le contenu des documents. Cette comparaison permet l'évaluation de la pertinence d'un document pour un profil donné. La recommandation proposé consistant, soit à organiser tous les documents, par ordre de pertinence, soit à sélectionner et ne proposer que les plus pertinents (cf. chapitre **??**).

Comme le montre l'état de l'art (cf. chapitre **??**), les systèmes de recommandation basés sur le contenu utilisent majoritairement une modélisation vectorielle des items. Dans ces systèmes, la description des profils et documents est modélisée sous la forme de vecteurs de mots. Ce type de modélisation permet de rendre la description des items manipulables par la machine et donc d'automatiser la tâche de comparaison, et donc de recommandation.

D'autres systèmes, plus récents se basent sur des outils de gestion des connaissances tels que les ontologies. Dans ce type système, nommés systèmes de recommandation basés sur la sémantique (cf. chapitre **??**), ce ne sont pas des vecteurs de mots, mais des vecteurs de concepts qui sont utilisés. Les concepts représentent dans une ontologie, des notions appartenant à un domaine. Les ontologies permettent l'organisation de ces notions, c'est-à-dire l'organisation des connaissances d'un domaine. Plusieurs mots pouvant faire référence à un même concept, ce type de système permet de prendre en compte et de contrôler l'ambiguïté du langage naturel.

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 le document. Selon la base théorique, le profil peut être considéré comme un document idéal, donc plus un document est similaire à ce document idéal (i.e. profil) plus il est pertinent (i.e. plus il correspond aux besoins de l'utilisateur).

La *similarité* est une fonction $Sim(x, y) : I \times I \rightarrow [0, 1]$. Elle permet d'évaluer le degré de similarité entre deux objets x et y. Dans notre cas x est un document et y un profil. La fonction de similarité satisfait les propriétés suivantes [**?**] :

- Positivité $\forall x, y \in I,\ Sim(x, y) \geq 0$
- Réflexivité $\forall x \in I,\ Sim(x, x) = 1$
- Symétrie $\forall x, y \in I,\ Sim(x, y) = Sim(y, x)$

La dernière propriété est sujet à débat dans la communauté [**?**]. Nous conservons l'axiome de symétrie car nous utilisons des algorithmes bien connus permettant la com-

paraison de vecteurs qui sont symétriques (e.g. similarité Cosinus, la similarité Jaccard et la distance Euclidienne).

La *pertinence* est une fonction $Rel(x, y) : I \times I \rightarrow [0, 1]$. Elle permet d'évaluer le degré de pertinence d'un document x vis-à-vis d'un profil y. Cette mesure de pertinence doit respecter les propriétés suivantes :

- Positivité $\forall x, y \in I \; Sim(x, y) \geq 0$
- Réflexivité $\forall x \in I \; Sim(x, x) = 1$

La pertinence est une notion provenant des sciences de l'information, largement utilisée dans le domaine de la recherche d'informations et des systèmes de recommandation [**?**]. Dans notre cas, la pertinence n'est pas binaire. Un item qui doit être recommandé peut plus ou moins correspondre au besoin d'informations d'un utilisateur. Nous souhaitons pouvoir organiser les documents proposés aux utilisateurs par ordre décroissant de leur pertinence, lors de la présentation des résultats (i.e. recommandation). La comparaison dans le contexte d'une modélisation vectorielle permet de produire des résultats non-binaires, c'est pourquoi nous utilisons le modèle vectoriel (i.e. VSM, Vector Space Model) pour estimer la pertinence.

Nous présentons dans ce chapitre une évolution de la méthode de comparaison vectorielle pour l'évaluation de la pertinence d'un document en fonction d'un profil utilisateur.

5.1/ DÉFINITION DES VECTEURS

Dans cette section nous présentons la façon dont sont créés les vecteurs de représentation des items sur la base de l'indexation présentée dans le chapitre précédent. Nous commençons par les vecteurs simples, qui sont des vecteurs de représentation ne prenant pas en compte toutes les connaissances de la base de connaissances. Puis, nous présentons les vecteurs étendus. Ces vecteurs permettent une meilleure prise en compte des connaissances de la base de connaissances et donc du contexte. Cette prise en compte a pour objectif l'amélioration des performances du système [**?**] [**?**].

5.1.1/ LES VECTEURS SIMPLES

La description des items (i.e. le résultat de l'indexation des items), est une connaissance contenue dans la base de connaissances du système (cf. chapitre **??**). Comme le présente la figure **??**, le module de connaissances système de la base de connaissances contient :

- La spécialisation éventuelle des items nécessaires à l'adaptation du système au domaine d'application. Dans notre exemple **Loi** et **Profil**.
- Les instances correspondantes aux items. Dans notre exemple **loi_098678** et **profil_000193**.

Ainsi complétée avec les modules de connaissances générales et de domaines, contenant les facettes descriptives, la base de connaissances permet la prise en charge des descriptions des items. Ainsi, dans notre exemple, que nous nommons le **cas 1**, illustré par la figure **??**, le profil **profil_000193** est qualifié par **Bourgogne_C** en ce qui concerne la facette de description **Localisation** (illustrée par la spécialisation **Localisation_CF** du concept **ConceptFacette**) et la Loi **loi_098678** par **Dijon_C**. Aucune autre facette de description n'est prise en compte dans notre exemple. Afin de faciliter la compréhension en améliorant la visibilité, seule une faible partie des connaissances présentes dans la facette de description **Localisation** sont présentées dans la figure.

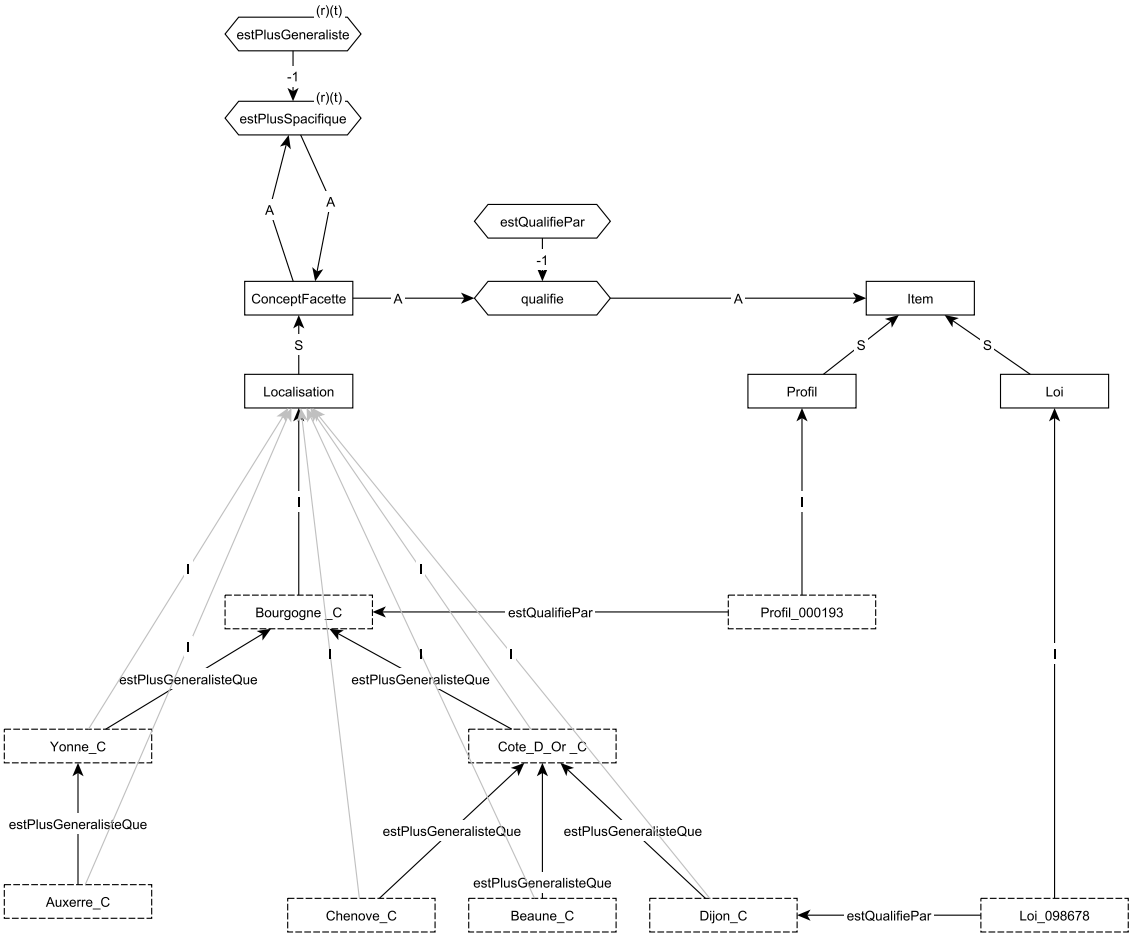


FIGURE 5.1 – Exemple d'indexation utilisant la facette localisation en G-OWL, cas 1

Selon la modélisation de la base de connaissances définie dans le chapitre **??**, nous pouvons représenter la base de connaissances illustrée dans l'exemple (cf. figure **??**) à l'aide d'ensembles. Nous présentons donc ci-dessous l'exemple illustré à l'aide de la méta-modélisation de Karlsruhe adaptée par [**?**].

- $sC = sC_{integrateur} \cup$ {Profil, Loi, Localisation},
- $\leq_C = \leq_{C\ integrateur} \cup$ {(Item, Profil), (Item, Loi), (ConceptFacette, Localiation)},

Les ensembles sC et $\leq_C$ définissent respectivement l'ensemble des concepts et l'ensemble de définition des relations de subsomption entre ces concepts. Ces ensembles sont les mêmes que ceux définis dans le modèle intégrateur, complétés de quelques spécialisations propres à l'exemple. Ainsi les conceptions Profil, Loi et Localisation sont ajoutées à l'ensemble des concepts, sC. Les relations de spécialisation définissant que Loi et Profil sont des spécialisations de la classe Item ainsi que Localisation une spécialisation de la classe ConceptFacettes sont ajoutées à l'ensemble des relations de subsomption, $\leq_C$.

- sI = {Bourgogne_C, Yonne_C, Auxerre_C, Cote_D_Or_C, Chenove_C, Dijon_C, Beaune_C, Loi_098678, Profil_000193},
- iR = {(estPlusSpecifiqueQue, (Auxerre_C, Yonne_C)), (estPlusSpecifiqueQue, (Yonne_C, Bourgogne_C)), (estPlusSpecifiqueQue, (Cote_D_Or_C, Bourgogne_C)), (estPlusSpecifiqueQue, (Chenove_C, Cote_D_Or_C)), (estPlusSpecifiqueQue, (Dijon_C, Cote_D_Or_C)), (estPlusSpecifiqueQue, (Beaune_C, Cote_D_Or_C)), (estQualifiePar, (Profil_000193, Bourgogne_C)), (estQualifiePar, (Loi_098678, Dijon_C))},
- iC = {(Localisation, Bourgogne_C), (Localisation, Yonne_C), (Localisation, Auxerre_C), (Localisation, Cote_D_Or_C), (Localisation, Chenove_C), (Localisation, Dijon_C), (Localisation, Beaune_C), (Profil, Profil_000193), (Loi, Loi_098678)},

Les ensembles sI, iR et iC sont respectivement les ensembles des instances, des relations entre instances et des relations entre instances et concepts.

Tous les ensembles ne sont pas présentés ici. En effet, notre exemple est une application du modèle intégrateur à un contexte d'utilisation visant à l'indexation de lois et de profils en fonction de la facette de description, localisation. Les autres ensembles correspondent exactement à ceux définis dans le modèle intégrateur (cf. chapitre **??**). Nous ne les redétaillons donc pas ici.

Le sous ensemble sC', $sC' \subseteq sC$, correspond à l'ensemble des facettes de description. Ainsi chaque $f \in sC'$ est une facette de description, c'est-à-dire une spécialisation de ConceptFacette. $f \in sC'$ ssi $\exists\,(ConceptFacette, f) \in \leq_C$. Chaque facette de description se voit associer un sous ensemble du vocabulaire d'indexation qui lui est propre. Ce sous ensemble correspond à un sous ensemble des instances de l'ontologie et en particulier à un sous ensemble des instances correspondant au concept ConceptFacette. Le sous ensemble d'instance du concept, ConceptFacette, correspond à l'ensemble des instances permettant la description des items est nommé sI', $sI' \subseteq sI$.

Dans la base de connaissances, le résultat de l'indexation des items, prend la forme d'une instanciation de relations entre les instances représentant ces items (i.e. les instances du concept Item) et les instances de ConceptsFacette (i.e. les instances de l'ensemble sI'), utilisées pour les décrire.

Nous définissons donc un item it_j appartenant à l'ensemble des items IT, comme un ensemble d'instances de concepts de l'ontologie appartenant à l'ensemble sI'. $\forall i_x \in sI'$, $it_j = \{i_1, i_2, ..., i_n\}$ avec $n \in [0; |sI'|]$, $it_j \in IT$ avec $j \in [0; |IT|]$.

L'ensemble des instances propres à une facette f est noté sI'_f, $sI'_f \subseteq sI' \subseteq sI$.

La description d'un item ne prenant en compte qu'une facette donnée est notée $it_{j,f}$. C'est un ensemble d'instances de concepts de l'ontologie apprenant à l'ensemble sI'_f $\forall i_x \in sI'_f, it_{j,f} = \{i_{1,f}, i_{2,f}, ..., i_{m,f}\}$ avec $m \in [0; |sI_f|]$.

L'exemple proposé concerne uniquement la facette Localisation et des items de type Profil et Loi. Ainsi, chaque loi ou profil ne se voit associé qu'une sélection des instances de l'ensemble $sI'_{Location}$. Appliqué à l'exemple (cf. figure **??**), cela donne les ensembles $sI'_{Location}$, $Loi\_098678_{Location}$ et $Profil\_000193_{Location}$ suivant :

- $sI'_{Location}$ = {Bourgogne_C, Yonne_C, Auxerre_C, Cote_D_Or_C, Chenove_C, Dijon_C, Beaune_C}
- $Loi\_098678_{Location}$ = {Dijon_C}
- $Profil\_000193_{Location}$ = {Bourgogne_C}

Avec $Loi\_098678_{Location} \in IT$ et $Profil\_000193_{Location} \in IT$.

Lors de la comparaison, sur la base de vecteurs simples, les ensembles correspondant au profil p et document d sont transformés en vecteurs de description. Ces vecteurs sont construits dans un espace vectoriel dans lequel chaque dimension correspond à un élément des ensembles p et d. Les éléments communs ne forment qu'une seule dimension. Appliqué à l'exemple (cf. figure **??**), cela donne les vecteurs $\overrightarrow{Loi\_098678_{Location}}$ et $\overrightarrow{Profil\_000193_{Location}}$ suivant :

- $\overrightarrow{Loi\_098678_{Location}} = \langle 1,\ 0 \rangle$
- $\overrightarrow{Profil\_000193_{Location}} = \langle 0,\ 1 \rangle$

Les vecteurs $\overrightarrow{Loi\_098678_{Location}}$ et $\overrightarrow{Profil\_000193_{Location}}$ sont donc, dans le cas exemple (cf. figure **??**), représentés dans un espace à deux dimensions. La première dimension représente la présence ou non de Dijon_C dans la description, la seconde celle de Bourgogne_C.

Lors de la création des vecteurs simples, les connaissances de la base de connaissances ne sont pas prises en compte. Ainsi, dans l'exemple, la loi concerne *Dijon* et le profil s'intéresse à la *Bourgogne* (cf. figure **??**). Lors de la comparaison, le système ne sachant pas que *Dijon* est une ville de *Bourgogne*, ne considérera pas la loi Loi_098678 comme étant pertinente pour le profil Profil_000193. En effet, l'évaluation de la similarité entre la loi et le profil, par la méthode de comparaison de vecteurs cosinus (cf. chapitre **??**), équivaut à la comparaison des vecteurs $\langle 0, 1 \rangle$ et $\langle 1, 0 \rangle$. Ces vecteurs étant orthogonaux, ils sont tout à fait dissimilaires. La loi est donc considérée comme ne correspondant pas du tout au besoin de l'utilisateur.

5.1.2/ LES VECTEURS ÉTENDUS

La modélisation vectorielle ne permet pas directement de prendre en compte les connaissances de la base de connaissances (i.e. dans notre exemple, il s'agit de la relation existante entre Dijon et Bourgogne. En effet, [**?**] montre que toutes les dimensions de l'espace de représentation des vecteurs sont orthogonales dans le modèle vectoriel et ainsi, que tous les éléments constituants les vecteurs sont considérés comme indépendants. Cette section traite de la prise en compte de ces connaissances de la base de connaissances par la méthode d'expansion de vecteurs (cf. chapitre **??**), lors de la modélisation vectorielle des items.

Dans les travaux sur les systèmes de recherche d'informations de [**?**] les requêtes des utilisateurs sont représentées sous forme de vecteurs. Afin d'améliorer les performances de ces systèmes par la prise en compte de connaissances externes au système, une méthode visant à ajouter de l'information dans les vecteurs a été introduite. Cette expansion de vecteurs, nommée expansion de requête (i.e. query expansion) permet, par exemple, de pallier certaines formes de complexités inhérentes aux langages naturels. Avec cette méthode, les vecteurs sont étendus par l'ajout de synonymes et de méronymes [1]. La connaissance des relations qui peuvent exister entre les termes proviennent de ressources externes au système. Dans notre exemple, il s'agit de connaissances lexicales qui peuvent, par exemple, provenir de la base de connaissances lexicales Wordnet

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

[**?**]. Plus récemment, cette méthode a été adaptée aux systèmes de recommandation par [**?**] sous la forme d'une expansion des vecteurs de profils à partir de connaissances sur le domaine traité. Dans ces travaux seuls les vecteurs décrivant le besoin d'informations sont étendus, cela dans l'objectif d'augmenter les performances des systèmes en augmentant le rappel (cf. annexe **??**). 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.

Dans notre exemple (cf. figure **??**), l'ajout d'instances en relation *directe* avec les instances déjà présentes dans le vecteur profil, comme cela est le cas dans les travaux précédents [**?**] [**?**], ne nous permet pas, à partir de *Bourgogne_C*, de connaître la relation avec *Dijon_C* et donc d'ajouter *Dijon_C* au vecteur profil. Seule *Cote_D_Or_C* et les trois autres départements de la région sont ajoutés au profil. La pertinence du document pour le profil serait donc toujours nulle. En effet cela revient à la comparaison des vecteurs $\langle 1,\ 1,\ 1,\ 1,\ 1,\ 0\rangle$ et $\langle 0,\ 0,\ 0,\ 0,\ 0,\ 1\rangle$.

Afin d'intégrer, la notion *Dijon_C* au vecteur de profil, il est nécessaire d'aller plus loin que la relation directe. Notre modèle intégrateur est basé sur une ontologie formelle utilisant des propriétés logiques permettant d'effectuer des raisonnements. Ce modèle présente les relations hiérarchiques **estPlusSpecifiqueQue** et **estPlusGeneralisteQue**, comme étant des relations transitives[2]. Ainsi, par raisonnement il nous est possible d'ajouter toutes les instances en relation via transitivité avec les instances déjà présentes dans le vecteur profil. Cela permet alors d'ajouter *Dijon* au vecteur et ainsi de prendre en compte que le document traite bien d'un contenu en relation avec le contenu souhaité par l'utilisateur. L'ajout par transitivité, au-delà des relations directes, ajoute non seulement *Dijon*, mais aussi les quatre mille autres communes de la région au vecteur de profil. La pertinence du document pour le profil ne serait donc pas nulle, mais tout de même très faible alors que dans ce cas la valeur devrait être relativement élevée.

Afin de pallier ce problème, nous proposons une approche consistant à étendre les vecteurs de description des items correspondant à l'offre d'informations (i.e. les lois dans notre exemple), en plus des vecteurs correspondant au besoin d'informations, les profils. De plus, afin de limiter la taille du vecteur, nous ajoutons les ancêtres de l'instance sélectionnée et non les descendants. Nous étendons donc les vecteurs par l'ajout des instances les plus générales en relation avec les instances présentes dans le vecteur via la relation **esPlusSpecifiqueQue**. Notre méthode se rapproche donc des méthodes utilisant la recherche de la profondeur de l'ancêtre commun le plus spécifique dans un graphe sémantique afin d'évaluer la distance sémantique entre deux nœuds [**?**] [**?**] [**?**].

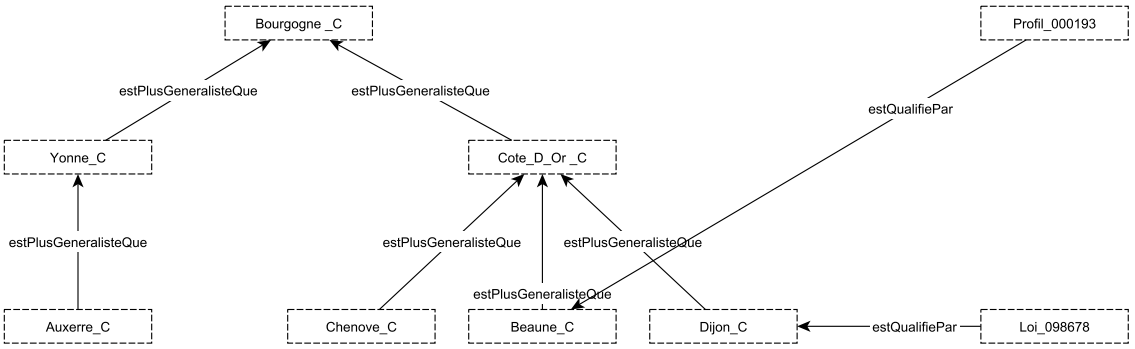
Cela donne dans notre exemple un vecteur document contenant *Dijon_C*, *Cote_D_Or_C* et *Bourgogne_C*, c'est à dire $\langle 1,\ 1,\ 1\rangle$ et un vecteur profil ne contenant que *Bourgogne_C*,

2. Soit une relation R entre éléments de l'ensemble E, R est transitive si $\forall x, y, z \in E, [(xRy \wedge yRz) \Rightarrow xRz]$

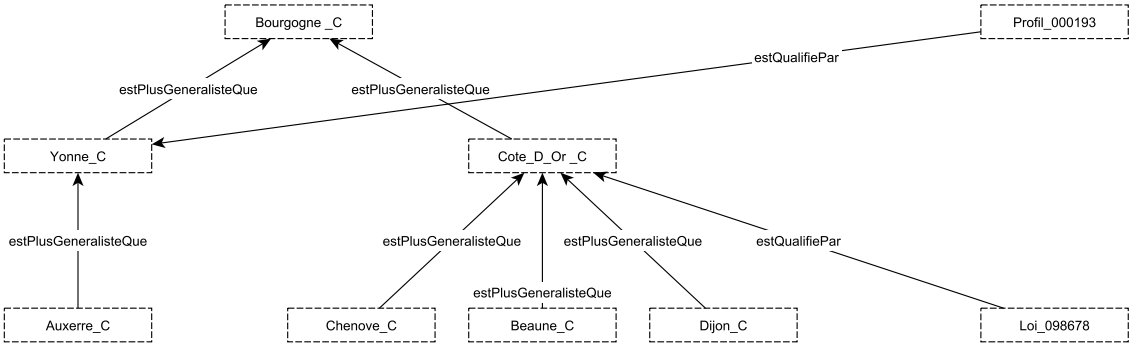
c'est à dire $\langle 0,\ 0,\ 1\rangle$. ce qui donne une similarité cosinus de 58%.

Notre méthode d'expansion nous permet en plus une meilleure prise en compte des cas suivants :

- **Cas 2 :** Contrairement à la figure **??** le document (i.e. la loi Loi_098678) ne traite plus de *Dijon_C* mais plus généralement de la *Cote_D_Or_C*, le profil (i.e. Profil_000193) n'est plus intéressé par la *Bourgogne_C*, mais plus précisément par l'*Yonne_C*. Ce cas est illustré par la figure **??**.
- **Cas 3 :** Contrairement à la figure **??** le profil (i.e. Profil_000193) n'est plus intéressé par la Bourgogne, mais plus précisément à la ville de Beaune (i.e. l'instance *Beaune_C*). Ce cas est illustré par la figure **??**.



Cas 3



Cas 2

FIGURE 5.2 – Exemples d'indexation utilisant la facette localisation en G-OWL, cas 2 et 3

Dans le **cas 2**

- Avec la méthode d'expansion de Voorhees [**?**], c'est-à-dire en ajoutant les concepts plus précis au profil, il faut ajouter les 455 villes de l'Yonne. Cela nous donne deux vecteurs 456 valeurs avec aucune valeur en commun. Soit une similarité de zéro.
- Avec la méthode d'Ijntema [**?**], c'est-à-dire en ajoutant les concepts en relation directe au profil, il faut ajouter vers les concepts les plus généraux directs c'est-à-dire Bourgogne ainsi que les plus spécifiques directs, c'est-à-dire les 455 villes de l'Yonne. Cela nous donne deux vecteurs 457 valeurs avec aucune valeur en commun. Soit une similarité de zéro.
- Avec notre méthode, c'est-à-dire en ajoutant les concepts les plus généraux en relation transitive à la fois au profil et aux documents, il faut ajouter Bourgogne à la fois au profil et au document. Cela nous donne deux vecteurs $\overrightarrow{Profil\_000193_{Location}} = \langle 1,\ 0,\ 1\rangle$ (i.e. $Profil\_000193_{Location}$ = {Bourgogne_C, Yonne_C}) et $\overrightarrow{Loi\_098678_{Location}} = \langle 0,\ 1,\ 1\rangle$ (i.e. $Loi\_098678_{Location}$ = {Bourgogne_C, Cote_D_Or_C}). Soit une similarité par méthode cosinus de 50%.

Dans le **cas 3**

- Avec la méthode d'expansion [**?**], c'est-à-dire en ajoutant les concepts plus précis au profil, rien n'est ajouté, car Beaune est l'instance du concept Location la plus précise de sa branche. Cela donne deux vecteurs $\overrightarrow{Profil\_000193_{Location}} = \langle 1,\ 0\rangle$ (i.e. $Profil\_000193_{Location}$ = {Beaune_C}) et $\overrightarrow{Loi\_098678_{Location}} = \langle 0,\ 1\rangle$ (i.e. $Loi\_098678_{Location}$ = {Dijon_C}). Soit une similarité de zéro.
- Avec la méthode [**?**], c'est-à-dire en ajoutant les concepts en relation directe au profil, il faut ajouter les concepts les plus généraux directs c'est a dire Côte d'Or il n'y a pas de concepts plus spécifiques directs, car Beaune est déjà l'instance du concept Location la plus précise de sa branche. Cela donne deux vecteurs $\overrightarrow{Profil\_000193_{Location}} = \langle 1,\ 0,\ 1\rangle$ (i.e. $Profil\_000193_{Location}$ = {Beaune_C, Cote_D_Or_C}) et $\overrightarrow{Loi\_098678_{Location}} = \langle 0,\ 1,\ 0\rangle$ (i.e. $Loi\_098678_{Location}$ = Dijon). Soit une similarité de zéro.
- Avec notre méthode, c'est-à-dire en ajoutant aux descriptions des items, les concepts plus généraux, en relation transitive avec les concepts déjà présents dans leur description, il est nécessaire d'ajouter Bourgogne et Côte d'Or à la fois au profil et au document. Cela donne deux vecteurs $\overrightarrow{Profil\_000193_{Location}} = \langle 1,\ 0,\ 1\rangle$ (i.e. $Profil\_000193_{Location}$ = {Bourgogne_C, Cote_D_Or_C, Beaune_C}) et $\overrightarrow{Loi\_098678_{Location}} = \langle 0,\ 1,\ 1\rangle$ (i.e. $Loi\_098678_{Location}$ = {Bourgogne_C, Cote_D_Or_C, Dijon_C}). Soit une similarité par méthode cosinus de 68%.

Contrairement à [**?**], dans notre cas l'expansion est automatique. Notre système intègre une base de connaissances basée sur le modèle intégrateur ce qui permet via le vocabulaire contrôlé, de limiter les problèmes d'ambiguïté du langage naturel. L'indexation des items est basé sur ce vocabulaire contrôlé, ce qui a permis l'automatisation de la tâche d'expansion. L'expansion est donc réalisée sans avoir recours à l'humain, ni à des processus complexes de désambiguïsation, mais en inférant à partir des informations de notre base de connaissances.

L'utilisation d'un algorithme de comparaison comme la similarité cosinus avec notre méthode d'expansion de vecteurs, permet une comparaison des documents et profils de façon analogue aux algorithmes d'évaluation de la distance sémantique entre deux nœuds d'une hiérarchie se basant sur la recherche de la profondeur de leur l'ancêtre commun le plus proche [**?**] [**?**] [**?**]. Notre méthode apporte les mêmes avantages que ces algorithmes à savoir la gestion des **cas 2** et **cas 3**. En effet, nous souhaitons prendre en compte que si un utilisateur est intéressé par une ville d'un département, il peut être intéressé par d'autres villes du même département, mais potentiellement moins. C'est le cas ici, la similarité n'étant pas de 100%, mais de 68%. De même *Bourgogne_C* et *Yonne_C*, sont plus dissimilaires que *Dijon_C* et *Beaune_C*, car leur ancêtre commun est plus haut dans la hiérarchie. Ainsi un utilisateur intéressé par la *Cote_D_Or_C*, peut être intéressé par un département voisin comme l'*Yonne_C*, mais potentiellement moins. C'est le cas ici, la similarité n'étant pas de 100% mais de 50%.

5.2/ ESTIMATION DE LA PERTINENCE

Les systèmes de recherche d'informations utilisant une approche vectorielle ainsi que les systèmes de recommandation qu'ils soient basés sur le contenu, sémantiques ou basés sur un filtrage collaboratif, utilisent des fonctions de comparaison de vecteurs. Ces fonctions permettent d'évaluer la similarité entre des documents, entre des profils ou entre des documents et des profils [**?**] [**?**] [**?**] [**?**]. Par exemple, l'évaluation de la pertinence d'un document au regard du besoin d'un utilisateur dans le système [**?**] se base sur une évaluation de la similarité des vecteurs à l'aide de l'algorithme de similarité cosinus. Ainsi, ces systèmes déduisent directement la pertinence d'un item avec la description du besoin du client comme nous l'avons expliqué en introduction de ce chapitre.

Premièrement, nous présentons ci-dessous les principales fonctions permettant la comparaison de vecteurs de description que sont la similarité cosinus, la distance Euclidienne ainsi que la similarité Jaccard étendue. Deuxièmement, nous présentons la formule de la similarité entre document et profil dans notre système. Enfin, nous déduisons de cette formule celle de la pertinence d'un document au regard d'un profil.

5.2.1/ SIMILARITÉ COSINUS

Nous introduisons la formule permettant le calcul de la similarité cosinus de deux vecteurs dans notre état de l'art **??** nous la re-exprimons ci-dessous avec les notations correspondantes.

La **similarité** entre le profil et un document peut alors être mesurée par le cosinus de l'angle formé par les deux vecteurs $\vec{d}$ et $\vec{p}$. Cette mesure est appelée similarité cosinus et est la plus couramment utilisée en recherche d'informations (i.e Information Retrieval) ou en catégorisation et regroupement de documents (i.e Text Clustering) :

$$Sim\left(\vec{d},\vec{p}\right) = \cos\theta = \frac{\vec{d}.\vec{p}}{\left|\vec{d}\right| \times \left|\vec{p}\right|} = \frac{\sum_{x=1}^{t} i_{d,x} \times i_{p,x}}{\sqrt{\sum_{x=1}^{t} i_{d,x}^2} + \sqrt{\sum_{x=1}^{t} i_{p,x}^2}} \tag{5.1}$$

soit θ l'angle entre les vecteurs $\vec{d}$ et $\vec{p}$.

Exemple dans un espace à deux dimensions :

$$Sim\left(\vec{d},\vec{p}\right) = \frac{d_1 p_1 + d_2 p_2}{\sqrt{d_1^2 + d_2^2} + \sqrt{p_1^2 + p_2^2}} \tag{5.2}$$

Le résultat est toujours compris entre 0 et 1 :

$$Sim\left(\vec{d},\vec{p}\right) = \frac{\vec{d}.\vec{p}}{\left|\vec{d}\right| \times \left|\vec{p}\right|} \in [0,1] \tag{5.3}$$

5.2.2/ DISTANCE EUCLIDIENNE

La distance Euclidienne est la mesure de distance la plus classique et la plus connue. C'est une métrique d'évaluation de la similarité classiquement utilisée pour l'algorithme des k-moyennes (i.e. K-means), elle est aussi très utilisée pour des tâches de regroupement (i.e. Clustering) :

$$Sim\left(\vec{d},\vec{p}\right) = \left(\sum_{x=1}^{t} \left|i_{d,x} - i_{p,x}\right|\right)^{\frac{1}{2}},\ t = \left|\vec{d}\right| \tag{5.4}$$

Pour des vecteurs de dimensions t, la distance euclidienne est un cas particulier de la distance de Minkowsky [**?**].

5.2.3/ SIMILARITÉ JACCARD ETENDUE

La similarité Jaccard Etendue [**?**] est basée sur les travaux de Jaccard [**?**]. Cet algorithme est tout comme la similarité cosinus très utilisé en recherche d'informations. La similarité Jaccard Etendue est définie comme le nombre d'objets communs divisé par le nombre total des objets moins le nombre d'objets communs :

$$Sim(\vec{b}, \vec{p}) = \frac{\vec{b}.\vec{p}}{|\vec{b}|^2 + |\vec{p}|^2 - \vec{b}.\vec{p}} \tag{5.5}$$

5.2.4/ SIMILARITÉ MULTI-FACETTES

Nous venons de présenter les différentes formules les plus couramment utilisées pour l'évaluation de la similarité entre deux vecteurs. Comme nous présentons en début de ce chapitre, la description du besoin ainsi que du contenu des documents prend dans notre système la forme de plusieurs vecteurs, un par facette d'indexation (i.e. facette de description). Nous présentons donc ci-dessous une formule permettant l'évaluation de la similarité entre un document et un profil correspondant à notre modélisation multi-vecteurs.

$$Similarite(\vec{b}, \vec{p}) = \frac{\sum(\omega_f \times Sim_f(\vec{b_f}, \vec{p_f}))}{\sum \omega_f} \tag{5.6}$$

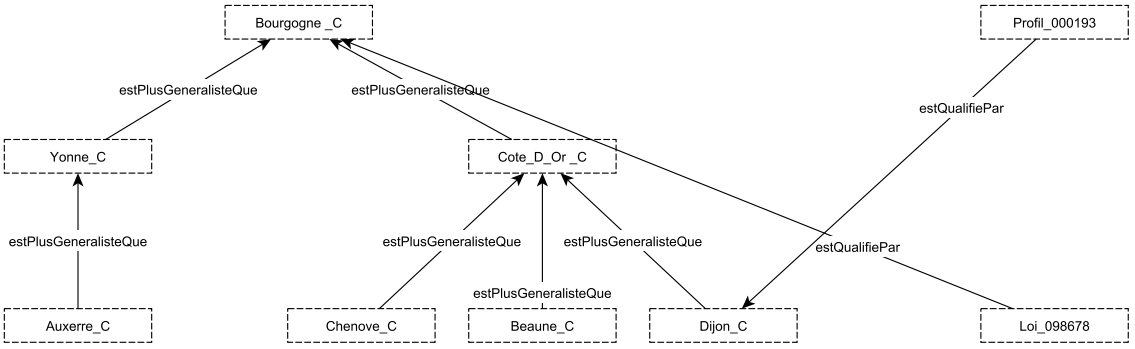
$Sim_f(\vec{b}, \vec{p})$ étant la mesure de similarité entre le profil $\vec{p}$ et le document $\vec{d}$ pour la facette f, telle que $f \in sC'$. Nous utilisons, lors de l'évaluation, alternativement les mesures : Cosinus, Jaccard et Euclide en tant que mesure de similarité $Sim_f(\vec{d}, \vec{p})$. ω_f est le coefficient de pondération défini pour la facette f. $\forall i_{x,f} \in sI'_f$, $\vec{p_f} = \langle i_{1,f}, i_{2,f}, \ldots, i_{n,f} \rangle$ et $\vec{d_f} = \langle i_{1,f}, i_{2,f}, \ldots, i_{m,f} \rangle$ avec n et $m \in [0; |sI'_f|]$. Il est donc possible en fonction de la facette, d'utiliser une mesure de similarité ainsi qu'une pondération différente.

5.2.5/ LA PERTINENCE POUR UNE FACETTE

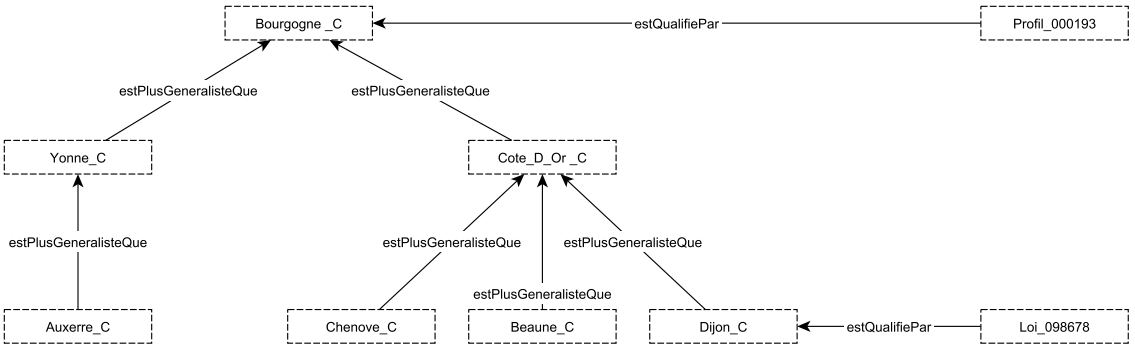
Nous venons de présenter notre méthode d'évaluation de la similarité avec prise en compte des connaissances de la base de connaissances. Nous présentons ici la distinction entre la similarité, mesure symétrique et la pertinence et proposons une méthode d'évaluation de la pertinence basée sur l'évaluation de la similarité. Contrairement à la plupart des systèmes utilisant une modélisation vectorielle pour la recommandation ou la recherche d'informations, nous distinguons ces deux notions. Cette distinction nous permet de prendre en compte la structuration hiérarchique de certains vocabulaires d'indexation contenue dans la base de connaissances du système de recommandation. En

effet, l'utilisation de termes provenant de vocabulaires contrôlés organisés hiérarchiquement permet une définition plus ou moins précise ou générale du besoin d'informations d'un utilisateur ou des informations contenues dans un document.

Nous illustrons ici, la notion de précision en reprenant notre exemple précédent, **cas 1** (cf. figure **??**). Ainsi, si un document traite d'informations à propos de la ville de *Dijon*, nous pouvons dire qu'en terme de précision sur la facette localisation (i.e. la facette *Localisation*), ce document est précis. En effet, *Dijon_C* est une instance au niveau le plus bas de notre hiérarchie de localisations. Le profil lui, s'intéressant à toute la région Bourgogne, l'instance *Bourgogne_C* est plus générale.



Cas 4



Cas 1

FIGURE 5.3 – Exemples d'indexation utilisant la facette localisation en G-OWL, cas 1 et 4

Nous considérons que cette différence influe sur la perception de la pertinence et qu'elle doit donc être prise en compte lors de son évaluation. Or, il n'est pas possible de prendre cette information en compte, lorsque, la pertinence est directement déduite de la similarité, du fait de la symétrie des mesures de similarité.
Ainsi, **cas 1**, si nous évaluons la pertinence du document, loi_098678 pour le profil, $profil\_000193$. Dans l'exemple présenté dans la figure **??**, nous obtenons un vecteur profil $\vec{p} = \langle 0,\ 0,\ 1\rangle$ (i.e. $Profil\_000193_{Location}$ = {Beaune_C,}) et un vecteur document $\vec{d} = \langle 1,\ 1,\ 1\rangle$ (i.e. $Loi\_098678_{Location}$ = {Bourgogne_C, Cote_D_Or_C, Dijon_C}). L'évaluation de la similarité cosinus entre ces deux vecteurs donne un résultat de 58%. En effet, si un utilisateur s'intéresse à la Bourgogne, nous pouvons penser qu'il va s'intéresser aux événements ayant lieu dans une des villes se situant dans cette région. Si nous déduisons directement la pertinence de la similarité, alors le document serait pertinent à 58%. Dans le cas inverse, **cas 4** (cf. figure **??**), l'utilisateur (i.e. $Profil\_000193$) s'intéresse à Dijon, et le document (i.e. Loi_098678) traite de la Bourgogne. Nous obtenons les vecteurs suivants : un vecteur document $\vec{d} = \langle 0,\ 0, 1\rangle$ (i.e. $Loi\_098678_{Location}$ = {Bourgogne_C}) et un vecteur profil $\vec{p} = \langle 1,\ 1,\ 1\rangle$ (i.e. $Profil\_000193_{Location}$ = {Bourgogne_C, Cote_D_Or_C, Dijon_C}). Du fait de la symétrie de la mesure de similarité, nous obtenons une mesure cosinus de 58%, qui, si nous déduisons directement la pertinence de la similarité, donne une pertinence de 58%. Donc, dans les **cas 1** et **cas 4** la pertinence est la même. Or, dans le **cas 4**, le besoin de l'utilisateur est plus précis que dans le **cas 1**. Il y a donc une perte de précision entre le besoin exprimé et l'information fournie qui doit être répercutée sur la valeur de la pertinence évaluée. La pertinence du document au regard du besoin de l'utilisateur doit être plus faible dans le **cas 4** que dans le **cas 1**, ce qui n'est pas possible lorsqu'elle est déduite directement des mesures classiques de similarité.

Afin de prendre en compte l'influence que peut avoir la différence de précision sur la perception de la pertinence, nous proposons la formule **??**. Ainsi, nous définissons la pertinence pour une facette f de la façon suivante :

$$Pert_f(\vec{d_f}, \vec{p_f}) = \frac{\omega'_{1,f} \times Sim_f(\vec{d_c}, \vec{s_f}) + \omega'_{2,f} \times Sim_f(\vec{p_f}, \vec{s_f})}{\omega'_{1,f} + \omega'_{2,f}} \tag{5.7}$$

Avec S_f le sous-ensemble commun d'éléments de l'ensemble d'instances en relation à la fois avec le profil $sI'_{p,f} = p_f$ et le document $sI'_{d,f} = d_f$ pour la facette f ; $S_f = sI'_{p,f} \cap sI'_{d,f}$. La facette f appartient à l'ensemble des classes définies pour la description des items sC' ; $f \in sC'$, $sC' = Localisation$ dans notre exemple (cf. figure **??**). L'ensemble des instances du profil correspondant à la facette f est un sous ensemble de l'ensemble des instance du profil $sI'_{p,f} \subseteq sI'_p$ de même pour les documents $sI'_{d,f} \subseteq dI'_d$. Les instances utilisées pour la description des documents et profils appartiennent à l'ensemble des instances des classes de description $sI'_d \subseteq sI'$ et $sI'_p \subseteq sI'$; $sI' \subseteq sI$. Dans notre exemple, sI' =

{Bourgogne_C, Yonne_C, Auxerre_C, Cote_D_Or_C, Chenove_C, Dijon_C, Beaune_C}, $sI'_{p,f} = Profil\_000193_{Location}$ = {Beaune_C,} et $sI'_{d,f} = Loi\_098678_{Location}$ = {Bourgogne_C, Cote_D_Or_C, Dijon_C}.

$\forall i_{x,f} \in S_f$ le vecteur $\vec{s_f}$ est composé des éléments de l'ensemble S_f ; $\vec{s_f} = \langle i_{1,f}, i_{2,f}, \cdots, i_{t,f} \rangle$ avec $t = |\vec{s_f}|$.

Avec cette méthode, il est possible de pondérer de plusieurs façons la différence de précision entre profils et documents, afin de l'adapter aux besoins. Dans notre cas, nous utilisons $\omega'_{1,f} = 1$ et $\omega'_{2,f} = 4$ car nous considérons que la perte de précision du profil par rapport au document ne doit pas influencer plus de 20% du résultat. Par contre, la perte de précision du document 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 la facette considérée.

5.2.6/ PERTINENCE MULTI-FACETTE

A partir de la formule **??** nous définissons la pertinence globale (cf. formule **??**) comme étant la somme des mesures de pertinence pour chacune des facettes, éventuellement pondérées. Cette méthode d'évaluation de la pertinence, nommée *PEnSIVE* (i.e. PrEciSIon Vsm Extended), est utilisée dans le système de recommandation afin de trier les résultats (i.e. les documents) proposés aux utilisateurs en fonction de leur profil.

$$PEnSIVE(\vec{d}, \vec{p}) = \frac{\sum(\omega_f \times Pert_f(\vec{d_f}, \vec{p_f}))}{\sum \omega_f} \tag{5.8}$$

5.3/ EXPÉRIMENTATION

Nous proposons un système de recommandation, dans lequel les items sont indexés sur la base de vocabulaires contrôlés, contenus dans une base de connaissances sémantique du domaine. Ces vocabulaires contiennent les termes permettant la description des items. Certains des vocabulaires d'indexation, nommés facettes, sont organisés de façon hiérarchiques. Nous proposons une méthode, nommée *PEnSIVE* (i.e. PrEciSIon Vsm Extended), qui permet de répondre à différents cas de figure détaillés dans les sections précédentes. Cette méthode permet d'évaluer la pertinence d'un document au regard du besoin d'un utilisateur en prenant en compte à la fois, la précision de la demande et les connaissances du domaine.

Cette section propose une évaluation de la méthode proposée. Nous avons défini une méthode couplant une expansion des vecteurs profil et document et une prise en compte de la différence de précision entre les descriptions fournies par ces vecteurs. Nous éva-

luons donc ici ces deux supports fondamentaux de notre approche.
Pour cela nous avons élaboré un jeu de tests comportant 10 profils de lecteurs et 70 documents. Cela correspond à la production quotidienne de documents de la société FirstEco. 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 expérimentées 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. La plupart des travaux auxquels nous faisons référence utilisent des outils de mesures classiques provenant du domaine de la recherche d'informations. Ces méthodes classiques permettent une évaluation objective, binaire (i.e. un document est recommandé ou ne l'est pas), or l'ordre dans lequel les documents sont proposés aux utilisateurs est au centre des systèmes de recherche d'informations tout autant que de celui des systèmes de recommandation. Cette section propose donc en complément de l'évaluation binaire une évaluation de l'ordre, nommée *corrélation de rang*. Nous proposons en annexe **??** un tour d'horizon des outils d'évaluation que nous utilisons ci-dessous.

- **Evaluation binaire :** Pour évaluer la recommandation produite par les différents algorithmes cette section se base sur les mesures classiques de précision [3] et de rappel [4].
  Pour cela il est nécessaire que les résultats produits par les algorithmes de recommandation soient binaires. Or ils fournissent des documents de façon triée à l'aide d'une valeur de pertinence, entre 0 et 1. Nous définissons donc un seuil au-delà duquel un item est recommandé et en dessous duquel il ne l'est pas. Le seuil de 0,5 à été choisi par expérimentation. C'est ce seuil qui est utilisé pour l'évaluation de la pertinence binaire ci-dessous.

  Afin de considérer à la fois la précision [5] et le rappel dont l'optimisation est l'objectif principal des recherches dans le domaine des systèmes de recommandation. La F-mesure est une combinaison pondérée de précision et de rappel qui produit des scores allant de 0 à 1. Elle est basée sur la mesure d'efficacité proposée par Van Rijsbergne [**?**].

- **Evaluation de l'ordre :** Pour évaluer l'ordre des documents 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

3. Nombre de documents pertinents retrouvés par rapport au nombre de documents total proposés en réponse d'une requête.
4. Nombre de documents pertinents retrouvés au regard du nombre de documents pertinents que possède la base de données.
5. Dans cette section, le mot *précision* fait uniquement référence à la mesure de la précision de la recommandation, dans le sens précision / rappel.

produisent des scores allant de -1 à 1. 0 étant l'absence de similitude, 1 la similitude complète et -1 l'inverse.

La notion de pertinence est subjective, notre évaluation est complétée par les résultats d'une enquête au près de 120 utilisateurs ayant utilisé notre prototype. Cette enquête constitue une évaluation subjective de l'outil, basée sur la méthode *PEnSIVE* de recommandation.

5.3.1/ EVALUATION OBJECTIVE DE L'ALGORITHME PENSIVE

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

TABLE 5.1 – Comparaison vecteurs étendus et non étendus par des mesures d'évaluation binaires et de corrélation de rang.

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 **??**. Les mesures de similarité Jaccard, Cosinus et la distance Euclidenne ont été utilisées selon les différentes façons de créer les vecteurs. Les résultats de la F1-mesure montrent que lorsque les vecteurs sont étendus afin de prendre en compte les connaissances de la base de connaissances (i.e. méthode **B**), les résultats sont au moins aussi bons qu'avec les vecteurs classiques (i.e. méthode **C**). L'évaluation de l'ordre des documents 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 [**?**], 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 ce qui concerne la précision. Nous confirmons ici les résultats de Middleton [**?**] et Ijntema [**?**] quant à l'intérêt de l'expansion de vecteurs, et nous montrons que notre approche automatique d'expansion ontologique s'inscrit dans ce constat.

Les résultats suivants s'intéressent à l'évaluation de l'apport fourni par la prise en compte de la différence de précision entre la description des profils et des documents lors de la mesure de la pertinence. La méthode *PEnSIVE*, permet de prendre en compte lors de l'évaluation de la pertinence d'un document, son adéquation avec le degré de spécificité du besoin exprimé par le profil de l'utilisateur. 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 (i.e. méthode **B**) et par notre méthode *PEnSIVE* (i.e. 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

TABLE 5.2 – Comparaison des mesures de pertinence avec et sans prise en compte des différences de précision des descriptions par évaluation binaire et de corrélation de rang.

Les résultats de l'évaluation de la recommandation proposée par la méthode de mesure de la pertinence basée sur la similarité directe et par notre méthode *PEnSIVE*, sont présentés dans la table **??**. Les deux méthodes présentées utilisent des vecteurs étendus.

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

5.3.2/ EVALUATION SUBJECTIVE DE L'ALGORITHME PENSIVE

Nous avons souhaité prendre en compte le point de vue des clients sur ce nouveau produit. Pour cela nous avons donc ouvert à un nombre restreint de 120 utilisateurs un prototype utilisant l'algorithme de recommandation *PEnSIVE* (cf. formule **??**). La mesure de similarité utilisée lors des tests est la similarité cosinus (cf. formule **??**). Cette évaluation subjective ne prend pas en compte la comparaison avec les méthodes **B** et **C** contrairement à l'évaluation objective précédente.

Les utilisateurs ont eu accès au système du 16 mai au 27 juin 2013, soit 7 semaines. Les publications de revues ont eu lieu deux fois par semaine au cours de cette période.

A la fin de ces semaines de tests, les utilisateurs se sont vus proposer un questionnaire dont les résultats sont présentés en annexe **??**. En ce qui concerne les résultats de la recommandation qui leur a été fournie, 10% l'ont trouvé très pertinente, 41% pertinente, 35% pas assez pertinente et enfin 14% peu pertinente. En ce qui concerne l'aspect pra-

tique de l'algorithme de recommandation comparativement à la réception des documents non triés, 60% l'ont jugé bien, 24% perfectible, 4% à revoir et 4% ont déclaré ne pas l'avoir utilisé.

Les évaluations objectives et subjectives témoignent de la pertinence de notre approche. Elles mettent en avant que les paramètres les plus efficaces pour la recommandation de documents dans notre contexte d'évaluation sont d'effectuer une expansion des vecteurs profils et documents selon la méthode proposée (cf. section **??**), et de prendre en compte les différences de précision entre l'expression du besoin et la description du contenu des documents comme le permet notre méthode *PEnSIVE*.

5.3.3/ CONCLUSION ET ÉVOLUTIONS

Dans les sections précédentes, nous présentons et évaluons l'algorithme *PEnSIVE* (cf. formule **??**). Bien que cette évolution des systèmes de recommandation vectorielle montre des résultats intéressants, elle s'avère imparfaite. Ainsi l'exemple présenté **??** est une simplification de la réalité visant à faciliter la compréhension. Cet exemple présente un cas simple de documents et de profils indexés sur la facette localisation avec une seule valeur. Or notre système permet la gestion de valeurs multiples pour chacune des facettes. Le **cas 5**, illustré par la figure **??** montre un exemple d'indexation multi-valeur. Ainsi dans cet exemple, sur la facette consternée, le profil s'intéresse à trois termes.

En ce qui concerne la perception de l'utilisateur, un document est pertinent quand au moins une des valeurs du profil correspond pour une même facette à au moins l'une des valeurs du document. Bien que notre algorithme donne de bons résultats, il existe dans les cas de facettes multivaluées, des effets de bords. Les profils que nous pourrions qualifier de larges **cas 5** (cf. figure **??**), car pour une facette donnée ils s'intéressent à plusieurs valeurs, sont, avec cet algorithme, considérés comme équivalents à des profils précis, **cas 6** (cf. figure **??**), c'est-à-dire, ne s'intéressant qu'à une valeur, mais très spécifique.

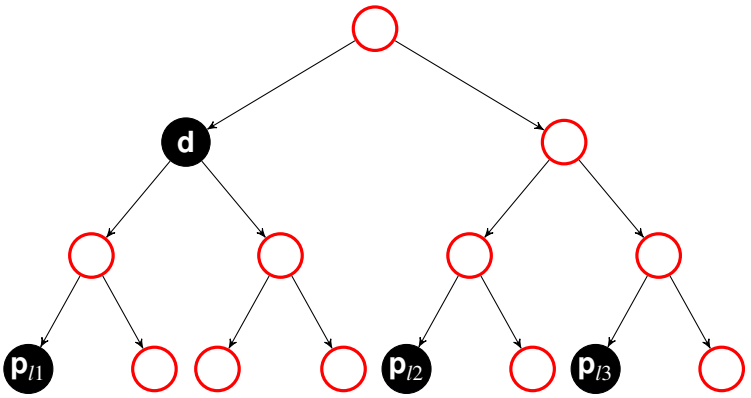


FIGURE 5.4 – Profil large, Cas 5

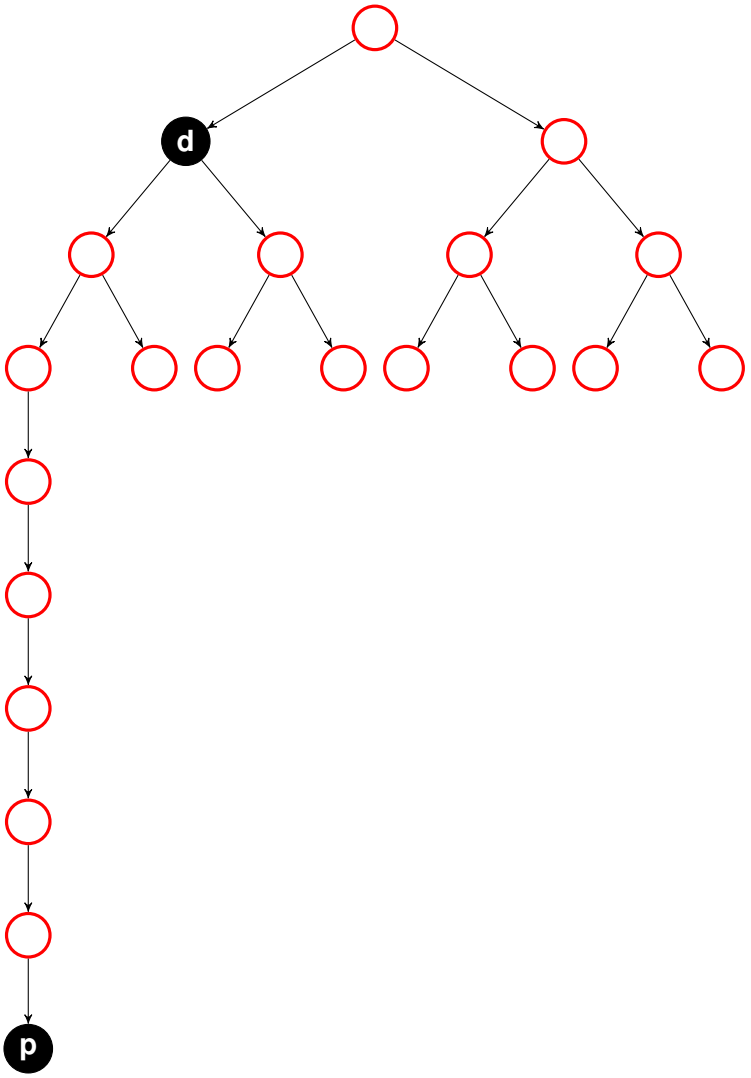


FIGURE 5.5 – Profil très précis, Cas 6

Ainsi notre algorithme fournit la même valeur de pertinence dans les **cas 5** et **cas 6**, alors que nous attendons une pertinence inférieure dans le **cas 6**. De même, la pertinence dans le **cas 5** est très inférieure à celle calculée dans le **cas 7**, alors que nous souhaiterions qu'elle soit équivalente, si ce n'est égale.

		Profil		
		p_{l1}	p_{l2}	p_{l3}
Document	a	**0,8**	0,5	0,5

TABLE 5.3 – Matrice de pertinence par appariement

Une solution simple et naïve permet de contenir ces effets indésirables. Dans les cas de facettes multivaluées, pour une facette donnée, il est possible de comparer deux à deux chaque profil et document. Suite à ces comparaisons, il est possible de sélectionner le meilleur résultat, comme dans le tableau **??**.

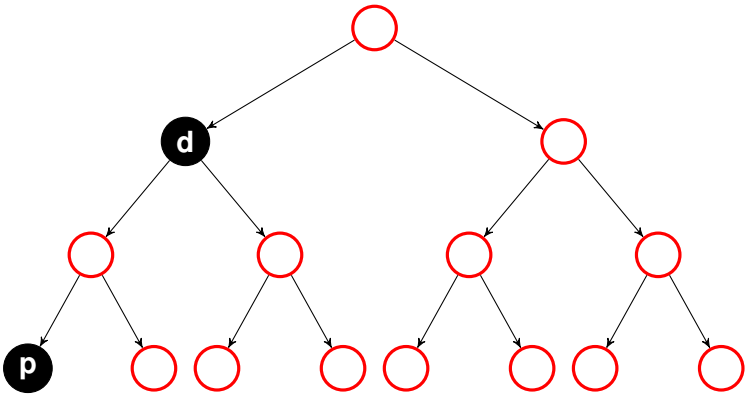


FIGURE 5.6 – Cas 7

Cette méthode permet facilement de gérer la différence de précision entre des profils et documents ayant potentiellement des valeurs multivaluées pour certaines facettes.

En conclusion, notre proposition comble un manque de l'état de l'art en ce qui concerne la gestion du degré de précision lors de la comparaison des descriptions du besoin et de l'offre. L'algorithme proposé montre de bons résultats au cours des évaluations malgré certains défauts. Une nouvelle version de cet algorithme, palliant les défauts rencontrés, est en cours d'évaluation. D'après nos premiers test, les modifications apportées permettent un gain en terme de qualité de la recommandation ainsi qu'en terme de temps de calcul. Ces travaux sont en cours de publication. Ils se basent sur la comparaison efficace de vecteurs binaires. Plusieurs vecteurs sont créés et pondérés pour chaque profil. Ils représentent les différents niveaux de précision du besoin. La pertinence est évaluée en fonction du taux de correspondance des vecteurs correspondant aux documents à recommander avec chacun des vecteurs du profil.

6

IMPLÉMENTATION

Ce chapitre présente l'implémentation et l'architecture de la solution développée ayant fait l'objet des chapitres précédents. Il montre comment cette architecture répond aux limites mises en avant dans l'ancienne revue, *FirstECO* tout en prenant en compte un certain nombre de nouvelles contraintes afin de donner naissance au nouveau produit de l'entreprise partenaire, *FirstECO Pro'fil*.

Ce chapitre présente l'implémentation et l'architecture de la solution développée ayant fait l'objet des chapitres précédents. Ces travaux ont été réalisés dans le cadre d'un financement CIFRE. L'entreprise Actualis Sarl partenaire et financeur de ces travaux est spécialisée dans la production et la distribution de revues de presse.

La revue *First Eco* est un outil de veille externalisé qui fournit quotidiennement des synthèses de l'information économique aux clients de l'entreprise. Les huit éditions sont distribuées par courriels, sous la forme de fichier PDF ou HTML. Chaque édition est composée d'articles rédigés par les experts de l'entreprise suite à un processus de veille qui est détaillé en annexe **??**. Au total ce sont plusieurs centaines d'articles qui sont produits tous les jours par l'entreprise, pour plusieurs milliers de lecteurs. Une description plus poussée de la revue *First Eco* est disponible dans l'introduction du document (cf. chapitre **??**).

L'offre de service fournie par l'entreprise présente trois principales limites :

- La limite **structurelle**. La revue classique (i.e. *First Eco*) se décline en huit sous-revues. Chacune est propre à une zone géographique. La structure des revues est fixe. En l'état, elles nécessitent un travail de filtrage des informations de la part des utilisateurs. Cette tâche est longue, il peut arriver que des utilisateurs passent à côté d'informations importantes par manque de temps.
- La limite **sémantique**. Aucune information ou métadonnée sur les revues ou sur les articles n'est conservée par le système. Les informations ne sont ni indexées ni qualifiées, de façon à faciliter leur exploitation ultérieure tant par la machine que l'humain.
- La limite **pragmatique**. La revue classique ne comprend pas d'outils permettant de comprendre l'usage, le comportement de lecture et le besoin des lecteurs. Le recours à des enquêtes est donc nécessaire, bien qu'il soit difficile de faire répondre les utilisateurs, et qu'elles ne permettent pas de donner une vision des cas particuliers de chaque utilisateur.

Ces verrous sont détaillés dans l'introduction du document (cf. chapitre **??**). La résolution de ces trois verrous nous permet de fournir un nouveau produit hautement compétitif. En plus des limites mises en avant précédemment, l'architecture mise en place doit permettre de répondre à un certain nombre de contraintes :

- **Qualité** : l'architecture proposée doit permettre de conserver la qualité de rédaction des articles ainsi que celle du relationnel client. Afin d'améliorer la qualité de la revue fournie aux lecteurs, il est nécessaire de prendre en compte la spécificité des items qui doivent être recommandés. En effet il s'agit ici, d'articles d'actualité qui,

contrairement à d'autres types d'items, n'ont de valeur que dans un court laps de temps.

- **Rapidité** : le système doit faire gagner du temps aux clients. Une enquête client de 2011 partiellement disponible en annexe **??** montre que les lecteurs de la revue ne consacrent pas plus de 5 minutes par jour à sa lecture, ce qui a pour conséquence la lecture d'un nombre limité d'articles, en moyenne 5. Il est donc important que dans ce court laps de temps ils aient accès le plus efficacement possible aux informations dont ils ont besoin. Le gain de temps pour le client ne doit pas se traduire par une surcharge de travail trop importante pour les experts.
- **Simplicité** : le système doit prendre en compte la nécessité de simplifier autant que possible les interactions requises entre les différents acteurs et processus intervenants.
- **Adaptabilité** : le système doit être adaptable à différents domaines d'application ainsi qu'au point de vue métier de l'entreprise sur ces domaines.
- **Évolutivité** : le système doit prendre en compte l'évolution du domaine d'application traité.

Notre objectif est donc de produire une solution permettant la personnalisation de la revue en fonction du besoin des utilisateurs. L'idée est de donner la bonne information à la bonne personne, et ainsi permettre un gain de temps aux clients. Nous proposons la mise en place d'un système de recommandation basé sur la sémantique, cas particulier des systèmes basés sur le contenu (cf. chapitre **??**).

En effet ce type de systèmes est le plus adapté afin de répondre à la problématique exposée précédemment. Comme le montre l'état de l'art (cf. chapitre **??**), les systèmes basés sur le contenu permettent de recommander des items sans nécessiter qu'ils soient au préalable évalués par un grand nombre d'utilisateurs. De plus contrairement aux systèmes de recommandation collaboratifs, un faible nombre d'utilisateurs n'impacte pas négativement les performances des systèmes basés sur le contenu.

Les systèmes de recommandation basés sur le contenu (cf. chapitre **??**) sont composés de deux modules principaux, correspondant à trois processus nécessaires au fonctionnement global du système de recommandation :

- **Indexation :**
  - **Indexation de l'offre d'information :** la création d'une représentation du contenu des items à recommander compréhensible et manipulable par la machine.
  - **Indexation du besoin d'information :** la création d'une représentation du besoin des clients compréhensible et manipulable par la machine.
- **Comparaison :** un processus de comparaison entre les représentations d'articles et de profils. L'objectif est d'évaluer automatiquement la pertinence des articles pour chacun des utilisateurs et donc de fournir aux clients les articles qui sont en adéquation avec leurs intérêts.

Le choix de la variante basé sur la sémantique des systèmes basés sur le contenu permet de répondre aux contraintes de facilité et d'adaptabilité. En effet les interactions sont simplifiées entre les acteurs utilisant le système, car ils utilisent en plus des modules des systèmes basés sur le contenu, un module, **base de connaissances**. La base de connaissances, c'est-à-dire le vocabulaire formel, contrôlé et commun, limite les ambiguïtés et permet aux humains comme à la machine de "comprendre" et de manipuler plus aisément le contenu des articles. De plus, la base de connaissances constitue une modélisation du domaine traité. Celle-ci permet de prendre en compte une modélisation du domaine dans lequel s'inscrit le système de recommandation et donc d'adapter le système au domaine ou à la vision métier à partir de laquelle la modélisation du domaine a été pensée.

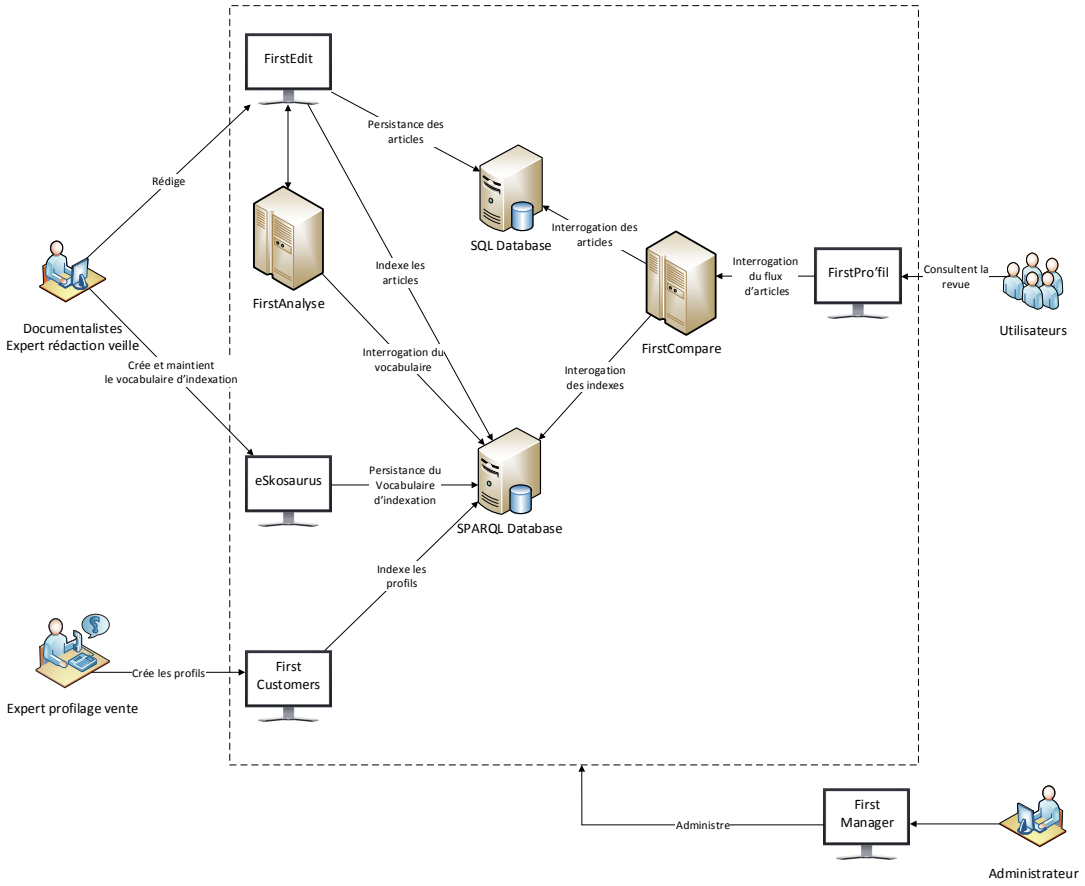


FIGURE 6.1 – Les outils mis en place dans l'architecture de la solution et leurs interactions

L'architecture de la solution développée et mise en production est composée de plusieurs outils illustrés par la figure **??**.

- Une application d'édition et d'indexation des articles, FirstEdit.
- Une application d'édition et de contrôle des vocabulaires d'indexation, eSkosaurus.
- Une application d'indexation et de contrôle des profils, FirstCustomers.
- Une application de consultation de revue personnalisée, FirstPro'fil.
- Un service de comparaison articles / profils, FirstCompare.
- Un service d'analyse de texte, FirstAnalyse.
- Un service de persistance des données de l'ontologie, SPARQL Database
- Un service de persistance des données relationnelles, SQL Database
- Une application qui permet d'administrer l'ensemble, FirstManager.

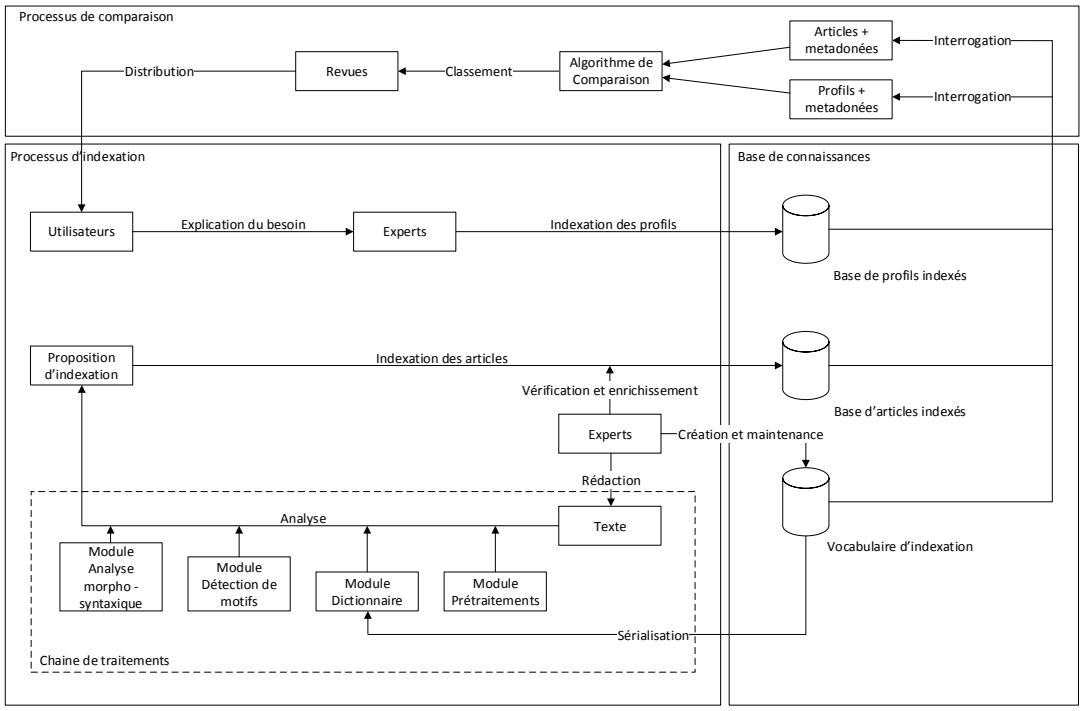


FIGURE 6.2 – Les principaux modules du systèmes et leurs processus

La figure **??** présente les trois principaux modules de la solution, c'est-à-dire les composants principaux d'un système de recommandation basés sur la sémantique. Le détail du contenu et du fonctionnement de ces modules est l'objet des sections suivantes.

Ainsi, dans ce chapitre, nous développons premièrement le module **base de connaissances** présenté dans la figure **??**. Ce module comprend une base d'articles indexés, une base de profils indexés ainsi d'une base des vocabulaires nécessaires à leur indexation. Il permet la persistance des données ontologiques via la base de données SPARQL (i.e. SPARQL Database). La création et la maintenance du vocabulaire d'indexation via l'outil eSkosaurus (cf. figure **??**) sont elles aussi traitées dans la première section.

Deuxièmement, nous développons le module d'**indexation**. Il comprend l'*indexation de l'offre d'information* dans la base d'articles indexés ainsi que l'*indexation du besoin d'information* dans la base de profils indexés, le tout étant réalisé sur la base du vocabulaire d'indexation contenu dans la base des vocabulaires. L'application FirstEdit et l'outil FirstAnalyse (illustré par le sous-module chaîne de traitement dans la figure **??**) permettent l'indexation des articles. L'application FirstCustomers permet l'indexation des profils utilisateurs (cf. figure **??**).

La troisième section présente le module de **comparaison**, c'est-à-dire la recommandation via le service d'évaluation de la pertinence, FirstCompare et l'interface de lecture des clients FirstEco Pro'fil (cf. figure **??**).

Pour finir, nous discutons cette architecture ainsi que les opportunités qu'elle apporte à l'entreprise et nous concluons.

6.1/ BASE DE CONNAISSANCES

Cette section présente le module **base de connaissances** du système (cf. figure **??**), dont la persistance est assurée par les outils SPARQL Database et SQL Database (cf. figure **??**). Cette base de connaissances repose sur le modèle intégrateur détaillé dans le chapitre **??**.

Le module base de connaissances est composé, d'une base d'articles indexés, d'une base de profils indexés ainsi que d'une base des vocabulaires d'indexation (cf. figure **??**).

- La base de profils indexés : contient les descriptions des profils des utilisateurs tels qu'ils ont été créés par les experts en charge des relations clients. Le processus d'indexation manuelle des profils est traité ci-dessous à la section **??**.
- La base d'articles indexés : contient les descriptions des articles rédigés par les rédacteurs documentalistes. Le processus de création et d'indexation semi-automatique, supervisé, des articles est traité ci-dessous à la section **??**.
- La base des vocabulaires d'indexation : contient les termes nécessaires à l'indexation des items. Les termes sont organisés sous la forme de vocabulaires contrôlés et structurés correspondant aux facettes descriptives des items. Le processus de définition des facettes ainsi que celui de définition des vocabulaires associés aux facettes font l'objet des sous-sections suivantes.

Nous présentons ci-dessous les différentes étapes du processus qui nous ont amené à la version stable du vocabulaire d'indexation utilisé aujourd'hui dans la solution en production.

- Etape 1 : définition des facettes.
- Etape 2 : définition des vocabulaires associés aux facettes.
- Etape 3 : évaluation et ajustements.

Ce processus en trois étapes a abouti à une version stable, mais non définitive, du vocabulaire d'indexation. L'architecture mise en place est pensée afin de prendre en compte le cycle de vie du vocabulaire d'indexation. Elle intègre des outils de maintenance et de management qui permettent d'ajouter ou de supprimer des facettes de description.

eSkosaurus (cf. figure **??**) est un outil de maintenance des vocabulaires associés afin justement de faciliter l'ajout, la suppression ou la réorganisation des termes.

Les étapes du processus sont détaillées dans les sous-sections suivantes.

6.1.1/ ETAPE 1 : DÉFINITION DES FACETTES

Les articles contiennent des informations complexes qui peuvent être appréhendées de différentes façons et être perçues sous différents angles. Les *facettes* de description sont les propriétés caractérisant un item. Un profil ou un article (i.e. un item) est décrit en fonction de ces différentes propriétés. Celles-ci sont plus ou moins pertinentes quand il s'agit de qualifier l'information pour indexer ou recommander l'article.

Par exemple, les facettes telles que le nom de l'auteur, le nombre de caractères ou de mots, la région géographique concernée, permettent une description des articles. Toutefois, en fonction de l'objectif d'utilisation, elles ne sont pas toutes pertinentes. Pour un système permettant de générer une revue papier à partir d'articles, le nombre de caractères de chacun des articles, est une caractéristique importante, car l'objectif du système est d'organiser les articles de façon à ce qu'ils tiennent dans la revue. Par contre, pour le lecteur, cette caractéristique sera beaucoup moins pertinente. Il sera probablement plus intéressé par la région géographique ou par le nom de l'auteur d'un article.

La recommandation étant basée sur le contenu des articles, nous nous intéressons ici aux dimensions descriptives substantielles des articles et non aux dimensions descriptives administratives. Bien que notre architecture prenne en compte un certain nombre d'informations administratives (e.g. identificateur, date de publication, auteurs, sources, type d'article, etc.) elles n'influencent pas la recommandation finale et ne sont donc pas détaillées ici.

Notre modèle intégrateur (cf. chapitre **??**) permet la qualification d'items sur la base de facettes de description basées sur des vocabulaires contrôlés. Afin de définir les dimensions descriptives nécessaires à notre système, les principaux schémas de métadonnées existants adaptés à notre contexte ont été étudiés. Cette courte étude est disponible dans l'annexe **??**.

Toutefois, ce n'est pas là le principal champ d'investigation. L'outil est conçu en vue d'être adaptable en fonction du domaine d'application. De plus, notre objectif est de l'adapter au cas d'utilisation précis de l'entreprise. Il est donc nécessaire de s'intéresser au point de vue des utilisateurs de l'application. C'est-à-dire, comprendre les méthodes de travail et les besoins des experts qui alimentent l'outil, ainsi que ceux des lecteurs de la revue. Pour les experts, cela passe par le dialogue direct et l'implication des personnes dans le processus de réflexion et pour les lecteurs par le recours à des enquêtes.

L'analyse des deux principaux schémas de métadonnées existants (cf. annexe **??**) a permis de mettre en avant deux types de facettes, *Subject* et *Coverage* pour la qualification substantielle des articles.

- La facette *Subject* a pour objectif de qualifier le sujet de la ressource à l'aide de mots-clefs, phrases de résumé ou codes de classement en référence aux termes provenant d'un vocabulaire contrôlé prédéfini.
- La facette *Coverage* a pour objectif de qualifier la couverture spatiale (point géographique, pays, régions, noms de lieux) ou temporelle d'une ressource.

L'étude des schémas de métadonnées existants doit être complétée par la connaissance métier des experts de l'entreprise (i.e. responsable de rédaction et son équipe de documentalistes) ainsi que de l'avis des utilisateurs afin de définir les facettes de description les plus adaptées. Pour plus d'informations, le travail des documentalistes est détaillé dans l'annexe **??**.

Lors de la rédaction des articles pour la revue FirstEco classique par les documentalistes, un à deux termes (i.e. mots-clés) sont ajoutés avant le titre ainsi les numéros de département concernés. Cela facilite la sélection des articles pertinents pour le lecteur et donc la lecture de la revue. Dans la revue classique, les articles ne sont pas indexés, et donc, ne sont pas organisés en fonction du besoin du lecteur. Les utilisateurs peuvent ainsi se faire une idée du contenu d'un article avant sa lecture afin de décider de le lire ou non. Dans la figure **??** on peut voir les numéros de département 59, 62 et 80 ainsi que les termes PRESSE, PECHE et DEVELEOPPEMENT ECONOMIQUE qualifiant trois articles de la revue FirstEco de la région Nord datés du 5 janvier 2011.

Les termes peuvent être apparentés à la facette de description *Subject* et les numéros de département à l'aspect couverture spatiale de la facette *Coverage* que nous avons mise en avant suite à l'étude des schémas NewsML et Dublin Core (cf. annexe **??**).

Les experts nous ont permis de préciser la couverture des termes (i.e. mots-clés) utilisés afin de qualifier le sujet des articles. Deux types d'informations distinctes sont utilisés par les experts afin de qualifier le sujet des articles. Dans certains cas, il s'agit du ou des *secteurs* économiques dans d'autres cas du ou des évènements économiques dont traite l'article, parfois il s'agit d'une combinaison des deux. Les évènements économiques sont par la suite nommés *thèmes* de l'article.

(59) PRESSE - La Presse Flamande va investir 3M€ pour installer cinq tours supplémentaires sur sa rotative à Hazebrouck

La Presse Flamande, société éditrice de L'Indicateur des Flandres, va investir 3M€ afin d'offrir aux lecteurs un journal en couleur et plus fourni. La société hazebrouckoise, qui imprime chaque semaine 14 titres de la presse hebdomadaire régionale (dont Le Phare Dunkerquois, La Croix du Nord ou encore Le Journal de Montreuil), va ainsi équiper sa rotative de cinq tours quadrichromie supplémentaires. Elles permettront d'augmenter la pagination des journaux jusqu'à 72 pages. Les nouveaux équipements, fabriqués aux Etats-Unis, seront réceptionnés en avril prochain. Le premier Indicateur tout couleur devrait sortir des presses courant juin. (www.lindicateurdesflandres.fr)

Source : L'Indicateur, 29/12, p.9 - Synthèse : First Eco

(62) PECHE - Le Boulonnais Euronor passe sous pavillon britannique

L'armateur britannique UK Fisheries Ltd vient d'acquérir Euronor, dernier armateur de pêche hauturière boulonnaise.
Euronor exploite six chalutiers de pêche hauturière pour le lieu noir et les espèces de grand fond en Ouest-Ecosse et mer d'Irlande. La société restera basée à Boulogne-sur-Mer.
Rappelons qu'Euronor, née début 2006 de la fusion des armements Le Garrec et Leduc, a été en 2010 la première pêcherie française à obtenir l'écolabel MSC (Marine Stewardship Council), certifiant qu'elle pratique "une pêche durable et bien gérée". (www.euronor.fr)

Source : Voix du Nord, 4/01 - Synthèse : First Eco

(80) DEVELOPPEMENT ECONOMIQUE - Un éco-village d'entreprises va voir le jour dans les prochains mois à Amiens

Le premier éco-village d'entreprises d'Amiens va sortir de terre courant 2011. Situé près du Pôle Jules Verne, il s'étendra sur 35.000m². Les six premières cellules seront disponibles début 2012. Les futures entreprises locataires pourront choisir d'en occuper une, deux voire trois. Avec cet éco-village, la CCI espère séduire des sociétés extérieures au territoire, en particulier d'Ile-de-France ou du Nord-Pas de Calais, désirant développer leur activité dans la région.

Source : Entreprises 80, N°138 - Synthèse : First Eco

FIGURE 6.3 – Extrait de la revue FirstEco Nord 05-01-2011

Les principales facettes permettant de définir l'information contenue dans les articles sont donc :

- Secteur Économique : ce critère correspond aux secteurs économiques traités par l'article. Par exemple : énergie, transport, Informatique, etc.
- Thème : ce critère correspond aux principaux évènements économiques traités par l'article. Par exemple : OPA, délocalisation, embauche, etc.
- Localisation : ce critère correspond aux villes, régions, départements et pays traités dans l'article.

Une enquête menée auprès des lecteurs valide le fort intérêt des clients pour une qualification des articles à l'aide de ces facettes. Des extraits des enquêtes sont disponibles en annexe **??**. Une fois les facettes définies, il est nécessaire de définir les vocabulaires qui y sont associés. C'est l'objet de la sous-section suivante.

6.1.2/ ETAPE 2 : DÉFINITION DES VOCABULAIRES ASSOCIÉS AUX FACETTES

La première étape décrite ci-dessus a permis la définition des facettes suivante : *Secteurs économiques*, *Thèmes* et *Localisations*. Il est donc nécessaire de définir les vocabulaires associés à ces facettes.

Afin de faciliter cette tâche, un outil permettant de créer, d'importer et de maintenir des vocabulaires contrôlés a été développé, eSkosaurus, illustré par la figure **??**.

Au cours du cycle de vie de la solution, le vocabulaire d'indexation peut être amené à évoluer. Par exemple, les systèmes de classification utilisée en bibliothèques que sont les CDU et CDD (cf. section **??**) sont régulièrement mises à jour. Ces mises à jour permettent d'intégrer de nouveaux termes, de nouveaux domaines, de nouvelles notions, ou

tout simplement de prendre en compte les évolutions des domaines préexistants dans la classification. L'architecture mise en place a été pensée pour permettre d'ajouter ou de supprimer des facettes. De même, l'architecture inclut des outils de maintenance des vocabulaires afin justement de faciliter l'ajout, la suppression et/ou la réorganisation des termes composant les vocabulaires associés aux facettes.

Comme l'illustre la figure **??**, cet outil est intégré à l'architecture, afin de permettre la maintenance du vocabulaire d'indexation au cours du cycle de vie de la solution. L'outil permet de visualiser en direct les relations entre les différents termes d'un vocabulaire comme cela est visible dans la partie gauche de la figure **??**. La partie de droite permet pour un terme donné l'édition de notes d'utilisation explicatives, de synonymes, de définitions ainsi que des relations avec les autres termes.

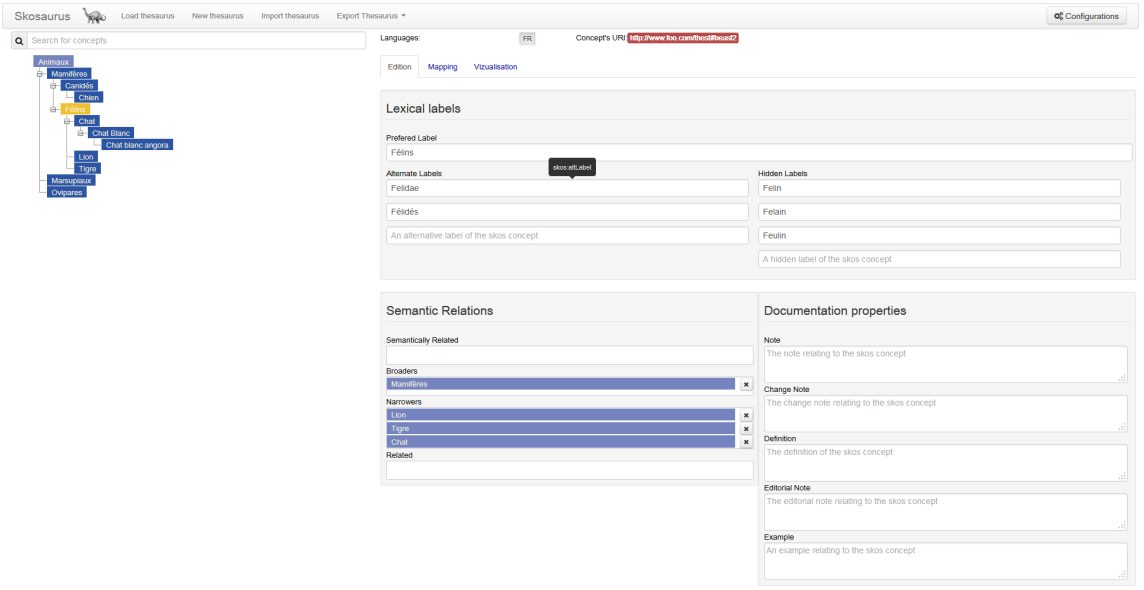


FIGURE 6.4 – Interface de l'application eSkosaurus

Le modèle unificateur défini dans le chapitre **??**, permet de structurer sur la base d'un même modèle formel unique, les différents types de vocabulaires contrôlés existants. Cela permet de faciliter leur intégration et leur utilisation dans le modèle intégrateur. Ce modèle unificateur est très proche de ce qui est permis par le langage SKOS [1]. En effet ce langage qui est une recommandation du W3C depuis le 18 aout 2009 vise à faciliter la publication et l'échange de vocabulaires contrôlés. Nombre de vocabulaires contrôlés au sein des entreprises ou publiés publiquement par celle-ci ou des organismes publics reposent sur ce langage. Notre outil est basé sur ce langage afin de faciliter l'intégration de vocabulaires existants dans notre base de connaissances reposant sur le modèle intégrateur. De plus, cela permet aussi de faciliter la publication éventuelle de certains des vocabulaires de l'entreprise.

1. http ://www.w3.org/TR/2009/REC-skos-reference-20090818/

La facette *Localisation* a été créée à partir de vocabulaires existants provenant de données ouvertes IGN [2] et Geonames [3]. Il en résulte un vocabulaire organisé hiérarchiquement selon quatre niveaux de granularité : Pays, Régions, Département et Communes. Chacun de ces niveaux peut être vu comme une sous facette composée d'une liste plate de termes. S'ils existent dans les vocabulaires étudiés, les synonymes et variations orthographiques sont conservés en tant qu'alias. Ils permettent ainsi de proposer le bon terme lors d'une recherche assistée par exemple.

Les vocabulaires des facettes *Secteurs économiques* et *Thèmes* résultent de la méthodologie suivante :

- Définition du domaine : durant cette phase, les documentalistes ont délimité les domaines couverts par les deux facettes. L'objectif est de définir les limites ainsi que d'identifier les grandes notions. Au terme de cette phase, 25 secteurs d'activité principaux ainsi que 5 thèmes ont été dégagés. Ils servent de base à la construction des vocabulaires.

- Analyse de l'existant : il s'agit de rechercher les vocabulaires existants couvrant tout ou partie des notions nécessaires mises en lumière à l'étape précédente. Dans notre cas les vocabulaires pertinents sont principalement le thésaurus Delphes [4] ainsi que la nomenclatures NAF [5] . Il s'agit de les évaluer : sont-ils suffisants ? Sont-ils assez riches ? Leur structure correspond-elle aux besoins ? Pour cela quelques essais d'indexation sont effectués sur un échantillon représentatif de documents. Bien que les vocabulaires cités ne soient pas directement utilisables, car ils ne répondent pas à certains critères, ils sont une base de travail quant à l'organisation et à la variété du lexique.

- Collecte des termes : l'objectif est ici de créer une liste de termes candidats pour chaque vocabulaire (i.e. candidats-descripteurs d'après la littérature). Pour cela, les termes issus de vocabulaires existants, ainsi que des documents à indexer sont analysés. La connaissance des experts ainsi que des lecteurs à travers leurs façons d'exprimer leurs besoins a été étudiée. Comme cela a été présenté précédemment (cf. section **??**) et illustré par la figure **??**, lors de la rédaction de la revue classique, les rédacteurs sont libres de placer un certain nombre de mots clés avant le titre d'un article afin d'en qualifier le contenu. Une des premières tâches consiste donc à extraire la *folksonomie* (cf. section **??**) de termes utilisés dans les plus de 20 000 revues ayant été produites depuis 2004. Cela nous fournit un vocabulaire d'indexation de plus de 3000 termes.

2. http ://data.ign.fr/
3. http ://www.geonames.org/
4. http ://www.indexpresse.fr/project/delphes/
5. http ://www.insee.fr/fr/methodes/default.asp ?page=nomenclatures/naf2008/naf2008.htm

- Classer les candidats-descripteurs : cette étape a pour objectif de regrouper les termes qui traitent de notions proches voire synonymes.
- Sélectionner les descripteurs : les termes descripteurs ainsi que non-descripteurs sont sélectionnés à partir de la liste des candidats-descripteurs. Dans chaque ensemble de termes synonymes ou quasi-synonymes représentant le même concept défini comme pertinent, un des termes est choisi comme descripteur, le terme principal. Généralement, c'est le plus utilisé qui est sélectionné. Les termes non choisis comme descripteurs, c'est-à-dire les termes synonymes ou quasi-synonymes, deviennent des non-descripteurs. Nous nommons le terme principal *alias principal* et le terme synonyme, *alias secondaires* dans le chapitre **??** illustrant le modèle intégrateur sur lequel repose la base de connaissances qui intègre les vocabulaires contrôlés propres à chaque facette.
- Construire si besoin des relations : l'étape précédente permet de mettre en place les relations d'équivalence entre les termes traitant d'un même concept (i.e. d'une même notion). Le vocabulaire peut être organisé à l'aide d'autres relations, notamment les relations hiérarchiques ainsi que celles de voisinages (cf. section **??**).
- Définir si besoin des notes : il est possible de définir des notes afin de préciser aux utilisateurs du vocabulaire les définitions ou cas d'utilisation de certains termes du vocabulaire.
- Tester : Comme lors de l'analyse de l'existant, le vocabulaire défini doit être testé. Cela consiste principalement à le mettre à l'épreuve en l'utilisant pour l'indexation de documents représentatifs du corpus pour lequel il a été créé. L'évaluation des vocabulaires est traitée dans la sous-section **??** suivante. L'objectif est de confirmer que, (i) le lexique et la structure relationnelle répondent aux besoins, (ii) les descripteurs choisis sont utiles pour l'indexation et la recherche de documents, (iii) le réseau de relations est bien représentatif de l'organisation du domaine couvert et (iv) éventuellement cette phase peut permettre d'identifier des termes manquants.

La sous-section suivante traite du retour sur expérience de l'utilisation d'un prototype basé sur les facettes et un vocabulaire défini lors des étapes 1 et 2.

6.1.3/ ETAPE 3 : ÉVALUATION ET AJUSTEMENTS

Suite à la définition des facettes ainsi que des vocabulaires qui y sont associés, un prototype les intégrant dans sa base de connaissances à été développé. Le prototype a fait l'objet de plusieurs semaines d'utilisation par les experts de l'entreprise ainsi qu'un panel de lecteurs sélectionnés. Le retour sur expérience des experts ainsi qu'une seconde enquête auprès des lecteurs ayant participé aux tests (cf. annexe **??**) a permis de valider

la liste de facettes définies précédemment et de mettre en avant une seconde liste de facettes permettant d'affiner le filtrage des informations. Des vocabulaires simples (i.e. listes plates de termes), ont été définis par les experts et associés aux facettes secondaires.

Les facettes secondaires sont :

- Temporalité : ce critère permet de prendre en compte la temporalité des évènements auxquels l'article fait référence. Il permet de définir si un évènement est passé, présent ou futur. Elle permet de compléter la facette *Coverage* mise en avant lors de l'étude des schémas existants (cf. annexe **??**) et donc préciser la couverture temporelle d'un article en plus de la couverture spatiale déjà prise en compte par la facette localisation.

- Taille des entreprises : ce critère permet de prendre en compte la taille des entreprises auxquelles l'article fait référence. C'est-à-dire : TPE (0-19 employés), PME (20-249 employés), ETI/Grande Entreprise (+250 employés).

- Type de site : ce critère permet de prendre en considération le type d'activité des sites concernés par l'information. C'est-à-dire : industrielle (usine), tertiaire (bureaux), logistique (entrepôt), commerciale (surface de vente) et publique / parapublique.

Les facettes secondaires sont moins importantes que les principales pour les utilisateurs (cf. annexe **??**). Elles sont prises en compte, mais influencent moins les résultats de la recommandation (cf. sous-section **??**). Elles permettent aux utilisateurs un filtrage plus précis de leurs besoins.

Le prototype permet de tester les vocabulaires associés aux facettes principales *Secteurs économiques*, *Thèmes* et donc de détecter et corriger des lacunes quant à la couverture ou à la structure de ceux-ci.

6.2/ INDEXATION

La recommandation automatique d'articles aux utilisateurs nécessite la compréhension par le système du contenu des articles ainsi que des besoins de chacun des lecteurs. La base de connaissances mise en place repose sur le modèle intégrateur introduit dans le chapitre **??**. Ce modèle formel, manipulable par la machine, permet l'indexation des items sur la base de vocabulaires contrôlés correspondant aux facettes de description dont le processus de définition a fait l'objet de la section précédente.

6.2.1/ INDEXATION DES ARTICLES

Le processus d'indexation des articles, illustré dans la figure **??**, utilise deux outils spécifiquement développés à cette fin, FirstEdit et FirstAnalyse (cf. figure **??**). FirstEdit est une application intégrée à l'architecture de la solution, dédiée à la rédaction et à l'indexation des articles. Elle permet l'alimentation du système en articles ainsi que la "compréhension" des articles par le système. C'est l'indexation qui permet aux systèmes de "comprendre" le contenu des articles et donc, permet leur recommandation. FirstEdit fait appel à l'outil, FirsAnalyse afin d'automatiser une partie de la tâche d'indexation.

En fonction de l'information traitée, différents types d'articles peuvent être rédigés.

- Articles locaux : information à portée purement locale.
- Articles nationaux : information à portée nationale.
- Articles flash : article court, peu d'informations disponibles, informations à moindre valeur ajoutée ou complément d'une information déjà publiée.
- Articles de type appels d'offres ou annonces légales.

L'ajout d'un article (cf. figure **??**) au flux d'articles à recommander se déroule de la façon suivante :

1. Rédaction d'un article.
2. Analyse automatique.
3. Supervision et enrichissement.

Rédaction d'un article. Après avoir détecté une information intéressante, regroupé et lu toutes les sources traitant cette information, les rédacteurs (i.e. documentalistes) synthétisent l'information sous la forme d'un article. Le processus de travail des documentalistes est détaillé en annexe **??**.

FirstEdit propose un formulaire de création des articles qui s'adapte en fonction du type d'article que le documentaliste souhaite rédiger. Les articles sont composés d'un titre, d'un corps, d'informations sur les sources et sur les entreprises citées.

Lors de la rédaction du titre, de premières informations sur les facettes **Localisation** et **Secteur d'activité** sont renseignées afin de présenter ces informations dans le titre ce qui permet de conserver une rétrocompatibilité avec la revue classique existante (cf. section **??**.

Le secteur est renseigné grâce à un système de complétion automatique. Afin de faciliter la rédaction, lors de la frappe, les termes proches présents dans le vocabulaire contrôlé pouvant correspondre sont proposés.

Pour certains types d'articles, comme les articles Flash, il n'y a pas de titre. Ce sont le ou les secteurs sélectionnés qui font office de titre.

Les champs, coordonnées de la ou des entreprises, ainsi que sources desquelles émanent les informations font aussi l'objet d'une complétion. Lors de la frappe, les sources ainsi que les entreprises déjà connues par le système sont proposées. Après rédaction, l'article est envoyé au système pour analyse.

Analyse automatique. Cette étape succède à la rédaction d'un article. Elle a pour objectif de permettre un gain de temps lors de l'indexation. Une partie du travail est réalisée automatiquement par la machine. Ainsi, suite à la rédaction d'un article, l'interface web de rédaction et d'indexation des articles, FirstEdit, interroge l'outil FirstAnalysis (cf. figure **??**). Les informations non structurées contenues dans les articles sont analysées par l'outil FirstAnalysis intégré à l'architecture de la solution (cf. figure **??**). Cet outil est basé sur la plate-forme GATE [**?**] et permet l'analyse automatique de documents textuels à des fins d'extraction d'informations.

L'analyse d'un texte est une tâche complexe pour la machine. En fonction du type d'analyse effectuée, et du niveau de compréhension recherché, elle peut même parfois l'être pour un humain. Deux types d'informations contenues dans les articles peuvent être distingués : les informations explicites (e.g. lieux, personnes, organisations, etc.) et les informations implicites (i.e. thèmes et secteurs économiques). Dans le système, le thème d'un article est une facette de description qui correspond aux événements économiques principaux traités par l'article. Le vocabulaire définissant les thèmes ainsi que les secteurs sont issus de la connaissance métier des experts de l'entreprise et sont contenus dans la base de connaissances du système.

Afin de conserver de bonnes performances lors de l'indexation automatique, seule l'indexation correspondant aux facettes exprimées de façons explicites est automatisée. La détection d'informations implicites est plus complexe et donne de moins bons résultats (cf. chapitre **??**). De mauvaises performances engendrerait plus de corrections de la part des documentalistes ce qui aurait pour conséquence une perte de temps plus importante que le gain. Bien que des travaux visant à une automatisation plus poussée de l'indexation des items soient présentés dans le chapitre **??**, la solution n'a pour l'heure pas été mise en production. L'outil d'analyse, FirstAnalyse fonctionne sur la base d'une succession de tâches effectuées sur un document textuel. Le résultat de chaque tâche alimente la tâche suivante. Cette suite de tâches appelée *chaîne de traitements* (i.e. pipeline) est illustrée par la figure **??**. Ainsi, les lieux traités par les articles sont automatiquement

extraits à l'aide de la chaîne de traitement suivante :

- Prétraitement :
  - Sentence splitters : ce traitement permet le découpage du texte en phrases.
  - Tokenizer : ce traitement le découpage des phrases en mots.
- Analyse morphosyntaxique (i.e. Part Of Speech tagger) : ce traitement fournit une analyse morphosyntaxique des phrases. Elle fournit des informations grammaticales sur les mots qui la composent (i.e. mode, temps, voix), ainsi que fonctionnelles (i.g. verbe, adjectif, articles, etc.).
- Dictionnaire (i.e. Gazetteer) : ce traitement permet une recherche de termes dans le texte. Il se base sur un dictionnaire de termes déjà connus. Nous utilisons notre base de connaissances lexicales afin d'alimenter le dictionnaire en termes.
- Détection de motifs (i.e. Règles JAPE [**?**]) : ce traitement permet à l'aide d'un moteur de reconnaissance et d'appariement de motifs (i.e. patrons) de générer des automates afin de détecter des structures syntaxiques prédéfinies. L'utilisation de notre base de connaissances permet au système de détecter des termes ayant une sémantique commune. Ainsi les patrons utilisés sont dits, lexico-sémantiques, par opposition aux patrons classiques lexico-syntaxiques.

Les patrons lexico-sémantiques utilisés sont définis manuellement. Leur objectif est de valider les lieux qui ont été sémantiquement détectés lors du traitement réalisé par le dictionnaire. Ils permettent de vérifier qu'un terme présent dans le texte et détecté par le dictionnaire comme étant un lieu, est bien présent dans un des contextes syntaxiques prédéfinis comme permettant l'expression d'un lieu. La simple présence d'un terme désignant un lieu n'est pas suffisante. En effet il n'est pas rare que des personnes aient comme nom de famille, le nom d'un lieu. Il existe des cas particuliers de villes ayant un nom correspondant à un mot couramment utilisé comme "une". Même avec la vérification basée sur des patrons faits main, des erreurs peuvent persister ; or, la qualité de la recommandation dépend en partie de la qualité de l'indexation. Les résultats, fruits de l'analyse du texte par la chaîne de traitements GATE utilisée par FirstAnalysis, sont donc présentés aux documentalistes afin d'être vérifiés, corrigés ou complétés.

Supervision et enrichissement. La figure **??** présente dans l'application web, FirstEdit, le résultat de l'analyse d'un article par FirstAnalysis. Le nom de la ville Neuville-en-Ferrain est le résultat de l'analyse automatique. Ce résultat est proposé au rédacteur, qui peut le corriger et/ou le compléter. Les autres informations, *thèmes*, *secteurs*, *taille de l'entreprise* et *activité du site* (i.e. les facettes descriptives implicites) doivent être complétées par le rédacteur.

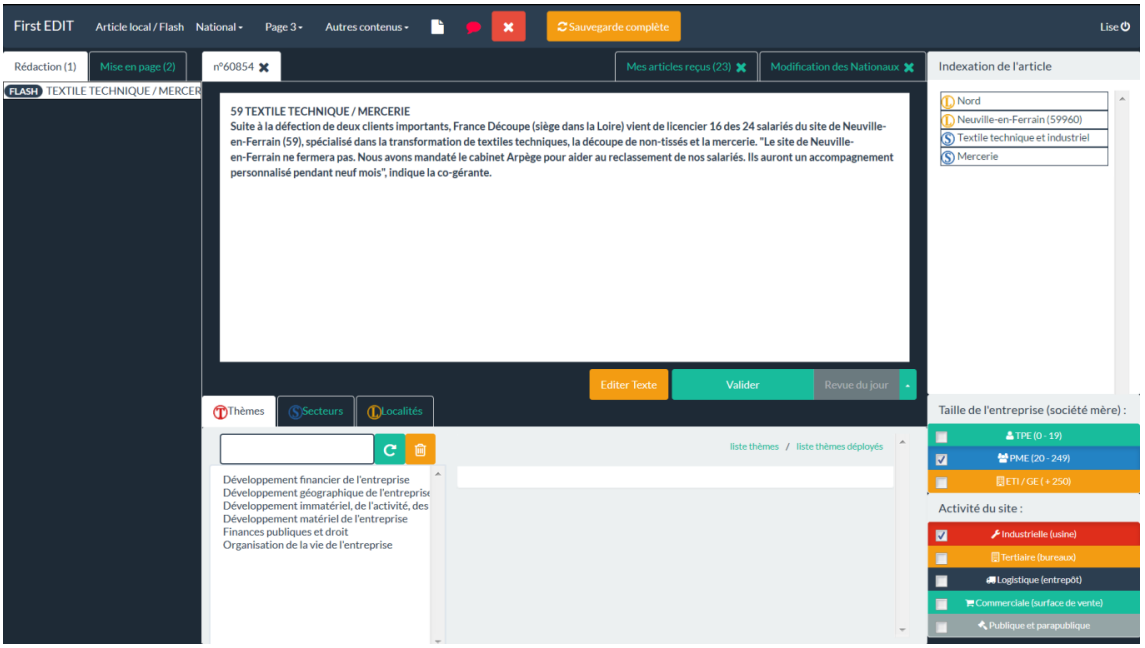


FIGURE 6.5 – Interface de supervision et d'enrichissement de l'indexation automatique

Un système de recherche de termes guidé par vocabulaire contrôlé est intégré à l'interface. Ce système est illustré par la figure **??**. Durant la frappe dans le champ, où a partir d'un mot ou d'un groupe de mots sélectionnés dans le texte de l'article, une recherche est effectuée dans l'ensemble des termes du vocabulaire contrôlé. Les termes proposés par le système de recherche sont présentés dans leur contexte. Dans le cas de *secteurs* économiques ou de *thèmes*, les vocabulaires sont organisés sous la forme de thésaurus. Chaque terme est donc présenté au sein de sa hiérarchie. La saisie d'un synonyme (i.e. alias, ou non-descripteur) renvoie directement au bon descripteur. Au survol d'un terme avec le curseur, les notes d'application associées s'affichent. Toutes ces fonctionnalités permettent d'exploiter au mieux les connaissances de la base dans l'objectif de rendre l'indexation plus facile et rapide, en simplifiant la recherche et la compréhension des termes utilisés pour qualifier les articles.

FIGURE 6.6 – Interface d'indexation des articles - Recherche guidée par vocabulaire contrôlé

Une fois les informations nécessaires à l'indexation vérifiées et complétées par le rédacteur, la validation des choix entraîne l'indexation effective de l'article dans la base de connaissances.

L'indexation des articles repose sur les vocabulaires contrôlés contenus dans la base de connaissances du système. Dans l'architecture de la solution (cf. figure **??**) la base de connaissances du système est représentée par l'élément SPARQL DataBase. Les articles y sont représentés par des instances, dans la *base d'articles indexés* de même que les concepts auxquels renvoient les termes du vocabulaire d'indexation dans la *base des vocabulaires d'indexation* (cf. figure **??**). L'indexation prend la forme d'une instanciation de relations entre ces instances (cf. chapitre **??**). Le contenu textuel des articles est conservé dans une base de données relationnelles classique représentée dans l'architecture (cf. figure **??**) par l'élément SQL DataBase.

Dans l'objectif de faciliter le travail des documentalistes ainsi que de conserver une qualité de rédaction élevée, une fois les articles rédigés et leur indexation validée, ils sont présentés dans le flux d'articles journalier du rédacteur. Il est possible à partir de ce flux de les partager avec d'autres rédacteurs ainsi que de les relire et éventuellement de les corriger avant la publication de la revue.

6.2.2/ INDEXATION DES PROFILS

L'indexation des profils, ou profilage, a pour objectif de fournir au système une représentation du besoin de chacun des utilisateurs. Une application dédiée à cette tâche est développée et intégrée à l'architecture de la solution illustrée par la figure **??**, FirstCustomers. Le processus d'indexation des profils est illustré dans la figure **??**.

Les experts de la relation client de l'entreprise sont en charge de la compréhension des besoins de chaque client. Les échanges entre experts et clients se font principalement

par téléphone. Au cours de ces conversations téléphoniques, les experts déterminent les besoins des clients. Cela contribue à créer un premier profil pour chaque client, et évite le problème d'un démarrage à froid, commun aux systèmes de recommandation basés sur le contenu.

Le processus d'indexation des profils est le même que celui des articles. Les profils sont représentés par des instances dans la *base de profils*. Les termes du vocabulaire d'indexation sont gérés sous la forme de concepts dans *base de vocabulaires d'indexation* (cf. figure **??**). L'indexation prend la forme d'une instanciation de relations entre ces instances (cf. chapitre **??**). Les bases de profils et de vocabulaires d'indexation sont des sous-ensembles de la base de connaissances (i.e. SPARQL DataBase dans la figure **??**).

L'interface web de l'outil FirstCustomers, permet aux experts de définir ou de modifier le profil des utilisateurs. Comme dans l'interface des rédacteurs, des champs de recherche guidée par la base de connaissances sont disponibles afin de trouver l'entrée correspondante dans le vocabulaire contrôlé (cf. figure **??**).

Un champ de prise de notes textuelles permet à l'expert de conserver certaines informations qui pourront lui être utiles lors de prochaines conversations, ou afin de terminer le profil une fois la conversation achevée.

Dans l'objectif de faciliter le travail des experts, différents indicateurs sont disponibles sur un utilisateur donné dans l'interface FirstCustomer. Ces indicateurs leur permettent de connaître les habitudes d'utilisation du client et éventuellement de détecter une sous-utilisation ou un profil ne correspondant pas ou plus au besoin du client. Ces indicateurs se basent sur le comportement du client, des outils d'analyse de ce comportement sont intégrés à l'interface de lecture de revue, traitée dans la section suivante.

6.3/ RECOMMANDATION

Le processus de recommandation illustré dans la figure **??** permet aux clients de consulter la revue de façon plus simple, plus efficace, plus rapide et en adéquation avec le temps qu'ils ont à y consacrer. Ce processus utilise les descriptions des articles ainsi que des profils contenus dans la base de connaissances et les compare à l'aide de l'algorithme présenté dans le chapitre **??**. Les résultats sont utilisés afin de composer la revue de l'utilisateur, en triant les articles en fonction de leur pertinence par rapport à son besoin. L'entreprise a besoin de comprendre les attentes des utilisateurs ainsi que leur façon d'utiliser la revue afin d'ajuster au mieux le produit. Pour ces raisons une interface web de consultation des revues est développée et intégrée à l'architecture de la solution, First Pro'fil (cf. figure **??**). Chaque jour une revue personnalisée, propre aux besoins du

lecteur, est mise automatiquement à sa disposition via cette interface web illustrée par la figure **??**.

FIGURE 6.7 – Interface de consultation de la revue

La page d'accueil contient les trois types d'articles principaux : locaux, flash et nationaux. Les articles locaux sont particulièrement mis en avant, ils correspondent aux informations à plus forte valeur. Chacun des trois flux d'articles présents en page d'accueil est organisé par ordre de pertinence. Le premier algorithme (cf. chapitre **??**) est utilisé afin d'évaluer la pertinence des articles en fonction du profil de l'utilisateur. Le niveau de pertinence est présenté aux utilisateurs, à l'aide d'une jauge circulaire illustrant le pourcentage de correspondance au profil de chacun des articles. Des pages sont spécifiquement dédiées aux articles nationaux ainsi qu'aux appels d'offres et annonces légales.

Le comportement de l'utilisateur lors de son utilisation de l'outil est enregistré. Chaque utilisateur donne de façon implicite des indications sur ses centres d'intérêt et besoins au moment de la consultation de la revue.

Si l'utilisateur souhaite affiner son flux, des filtres sont à sa disposition comme l'illustre la figure **??**. Les filtres utilisés par les lecteurs sont enregistrés.

Les articles présentés sur la page d'accueil ne sont pas directement visibles dans leur intégralité. L'utilisateur peut ainsi choisir de lire ou non l'article entièrement en se basant sur le titre, les métadonnées, et les premières lignes du corps de l'article. Dans ce contexte, les métadonnées sont les termes sélectionnés lors de l'indexation de l'article pour qualifier son contenu. L'utilisateur doit cliquer sur l'article s'il souhaite lire l'information dans son intégralité. Ce comportement est enregistré.

Différentes actions peuvent être utilisées sur chacun des articles. L'utilisateur peut ainsi choisir d'imprimer l'article, de l'envoyer par mail à un de ses contacts ou encore l'ajouter à sa sélection. Ce comportement est enregistré. En effet l'outil intègre un espace de travail, permettant au lecteur de conserver les articles les plus intéressants et de facilement contacter les entreprises concernées par l'information ainsi que de conserver des notes sur l'avancée du dossier.

Il est possible, grâce à l'interface, de consulter des articles passés. L'utilisateur peut déterminer une période. Tous les articles produits durant cette période sont ainsi organisés par ordre de pertinence et présentés à l'utilisateur. De même afin de limiter la perte d'information pour des utilisateurs ne pouvant se connecter tous les jours, la revue est générée non pas à partir de l'ensemble des articles de la journée, mais à partir de l'ensemble des articles publiés depuis la dernière consultation de l'utilisateur.

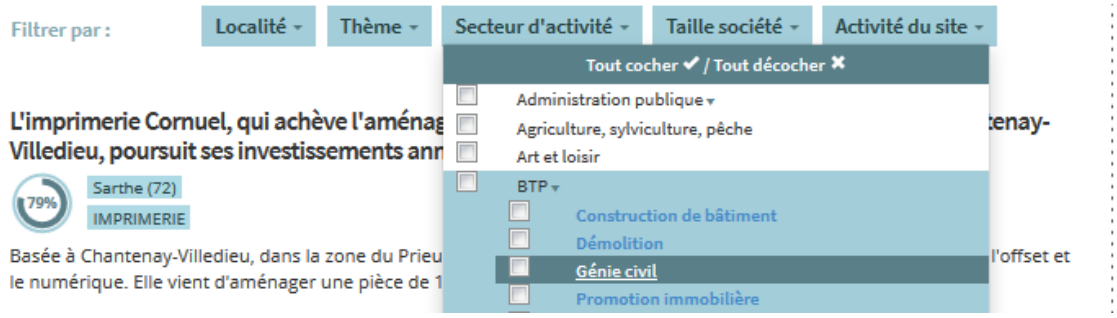


FIGURE 6.8 – Filtres de l'interface de lecture

6.4/ SYNTHÈSE

En introduction de ce chapitre, les limites de la revue *FirstECO* classique, ainsi que différentes contraintes auxquelles la nouvelle architecture doit être capable de répondre, ont été listées. Nous reprenons ci-dessous un à un chacun de ces points, en synthétisant les apports de l'architecture présentée ci-dessus.

Ainsi l'architecture proposée permet de répondre à l'ensemble des limites soulignées en introduction :

- Limite **structurelle** : la structure fixe a laissé place à un flux d'articles organisé en fonction de la correspondance de chacun avec les besoins du lecteur. Chaque lecteur a donc une revue personnalisée qui lui est propre.
- Limite **sémantique** : les articles, ainsi que les besoins des utilisateurs sont qualifiés de façon à faciliter leur exploitation tant par la machine que l'humain.
- Limite **pragmatique** : l'interface de lecture permet d'obtenir des retours vis-à-vis des utilisateurs. A partir de cette interface toute une série de comportements est enregistrée. Cela permet de savoir quel usage est fait de la revue par l'abonné. Il est toutefois à noter que bien que l'architecture le permette, le lecteur n'est pour le moment pas profilé en fonction de sa lecture. Seul le profil créé manuellement par les experts de l'entreprise influence pour le moment la recommandation qui est faite à un utilisateur. Sur la base des informations sur le comportement de l'utilisateur, des indicateurs permettent aux experts de savoir s'il est nécessaire, ou non, de recontacter l'utilisateur afin d'améliorer son profil. Ceci a pour objectif d'améliorer la relation client.

L'architecture proposée permet de plus de répondre à l'ensemble des contraintes présentées en introduction :

- **Qualité** : l'interface de rédaction est pensée afin de conserver ce qui faisait la qua-

lité des articles de la revue classique. Les métadonnées (i.e. termes qualifiant les articles) sont aujourd'hui, moins ambiguës, car basées sur un vocabulaire contrôlé commun et mieux mis en valeur dans l'interface de consultation de la revue. De plus l'outil d'édition intègre la possibilité d'échanger des articles entre documentalistes ainsi que de les corriger.

L'étude automatique du comportement de lecture de l'utilisateur permet de faire remonter rapidement aux experts différents problèmes, par exemple un profil mal-défini. La qualité de la relation client est meilleure, la définition des profils ainsi que l'analyse du comportement des utilisateurs sur la plateforme permettent aux experts de l'entreprise de mieux connaître le besoin de leurs clients.

- **Rapidité** : le système propose une interface de consultation des revues pour les lecteurs. Dans cette interface les articles sont triés par ordre de pertinence par rapport aux besoins des lecteurs. De plus des informations visuelles, comme les termes qualifiant un article ou le degré de correspondance de l'article par rapport au besoin du lecteur sont présentés, ce qui doit permettre à l'utilisateur une lecture plus rapide de la revue, un accès plus rapide à l'information correspondant à son besoin.

  Bien que le système permette un gain de temps pour l'utilisateur ; l'information est organisée en fonction de son besoin, il n'a donc pas à filtrer lui-même les informations ; la mise en place de cette architecture a pour conséquence une augmentation du temps de rédaction ainsi que du temps d'interaction avec les clients dû au besoin d'indexation. Le temps nécessaire à l'indexation des items varie pour les articles entre 30 minutes et une heure par jour alors que le temps nécessaire pour les profils est de 5 à 15 minutes par client. Cette problématique a été prise en compte lors de la définition de la solution, aussi les interfaces ont été pensées pour faciliter le travail des experts, elles intègrent différents outils comme le champ de recherche avec visualisation du contexte d'un terme, disponible sur les facettes dont le vocabulaire est organisé hiérarchiquement et riche (cf. figure **??**). La définition des interfaces a été réalisée en collaboration avec les experts dans l'objectif de fournir la meilleure ergonomie possible. De plus une partie de la tâche d'indexation a été automatisée afin de permettre aux documentalistes rédigeant les articles de consacrer un maximum de leur temps à la veille ainsi qu'à la synthèse de l'information qui est le cœur de leur expertise.

- **Simplicité** : le système intègre une base de connaissances contenant un vocabulaire défini de façon ad'hoc au besoin de l'application. Ainsi les interactions entre les différents acteurs sont simplifiées par la présence de ce vocabulaire contrôlé, limitant l'ambiguïté.

  Ce vocabulaire est exploité afin de faciliter le travail des experts, lors de l'indexation.

En effet un système de recherche guidé par le vocabulaire est développé afin de faciliter l'indexation, en simplifiant la recherche et la compréhension des termes utilisés pour qualifier les items.

- **Adaptabilité** : ce système peut être adapté facilement à d'autres domaines. Il suffit pour cela d'adapter le contenu de la base de connaissances et en particulier de la base de vocabulaire d'indexation au nouveau domaine. Une approche analogue à celle développée dans la section **??** permet de définir de nouvelles facettes ainsi que les vocabulaires correspondants.

- **Évolutivité** : l'architecture mise en place intègre un outil, eSkosaurus (cf. figure **??**) qui permet de prendre compte le cycle de vie du vocabulaire au sein de la solution. C'est un outil de maintenance qui permet l'ajout, la suppression et/ou la réorganisation des termes du vocabulaire d'indexation.

  La solution implémentée est pensée afin d'être une base à partir de laquelle, des évolutions sont possibles. Il reste dans l'architecture, telle qu'elle est exploitée aujourd'hui, des tâches manuelles. L'architecture est donc pensée afin de faciliter la collecte d'informations permettant par la suite une automatisation plus poussée de certaines tâches. Notamment (i) une indexation automatique des articles et des profils sur toutes les facettes qui ne l'est que pour certaines aujourd'hui et (ii) une automatisation de la prise en compte de l'évolution des besoins des utilisateurs.

  Le chapitre **??** présente une proposition d'automatisation plus poussée de l'indexation des articles. Elle n'est pour le moment pas implémentée dans la solution utilisée en production. Avec cette méthode il serait aussi possible d'automatiser la création du premier profil des utilisateurs. Il est, pour l'heure, créé manuellement afin de répondre au problème de démarrage à froid. Pour cela, il nous est possible d'utiliser une approche semblable à celle utilisée sur les articles, par l'analyse du texte, mais dans le cas des profils cela se ferait sur la base des notes prises par l'expert lors de ses entretiens avec le client.

  L'interface de lecture intègre des fonctionnalités d'enregistrement du comportement de lecture. Il est ainsi possible de faire évoluer le profil d'un lecteur au cours du temps afin d'affiner son besoin ou de prendre en compte un changement des besoins, sans l'intervention manuelle des experts. Il est possible avec notre système d'utiliser des profils complexes, c'est-à-dire des profils composés de sous-profils (cf. chapitre **??**). Ainsi un utilisateur pourrait se voir attribuer un ou plusieurs autres profils, déduits de son comportement de lecture. Cela permet notamment de gérer différents aspects du besoin de l'utilisateur. Un profil de long terme, basé sur son comportement depuis qu'il utilise l'application, complété par un profil de court terme, ne se basant que sur son comportement des derniers jours permet de gérer un changement ponctuel des besoins de l'utilisateur.

Une enquête récente (cf. annexe **??**) a montré des résultats probants quant aux apports de cette solution pour les clients.

// je n'ai pas encore les informations pour développer

Selon les premières retombées enregistrées, le produit *FristECO Pro'fil*, fruit de l'architecture présentée ci-dessus semble en parfaite adéquation avec les attentes des clients de l'entreprise. *FristECO Pro'fil* représente aujourd'hui environ 80% des nouvelles ventes enregistrées par l'éditeur.

7

CONCLUSION

Les travaux présentés dans ce document étudient la conception d'une architecture ayant pour objectif la recommandation d'informations. Ce travail s'appuie sur la mise en place d'un système de recommandation. Ces systèmes sont des outils logiciels qui assistent les humains afin de faciliter et d'optimiser l'exploitation de vastes quantités d'informations. En effet, l'accroissement exponentiel de la quantité d'information accessible aux humains tend à imposer et à généraliser l'utilisation de ce type de systèmes. Que ce soit sur des plateformes qui proposent l'écoute de musiques, la vente de produits, le visionnage de vidéos ou la lecture d'informations, l'espace informationnel proposé est souvent bien trop vaste pour qu'un humain seul puisse accéder efficacement aux contenus qui correspondent à ses goûts ou besoins. C'est pourquoi la conception et l'amélioration des systèmes de recommandation sont donc devenues un domaine de recherche important ces vingt dernières années.

Les travaux présentés dans cette thèse ont été réalisés sur la base d'un cas d'application précis. En effet, ils sont le fruit d'une collaboration entre la société Actualis Sarl du groupe FirstECO [1] et l'équipe Checksem [2] du laboratoire LE2I [3] (i.e. Laboratoire Électronique, Informatique et Image) de l'université de Bourgogne [4]. L'entreprise Actualis Sarl partenaire et financeur de ces travaux est spécialisée dans la production et la distribution de revues de presse. La revue *First Eco* est un outil de veille externalisé qui fournit quotidiennement des synthèses de l'information économique régionale (e.g. investissements, recrutements, redressements judiciaires, ainsi qu'une sélection d'informations légales et d'appels d'offres). Nos travaux ont pour objectif de répondre à la limite principale de l'offre de service de l'entreprise. En effet, la vaste quantité d'informations produite de façon quotidienne est proposée aux clients de l'entreprise. Mais toutes les informations produites ne correspondent pas aux besoins de tous les clients. Nous avons donc conçu un système de recommandation afin de faciliter l'accès à l'information des clients de l'entreprise. L'objectif a été de fournir la bonne information à la bonne personne, afin de limiter la surcharge d'informations et donc de limiter au maximum le filtrage par les clients des informations reçues ; ceux-ci ne pouvant consacrer que peu de temps à cette tâche longue et fastidieuse.

7.1/ CONTRIBUTIONS

Afin de répondre aux besoins de l'entreprise, une architecture complète est proposée et développée. Bien que cette architecture repose sur de nombreux outils, elle n'est composée que de trois principaux éléments :

1. http ://www.firsteco.fr/
2. http ://checksem.u-bourgogne.fr/www/
3. http ://le2i.cnrs.fr/
4. http ://www.u-bourgogne.fr/

- Une base de connaissances
- Des processus d'indexation
- Un processus d'évaluation de la pertinence

Sur chacun de ces éléments, nous avons apporté une contribution nous permettant d'atteindre nos objectifs tout en comblant des manques mis en lumière dans les outils et approches existants.

L'architecture développée constitue l'apport principal de ce projet. Elle est adaptable à différents contextes métier et domaines afin de fournir un système de recommandation sémantique, efficace et simple d'utilisation. Son adaptation au domaine de l'économie dans le contexte métier de l'entreprise partenaire, First ECO, a permis la commercialisation d'un nouveau produit hautement compétitif, *FirstECO Pro'fil*. Les paragraphes ci-dessous détaillent les apports sur les trois principaux éléments de l'architecture.

Le premier élément de l'architecture est la base de connaissances. L'ensemble du fonctionnement du système repose sur cette base de connaissances. Elle contient le vocabulaire d'indexation et forme une modélisation du domaine traité correspondant au contexte métier d'utilisation du système.
Notre première contribution consiste à proposer un modèle pour la base de connaissances du système répondant à différentes contraintes.
Il doit permettre la modélisation de domaines riches et au plus près de la vision métier des experts. Il doit proposer un vocabulaire modélisant le domaine qui soit aisément accessible et utilisable par la machine comme par l'humain. Il doit pouvoir être maintenu. Il doit permettre la maintenance ainsi que l'adaptation rapide et simple du vocabulaire.
C'est pourquoi notre modèle propose (i) la description des items à l'aide de facettes de descriptions. Ce qui a pour avantage de permettre la gestion de domaines complexes tout en conservant des descriptions aisément accessibles pour un humain. Chaque facette est composée d'une ressource terminologique de type langage documentaire (i.e. liste, taxonomie ou thésaurus). Ces ressources ont en premier lieu été pensées pour une utilisation humaine, et sont donc facilement accessibles pour des humains. Le modèle permet la réutilisation de ressources terminologiques existantes. (ii) Ce modèle repose sur une ontologie, il est donc formel et manipulable par la machine. Il repose sur la logique de description ce qui permet l'utilisation de contraintes logiques. (iii) Ce modèle gère chacune des ressources terminologiques selon le principe de l'abstraction conceptuelle, ce qui permet de faciliter la maintenance et l'évolution du vocabulaire qu'il contient. (iv) Lors de l'utilisation de ressources terminologiques simples, comme des listes, le modèle permet facilement d'en augmenter l'expressivité. Le modèle permet par exemple, l'ajout de contraintes ou la réorganisation hiérarchique des termes.

Le second élément de l'architecture est le processus d'indexation. Il permet de fournir

au système une description du besoin et de l'offre d'information. Cette description repose sur le vocabulaire contrôlé et structuré de la base de connaissances. Notre seconde contribution consiste à proposer une approche pour l'automatisation du processus d'indexation. Notre approche considère l'indexation comme une tâche d'indexation multi-label (ou multi-label hiérarchique dans le cas de vocabulaires d'indexation organisés de façon hiérarchique). L'indexation multi-label nécessite l'apprentissage d'un modèle prédictif. Pour cette tâche, nous utilisons les connaissances de la base de connaissances, qui contient les items déjà indexés. C'est-à-dire, les descriptions des profils et documents utilisés lors du fonctionnement du système. La base de connaissances de notre système est une ontologie. Nous considérons le modèle prédictif comme une connaissance à y intégrer. Pour cela, nous proposons de la traduire sous la forme de contraintes logiques intégrables dans l'ontologie. L'indexation est donc automatisée, elle est le résultat d'un processus d'inférence logique produit par des raisonneurs sur la base de connaissances du système. A notre connaissance aucun travail similaire n'a été réalisé. Nous démontrons la faisabilité de cette approche dont le but est de conserver le modèle de prédiction au plus près de la modélisation métier du domaine et donc au plus près de la connaissance des experts. C'est une première étape vers des systèmes plus évolutifs, à même de gérer efficacement l'évolution du vocabulaire d'indexation et ses répercussions sur le processus d'indexation.

Le troisième élément de l'architecture est le processus de comparaison. Il permet l'évaluation de la pertinence d'un document, par la comparaison de la description de l'offre d'information qu'il contient et celle du besoin d'information d'un utilisateur. Notre troisième contribution consiste à proposer un algorithme de comparaison qui exploite pleinement la richesse des descriptions d'items permise par le modèle sur lequel repose la base de connaissances. En effet, cette modélisation permet de décrire différents aspects (i.e. facettes) d'un item. Les vocabulaires utilisés pour la définition des facettes peuvent être organisés sous la forme de simples listes, mais aussi structurés sous la forme de taxonomies ou de thésaurus. Les items peuvent ainsi être décrits sur chacune des facettes à l'aide de termes plus ou moins précis. Les algorithmes de comparaisons classiques déduisent directement la pertinence de la précision et ne peuvent donc pas prendre en compte la différence de précision qu'il peut y avoir entre l'expression du besoin de celle de l'offre d'information. Notre algorithme comble un manque de l'état de l'art en ce qui concerne la prise en compte du degré de précision de la description des items.

Bien que l'architecture ait été mise en place dans le contexte de la recommandation d'articles de synthèse d'informations économiques régionales, elle a été pensée pour être évolutive et adaptable à d'autres domaines. Les perspectives offertes par le système sont détaillées dans la section suivante.

7.2/ PERSPECTIVES

Les travaux réalisés durant cette thèse ouvrent un nombre important de perspectives. Premièrement nous présentons les perspectives industrielles. Elles consistent principalement en la réutilisation de l'outil, adapté à des domaines différents. Ensuite, nous présentons les perspectives techniques. Elles concernent les évolutions possibles du système. Beaucoup de ces évolutions sont facilement réalisables, car pensées dès la conception. Enfin, nous présentons les perspectives de recherche.

L'architecture du système peut être aisément adaptée afin de permettre de fournir des recommandations dans un autre contexte d'application que la recommandation d'articles d'informations économiques régionales. La contrainte consistant à pouvoir utiliser l'architecture avec d'autres types d'items ou d'autres domaines a été prise en compte lors de sa conception. Adapter l'architecture consiste principalement à adapter la base de connaissances. Le modèle sur lequel elle repose est spécifiquement étudié pour cela. De plus, un outil dédié, eSkosaurus, a été développé à cet effet et est intégré à l'architecture. Des perspectives d'adaptation de l'outil à la recommandation d'informations médicales à des fins de veille sont en cours d'étude.

Diverses améliorations ont été pensées lors de la conception du système. Toutes n'ont pas été implémentées, pour des raisons de temps, de coûts, ou simplement parce que celles-ci n'étaient pas nécessaires afin de fournir un produit fonctionnel à l'entreprise.

Ainsi, l'algorithme de recommandation utilisé permet la définition de profils complexes. Ces profils ne sont pour le moment pas utilisés dans le produit final tel qu'il a été implémenté. Un utilisateur peut être vu comme une composition de plusieurs profils ce qui permet de répondre à des besoins encore plus particuliers et précis. En effet, un lecteur peut être intéressé par deux cas bien spécifiques. Il est alors possible avec notre architecture de lui créer deux sous-profils. Par exemple, le sous-profil 1, concernant les créations de sites dans l'industrie manufacturière et le sous-profil 2 concernant les OPA dans le secteur de l'énergie. L'utilisation d'une combinaison de deux profils, plutôt que d'un seul profil, va avoir pour conséquence une dévalorisation des articles ne traitant pas d'un des deux cas, mais d'une combinaison des deux cas. Dans notre exemple, les articles traitant d'OPA dans le secteur de l'industrie manufacturière ou de la création de sites dans celui de l'énergie. Sans la création de sous-profils, ils auraient la même importance que les informations réellement souhaitées par l'utilisateur. Avec notre outil, il nous est de plus possible de pondérer les différents sous-profils. Ainsi le lecteur qui s'intéresse plus au sous-profil 1 qu'au sous-profil 2, verra les informations correspondant y apparaître prioritairement dans son flux.

L'interface de lecture des revues, FirstECO pro'fil, intègre des fonctionnalités d'enregistrement du comportement de lecture. Ces fonctionnalités ne sont pour le moment utili-

sées que pour faire remonter quelques informations sur le comportement de lecture (e.g. nombre de connexions, temps passé sur l'outil, etc.) aux experts ayant en charge la relation client. Ces fonctionnalités ont été mises en place afin de permettre une utilisation future, plus poussée de ces connaissances sur le comportement de lecture des clients. Il est ainsi possible à l'aide de ces informations de faire évoluer le profil d'un lecteur au cours du temps afin d'affiner son besoin ou de prendre en compte l'évolution de son besoin. Il est possible avec notre système de créer des profils complexes, composés de sous-profils. Cela permet par exemple de distinguer les profils déduits du comportement de lecture, du profil de base du lecteur. Un lecteur peut ainsi se voir attribuer un profil de long terme, basé sur son comportement depuis qu'il utilise l'application et un profil de court terme, ne se basant que sur son comportement des derniers jours. Cela permet de détecter un besoin d'information ponctuel. Ces sous-profils peuvent être pondérés les uns par rapport aux autres en fonction de la fréquence de certains comportements. Si à chaque lecture de sa revue, le lecteur ressent l'obligation de filtrer le flux sur une combinaison de critères semblables, alors, cela doit être pris en compte et influencer son flux de façon importante. D'autre combinaisons, utilisées avec une fréquence moindre, peuvent aussi influencer son flux, mais de façon moins importante.

Notre algorithme, *PEnSIVE* offre de bonnes performances et comble un manque de l'état de l'art en ce qui concerne la gestion du degré de précision lors de la comparaison des descriptions du besoin et de l'offre. Une nouvelle version de cet algorithme est en cours d'évaluation. Les modifications apportées permettent un gain en terme de qualité de la recommandation ainsi qu'en terme de temps de calcul. Ces travaux sont en cours de publication. Ils se basent sur la comparaison efficace de vecteurs binaires. Plusieurs vecteurs sont créés et pondérés pour chaque profil. Ils représentent les différents niveaux de précision du besoin. La pertinence est évaluée en fonction du taux de correspondance des vecteurs correspondant aux documents à recommander avec chacun des vecteurs du profil.

Notre approche pour l'automatisation de l'indexation des items sur la base de vocabulaires contrôlés et structurés est une première étape vers des systèmes plus évolutifs, à même de gérer efficacement l'évolution du vocabulaire d'indexation et ses répercussions sur le processus d'indexation. Cette approche a été implémentée et évaluée, ce qui a montré sa faisabilité, mais a aussi mis en avant qu'en l'état, elle n'est pas directement utilisable dans un contexte industriel. En effet, les résultats montrent que les temps de calcul sont trop importants. Afin d'améliorer les temps de calcul, une première tentative visant à réaliser les raisonnements sur des sous-ontologies et non sur l'ontologie globale a été réalisée. Cette tentative a montré des gains très importants en terme de temps de calcul. Dans cette approche, pour chaque item à indexer, seules les informations nécessaires sont extraites de l'ontologie afin de créer une sous-ontologie sur laquelle le raisonnement inférant l'indexation est réalisé. D'autres approches sont à étudier, par exemple

l'utilisation de chaînage arrière (i.e. backward chaining).

La phase de hiérarchisation réalisée par les raisonneurs permet de réorganiser les relations entre les classes sur la base du modèle prédictif appris. L'organisation, notamment hiérarchique des classes entre elles, qu'elle soit préexistante (i.e. modélisation du domaine créé par les experts) ou déduite de la phase de hiérarchisation est directement prise en compte lors de la résolution et influence donc directement la classification. Il s'agit donc d'une classification multi-label hiérarchique. A l'inverse, dans notre implémentation la hiérarchie des classes n'est pas prise en compte lors de la création du modèle prédictif. L'utilisation de cette connaissance lors de la création du modèle prédictif doit permettre d'en améliorer la qualité. Cette perspective de recherche semble pertinente fin d'améliorer les performances des systèmes d'indexation multi-label hiérarchiques.

Afin de poursuivre les travaux présentés dans ce document, deux thèses sont actuellement en cours. Ces travaux portent principalement sur l'indexation de documents à l'échelle du web, en adaptant la méthode d'indexation automatique basée sur une classification multi-label hiérarchique.

Rafael PEIXOTO poursuit ainsi les objectifs visant à gérer efficacement l'évolution du vocabulaire d'indexation et ses répercussions sur le processus d'indexation par l'amélioration des performances de la méthode d'indexation automatique proposée dans cette thèse. En effet, l'amélioration est nécessaire afin de permettre son utilisation avec de vastes quantités de données. Pour cela, une partie des calculs nécessaires à la création du modèle est réalisée à l'aide d'une architecture MapReduce.

Thomas HASSAN travaille sur la recherche et le recoupement d'informations à l'échelle du web. Alors que le système de recommandation tel qu'il a été mis en place dans l'entreprise ne recommande que les articles produits par l'entreprise. Les travaux de M. Hassan visent à permettre la recommandation d'informations venant directement du web. Afin de prendre en compte la possible non-véracité des informations qui y circulent, il propose l'utilisation de notre méthode de classification multi-label à des fins de recoupement d'informations. La véracité d'une information est ainsi vérifiée par recoupement.

L'intérêt du travail réalisé durant cette thèse est double. Premièrement, ce travail a permis à l'entreprise partenaire, first ECO, de commercialiser un nouveau produit hautement compétitif. *FristECO Pro'fil* représente aujourd'hui environ 80% des nouvelles ventes enregistrées par l'éditeur. Deuxièmement, ce travail a répondu à certains manques de l'état de l'art ainsi qu'à certaines limites des outils et approches existantes. De plus, il permet l'émergence de nouvelles pistes de recherche, conduisant à la réalisation des deux thèses actuellement en cours.

TABLE DES FIGURES

LISTE DES TABLES

I

ANNEXES

A

MODÉLISATION BASÉE SUR LE MODÈLE KARLSRUHE

Cette annexe présente le méta-modèle inspiré de celui de Karlsruhe. Ce méta-modèle est utilisé dans le document afin de formaliser le modèle intégrateur développé lors de la conception de la base de connaissances du système de recommandation.

A.1/ FORMALISATION D'UNE BASE DE CONNAISSANCES

Nous présentons ici un méta-modèle qui s'inspire de celui de Karlsruhe utilisé notamment par Ehrig, [**?**] et étendu aux base de connaissances utilisant une expressivité logique $\mathcal{SHOIN}(\mathcal{D})$ par Pittet [**?**]. l'expressivité logique $\mathcal{SHOIN}(\mathcal{D})$ correspond au langage OWL-DL. Le modèle de Karlsruhe, tel que présenté dans les travaux de Perrine Pittet peut être aisément adapté aux différents niveaux d'expressivité logique des langages OWL-Lite, OWL-DL ou OWL2.

Nous utilisons dans ce mémoire ce modèle afin de formaliser la base de connaissances du système. En effet ce modèle permet de manipuler une représentation ensembliste d'une base de connaissances sur la base de structures mathématiques. Cette représentation facilite l'explication du fonctionnement de certains processus au cœur de notre système.

A.1.1/ DÉFINITION DU MODÈLE

Nous souhaitons donc modéliser une base de connaissances reposant sur une ontologie formelle. Une ontologie est un ensemble d'axiomes [**?**] qui peut être défini comme une structure mathématique. La structure est définie ici comme un n-uplet [**?**] $\mathcal{S} = (\Omega, \Sigma, \Phi, E)$:

- Soit Ω un ensemble appelé l'ensemble sous-jacent de $\mathcal{S}$.
- Soit Σ une collection d'axiomes de signature $\{\sigma_i : i \in I_1\}$ ou $\sigma_i \subseteq \Omega^{m_i}$ pour $m_i \geq 1$.
- Soit Φ une collection d'axiomes de fonction $\{\varphi_i : i \in I_0\}$ ou $\varphi_i : \Omega^{n_i} \to \Omega$ pour $n_i \geq 1$.
- Soit E une collection d'éléments distincts $\{\varepsilon_i : i \in I_2\} \subseteq \Omega$.

Les ensembles I_0 I_1 et I_2 peuvent être vide. n_i et m_i sont les arités respectives de φ_i et σ_i.

Une ontologie est donc une structure $\mathcal{S}_0 = (\Omega_0, \Sigma_0, \Phi_0, E_0)$ telle que :

- L'ensemble Ω_0 contient :
  - sC, un ensemble des concepts.
  - sT, un ensemble des types de données.
  - sR, un ensemble des rôles.
  - sA, un ensemble des attributs.
  - sI, un ensemble des instances.

- sV, un ensemble des valeurs de données.
- sK_R, un ensemble des caractéristiques de rôle. Les valeurs contenues dans cet ensemble influenceront le degré d'expressivité logique de l'ontologie, par exemple pour une ontologie OWL-DL, d'expressivité $\mathcal{SHOIN}(\mathcal{D})$ sK_R = { Symmetric, Functional, Inverse Functional, Transitive }.
- sK_A, un ensemble des caractéristiques d'attribut. Les valeurs contenues dans cet ensemble influenceront le degré d'expressivité logique de l'ontologie, par exemple pour une ontologie OWL-DL, d'expressivité $\mathcal{SHOIN}(\mathcal{D})$ sK_A = { Functional }.
- $\leq_C$, un ordre partiel, nommé subsomption de concepts.
- $\leq_T$, un ordre partiel, nommé subsomption de types de données.
- $\leq_R$, un ordre partiel, nommé subsomption de rôles.
- $\leq_A$, un ordre partiel, nommé subsomption de d'attributs.

Les ensembles $sC, sT, sR, sA, sI, sV, sK_R, sK_A$ sont disjoints.

Ainsi $\Omega_0 = \{(sC, \leq_C), (sT, \leq_T), (sR, \leq_R), (sA, \leq_A), sI, sV, sK_R, sK_A\}$,

- L'ensemble des signatures Σ_0 contient :
  - $\sigma_R : sR \rightarrow sC^2$, un ensemble d'axiomes de signatures de rôle.
  - $\sigma_A : sA \rightarrow sC \times sT$, un ensemble d'axiomes de signatures d'attribut.

  Ainsi $\Sigma_0 = \{\sigma_R, \sigma_A\}$,
- L'ensemble des fonctions Φ_0 contient :
  - $iC : sC \rightarrow 2^{sI}$, un ensemble d'axiomes contenant les instantiations de concepts.
  - $iT : sT \rightarrow 2^{sV}$, un ensemble d'axiomes contenant les instantiations de type de données.
  - $iR : sR \rightarrow 2^{sI \times sI}$, un ensemble d'axiomes contenant les instantiations de rôles.
  - $iA : sA \rightarrow 2^{sI \times sV}$, un ensemble d'axiomes contenant les instantiations d'attributs.
  - $K_R : sR \rightarrow 2^{sKR}$, un ensemble d'axiomes contenant les caractérisations de rôles.
  - $K_A : sA \rightarrow 2^{sKA}$, un ensemble d'axiomes contenant les caractérisations d'attributs.
  - $\varepsilon_C : sC \rightarrow 2^{sC}$, un ensemble d'axiomes contenant les concepts équivalents.
  - $\varepsilon_R : sR \rightarrow 2^{sR}$, un ensemble d'axiomes contenant les rôles équivalents.

- $\varepsilon_A : sA \rightarrow 2^{sA}$, un ensemble d'axiomes contenant les attributs équivalents.
- $\varepsilon_I : sI \rightarrow 2^{sI}$, un ensemble d'axiomes contenant les instances équivalentes.
- $\delta_C : sC \rightarrow 2^{sC}$, un ensemble d'axiomes contenant des concepts disjoints.
- $\delta_I : sI \rightarrow 2^{sI}$, un ensemble d'axiomes contenant des instances définies comme différentes.
- $-_C : sC \rightarrow 2^{sC}$, un ensemble d'axiomes spécifiant des concepts comme complément d'autres concepts de l'ensemble sC.
- $-_R : sR \rightarrow 2^{sR}$, un ensemble d'axiomes spécifiant comme étant inverses de rôles de l'ensemble sR.
- $maxCard_R : sR \rightarrow \mathbb{N}$, un ensemble d'axiomes définissant les restrictions de cardinalité maximale des rôles.
- $minCard_R : sR \rightarrow \mathbb{N}$, un ensemble d'axiomes définissant les restrictions de cardinalité minimale des rôles.
- $\sqcap_C : sC \rightarrow 2^{sC}$, un ensemble d'axiomes nommé intersection de concepts.
- $\sqcup_C : sC \rightarrow 2^{sC}$, un ensemble d'axiomes nommé union de concepts.
- $\sqcup_I : sI \rightarrow 2^{sI}$, un ensemble d'axiomes nommé énumération d'instances.
- $\sqcup_V : sV \rightarrow 2^{sV}$, un ensemble d'axiomes nommé énumération de valeurs de données.
- $\rho_{\exists R} : sR \rightarrow 2^{sC}$, un ensemble d'axiomes définissant les restrictions existentielles de rôles.
- $\rho_{\forall R} : sR \rightarrow 2^{sC}$, un ensemble d'axiomes définissant les restrictions universelles de rôles.
- $\rho_R : sR \rightarrow 2^{sI}$, un ensemble d'axiomes définissant les restrictions de valeur de rôles.
- $\rho_{\exists_A} : sA \rightarrow 2^{sT}$, un ensemble d'axiomes définissant les restrictions existentielles d'attributs.
- $\rho_{\forall_A} : sA \rightarrow 2^{sT}$, un ensemble d'axiomes définissant les restrictions universelles d'attributs.
- $\rho_A : sA \rightarrow 2^{sV}$, un ensemble d'axiomes définissant les restrictions de valeur d'attributs.

Ainsi $\Phi_0 = \{iC, iT, iR, iA, K_R, K_A, \varepsilon_C, \varepsilon_R, \varepsilon_A, \varepsilon_I, \delta_C, \delta_I, -_C, -_R, maxCard_R, minCard_R, \sqcap_C, \sqcup_C, \sqcup_I, \sqcup_V, \rho_{\exists R}, \rho_{\forall R}, \rho_R, \rho_{\exists_A}, \rho_{\forall_A}, \rho_A\}$,

- L'ensemble des éléments distincts E_0 contient :
  - **TopConcept** un concept spécial, subsumant tous les concepts.

- **BottomConcept** un concept spécial, subsumé par tous les concepts.
- **TopAttribute** un attribut spécial, subsumant tous les Attributs.
- **BottomAttribute** un attribut spécial, subsumé par tous les Attributs.
- **TopRole** un rôle spécial, subsumant tous les rôles.
- **BottomRole** un rôle spécial, subsumé par tous les rôles.
- **TopDataType** un type de donnés spécial, subsumant tous les typese de données.
- **BottomDataType** un type de données spécial, subsumé par tous les types de données.

Ainsi, E_0 = {TopConcept, BottomConcept, TopAttribute, BottomAttribute, TopRole, BottomRole, TopDataType, BottomDataType}

B

PRÉSENTATION DU TRAVAIL DES DOCUMENTALISTES

Cette annexe présente le travail des documentalistes de l'entreprise. La solution mise en place dans l'entreprise repose en partie sur le savoir-faire métier de ces experts. Ils ont en charge la veille et la production d'articles de synthèse. Ils ont grandement participé à la création des vocabulaires contrôlés, durant le projet.

B.1/ MÉTHODE DE TRAVAIL DES DOCUMENTALISTES

Cette annexe présente quelques aspects du travail de documentaliste. Dans l'entreprise les documentalistes rédigent les articles qui seront par la suite proposés aux clients. Leur travail consiste à :

1. Veiller des sources d'informations.
2. Sélectionner les articles intéressants proposés par ces sources.
3. Extraire les éléments pertinents des articles sources. Cela peut inclure des tâches supplémentaires comme :
   - Recouper les informations entre différentes sources si différentes sources sont disponibles.
   - Approfondir la recherche dans certains cas où l'information est trop partielle.
4. Synthétiser cette information sous la forme d'articles proposés aux clients.

Nous détaillons ces points dans les sections suivantes.

B.1.1/ SOURCES ET SUPPORTS DE L'INFORMATION ÉCONOMIQUE

Chaque documentaliste est en charge de la veille d'un espace géographique et la définition de ses sources d'information. Une veille performante et de qualité doit prendre en compte la multiplicité des sources d'information ainsi que leur évolution.

Les supports utilisés par les documentalistes de l'entreprise sont les suivants :

- La presse papier (presse quotidienne régionale, presse quotidienne nationale, presse professionnelle, presse hebdomadaire)
- La presse web
- Le web en général, qu'il s'agisse de sites d'actualités ou de sites vitrines
- Les outils de relations presse, que sont les dossiers de presse et les communiqués de presse
- La transmission orale de l'information.

Ces différentes sources transmettent chaque jour des quantités importantes d'information qu'il faut sélectionner, filtrer, pour ne garder que les plus pertinentes.

B.1.2/ SÉLECTION DES INFORMATIONS

Le processus de sélection d'une information dépend en partie de sa source. Une source traitant uniquement d'informations économiques ne sera pas traitée de la même façon qu'une source proposant de l'information généraliste. Mais les deux types sont veillés. L'analyse de sources spécialisées, plus proche du domaine de travail de l'entreprise sera ainsi plus facile et rapide à analyser qu'une source généraliste.

Les informations proposées par les différentes sources sont parcourues à la recherche de mots clés. Une partie de ce travail est fait de façon automatisée par les logiciels de veille, qui font un premier filtrage automatique des sources d'informations numériques à l'aide de règles définies par les documentalistes. Les règles sont des combinaisons plus ou moins élaborées de mots clés, par exemple : *entreprise* ou *société* et *Nord* ou *Pas-de-Calais* sauf *chasse*. En ce qui concerne les sources imprimées, ou orales, par contre, le travail est intégralement fait par le documentaliste.

Lorsque le documentaliste-rédacteur recherche des informations sur un support papier, il dispose, pour permettre son analyse, d'un titre, d'un chapô [1] et du texte intégral. L'analyse est donc rapide puisque le nombre de mots-clés à sa disposition est important. La sélection s'opère alors en deux temps :

- lecture diagonale de l'ensemble du texte
- sélection ou rejet

Si le documentaliste-rédacteur est face à un support web, il ne dispose en revanche, dans la majorité des cas que d'un titre et d'un chapô. Aussi, la sélection est plus complexe. Elle s'opère alors en quatre temps :

- analyse du titre
- clic sur le titre
- analyse du texte
- sélection ou rejet du texte

L'analyse des contenus s'effectue en outre selon plusieurs niveaux de lecture. Les documentalistes cherchent en premier lieu des termes typiquement liés à l'univers économique, comme *entreprise*, *société*, *PME*, *investissement*, *zone d'activité*, ou encore des mots faisant référence à des événements (i.e. thèmes) ou des noms des principaux secteurs d'activités économiques.

1. Texte court coiffant un article, permettant d'amener le lecteur à entrer dans l'article.

Un vocabulaire plus lointain est aussi pris en compte. L'expérience montre qu'ils pourraient générer des réponses pertinentes. Ainsi, des termes comme *pôle*, *conseil communautaire*, *conseil municipal*, *projet* ou encore *activité*, génèrent une attention particulière de la part des documentalistes. Combinés à un mot-clé économique, ces termes provoquent une analyse complète du contenu.

Un mot-clé, même appartenant à l'univers de l'économie, ne suffit pas à provoquer la sélection de l'information. Il induit, tout au plus, une lecture un peu plus approfondie du contenu. Pour que s'opère la sélection, il faut que le contenu combine plusieurs termes pertinents. Le processus de sélection est donc basé sur une recherche booléenne.

L'analyse booléenne n'est pas la même avec tous les supports. En effet, les niveaux de lecture étant différents en fonction du support, l'analyse l'est également.

Ainsi, un support à vocation généraliste avec une couverture nationale (comme Le Monde ou Le Figaro) va induire une première recherche avec des mots-clés relevant de la géolocalisation (ex-recherche : Nord-Pas-de-Calais), puis dans un deuxième temps, une recherche avec des mots-clés typiquement liés à l'économie (entreprise ou investissement).

Un support avec une couverture nationale, mais à vocation économique (La Tribune ou Les Echos) va plus simplement induire une recherche géolocalisée.

Un support local, mais généraliste (comme La Voix du Nord, Ouest France, Le Parisien, etc.) n'imposera pas de géolocalisation puisqu'il est local et va donc induire une recherche de mots-clés typiquement liés à l'économie.

Un support local économique (La Gazette Nord-Pas de Calais, Bref Rhône-Alpes ou La Lettre API) implique une simple recherche de vocabulaire économique, puis une hiérarchisation de l'information en fonction de sa valeur ajoutée pour les clients.

Il est donc beaucoup plus aisé d'analyser un support à la fois local et économique puisqu'il réunit tous les paramètres de la requête et effectue en quelque sorte un pré-tri des informations. Il est en revanche beaucoup plus complexe d'analyser un support à la fois national et généraliste, puisqu'il implique d'opérer plusieurs requêtes (vérifier plusieurs critères), à la fois géographiques et sectorielles.

B.1.3/ EXTRACTION DES INFORMATIONS

L'objectif une fois une information sélectionnée est d'en rédiger une synthèse. Pour cela il faut extraire de la source d'information, par exemple un article de journal en ligne, les éléments à conserver. Pour cela, les principes des *5W* ou en français *QQOQCP* est utilisé. Les éléments conservés sont ceux qui répondent aux questions : *Qui ? Quoi ? Où ? Quand ? Comment ? Pourquoi ?*

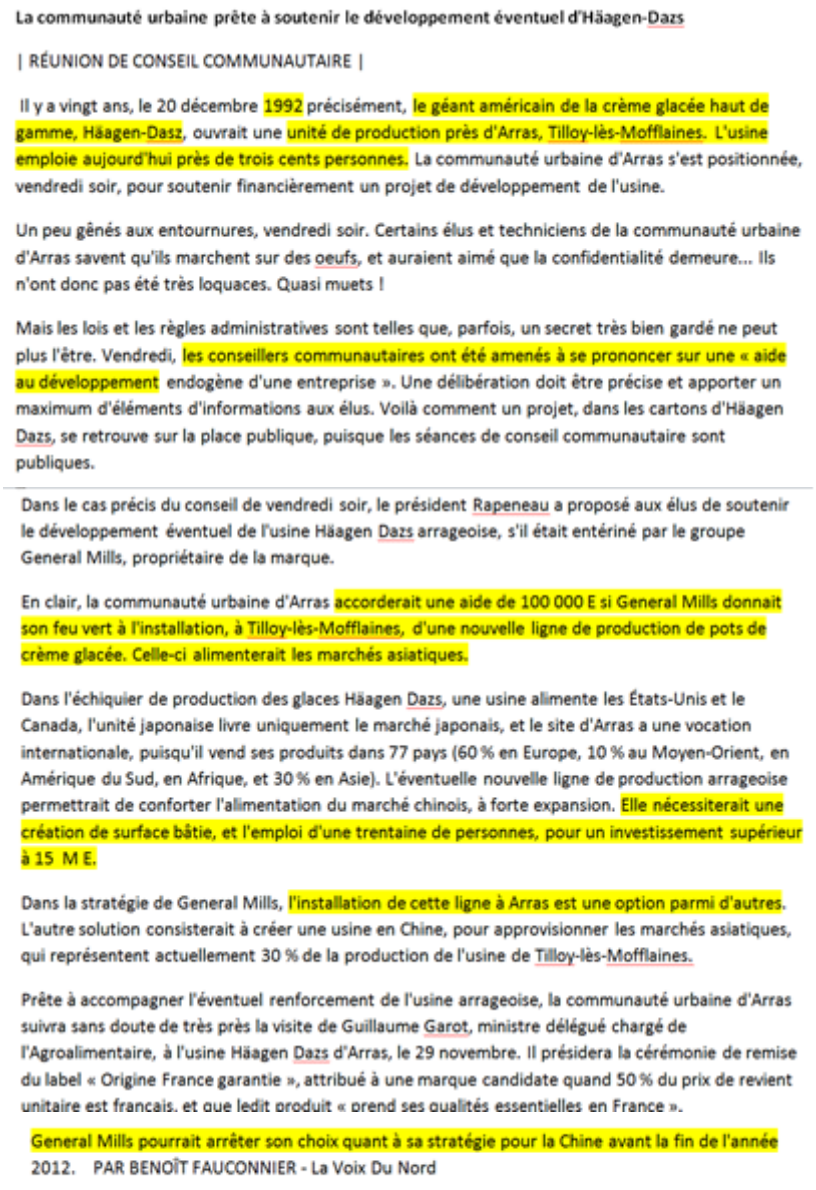
Des suppléments d'informations par rapport à ce qui est présent dans la source sont parfois ajoutés.

Il est possible que le documentaliste puisse répondre à ce qui est nommé le 2e *Quoi*. Une deuxième information trouvée sur un autre support montre que l'entreprise en question dans l'article réalise simultanément une deuxième action. Il faut donc recouper les informations.

Si l'entreprise a fait l'objet d'un article récent, un rappel de ce qui a été dit dans le précédent article sera introduit afin de développer le *Qui*.

Dans certains cas il peut tout simplement s'agir de renvoyer l'utilisateur vers un document, via un lien hypertexte par exemple.

Enfin, parfois l'article source est incomplet, imparfait, voire difficile à comprendre, mais il a tout de même été sélectionné comme pertinent à la phase précédente. Il est donc nécessaire de trouver plus d'informations en recroisant différentes sources, voire parfois en contactant directement les protagonistes.



La communauté urbaine prête à soutenir le développement éventuel d'Häagen-Dazs

| RÉUNION DE CONSEIL COMMUNAUTAIRE |

Il y a vingt ans, le 20 décembre 1992 précisément, le géant américain de la crème glacée haut de gamme, Häagen-Dasz, ouvrait une unité de production près d'Arras, Tilloy-lès-Mofflaines. L'usine emploie aujourd'hui près de trois cents personnes. La communauté urbaine d'Arras s'est positionnée, vendredi soir, pour soutenir financièrement un projet de développement de l'usine.

Un peu gênés aux entournures, vendredi soir. Certains élus et techniciens de la communauté urbaine d'Arras savent qu'ils marchent sur des oeufs, et auraient aimé que la confidentialité demeure... Ils n'ont donc pas été très loquaces. Quasi muets !

Mais les lois et les règles administratives sont telles que, parfois, un secret très bien gardé ne peut plus l'être. Vendredi, les conseillers communautaires ont été amenés à se prononcer sur une « aide au développement endogène d'une entreprise ». Une délibération doit être précise et apporter un maximum d'éléments d'informations aux élus. Voilà comment un projet, dans les cartons d'Häagen Dazs, se retrouve sur la place publique, puisque les séances de conseil communautaire sont publiques.

Dans le cas précis du conseil de vendredi soir, le président Rapeneau a proposé aux élus de soutenir le développement éventuel de l'usine Häagen Dazs arrageoise, s'il était entériné par le groupe General Mills, propriétaire de la marque.

En clair, la communauté urbaine d'Arras accorderait une aide de 100 000 E si General Mills donnait son feu vert à l'installation, à Tilloy-lès-Mofflaines, d'une nouvelle ligne de production de pots de crème glacée. Celle-ci alimenterait les marchés asiatiques.

Dans l'échiquier de production des glaces Häagen Dazs, une usine alimente les États-Unis et le Canada, l'unité japonaise livre uniquement le marché japonais, et le site d'Arras a une vocation internationale, puisqu'il vend ses produits dans 77 pays (60 % en Europe, 10 % au Moyen-Orient, en Amérique du Sud, en Afrique, et 30 % en Asie). L'éventuelle nouvelle ligne de production arrageoise permettrait de conforter l'alimentation du marché chinois, à forte expansion. Elle nécessiterait une création de surface bâtie, et l'emploi d'une trentaine de personnes, pour un investissement supérieur à 15 M E.

Dans la stratégie de General Mills, l'installation de cette ligne à Arras est une option parmi d'autres. L'autre solution consisterait à créer une usine en Chine, pour approvisionner les marchés asiatiques, qui représentent actuellement 30 % de la production de l'usine de Tilloy-lès-Mofflaines.

Prête à accompagner l'éventuel renforcement de l'usine arrageoise, la communauté urbaine d'Arras suivra sans doute de très près la visite de Guillaume Garot, ministre délégué chargé de l'Agroalimentaire, à l'usine Häagen Dazs d'Arras, le 29 novembre. Il présidera la cérémonie de remise du label « Origine France garantie », attribué à une marque candidate quand 50 % du prix de revient unitaire est français, et que ledit produit « prend ses qualités essentielles en France ».

General Mills pourrait arrêter son choix quant à sa stratégie pour la Chine avant la fin de l'année 2012. PAR BENOÎT FAUCONNIER - La Voix Du Nord

FIGURE B.1 – Exemple d'article source avec sélection des informations à extraire

La figure **??** est un exemple d'article source dans lequel les informations sélectionnées comme étant importantes à extraire pour les documentalistes sont mises en surbrillance.

B.1.4/ RÉDACTION DE LA SYNTHÈSE

La synthèse débute la plupart du temps par une rapide description du *Qui* (i.e. nom, activité, parfois brefs éléments historiques), généralement un *Où* (pour localiser le *Qui*). Ensuite, les rédacteurs introduisent le *Quoi* (i.e. action, raison d'être de la synthèse). Puis, ils détaillent le *Quoi* en apportant la réponse aux questions *Quand*, *Comment*, *Pourquoi* et *Où* (si l'action ne se déroule pas dans le même lieu que celui cité en introduction). Enfin, ils ajoutent très souvent des éléments complémentaires. La plupart du temps concernant le *Qui* mais sans rapport avec le *Quoi* (e.g. le chiffre d'affaires, l'effectif de l'entreprise, etc.).

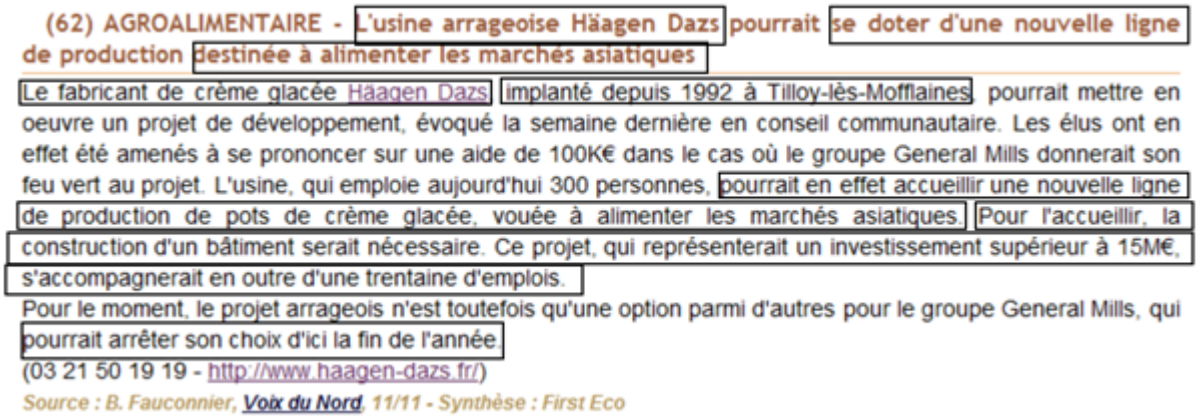
Le titre est introduit par un ou plusieurs termes définissant généralement les secteurs d'activités concernés par l'information. Il pouvait s'agir avant la mise en place de l'architecture dont cette thèse fait l'objet, dans certains cas de termes définissant un événement économique (i.e. thème) et non un secteur économique. Ces termes sont suivis d'une phrase répondant aux questions : *Qui*, *Quoi*, *Où*, *Quand* et parfois *Pourquoi*, comme le montre la figure **??**.

Spécifier le secteur ainsi que beaucoup d'informations dans le titre a pour objectif de permettre à l'utilisateur de savoir rapidement si l'article correspond à ses attentes. Cela était d'autant plus important avant la mise en place de la nouvelle architecture et donc du système de recommandation d'articles aux clients.

Afin de faciliter encore le travail aux clients, avant la mise en place de la revue personnalisée, les articles étaient hiérarchisés en fonction de leur importance pour les clients par leurs rédacteurs. Ainsi dans chaque revue régionale, les articles étaient organisés en fonction :

1. du *Où*, c'est-à-dire de la localisation concernée par l'article.
2. du *Qui* et du *Quoi* concernés par l'article.

Le *Qui* prend à la fois en compte, l'entreprise en elle-même (sa taille, son importance, son chiffre d'affaires) et le secteur économique concerné par son activité. Le quoi, concerne la raison d'être de l'article, c'est-à-dire, généralement l'événement économique (i.e. le thème) qui est la raison d'être de l'article. Certains secteurs d'activité ainsi que certains événements économiques intéressent évidement plus les lecteurs que d'autres. Ainsi, une entreprise industrielle qui va investir à toutes les chances de se retrouver en une.



(62) AGROALIMENTAIRE - L'usine arrageoise Häagen Dazs pourrait se doter d'une nouvelle ligne de production destinée à alimenter les marchés asiatiques

Le fabricant de crème glacée Häagen Dazs, implanté depuis 1992 à Tilloy-lès-Mofflaines, pourrait mettre en oeuvre un projet de développement, évoqué la semaine dernière en conseil communautaire. Les élus ont en effet été amenés à se prononcer sur une aide de 100K€ dans le cas où le groupe General Mills donnerait son feu vert au projet. L'usine, qui emploie aujourd'hui 300 personnes, pourrait en effet accueillir une nouvelle ligne de production de pots de crème glacée, vouée à alimenter les marchés asiatiques. Pour l'accueillir, la construction d'un bâtiment serait nécessaire. Ce projet, qui représenterait un investissement supérieur à 15M€, s'accompagnerait en outre d'une trentaine d'emplois.

Pour le moment, le projet arrageois n'est toutefois qu'une option parmi d'autres pour le groupe General Mills, qui pourrait arrêter son choix d'ici la fin de l'année.

(03 21 50 19 19 - http://www.haagen-dazs.fr/)

*Source : B. Fauconnier, **Voix du Nord**, 11/11 - Synthèse : First Eco*

FIGURE B.2 – Résultat de la synthèse de l'article de la Figure **??**

C

EVALUATION D'UN SYSTÈME DE RECOMMANDATION

Cette annexe présente et détaille les différentes méthodes d'évaluation utilisées dans cette thèse. L'évaluation des systèmes de recommandation est une tâche complexe. Nous présentons et expliquons dans cette annexe les approches d'évaluation objectives et subjectives. Nous détaillons principalement les approches objectives d'évaluation binaire et de corrélation de rang.

C.1/ EVALUATION D'UN SYSTÈME DE RECOMMANDATION

Le problème de l'évaluation des systèmes de recommandation est complexe. Parmi les différents facteurs induisant cette complexité, nous noterons la difficulté inhérente à la comparaison de systèmes souvent développés de façon ad'hoc à des cas d'application concrets distincts, ainsi que la multiplicité des acteurs et processus influençant la recommandation et sa perception. Nous présentons dans cette annexe, les outils permettant l'évaluation d'un système dans son ensemble.

La première section présente les outils d'évaluation objective, binaires, ordinaux et comportementaux. En suite, la seconde section présente succinctement ce en quoi consiste l'évaluation subjective.

C.1.1/ LES MÉTHODES D'ÉVALUATION OBJECTIVE

Selon Spinoza, le jugement objectif ne fait pas intervenir les sentiments ou l'imagination de celui qui l'énonce [**?**]. Plus tard, Kant étendra la notion de jugement objectif a tout jugement valable universellement [**?**]. Ainsi, les méthodes d'évaluation objective tentent d'évaluer les performances d'un système or de toute subjectivité, c'est à dire, en se basant sur des faits et non des opinions.

C.1.1.1/ LES MESURES D'ÉVALUATION CLASSIQUES : BINAIRE

Les deux mesures les plus populaires sont la precision (i.e. précision) et le recall (i.e. rappel). Ces mesures permettent l'évaluation des capacités du système à diminuer le bruit dans les réponses qu'il fournit (i.e. donner le moins possible de documents non pertinents) ainsi qu'à limiter les silences (i.e. limiter le nombre documents pertinents non fournis à l'utilisateur). Précision et rappel sont des mesures qui donnent des résultats entre 0 et 1, L'objectif est de permettre aux systèmes de tendre vers 1. Lors de la recommandation d'un item, quatre cas sont possibles :

1. Les vrais positifs vp : l'item est recommandé, et il correspond bien au profil.
2. Les faux positifs fp : l'item est recommandé, mais n'aurait pas dû, car il ne correspond pas au profil (i.e. baisse de la précision).
3. Les faux négatifs fn : l'item n'est pas recommandé, mais il aurait dû car il correspond au profil (i.e. silence donc baisse du rappel).
4. Les vrais négatifs vn : l'item n'est pas recommandé, ce qui est correct, car il ne correspond pas au profil.

$$Precision = \frac{vp}{(vp + fp)} \tag{C.1}$$

$$Rappel = \frac{vp}{(vp + fn)} \tag{C.2}$$

Toujours en utilisant les quatre cas précédemment expliqués, deux autres mesures existent. La première est le True Negative Rate aussi appelé Specificity (i.e. spécificité). La mesure True Positive Rate est déjà définie, il s'agit du recall ou sensitivity (i.e. sensibilité). La deuxième est Accuracy (i.e. incertitude) qui permet d'évaluer la marge d'erreur de l'instrument, c'est-à-dire du système de recommandation dans notre cas.

$$TrueNegativeRate = \frac{vn}{(vn + fp)} \tag{C.3}$$

$$Accuracy = \frac{(vp + vn)}{(vp + vn + fp + fn)} \tag{C.4}$$

L'optimisation de la précision et du rappel étant la problématique centrale des systèmes de rechercher d'informations autant que de recommandation, une mesure permettant de prendre en compte les deux en en pondérant éventuellement l'une par rapport à l'autre a été proposée par Rijsbergen en 1979 [**?**]. C'est la F-measure (i.e. F-mesure) ou encore F-score, généralement notée F_β.

$$F_\beta = \left(1 + \beta^2\right) \times \frac{Precision \times Recall}{\beta^2 \times Precision \times Recall} \tag{C.5}$$

Aujourd'hui, devant la multiplicité et la redondance de l'information disponible sur internet il est souvent considéré que la précision est plus importante que le rappel pour les systèmes utilisant de très vastes quantités d'information. La $F_{0,5}$-mesure est généralement utilisée pour donner plus d'importance à la précision qu'au rappel, le cas où les deux sont considérés comme tout autant important est appelé F_1-mesure. Ces mesures binaires bien que très utilisées ne semblent être totalement adaptées à notre problématique. En effet, nos résultats sont fournis de façon classée par ordre de pertinence. Le système ne dit pas à l'utilisateur "cet article A1 correspond à votre profil, alors que cet article A2 non", mais bien "cet article A1 correspond à 67,3% avec votre profil alors que celui-ci, A2, à seulement 12,6%".

Afin de pallier ce problème, Athena [**?**] a défini un seuil au-dessus duquel les résultats sont dits positifs et en dessous duquel ils sont dits négatifs. Il y a là une perte considérable de granularité. Ce que nous cherchons à évaluer est la pertinence de la pertinence mesurée par le système ; en d'autres termes est-ce qu'un article fourni à un utilisateur avec

une pertinence forte, par exemple 67,3%, mérite bien cette note, ou mérite-t-il plus, ou moins ? Une évaluation aussi précise semble impossible. Car, si le degré de pertinence d'un article est bien évaluable à la seule lumière du profil de l'utilisateur, il l'est seulement de façon floue. En effet, en se basant sur un nombre de classes de pertinence limité (e.g. très pertinent, pertinent, peu pertinent et non pertinent), l'évaluation est possible, par un humain par exemple. Par contre, mesurer précisément que le système à considéré que l'article était pertinent à 67,3% alors qu'il devrait l'être à 68,7% semble impossible. Ainsi, il semblerait que cette pertinence, ou la pertinence de cette pertinence ne soit évaluable qu'à la lumière des autres articles. C'est ici que le ranking (i.e. classement) intervient. En fonction des méthodes utilisées, un article très pertinent peut être à 20% et un peu pertinent à 0,5%, alors que pour une autre méthode, un peu pertinent sera à 62,5% et un très pertinent à 98%. Nous proposons donc de compléter les mesures d'évaluation binaires par des mesures de corrélations de rang.

C.1.1.2/ EVALUATION DE LA CORRÉLATION DE RANG : ORDINALE

Nous proposons d'utiliser des outils permettant l'évaluation de la corrélation entre variables ordinales afin de compléter les évaluations binaires. Elles permettent de pallier les problèmes d'évaluation dus à la définition d'un seuil lors d'une évaluation binaire. En effet, les articles représentés par leurs identifiants peuvent être vus comme des valeurs ordinales, car il y a une relation d'ordre entre eux. Ici nous faisons abstraction des valeurs de pertinence données aux articles. Nous considérons qu'il est impossible de savoir s'il y a le même écart entre la première valeur et la deuxième, qu'entre la deuxième et la troisième. La seule information prise en compte est que le premier est mieux classé que le deuxième, que le deuxième l'est mieux que le troisième et ainsi de suite. Ainsi comme avec les mesures classiques les résultats des algorithmes sont comparés à un résultat idéal. Mais contrairement aux mesures classiques, ce ne sont pas les articles sélectionnés et non sélectionnés dans les deux cas qui sont comparés, mais l'ordre des articles fournis aux utilisateurs. Des mesures permettent de comparer le tri réalisé par les algorithmes avec le tri idéal, et ainsi évaluer la qualité de la recommandation. C'est notamment le cas du Spearman's rho ainsi que du Kendall's tau.

Premièrement, prendre les résultats des variables (i.e. dans notre cas, c'est le tri des articles pour une méthode et un profil donné) et les traduire en rang. Attention, si deux articles ont le même rang, il faut leur donner la moyenne du rang. La plus forte valeur prend le rang 1.

Une fois les rangs calculés, il ne reste plus qu'à utiliser une mesure de corrélation linéaire classique entre les variables.

Spearman's rho Le coefficient de corrélation de Spearman est une application du coefficient de corrélation linéaire de Pearson adapté à des valeurs ordinales.

Le rho de Spearman varie entre -1 et 1 ; $\rho \in [1; 1]$. -1 signifiant un classement exactement inverse, 1 un classement exactement identique et 0 l'absence de corrélation.

$$\rho = 1 - \frac{6 \sum_{i=1}^{n} d_i^2}{n^3 - n} \tag{C.6}$$

Soit, n le nombre de valeurs, i le rang d'observation et d la différence entre le rang de la valeur et le rang d'observation.

Kendall's tau Comme le coefficient de corrélation de Spearman, le coefficient de corrélation de Kendall permet d'évaluer la corrélation existant entre deux variables ordinales.

le tau de Kendall varie entre -1 et 1 ; $\tau \in [1; 1]$. l'interprétation des résultats est la même que pour le rho de Spearman.

$$\tau = \frac{2(n_c - n_d)}{n(n-1)} \tag{C.7}$$

Soit, n le nombre de valeurs, n_c le nombre de concordances, c'est à dire le nombre de paires de valeurs dans le même ordre et n_d le nombre de discordances c'est à dire le nombre de paires différentes.

Devant la difficulté d'évaluation, certains systèmes n'évaluent pas réellement les résultats, mais le comportement des utilisateurs face à ces résultats.

C.1.1.3/ ÉTUDE DU COMPORTEMENT UTILISATEUR

L'étude du comportement des utilisateurs sur la plateforme peut fournir des informations pertinentes. Ainsi, par hypothèse, la qualité du système doit influencer son comportement d'utilisation et de navigation. Différents facteurs peuvent être pris en compte. Par exemple, le temps passé sur la plateforme, le nombre de connexions à l'outil, le nombre de clics et d'articles lus, le temps passé sur chacun des articles, le temps entre deux clics. Toutes ces activités peuvent être interprétés afin de donner des informations sur la qualité et l'utilité du système.

Toutefois, bien que les données soient objectives, leur exploitation est soumise à interprétation. Si un utilisateur passe beaucoup de temps à utiliser l'outil, est que c'est un bon outil, agréable à utiliser ? Où est-il obligé d'y passer plus de temps, car l'outil fonctionne mal et ne lui fournit pas directement les informations qu'il cherche ?

Nous distinguons ici deux types de comportement de la part de l'utilisateur sur la plate-

forme.

1. Un comportement qui fournit des informations dites explicites. L'utilisateur donne explicitement une information qui n'a pas à être interprétée, elle est donc plus simple à manipuler.

2. Un comportement, où l'utilisateur n'a pas pour objectif de fournir des informations, mais par lequel il en transmet tout de même. Ces informations sont plus difficiles à manipuler. Les conséquences sur la qualité du système plus difficile à déduire.

Ce sont les mêmes informations fournies par l'utilisateur qui peuvent être utilisées afin d'évaluer son besoin.

C.1.2/ LES MÉTHODES D'ÉVALUATION SUBJECTIVES

Les enjeux d'une évaluation sont importants. Les résultats vont influencer l'avenir de l'outil qui est évalué, ainsi l'objectivité est souvent recherchée en priorité. Néanmoins, les systèmes de recommandation servent à fournir de l'information à leurs utilisateurs. Kant a énoncé qu'un jugement objectif était un jugement valable universellement [**?**], c'est-à-dire indépendamment d'une personne ou d'un groupe de personnes. Or, l'outil s'adressant justement à un groupe de personnes. L'exploitation du jugement subjectif d'un échantillon représentatif de ce groupe de personnes semble donc pertinente.

L'organisation d'un test subjectif est une tâche complexe, car la pertinence des résultats dépendent de la façon de celui-ci est mené. La principale mesure d'évaluation subjective d'un système consiste à demander l'avis des utilisateurs. Cette demande se fait principalement par l'intermédiaire de questionnaires. Nous avons présenté l'étude du comportement de l'utilisation comme une évaluation objective. En effet, la subjectivité est introduite par la façon de poser la question. Il n'y a donc pas de subjectivité pour l'utilisateur qui se contente d'utiliser l'outil.

Les commentaires des utilisateurs, ainsi que leurs réponses aux questionnaires peuvent être utilisés quantitativement, avec des questionnaires à choix multiples (i.e. questions fermées) et des analyses statistiques afin d'évaluer les performances d'un système du point de vue des utilisateurs.

Toutefois afin d'en garantir la qualité, certaines "règles" doivent être respectées. Ainsi il faut faire attention aux doubles questions, par exemple "est-ce que le système de recommandation fournit des éléments nouveaux et pertinents ?". Dans ce cas, que va répondre l'utilisateur s'il trouve que le système recommande bien des éléments nouveaux, mais non pertinents pour lui ? Il faut aussi faire attention aux questions qui pourraient trop influencer la réponse de l'utilisateur, par exemple "est-ce que le système est bon ?". Il est

plus pertinent de demander à l'utilisateur de noter le système avec une note entre 0-5 par exemple, ou par le choix d'un mot dans une liste (e.g. complètement d'accord, plutôt d'accord, ni en accord ni en désaccord, plutôt en désaccord, totalement en désaccord). Les concepts à étudier sont la satisfaction, l'utilité ainsi que la difficulté ou la complexité de l'outil. Ce sont des concepts difficilement évaluables avec une seule question. Il est donc préférable d'en utiliser plusieurs.

D

REPRÉSENTER ET RAISONNER À PARTI DE CONNAISSANCES

Cette annexe présente un certain nombre de notions et de concepts issus du domaine du Web Sémantique et utilisés dans les travaux exposés dans ce document. Cette annexe traite principalement de la logique de description et des raisonnements logiques. Elle est fortement inspirée du travail de M. Christophe Cruz [**?**] et a été coécrite avec Fayrouz Soualah Alila [**?**]. Cette annexe et partiellement commune à nos deux thèses.

D.1/ LES LOGIQUES DE DESCRIPTION

La logique descriptive est une famille de formalismes utilisés pour représenter une base de connaissances d'un domaine d'application d'une façon structurée et formelle [**?**]. Les logiques de description ont deux objectifs : Tout d'abord représenter les connaissances d'un domaine, c'est donc la partie description ; et ensuite raisonner à partir de ces connaissances, c'est la partie logique.
Cette annexe traite des logiques descriptives ainsi que les aspects de raisonnement logique. Leur utilisation conjointe permet la définition de bases de connaissances et à partir de ces connaissances la déduction de nouveaux faits.

D.1.1/ LES LANGAGES FORMELS ET LE WEB SÉMANTIQUE

Les logiques de description sont utilisées pour de nombreuses applications, parmi lesquelles nous pouvons citer les domaines suivants : traitement automatique des langues, l'ingénierie logicielle (i.e. représentation de la sémantique des diagrammes de classe UML) et le Web sémantique pour la représentation d'ontologies. Le courant des recherches, qui s'est nourri d'études effectuées sur la logique des prédicats et les réseaux sémantiques, a donné naissance à une famille de langages de représentation appelés logiques de description $\mathcal{LD}$. Ces langages ont été introduits dans les années 80 dans le but de rendre la représentation de connaissances plus naturelle qu'en logique du premier ordre. Nous nous intéressons donc ici aux logiques de description en tant que langage de modélisation de données.
Une caractéristique fondamentale des logiques de description est qu'elles ont une sémantique formelle conforme au cadre des logiques de [**?**]. Les connaissances d'un domaine y sont représentées par des entités qui correspondent à une description syntaxique à laquelle est associée une sémantique :
(1) Les éléments du monde réel y sont représentés à l'aide de concepts, rôles et individus. Les concepts ainsi que les rôles sont organisés de façon hiérarchique à l'aide de relations de subsomption. L'utilisation de ce type de logique permet la réalisation de tâches de raisonnement.
(2) La sémantique des logiques de description est clairement définie et associée à la description des concepts ainsi que des rôles. Cette sémantique affecte la façon de manipuler ces éléments. Cette sémantique est définie comme suit :
Soit $\mathfrak{C} = \{C1, C2, \ldots\}$ un ensemble fini de concepts atomiques, $\mathfrak{R} = \{R1, R2, \ldots\}$ un ensemble fini de rôles atomiques et $\mathfrak{I} = \{a1, a2, \ldots\}$ un ensemble fini d'individus. Si $\mathfrak{C}$, $\mathfrak{R}$, $\mathfrak{I}$ sont disjoints deux à deux, $\mathcal{S} = \langle \mathfrak{C}, \mathfrak{R}, \mathfrak{I} \rangle$ est une signature. Une fois qu'une signature $\mathcal{S}$ est fixée, un modèle d'interprétation $\mathcal{I}$ pour $\mathcal{S}$ est défini comme un couple $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, où :

- $\Delta^{\mathcal{I}}$ est un ensemble non vide, le domaine d'interprétation.
- $\cdot^{\mathcal{I}}$ est la fonction d'interprétation :
  - un élément $a_i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ à chaque individu $a_i \in \mathfrak{I}$,
  - un sous-ensemble $C_i^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ à chaque concept atomique $C_i \in \mathfrak{C}$
  - une relation $R_i^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ à chaque rôle atomique $R_i \in \mathfrak{R}$.

D.1.1.1/ LES BASES DE CONNAISSANCE

Une base de connaissance est composée d'une partie terminologique, qualifiée de *TBox* (i.e. Terminology Box), et d'une partie assertionnelle, qualifié de *ABox* (i.e. Assertions Box). La *ABox* décrit les individus et leurs relations (quel individu appartient à quel concept nommé, quel individu est lié à quel autre à travers quel rôle). Typiquement, la base de connaissances standard utilisée par les logiques de description est définie de la manière suivante :
Étant donné un langage de description $\mathcal{L}$ et une signature $\mathcal{S}$, une base de connaissances Σ dans $\mathcal{L}$ est une paire $\Sigma = \langle \mathcal{T}, \mathcal{A} \rangle$ telle que :

- $\mathcal{T}$ la *TBox*, est un ensemble fini, qui peut être vide, d'expressions appelées GCI (i.e. General Concept Inclusion) de la forme $C \sqsubseteq D$ où C et D sont des concepts sans restriction. La *TBox* contient les définitions de concepts. Un nouveau concept peut être défini en fonction des concepts déjà présents dans la *TBox*. Seuls des axiomes d'équivalence $C \equiv D$ et d'inclusion $C \sqsubseteq D$ sont utilisés pour la constituer. Par exemple, $Femme \equiv Personne \wedge Femelle$ [**?**]. De façon générale, $C \equiv D$ ssi $C \sqsubseteq D$ et $D \sqsubseteq C$.
  Une interprétation $\mathcal{I}$ d'une *TBox* satisfait $C \sqsubseteq D$ si $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ et elle satisfait $C \equiv D$ si $C^{\mathcal{I}} \subseteq D^{\mathcal{I}} \wedge D^{\mathcal{I}} \subseteq C^{\mathcal{I}}$.
- $\mathcal{A}$ la *ABox*, un ensemble fini, qui peut être vide, d'expressions de la forme $C(a)$ ou $R(a,b)$, où C est un concept sans restriction, R est un rôle qui n'est pas nécessairement atomique, et a, b appartiennent à $\mathfrak{I}$. Les formules de *A* sont appelées des assertions.
  Une interprétation $\mathcal{I}$ d'une *ABox* associe à chaque constante a un élément $a^{\mathcal{I}}$ de $\Delta^{\mathcal{I}}$. $\mathcal{I}$ est un modèle de la *ABox* si pour toute assertion $C(a) \Rightarrow a^{\mathcal{I}} \in C^{\mathcal{I}}$ et $R(a,b) \Rightarrow (a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$

Un modèle étant une interprétation qui satisfait tous les axiomes d'une base de connaissances.

Les logiques de description adoptent l'hypothèse de nom unique, cela implique que les individus possèdent toujours des noms différents. Deux individus ayant le même nom sont confondus, ils forment donc le même individu. De plus, les bases de connaissances reposant sur les logiques de description adoptent la sémantique du monde ouvert. L'hypothèse du monde ouvert implique que les informations peuvent être incomplètes. Ainsi, ce qui ne peut pas être prouvé à partir des informations disponibles n'est pas nécessairement faux. Contrairement à l'hypothèse de monde clos. Celui-ci implique que l'information dans la base de connaissances est nécessairement complète et donc que ce qui ne peut pas être prouvé à partir des informations disponibles est faux. Par exemple, la base de connaissances est composée des individus suivants Homme(Jean), Homme(Paul) et de l'assertion de rôle suivant : possèdeEnfants(Jean, Paul). La question est : est-ce que tous les enfants de Jean sont des hommes ? La réponse est vraie dans l'hypothèse d'un monde clos, tel qu'il est adopté dans le domaine des bases de données relationnelles. Par contre, le résultat est inconnu pour la sémantique du monde ouvert. Car aucune information n'est disponible stipulant que Paul est le seul enfant de Jean.

D.1.1.2/ LES FAMILLES DE LOGIQUES DE DESCRIPTION

Les $\mathcal{LD}$ forment une famille de langages de représentation de connaissances. Dans le formalisme des $\mathcal{LD}$, un concept représente un ensemble d'individus et un rôle représente une relation binaire entre individus. Un concept correspond à une entité générique d'un domaine d'application et un individu à une entité particulière, instance d'un concept.
Concepts, rôles et individus obéissent aux principes suivants :

- Un concept et un rôle possèdent une description structurée, élaborée à partir d'un certain nombre de constructeurs. Une sémantique est associée à chaque description de concept et de rôle par l'intermédiaire d'une interprétation.
- Les connaissances sont prises en compte selon plusieurs niveaux : la représentation et la manipulation des concepts et des rôles relèvent du niveau terminologique *TBox* alors que la description et la manipulation des individus relèvent du niveau factuel ou niveau des assertions *ABox*.
- La relation de subsomption permet d'organiser concepts et rôles par niveau de généralité : intuitivement, un concept C subsume un concept D si C est plus général que D au sens où l'ensemble d'individus représenté par C contient l'ensemble d'individus représenté par D. Une base de connaissances se compose alors d'une hiérarchie de concepts et d'une (éventuelle) hiérarchie de rôles.

Exemple : Cet exemple décrit les relations entre membres d'une famille :

- Concepts : *Femme*, *Homme*, etc.
- Rôles : *epousDe*, *pereDe*, etc.
- Individus : *marie*, *pierre*, *jean*, etc.

Un concept et un rôle possèdent une description structurée définie à partir d'un certain ensemble de constructeurs. Concepts, rôles et individus de l'exemple précédent sont décrits comme suit au niveau de la *TBox* et de la *ABox* :

- *TBox*, décrit les concepts :

$$\begin{gathered} \textit{Femelle} \sqsubseteq \top \sqcap \neg \textit{Male} \\ \textit{Male} \sqsubseteq \top \sqcap \neg \textit{Femelle} \\ \textit{Animal} \equiv \textit{Male} \sqcup \textit{Femelle} \\ \textit{Humain} \sqsubseteq \textit{Animal} \\ \textit{Femme} \equiv \textit{Humain} \sqcap \textit{Femelle} \\ \textit{Homme} \equiv \textit{Humain} \sqcap \neg \textit{Femelle} \end{gathered}$$

- *ABox*, fournit des instances des concepts et des rôles :

$$\begin{gathered} \textit{marie : Femme} \\ \textit{pierre : Homme} \\ \textit{jean : Homme} \\ \textit{epouseDe(pierre, marie)} \\ \textit{pereDe(jean, pierre)} \end{gathered}$$

Il existe plusieurs familles de langages de description des concepts, des rôles et des individus. La logique de description $\mathcal{AL}$ (Attribute Language) est la logique minimale, dite aussi logique attributive, et elle a été définie par [**?**]. Elle est dite minimale car une logique d'expressivité inférieure n'aurait pas d'intérêt pratique [**?**].
Les descriptions possibles dans le langage $\mathcal{AL}$ sont présentées par le tableau **??**. On suppose que A est un concept atomique et que C et D sont des concepts atomiques ou complexes.

Le langage de base $\mathcal{AL}$ est défini à partir des éléments syntaxiques suivants :

- Le concept *Top* ($\top$) dénote le concept le plus général et le concept *Bottom* ($\bot$) le concept le plus spécifique. Intuitivement, l'extension de Top inclut tous les individus possibles tandis que celle de Bottom est vide.

Syntaxe	Définition
A	Concept atomique
$\top$	Le concept universel Top
$\bot$	Le concept vide Bottom
$\neg A$	Le Complément de concept atomique
$C \sqcap D$	Conjonction de concepts
$\forall R.C$	Quantificateur universel
$\exists R.\top$	Quantificateur existentiel non typé

TABLE D.1 – La grammaire du langage de description de concepts selon $\mathcal{AL}$

- Le constructeur *not* ($\neg$) correspond à la négation et ne porte que sur les concepts primitifs (atomiques). Le constructeur *and* ($\sqcap$) permet de définir une conjonction d'expressions conceptuelles.
- La quantification universelle *all* ($\forall R.C$) précise le co-domaine du rôle R et la quantification existentielle non typée *some* ($\exists R$) introduit le rôle R et affirme l'existence d'au moins un couple d'individus en relation par l'intermédiaire de R.

On obtient alors le langage $\mathcal{AL} = \{\top, \bot, A, \neg A, C \sqcap D, \forall r.C, \exists r\}$ qui peut être par la suite enrichi par plusieurs autres constructeurs.
La sémantique du langage $\mathcal{AL}$ fait appel à la théorie des ensembles. Essentiellement, à chaque concept est associé un ensemble d'individus dénotés par ce concept. Une interprétation suppose donc l'existence d'un ensemble non vide Δ qui représente des entités du monde décrit. Une fonction d'interprétation (décrite précédemment) associe à un concept C un sous-ensemble $C^{\mathcal{I}}$ de $\Delta^{\mathcal{I}}$ et à un rôle R un sous-ensemble $R^{\mathcal{I}}$ de $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ [**?**].
La sémantique de $\mathcal{AL}$ est définie alors comme dans le tableau **??**.

La logique $\mathcal{AL}$ n'est généralement pas suffisamment expressive pour représenter certaines connaissances. Il existe alors plusieurs autres constructeurs que l'on peut ajouter à ce langage pour le rendre plus expressif. Notamment, la logique $\mathcal{ALC}$ qui est l'extension de la logique $\mathcal{AL}$ par la négation complète C (i.e. négation de concepts non primitifs, définis) [**?**]. $\mathcal{AL}$ constitue la logique descriptive de base, $\mathcal{ALC}$ la logique descriptive minimale : $\mathcal{ALC} = \mathcal{AL} \cup \{\neg C\}$.
Les différentes logiques de description qui existent aujourd'hui sont des combinaisons des différents constructeurs du tableau **??** avec la logique de base $\mathcal{ALC}$.

Le tableau **??** décrit la syntaxe et la sémantique des différents constructeurs des logiques de description.

Nous pouvons ainsi obtenir d'autres langages plus expressifs, tels qu'avec l'utilisation

Syntaxe	Sémantique
A	$A^{\mathcal{I}}$
$\top$	$\Delta^{\mathcal{I}}$
$\bot$	$\emptyset$
$\neg A$	$\Delta^{\mathcal{I}} \setminus A^{\mathcal{I}}$
$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
$\forall R.C$	$\{x \in \Delta^{\mathcal{I}} \mid \forall y \in \Delta^{\mathcal{I}}, (x,y) \in R^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}}\}$
$\exists R.\top$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}}, (x,y) \in R^{\mathcal{I}}\}$

TABLE D.2 – La sémantique du langage de description de concepts selon $\mathcal{AL}$

des constructeurs suivants :

- Le constructeur $\mathcal{U}$ permet de définir la disjonction de concepts, notée $C \cup D$. L'extension correspondante d'$\mathcal{AL}$ est $\mathcal{ALCU}$.
- Le constructeur $\mathcal{R}$ permet de définir la conjonction de rôles, notée $R_1 \cap R_2$, les rôles R_1 et R_2 étant primitifs. L'extension correspondante d'$\mathcal{AL}$ est $\mathcal{ALCR}$.
- La quantification existentielle typée $\mathcal{E}$, notée $\exists R.C$ (la quantification $\exists R.C$ introduit un rôle R de co-domaine C et impose l'existence d'au moins un couple d'individus (x, y) en relation par l'intermédiaire de R, où C est le type de y). L'extension correspondante d'$\mathcal{AL}$ est $\mathcal{ALCE}$.
- Le constructeur $\mathcal{N}$, noté $\geq nR$ (i.e. au moins nR) et $\leq nR$ (i.e. au plus nR), fixent la cardinalité minimale et maximale du rôle auquel ils sont associés. L'extension correspondante d'$\mathcal{AL}$ est $\mathcal{ALCN}$.

Certaines logiques sont équivalentes, notamment $\mathcal{ALC}$ et $\mathcal{ALUE}$. Ces deux dernières logiques augmentées par $\mathcal{R}^+$ sont notées $\mathcal{S}$.

Toutes les logiques de description qui existent sont des combinaisons des différents constructeurs du tableau **??**. Les principales familles sont illustrées dans le tableau **??**.

D.1.1.3/ LE LANGAGE ONTOLOGIQUE OWL

Le langage OWL, proposé par le consortium W3C, est le langage de représentation des connaissances permettant la définition d'ontologies. Il est basé sur les travaux ayant eu lieu dans le domaine des logiques de description précédemment introduites. Cela permet la mise en place de processus de vérification de cohérence et de consistance ainsi que de déduction de connaissances à partir des connaissances existantes.

Lettre	Définition
$\mathcal{F}$	Rôles fonctionnels
$\mathcal{E}$	Quantifications existentielles typées
$\mathcal{U}$	Disjonction de rôles
$\mathcal{C}$	Négation complète
$\mathcal{S}$	Abréviation de la logique $\mathcal{ALC}$ avec la transitivité des rôles $\mathcal{R}^+$
$\mathcal{H}$	Hiérarchie de rôles
$\mathcal{O}$	Classes nominales ou énumération par individus
$\mathcal{I}$	Rôles inverses
$\mathcal{N}$	Restrictions de cardinalité
$\mathcal{Q}$	Restrictions de cardinalité qualifiée
$(\mathcal{D})$	Support des types primitifs

TABLE D.3 – Constructeurs des logiques de description

Constructeurs	Syntaxe	Sémantique
$\mathcal{F}$	$\leq 1R$	$\{x \in \Delta^I \mid y, (x,y) \in R^I \mid \leq 1\}$
$\mathcal{E}$	$\exists R.C$	$\{x \in \Delta^I \mid \exists y \in \Delta^I, (x,y) \in R^I \Rightarrow y \in C^I\}$
$\mathcal{U}$	$C_1 \sqcup C_2$	$C_1^I \cup C_2^I$
$\mathcal{C}$	$\neg C$	$\Delta^I \setminus C^I$
$\mathcal{H}$	$R_1 \sqsubseteq R_2$	$R_1^I \subseteq R_2^I$
$\mathcal{O}$	I	$I^I \subseteq \Delta^I$ avec $\lvert I^I \rvert = 1$
$\mathcal{I}$	R^{-1}	$\{(x,y) \in R^I \mid (y,x) \in R^I\}$
$\mathcal{N}$	$\geq nR$	$\{x \in \Delta^I \mid y, (x,y) \in R^I \mid \geq n\}$
$\mathcal{N}$	$\leq nR$	$\{x \in \Delta^I \mid y, (x,y) \in R^I \mid \leq n\}$
$\mathcal{N}$	$= nR$	$\{x \in \Delta^I \mid y, (x,y) \in R^I \mid = n\}$
$\mathcal{Q}$	$\geq nR.C$	$\{x \in \Delta^I \mid y \in C^I, (x,y) \in R^I \mid \geq n\}$
$\mathcal{Q}$	$\leq nR.C$	$\{x \in \Delta^I \mid y \in C^I, (x,y) \in R^I \mid \leq n\}$
$\mathcal{Q}$	$= nR.C$	$\{x \in \Delta^I \mid y \in C^I, (x,y) \in R^I \mid = n\}$
$\mathcal{R}$	$R_1 \sqcap R_2$	$R_1^I \cap R_2^I$

TABLE D.4 – Constructeurs : syntaxe et sémantique

Les ontologies sont une structure de données largement utilisée pour la réalisation de bases de connaissances.

Le langage OWL est composé de trois sous-langages : OWL-Lite, OWL-DL et OWL-Full. OWL-Lite correspond essentiellement à la famille $\mathcal{SHIF}$ alors que OWL-DL, qui signifie «Description Logic», correspond essentiellement à la famille $\mathcal{SHOIN}$. Plus précisément, il s'agit des familles $\mathcal{SHIF}(\mathcal{D})$ et $\mathcal{SHOIN}(\mathcal{D})$. La distinction est faite sur les deux types de rôles : les rôles qui lient deux individus et les rôles qui associent un

Composition	**Famille**
$\mathcal{AL}$	Logique de base
$\mathcal{ALC}$	Logique minimale
$\mathcal{ALC} + \mathcal{N}$	Logique $\mathcal{ALCN}$
$\mathcal{ALC} + \mathcal{Q}$	Logique $\mathcal{ALCQ}$
$\mathcal{ALC} + \mathcal{F}$	Logique $\mathcal{ALCF}$
$\mathcal{ALC} + \mathcal{R}^+$	Logique $\mathcal{S}$
$\mathcal{S} + \mathcal{H}$	Logique $\mathcal{SH}$
$\mathcal{SH} + \mathcal{I} + \mathcal{F}$	Logique $\mathcal{SHIF}$
$\mathcal{SH} + \mathcal{I} + \mathcal{Q}$	Logique $\mathcal{SHIQ}$
$\mathcal{SH} + \mathcal{O} + \mathcal{I} + \mathcal{N}$	Logique $\mathcal{SHOIN}$
$\mathcal{SH} + \mathcal{O} + \mathcal{I} + \mathcal{Q}$	Logique $\mathcal{SHOIQ}$

TABLE D.5 – Familles de logique de description

individu avec un type primitif (e.g. entier, chaîne de caractères, etc.), d'où le $(\mathcal{D})$, pour Data Property. OWL-Lite est un sous-langage d'OWL-DL. OWL-DL est une version décidable du langage OWL. Il permet donc la réalisation de raisonnements, contrairement à OWL-Full.

En 2009, le consortium W3C a officiellement lancé OWL 2, qui se distingue de la première version par un pouvoir expressif augmenté et une élimination de la décomposition en trois sous-langages. Ainsi OWL-Full n'existe plus, alors que OWL-Lite et OWL-DL sont considérés comme des profils de OWL 2 (on parle de profil lorsqu'on élimine de OWL 2 certains constructeurs limitants sans pouvoir expressif). Depuis 2012, OWL 2 est recommandé par W3C et permet l'expressivité de la logique $\mathcal{SROIQ}(\mathcal{D})$ [**?**].

L'évolution de l'expressivité depuis les logiques descriptives les moins expressives jusqu'aux logiques les plus complexes utilisées dans le domaine du Web sémantique est présentée dans la figure **??**.

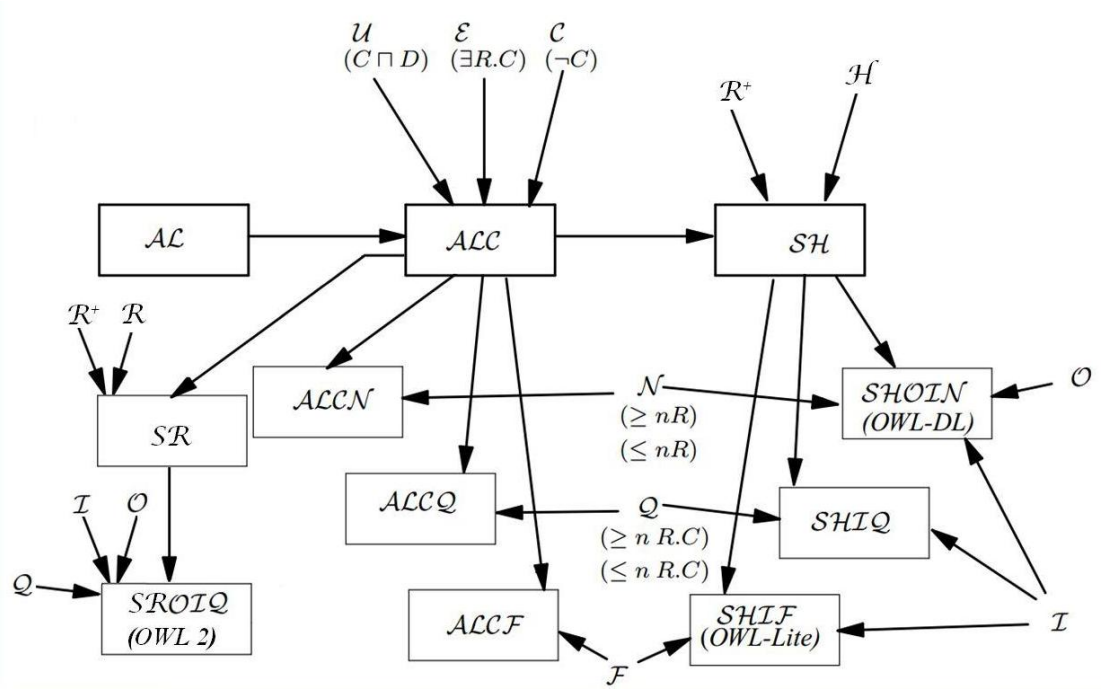


FIGURE D.1 – Evolution des logiques de description [**?**]

Les langages OWL-DL et OWL-Lite se basent respectivement sur les logiques $\mathcal{SHOIN}$ et $\mathcal{SHIF}$ qui sont une augmentation de l'expressivité de la logique $\mathcal{SH}$, elle-même basée sur la logique $\mathcal{S}$ à laquelle la gestion des hiérarchies de rôles a été ajoutée. Tandis que le langage OWL 2 a l'expressivité de la logique $\mathcal{SROIQ}$ qui est une augmentation de l'expressivité de la logique $\mathcal{SR}$ basée sur la logique $\mathcal{S}$ à laquelle contrairement la logique $\mathcal{SH}$ les inclusion de rôles R ont été ajoutées en plus des hiérarchies de rôles.

Constructeurs	**Syntaxe**	**Exemple**
intersectionOf	$C_1 \sqcap C_2$	$Humain \sqcap Homme$
unionOf	$C_1 \sqcup C_2$	$Docteur \sqcup Avocat$
complementOf	$\neg C$	$\neg Homme$
oneOf	$\{a_1 ... a_2\}$	$\{marie, pierre, jean\}$
allValueFrom	$\forall R.C$	$\forall possedeEnfant.Docteur$
someValueFromFrom	$\exists R.C$	$\exists possedeEnfant.Avocat$
hasValue	$\exists R.\{a\}$	$\exists citoyenDe.\{France\}$
minCardinality	$(\geq nR)$	$(\geq\ 2\ possedeEnfant)$
maxCardinality	$(\leq nR)$	$(\leq\ 1\ possedeEnfant)$
inverseOf	R^-	$possedeEnfant^-$

TABLE D.6 – Constructeurs OWL

Les tableaux **??** et **??** présentent les constructeurs du langage OWL et les axiomes du

Axiome	Syntaxe	Exemple
subClassOf	$C_1 \sqsubseteq C_2$	$Humain \sqsubseteq Animal \sqcap Bipede$
equivalentClass	$C_1 \equiv C_2$	$Homme \equiv Humain \sqcap Male$
subPropertyOf	$R_1 \sqsubseteq R_2$	$possedeFille \sqsubseteq possedeEnfant$
equivalentProperty	$R_1 \equiv R_2$	$Cout \equiv Prix$
disjointWith	$C_1 \sqsubseteq \neg C_2$	$Male \sqsubseteq \neg Femelle$
sameAs	$\{a_1\} \equiv \{a_2\}$	$\{president\_hollande\} \equiv \{francois\_hollande\}$
differentFrom	$\{a_1\} \sqsubseteq \neg\{a_2\}$	$\{marie\} \sqsubseteq \neg\{jean\}$
transitiveProperty	R rôle transitif	$estAncetreDe$ rôle transitif
functionalProperty	$\top \sqsubseteq (\leq 1R)$	$\top \sqsubseteq (\leq 1\ estMereDe)$
inverseFunctionalProperty	$\top \sqsubseteq (\leq 1R^-)$	$\top \sqsubseteq (\leq 1\ aPourMere^-)$
symmetricProperty	$R \equiv R^-$	$estMarieAvec \equiv estMarieAvec^-$

TABLE D.7 – Axiomes OWL 2

langage OWL 2 ainsi que leur correspondance en logiques de description.
Différentes syntaxes peuvent être utilisées afin de sérialiser une ontologie OWL, par exemple en Turtle ou en XML.
L'exemple 1 décrit la syntaxe XML pour la définition d'une classe étant l'intersection des classes *Homme* et *Humain* (*Humain* $\sqcap$ *Homme*) :

```
<owl:Class>
        <owl:intersectionOf rdf:parserType="Collection">
                <owl:Class rdf:about="#Humain"/>
                <owl:Class rdf:about="#Homme"/>
        </owl:intersectionOf>
</owl:Class>
```

L'exemple 2 décrit la syntaxe XML pour la définition d'une restriction sur la cardinalité minimum d'un rôle ($\geq$ 2 *possede-enfant.Thing*) :

```
<owl:Restriction>
        <owl:OnProperty rdf:Ressource="#possede-enfant"/>
        <owl:minCardinality rdf:datatype="&xsd;NonNegativeInterger"> 2
        <owl:minCardinality>
</owl:Restriction>
```

D.1.2/ LES RAISONNEMENTS

Le raisonnement est un processus qui permet de produire de nouveaux résultats ou de vérifier des faits. Les systèmes de représentation de connaissances basés sur les lo-

giques de description permettent la réalisation de raisonnements. En effet, dans le terme *logique de description*, la *description* signifie le fait de représenter les connaissances et la *logique* le fait de raisonner à partir de ces connaissances. Les bases de connaissances composées de *TBox* $\mathcal{T}$ et de *ABox* $\mathcal{A}$ sont des systèmes de représentation des connaissances d'un domaine. Différents types de raisonnements peuvent être effectués à partir de ces connaissances, ils sont nommés inférences logiques.

D.1.2.1/ INFÉRENCES

Parmi les différentes inférences logiques réalisables sur une base de connaissances, nous distinguons les trois cas suivants :

1. Les inférences qui permettent de découvrir des connaissances implicites à partir des connaissances explicites contenues dans la base de connaissances.
2. Les inférences qui permettent de vérifier la cohérence de la *TBox*.
3. Les inférences qui permettent de vérifier la consistance de la *ABox*.

La découverte de nouvelles connaissances correspond à la :

- découverte de propriétés d'individus.
- découverte de relations entre individus,
- découverte de relations de subsomption ou d'équivalence entre concepts permettant de produire la hiérarchie des concepts (i.e. hiérarchisation).
- découverte de relations entre concepts et instances. Notamment la classification au plus spécifique concept pour chaque individu (i.e. réalisation).

Une base de connaissances $\Sigma = \langle \mathcal{T}, \mathcal{A} \rangle$ est incohérente si au moins un des concepts C appartenant à sa *TBox* $\mathcal{T}$ est insatisfiable. C'est-à-dire si pour chaque interprétation $\mathcal{I}$ le nombre d'instances du concept C est égale à zéro. Il n'existe pas de C tel que $\Sigma \models C \equiv \bot$. Un concept C est satisfiable s'il existe un modèle d'interprétation $\mathcal{I}$ de $\mathcal{T}$ pour lequel $C^{\mathcal{I}}$ est non vide. Dans ce cas $\mathcal{I}$ est un modèle de C.

Une base de connaissances $\Sigma = \langle \mathcal{T}, \mathcal{A} \rangle$ est inconsistante s'il n'existe pas d'interprétation $\mathcal{I}$ qui soit un modèle pour sa *TBox* $\mathcal{T}$ et sa *ABox* $\mathcal{A}$.

D.1.2.2/ APPROCHES DE RAISONNEMENT

A partir de la technique employée dans le développement des algorithmes de décision pour le problème d'inférence, nous pouvons les catégoriser en deux groupes :

Le premier groupe contient des algorithmes appelés algorithmes structurels. Ces algorithmes comparent la structure syntaxique des concepts pour résoudre le problème de subsomption de concept dans quelques logiques de description primitives. Ces algorithmes, toutefois, ne sont pas applicables pour les logiques de description plus complexes, incluant notamment la négation et la disjonction.
Le deuxième groupe contient des algorithmes appelés algorithmes de tableaux. Les algorithmes de tableaux ont été donnés la première fois par [**?**] et sont aujourd'hui l'outil principal pour la résolution des problèmes de satisfaisabilité et de subsomption de concept dans les logiques de description.
Dans ce qui suit nous décrivons brièvement ces deux catégories d'algorithmes.

Raisonnement basé sur la comparaison structurelle.
Les algorithmes de raisonnement basé sur la comparaison structurelle ne sont applicables que sur des logiques primitives de type $\mathcal{FL}^-$ (i.e. équivalent à $\mathcal{ALC}$ sans la négation des concepts atomiques). Les algorithmes du calcul de subsomption sont basés sur la comparaison structurelle entre les expressions de concepts. L'idée de cette approche est que si deux expressions de concept sont faites de sous-expressions, alors elles peuvent être comparées séparément en comparant une sous-expression d'un concept avec toutes celles de l'autre.
Afin de vérifier la subsomption dans les logiques $\mathcal{FL}^-$ l'algorithme s'exécute en deux phases. Premièrement, les concepts sont réécrits dans une forme normale (i.e. déplier les concepts et factoriser les rôles), et ensuite leurs structures sont comparées :

- Toutes les conjonctions emboîtées sont égalisées : $A \sqcap (B \sqcap C) \Leftrightarrow A \sqcap B \sqcap C$.
  Toutes les conjonctions de quantifications universelles sont factorisées : $\forall R.C \sqcap \forall R.D \Leftrightarrow \forall (C \sqcap D)$.
  Les concepts réécrits sont logiquement équivalents avec les précédents, donc la subsomption est préservée par cette transformation.

- Soient $C = C_1 \sqcap C_2 \sqcap \ldots \sqcap C_m$ et $D = D_1 \sqcap D_2 \sqcap \ldots \sqcap D_n$, alors D subsume C si et seulement si pour chaque D_i , il existe un C_j avec :
  (1) Si D_i est un concept atomique, ou est bien de la forme $\exists R$, alors $D_i = C_j$.
  (2) Si D_i est un concept de forme $\forall R.D'$, $C_j = \forall R.C'$ (le même rôle atomique R), alors $C' \sqsubseteq D'$.

Raisonnement basé sur l'algorithme de tableaux.
Cette technique de raisonnement est basée sur les calculs de tableaux pour la logique des prédicats du premier-ordre. Les structures de tableaux obtenues en raisonnant avec un langage donné des logiques de description sont soigneusement analysées, et les

vérifications redondantes dans les tableaux sont éliminées afin de donner une limite supérieure stricte sur la complexité de la méthode.
L'idée principale du calcul de tableau est de vérifier si une formule donnée F est une conséquence logique d'une théorie donnée T. On essaye de construire, en utilisant des règles de propagation, le modèle le plus générique de T où F est faux. Si le modèle est construit avec succès, alors la réponse est NON (parce que F n'est pas une conséquence logique de T) ; si le modèle construit n'est pas un succès, alors la réponse est OUI (parce qu'il n'existe pas un modèle de T avec F faux, donc F est réellement une conséquence logique de T). Les règles de propagation viennent directement de la sémantique de constructeurs.
D'une manière générale, l'algorithme de tableaux appliqué dans une logique de description essaye de prouver la satisfaisabilité d'une expression de concept D en démontrant l'existence d'une interprétation $\mathcal{I}$ dans laquelle $D^{\mathcal{I}} \neq \emptyset$.
Cette technique de raisonnement permet aujourd'hui de proposer des algorithmes de décision de satisfaisabilité et de subsomption qui sont corrects et complets pour les langages très expressifs des logiques de description.

D.1.2.3/ COMPLEXITÉ DE L'INFÉRENCE

Le complexité du raisonnement et par conséquent, le temps de calcul et la quantité de mémoire nécessaire à sa réalisation, dépendent de la taille (i.e. quantité de données) de la base de connaissances ainsi que de son niveau d'expressivité.

Le site http ://www.cs.man.ac.uk/ẽzolin/dl/ présente un aperçu non exhaustif de la complexité du raisonnement au niveau terminologique et assertionnel en fonction de l'expressivité de la logique utilisée [**?**].

Nous présentons ci-dessous un aperçu des différentes classes de complexité ainsi qu'un tableau associant la complexité à l'expressivité logique (cf. tableau **??**).

P : la classe des problèmes de décision prenant en entrée un énoncé de problème et produisant en sortie une réponse positive ou négative (i.e. oui ou non, 0 ou 1, vrai ou faux) requiert un temps polynomial par rapport à la taille du problème pour obtenir une solution à l'aide une machine de Turing déterministe. On qualifie alors le problème de polynomial. Ce problème est de complexité $O(n^k)$, pour un certain k.

NP : la classe des problèmes qui nécessitent un temps polynomial pour trouver une solution avec une machine de Turing non déterministe. Les calculs d'une machine de Turing déterministe forment une suite, tandis que les calculs d'une machine de Turing non déterministe forment un arbre, dans lequel chaque chemin correspond à une suite de calculs possibles.

PSpace : la classe des problèmes de décision qui requièrent une quantité de mémoire polynomiale pour résoudre un problème avec une machine de Turing déterministe ou non déterministe.

ExpTime : la classe des problèmes de décision solvables par une machine de Turing déterministe en un temps exponentiel par rapport à la taille du problème.

NExpTime : la classe des problèmes de décisions solvables par une machine de Turing non-déterministe en un temps exponentiel par rapport à la taille du problème.

Type d'inférence	Langage				
	$\mathcal{ALC}$	$\mathcal{S}$	$\mathcal{SH}$ $\mathcal{SHIF}$	$\mathcal{SHOIN}$	$\mathcal{SROIQ}$
Satisfiabilité de concept	PSpace Complet	PSpace Complet	EXPTime Complet	NExpTime complet	NExpTime complet
Consistence de la ABox	PSpace Complet	-	EXPTime Complet	NExpTime complet	NExpTime complet

TABLE D.8 – Complexité du raisonnement en fonction de la logique de description et donc de son expressivité0

D.1.2.4/ LES RAISONNEURS POUR OWL

Actuellement, il existe plusieurs moteurs d'inférence, la plupart conçus pour raisonner sur les logiques de description, mais qui acceptent en entrée des fichiers OWL/RDF(S). Parmi ceux-ci, on peut citer Fact++ [1], HermiT [2], RacerPro [3], StarDog [4] et Pellet [5], etc.

	FaCT++	HermiT	RacerPro	Pellet	StarDog
Expressivité	$\mathcal{SROIQ}(\mathcal{D})$	$\mathcal{SROIQ}(\mathcal{D})$	$\mathcal{SHIQ}$	$\mathcal{SROIQ}(\mathcal{D})$	$\mathcal{SROIQ}(\mathcal{D})$
Regles	-	+ SWRL	+ SWRL	+ SWRL	+ SWRL
Language	C++	JAVA	JAVA	JAVA	JAVA
Methode	tableau	hypertableau	tableau	tableau	tableau
Raisonnement sur ABox	+	+	+	+	+

TABLE D.9 – Moteurs d'inférence

1. http ://owl.man.ac.uk/factplusplus/
2. http ://hermit-reasoner.com/
3. http ://franz.com/agraph/racer/
4. http ://stardog.com/
5. http ://clarkparsia.com/pellet/

Certains moteurs d'inférence ne peuvent raisonner qu'au niveau terminologique, alors que des moteurs comme Pellet et RacerPro permettent de raisonner aussi sur les instances de concepts. Nous les présentons ci-dessous :

- *FaCT++ (Fast Classification of Terminologies)* : est un raisonneur OWL-DL ($\mathcal{SHOIN}(\mathcal{D})$) supportant un sous ensemble de OWL 2 ($\mathcal{SHOINQ}(\mathcal{D})$). Il a été implémenté en C++ et utilise un algorithme de tableaux optimisé pour de meilleures performances.
- *HermiT* : est un raisonneur codé en JAVA, utilisant les hypertableaux. Il a une expressivité allant jusqu'à ($\mathcal{SHOINQ}(\mathcal{D})$).
- *RacerPro (Renamed Abox and Concept Expression Reasoner)* : est un raisonneur utilisant les hypertableaux calculus optimisé pour la logique de description $\mathcal{SHOIQ}$. Il peut traiter des ontologies OWL-Lite et DL, mais ignore les types définis par l'utilisateur et les concepts énumérés.
- *Pellet* : est le premier raisonneur à supporter entièrement OWL-DL. La nouvelle version, Pellet 2.0 est compatible avec le langage OWL 2 et intègre diverses techniques d'optimisation, y compris pour les nominaux, les requêtes conjonctives, etc. Pellet est codé en JAVA et fonctionne sur la base de tableaux.
- *StarDog*, est un triple store intégrant le raisonneur Pellet.

Le tableau **??** résume les propriétés de ces raisonneurs.

E

LES SCHÉMAS DE MÉTADONNÉES

Cette annexe présente une courte étude de deux schémas de gestion des métadonnées. Ces deux schémas ont été choisis pour leur importance, ce sont des schémas de référence. Le premier Dublin Core est très populaire pour la gestion de documents. Le second NewsML a été spécifiquement développé par les plus grandes agences de presses pour la gestion et la structuration des informations. Cette étude a pour objectif de déterminer, les facettes de description qui peuvent être importantes à prendre en compte afin de décrire au mieux les articles que le système doit recommander.

E.1/ LES SCHÉMAS DE MÉTADONNÉES

Divers schémas pour la gestion de métadonnées existent. Nous qualifions de schéma de métadonnées une organisation plus ou moins structurée des différents types de métadonnées permettant de qualifier une ressource. Cette organisation prend la forme de facettes, ou dimensions descriptives de la ressource. Il est possible d'en distinguer deux types, les facettes substantielles et les facettes administratives.

Les facettes substantielles portent sur le contenu de la ressource, sa substance. Elles servent à décrire l'information véhiculée par celui-ci. Les facettes administratives portent sur la ressource, mais pas sur son contenu. Elles servent à faciliter la gestion de la ressource.

Le schéma le plus connu est Dublin Core. Des schémas plus récents et plus adaptés à la gestion de news existent, par exemple NewsML. Ces schémas sont proches, mais regroupent des réalités variées. Parfois, il s'agit d'indexer les documents, d'autres fois de les annoter afin de faciliter leur indexation par d'autres systèmes et enfin, il peut aussi s'agir de les structurer. L'objectif commun est de faciliter la compréhension et donc l'accès à ces ressources.

E.1.1/ DUBLIN CORE

Dublin Core est le fruit d'un travail de concertation entre des spécialistes de l'information et la communication et des spécialistes de l'informatique. L'initiative ayant pour objectif la convergence des éléments de métadonnées. Le Dublin Core a un statut officiel au sein du W3C et de la norme ISO 23950.

Ce schéma connaît une popularité certaine du fait de sa simplicité et de son ancienneté. Dublin Core n'utilise que 15 éléments de description (i.e. facettes) pour annoter les documents a l'aide de métadonnées :

- Title : Titre principal du document
- Creator : Noms des personnes, des organisations et/ou des services à l'origine de la rédaction du document
- Subject : Mots-clefs, phrases de résumé, ou codes de classement
- Description : Résumé, table des matières, référence à une représentation graphique du contenu ou texte libre sur le contenu
- Publisher : Noms des personnes, des organisations et/ou des services à l'origine de la publication du document

- Contributor : Noms de personnes, des organisations et/ou de services qui ont contribué à l'élaboration du document
- Date : Dates d'un évènement dans le cycle de vie du document
- Type : Genre du contenu
- Format : Format du document
- Identifier : Identificateur unique URI ou numéros ISBN
- Source : Identificateurs des ressources dont dérivent le document
- Language : Identificateurs de la langue du contenu du document
- Relation : Identificateurs de ressources en lien avec le document (il existe des raffinements afin de préciser le type de lien)
- Coverage : Couverture spatiale (e.g. point géographique, pays, régions, noms de lieux) ou temporelle
- Rights : Énoncé de gestion des droits pour la source ou référence du service fournissant cette information. Droits de propriété intellectuelle, les droits d'auteur et divers droits de propriété

Aucun vocabulaire contrôlé n'est directement associé aux facettes. Il est toutefois conseillé d'utiliser des vocabulaires existants. Par exemple *Language* peut être associé au vocabulaire défini par la norme RFC 1766 [1] des codes de langue IETF [2], *Coverage* peut être associé à un thésaurus géographique comme le TGN [3].

Les facettes proposées par le schéma Dublin Core sont générales et peuvent s'appliquer à tous types de documents. Elles sont principalement administratives à l'exception de *Title*, *Subject*, *Description* et *Coverage*.

E.1.2/ NewsML-G2

Faciliter la diffusion et la recherche d'informations est une problématique importante pour les journalistes et les grandes instances de presse. Structurer l'information est essentiel afin de faciliter sa diffusion ainsi que son utilisation. Ainsi des standards ont été mis en place dans le secteur de la presse. NewsML-G2 est un standard XML proposé par l'IPTC (i.e. International Press Telecommunication Council) et utilisé par les agences de presse comme l'AFP et Thomson Reuters afin de représenter leurs informations (nouvelles /

1. http ://tools.ietf.org/html/rfc1766
2. Internet Engineering Task Force
3. http ://www.getty.edu/research/tools/vocabularies/tgn/

news) de façon structurée. Ce standard est le fruit de concertations entre les plus grandes agences européennes et américaines. Il permet de transmettre le contenu journalistique ainsi que les métadonnées qui peuvent y être associées de façon structurée.

Différents types de documents peuvent être représentés par ce langage : Texte, Image, Vidéo ainsi que des documents plus complexes, Multimédia, Infographie fixe ou animée.

Tous les documents sont structurés à partir d'un nœud principal *newsMessage* contenant un entête *header* suivie d'un *itemSet*. Un *itemSet* est une liste de *newsItem*. Les documents de type textes, images et infographies fixes ainsi que vidéos et infographie animée ne contiennent qu'un seul *newsItem* dans leur *itemSet*. Seuls les documents de type multimédia contiennent plusieurs *newsItem*.

Le nœud *newsMessage* contient des informations de base nécessaires aux outils utilisés pour la manipulation de données *XML*. Il comprend des attributs pour les déclarations d'espace de nom et d'autres informations comme l'emplacement du schéma.

L'entête contient obligatoirement la date de transmission de l'information, d'autres informations optionnelles peuvent y être placées, comme le processus de transmission utilisé.

Un *newsItem* contient des métadonnées (e.g. titre, genre, sujet, etc.) ainsi que, selon les cas, un contenu textuel, d'éventuels liens vers une ressource externe (images ou vidéos). Les différents liens permettent pour une seule ressource d'avoir accès à différents formats, tailles ou résolutions). Dans le cas d'une vidéo, il peut également y avoir des liens vers une image d'illustration ou de prévisualisation. Seuls les documents multimédia contiennent plusieurs *newsItem* dans leur *itemSet*. Le premier *newsItem* de la liste est dit *Main news item* doit proposer un contenu textuel ainsi que des liens vers les *newsItem* suivants.

Voici le contenu d'un document textuel d'exemple :

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <newsItem guid="urn:newsml:acmenews.com:20131121:US-FINANCE-FED" version="3" standard="
       NewsML-G2" standardversion="2.15" xml:lang="en-US">
3    <catalogRef href="http://www.iptc.org/std/catalog/catalog.IPTC-G2-Standards_22.xml"/>
4    <catalogRef href="http://catalog.acmenews.com/news/ANM_G2_CODES_2.xml"/>
5    <rightsInfo>
6      <copyrightHolder uri="http://www.acmenews.com/about.html#copyright">
7        <name>Acme News and Media LLC</name>
8      </copyrightHolder>
9      <copyrightNotice>(c) 2013 Copyright Acme News and Media LLC</copyrightNotice>
10   </rightsInfo>
11   <itemMeta>
12     <itemClass qcode="ninat:text"/>
13     <provider uri="http://www.acmenews.com/about/"/>
14     <versionCreated>2013-11-21T16:25:32-05:00</versionCreated>
15     <embargoed>2013-11-26T12:00:00-05:00</embargoed>
```

```
    <pubStatus qcode="stat:usable"/>
  </itemMeta>
  <contentMeta>
    <contentCreated>2013-11-21T15:21:06-05:00</contentCreated>
    <contentModified>2013-11-21T16:22:45-05:00</contentModified>
    <located qcode="geoloc:NYC">
      <name>New York, NY</name>
    </located>
    <creator uri="http://www.acmenews.com/staff/mjameson">
      <name>Meredith Jameson</name>
    </creator>
    <infoSource qcode="is:AP">
      <name>Associated Press</name>
    </infoSource>
    <language tag="en-US"/>
    <subject qcode="medtop:04000000">
      <name>economy, business and finance</name>
    </subject>
    <subject qcode="medtop:20000350">
      <name>central bank</name>
    </subject>
    <subject qcode="medtop:20000379">
      <name>money and monetary policy</name>
    </subject>
    <slugline>US-Finance-Fed</slugline>
    <headline> Fed to halt QE to avert "bubble"</headline>
  </contentMeta>
  <contentSet>
    <inlineXML contenttype="application/nitf+xml">
      <nitf xmlns="http://iptc.org/std/NITF/2006-10-18/">
        <body>
          <body.head>
            <hedline>
              <hl1>Fed to halt QE to avert "bubble"</hl1>
            </hedline>
            <byline>By Meredith Jameson, <byttl>Staff Reporter</byttl></byline>
          </body.head>
          <body.content>
            <p>(New York, NY - November 21) Et, sent luptat luptat, commy
              Nim zzriureet vendreetue modo
              dolenis ex euisis nosto et lan ullandit lum doloreet vulla
              feugiam coreet, cons eleniam il ute facin veril et aliquis ad
              minis et lor sum del iriure dit la feugiamcommy nostrud min ulla
              autpat velisl duisismodip ero dipit nit utpatum sandrer cipisim
              nit lortis augiat nulla faccum at am, quam velenis nulput la
              auguerostrud magna commolore eliquatie exerate facilis
              modiamconsed dion henisse quipit at. Ut la feu facilla feu
              faccumsan ecte modoloreet ad ex el utat.
            </p>
```

```
65                    <p>Ugiating ea feugait utat, venim velent nim quis nulluptat num
66                       Volorem inci enim dolobor eetuer sendre ercin utpatio dolorpercing
67                       Et accum nullan voluptat wisis alit dolessim zzrilla commy nonulpu
68                       tpatinis exer sequatueros adit verit am nonse exerili quismodion
69                       esto cons dolutpat, si.
70                    </p>
71                 </body.content>
72              </body>
73           </nitf>
74        </inlineXML>
75     </contentSet>
76  </newsItem>
```

Listing E.1 – Document textuel d'exemple

Les *newsItem* permettent de manipuler le contenu journalistique ainsi que les métadonnées qui lui sont associées et des informations complémentaires nécessaires à son traitement. Les *newsItem* contiennent des :

- ***itemMeta*** **:** permet de définir des métadonnées sur l'élément *newsItem*. Pour cela différentes facettes sont proposées. Ainsi la classe de l'élément (e.g. texte, image, vidéo, etc.), le fournisseur, le créateur et la date de création peuvent y être renseignés. Des informations complémentaires concernant l'état de l'élément peuvent aussi être gérées. Par exemple les consignes d'embargo, l'état de la publication ou des commentaires éditoriaux.
- ***contentMeta*** **:** permet, lui aussi, de définir des métadonnées sur l'élément *newsItem*. Les facettes proposées sont par contre propres au contenu journalistique, ainsi le titre, les sujets, le genre ainsi que la langue utilisée y sont décrits.
- ***contentSet*** **:** permet de gérer le contenu journalistique de l'élément *newsItem*. Dans l'exemple cela prend la forme de texte contenu dans des balises HTML.

Afin de décrire de façon commune et non équivoque certaines facettes de description spécifique au domaine journalistique, des vocabulaires contrôlés sont utilisés comme référence.

Ces vocabulaires contrôles sont organisés sous la forme de listes plates ou de taxonomies.

Ainsi le statut de la publication défini par le champ *pubStatus* du bloc *itemMeta* n'est pas un champ libre. Il doit contenir une des trois valeurs définies dans un vocabulaire contrôlé et partagé de statuts prédéfinis (e.g. *usable*[4] ; *withheld*[5] et *canceled*[6]).

4. http://cv.iptc.org/newscodes/pubstatusg2/usable
5. http://cv.iptc.org/newscodes/pubstatusg2/withheld
6. http://cv.iptc.org/newscodes/pubstatusg2/canceled

NewsML permet de gérer des données communes à tous types de documents. Nous ne détaillons pas ici tout ce qu'il est possible de gérer à l'aide de NewsML. Pour plus d'informations, vous pouvez vous référer à la documentation officielle [7].

Le *genre* de chacun des documents, c'est-à-dire le style de contenu est défini dans le bloc *contentMeta* de chaque *newsItem*. Un vocabulaire contrôlé contenant 47 genres différents a été défini http://cv.iptc.org/newscodes/genre/ (e.g. Retrospective, Opinion, Interview, Music, Biography, Advice, etc.). Un élément *newsItem* peut être de plusieurs genres, l'importance de chacun des genres peut être définie à l'aide d'un attribut *rank* (i.e. le rang). Le genre est d'autant plus important que le rang est faible.

Les sources *infoSource* de chacun des documents, c'est-à-dire toute personne ou organisation ayant distribué ou compilé tout ou partie des informations utilisées dans le contenu. Une propriété rôle peut être utilisée afin de spécifier si la source est à l'origine du contenu ou à l'origine de l'information. Là encore un vocabulaire ne contenant que les deux termes nécessaires a été défini. Par contre en ce qui concerne les sources aucun vocabulaire contrôlé n'est proposé, le contenu est libre.

Les mots clés *keyword* peuvent être spécifiés dans le *contentMeta* de chaque *newsItem*. Ils sont définis par NewsML-G2 comme des "termes en texte libre pouvant être utilisés pour l'indexation ou la recherche du contenu par des moteurs de recherche textuelle".

Les fournisseurs *provider* permettent de spécifier le tiers responsable de la publication du document. Il est défini dans le bloc *contentMeta* de chaque *newsItem*. Un élément fils *broader* permet de préciser l'appartenance de l'entité à une entité plus large. Un vocabulaire contrôlé permettant de gérer les fournisseurs a été défini, il en contient 55, http://cv.iptc.org/newscodes/newsprovider/.

Le rôle dans le workflow *role*, permet de définir le rôle éditorial du document. Il est défini dans le bloc *contentMeta* de chaque *newsItem*. Un vocabulaire contrôlé contenant les différents rôles possibles a été défini http://cv.iptc.org/newscodes/edrole/. Ce rôle peut évoluer au fur et à mesure des mises à jour. Ainsi un document *Alerte* (i.e. texte très court de très haute priorité) peut devenir *Urgent* (i.e. texte court sur un développement important d'un sujet majeur) puis *Lead* (i.e. récapitulatif d'un sujet majeur).

Le sujet *subject*, dans la section *contentMeta* des *newsItem* est utilisé pour spécifier les sujets des documents. Un vocabulaire contrôlé définissant une liste de sujets est disponible http://cv.iptc.org/newscodes/mediatopic/. Néanmoins ce champ n'est pas forcement rempli à l'aide de données provenant de ce vocabulaire, des mots en langage naturel peuvent aussi être utilisés. l'attribut *rank* des éléments *subject* permet d'en définir l'importance. Des attributs *type*, *why* et *how* sont aussi disponibles. *type* permet de définir le type de sujet (e.g. personne, organisation, événement, etc.). Ce champ est libre et

7. http ://www.iptc.org/site/News_Exchange_Formats/NewsML-G2/

peut contenir des mots en langage naturel, néanmoins certaines agences comme l'AFP utilisent les termes du vocabulaire contrôlé http://cv.iptc.org/newscodes/cpnature/. *why* permet de donner une raison à l'utilisation de l'élément *subject*. Trois valeurs sont possibles, *direct* permet de spécifier que les métadonnées sujet ont été extraites du contenu, *ancestor* qu'elles sont héritées de concepts associés au contenu, *inferred* qu'elles ont été obtenues par la recherche dans un thésaurus. *how* permet de spécifier comment les métadonnées *subject* ont été extraites du contenu. Trois valeurs sont possibles, *person* selon les cas. Si elles ont été extraites à la main par une personne, *assisted* si elles ont été extraites par une personne assistée par un outil et *tool* si elles ont été extraites automatiquement par un outil.

En ce qui concerne les informations spécifiques aux documents de type texte, le nombre de mots peut être spécifié.

Parmi les informations que nous ne détaillons pas, la possibilité de gérer :

- Créateurs et contributeurs
- Corrections et des notes
- Date de création du document
- Date de création de la version du document
- Date de transmission
- Date de création du contenu
- Embargo
- Publics exclus
- Identifiant et numéro de version
- Langue du contenu et des métadonnées
- Titre et Sous-Titre
- Niveau d'urgence

La majorité des facettes proposées sont administratives et non substantielles. Seuls *Subject*,*Title*, *Slugline* et *Headline* peuvent être vues comme des facettes de description substantielles des ressources. Un vocabulaire contrôlé est associé à la facette *Subject*, il est disponible sur le site de l'IPTC [8].

8. http ://cv.iptc.org/newscodes/subjectcode/

E.1.3/ SYNTHÈSE

Contrairement à Dublin Core, NewsML a aussi pour objectif de structurer l'information afin d'en faciliter la diffusion et la réutilisation. Spécialisé pour une utilisation sur des ressources du type articles d'actualité, NewsML propose une grande variété de facettes administratives adaptées, mais peu de facettes substantielles.

Parmi les facettes substantielles disponibles dans ces deux schémas, seules deux ont retenu notre attention, *Subject* et *Coverage*. Elles permettent de définir le sujet, le lieu et la temporalité de l'information véhiculée par un document. Contrairement à Dublin Core, NewsML comprend un vocabulaire contrôlé associé à la facette *Subject*, son utilisation n'est toutefois pas obligatoire. Son étude a montré qu'il n'était pas adapté à notre cas d'application. Bien que spécialisé au domaine de la gestion de l'actualité, il est trop général pour être utilisé directement dans le cadre spécifique de la gestion de l'actualité économique.

F

ENQUÊTES UTILISATEURS

Cette annexe présente trois enquêtes ayant eu lieu aux différentes étapes du projet. Cette annexe ne montre pas l'intégralité des résultats des différentes enquêtes. Seuls les résultats ayant un rapport direct avec le projet sont présentés dans cette annexe.

F.1/ RÉSULTATS DES ENQUÊTES

Cette annexe présente une partie des résultats de trois enquêtes ayant eu lieu aux différentes étapes du projet. La première enquête, l'enquête préliminaire, montre le comportement de lecture des utilisateurs, leur intérêt pour une revue personnalisée ainsi que les critères de personnalisation qu'ils proposent et/ou valident. La seconde enquête porte sur l'évaluation par un groupe restreint d'utilisateurs d'un prototype de la solution finale. La troisième enquête, dont une partie des résultats ne nous est pas encore parvenue, porte sur l'évaluation de la solution commercialisée. Cette annexe ne montre pas l'intégralité des résultats des différentes enquêtes. Seuls les résultats ayant un rapport direct avec le projet sont présentés dans cette annexe.

F.1.1/ ENQUÊTE PRÉLIMINAIRE 2011

Cette première enquête a eu lieu au début du projet afin de mieux comprendre les intérêts et besoins des utilisateurs. Cette enquête a été envoyée à :

- 4635 lecteurs abonnés, dont 513 ont répondu, soit un taux de réponse des clients de 11,07%
- 4745 lecteurs prospects, dont 97 ont répondu, soit un taux de réponse des prospects de 2%

F.1.1.1/ TEMPS DE LECTURE

Le lectorat, composé principalement de décideurs, dispose de peu temps à consacrer à la recherche d'informations. C'est d'ailleurs pour cette raison que les revues First ECO – qui synthétisent l'actualité économique régionale – l'intéressent. Les revues personnalisées permettraient de franchir un cap supplémentaire en offrant aux lecteurs la possibilité de sélectionner ce qui les concerne personnellement, et donc de gagner du temps.

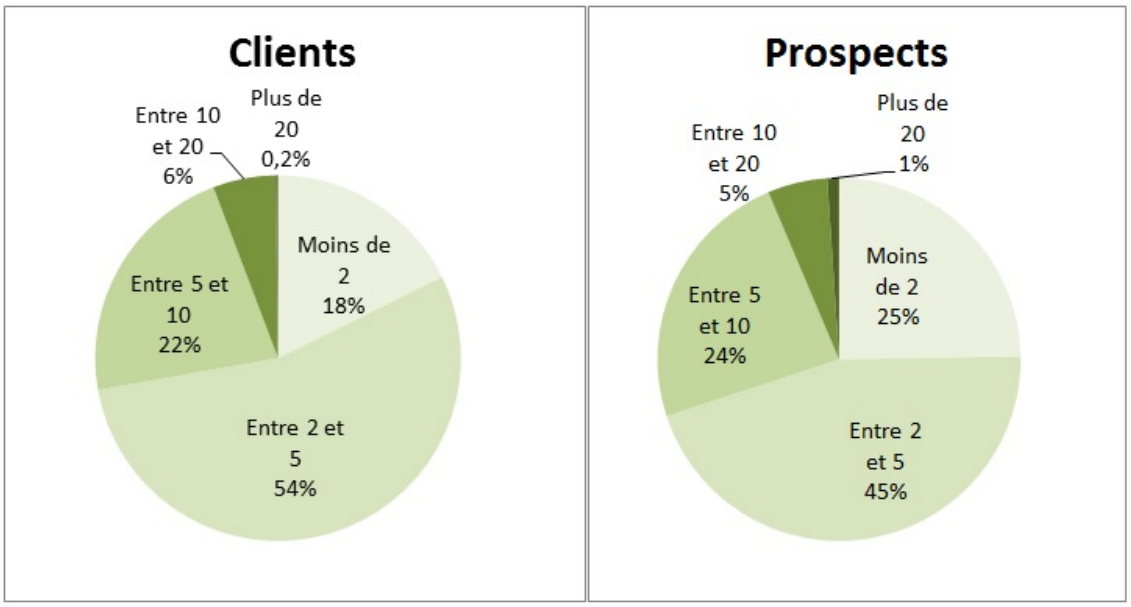


FIGURE F.1 – Réponses à la question : EN MOYENNE, COMBIEN DE TEMPS CONSACREZ-VOUS A LA LECTURE D'UNE EDITION CHAQUE JOUR ? (CHOIX UNIQUE)

F.1.1.2/ NOMBRE D'ARTICLES LUS

Les lecteurs sont actuellement contraints de repérer les articles qui les intéressent au sein de nos revues. Outre un gain de temps, les revues personnalisées leur apporteraient un confort de lecture accru.

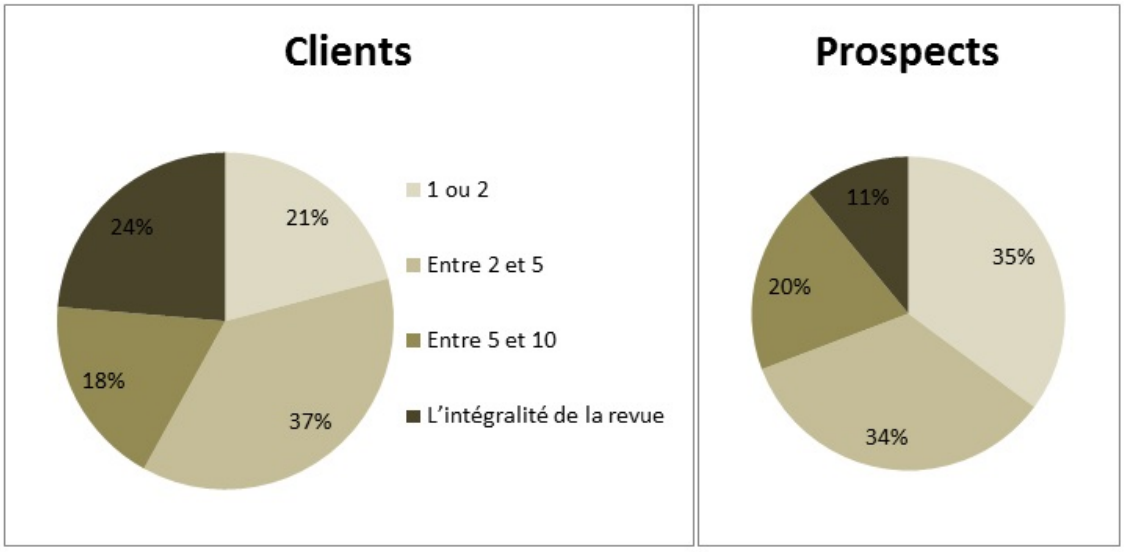


FIGURE F.2 – Réponses à la question : EN MOYENNE, COMBIEN LISEZ-VOUS D'ARTICLES PAR JOUR ? (CHOIX UNIQUE)

F.1.1.3/ COMPORTEMENT DE LECTURE

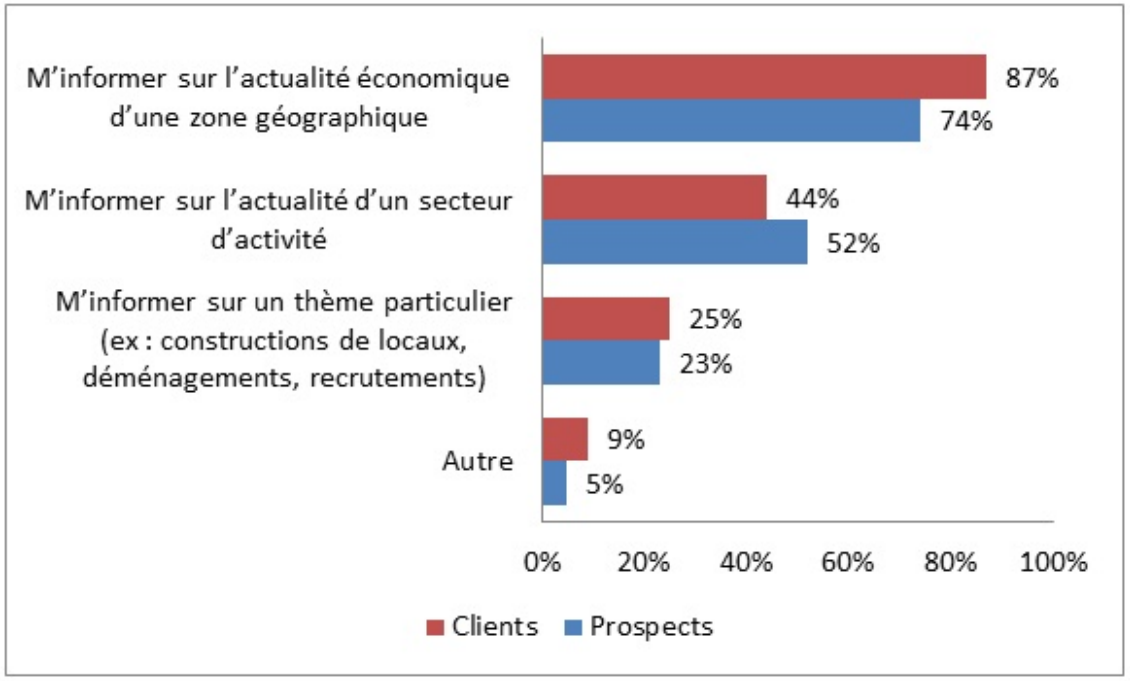


FIGURE F.3 – Réponses à la question : A QUELLES(S) FIN(S) LISEZ-VOUS FIRST ECO ? (CHOIX MULTIPLES - REPONSE OBLIGATOIRE)

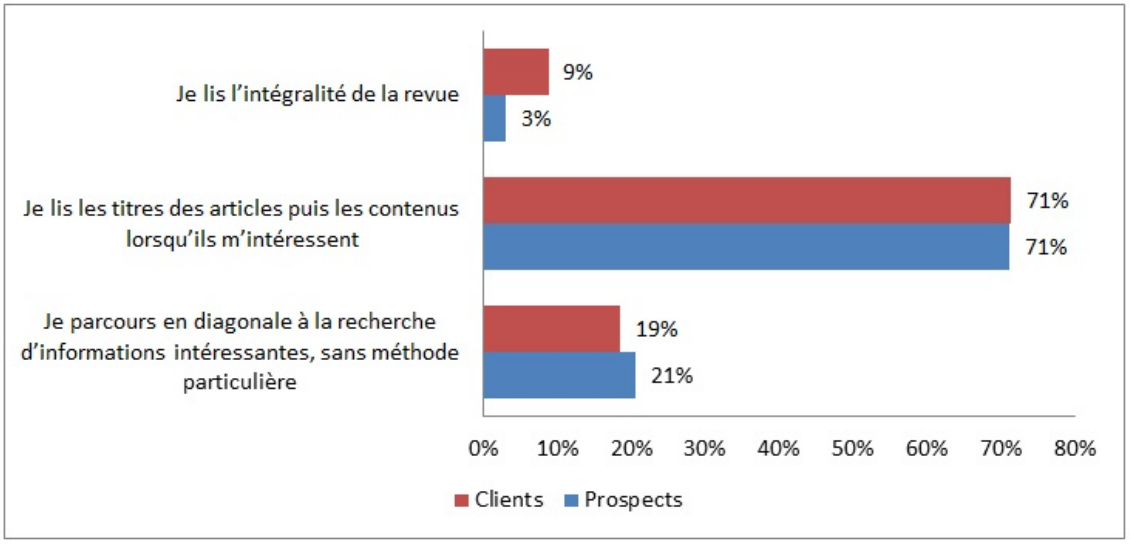


FIGURE F.4 – Réponses à la question : LORS DE VOTRE CONSULTATION DE FIRST ECO, QUELLES SONT VOS HABITUDES DE LECTURE ? (CHOIX UNIQUE)

F.1.1.4/ INTÉRÊT POUR LE SERVICE DE REVUES PERSONNALISÉES

Présentation du nouveau service telle qu'elle a été donnée aux utilisateurs : First ECO projette de mettre en place un service de revues personnalisées. Vous pourriez ainsi recevoir l'information qui vous intéresse, en fonction de critères que vous choisiriez :

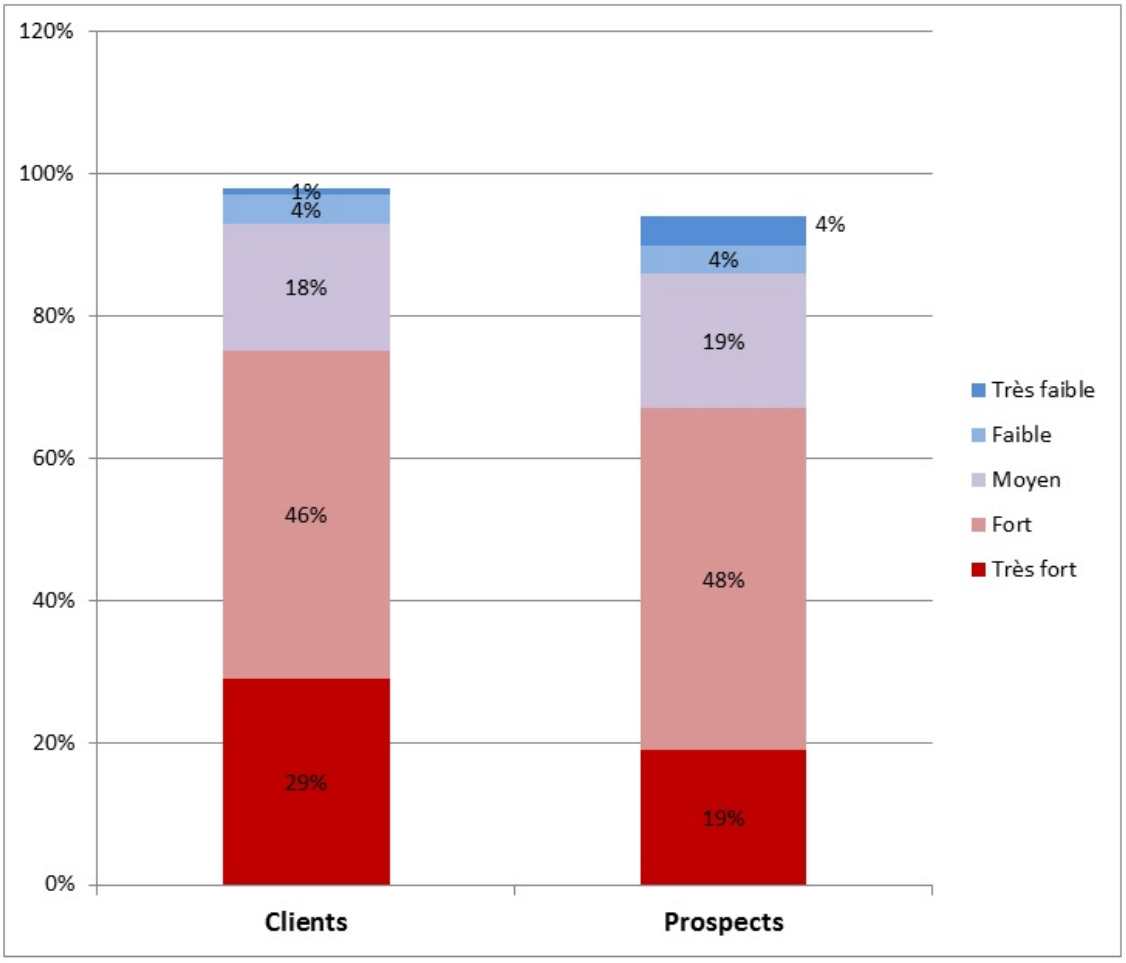


FIGURE F.5 – Réponses à la question : COMMENT QUALIFIERIEZ-VOUS VOTRE INTERET POUR LE SERVICE DE REVUES PERSONNALISEES ? (CHOIX UNIQUE)

75% des clients ayant répondu à l'enquête ont un intérêt **fort** ou **très fort** vis-à-vis d'un service de revue personnalisée. Par ailleurs, le nouveau service devrait permettre d'augmenter le taux de transformation des prospects en clients, dans la mesure où près de 70% de ces derniers manifestent également un intérêt **fort** ou **très fort**.

F.1.1.5/ CRITÈRES DE PERSONNALISATION

Les critères proposés par les clients sont principalement les secteurs économiques et des événement économiques. Mais d'autres réponses sont intéressantes, par exemple :

- Taille des entreprises concernées
- Surface des bâtiments concernés
- Montant des projets
- Echéance des projets
- Mots-clés
- Noms d'entreprises

F.1.2/ ENQUÊTE SUR PROTOTYPE 2013

Cette seconde enquête ayant pour objectif de recueillir les sentiments des testeurs sur notre futur produit, fait suite à une phase de tests auprès de 120 testeurs. Ces tests ont eu lieu du 16 mai au 27 juin 2013, durant cette période une revue personnalisée était produite, 2 fois par semaine. En date du 9 juillet, un quart des sondés (i.e. 30 sur 120) avaient rempli ce questionnaire.

Les profils créés sont basés sur :

- la zone géographique à surveiller
- les secteurs d'activités préférés
- les thèmes d'articles préférés

Au total, 120 personnes sont profilées et participent au test :

- 80% de lecteurs abonnés et 20% d'anciens abonnés ou anciens prospects
- 85% de multi-éditions classiques et 15% de mono-édition

F.1.2.1/ AVIS GLOBAL SUR LE CLASSEMENT AUTOMATIQUE DES ARTICLES ET LA QUALITÉ DE L'APPLICATION

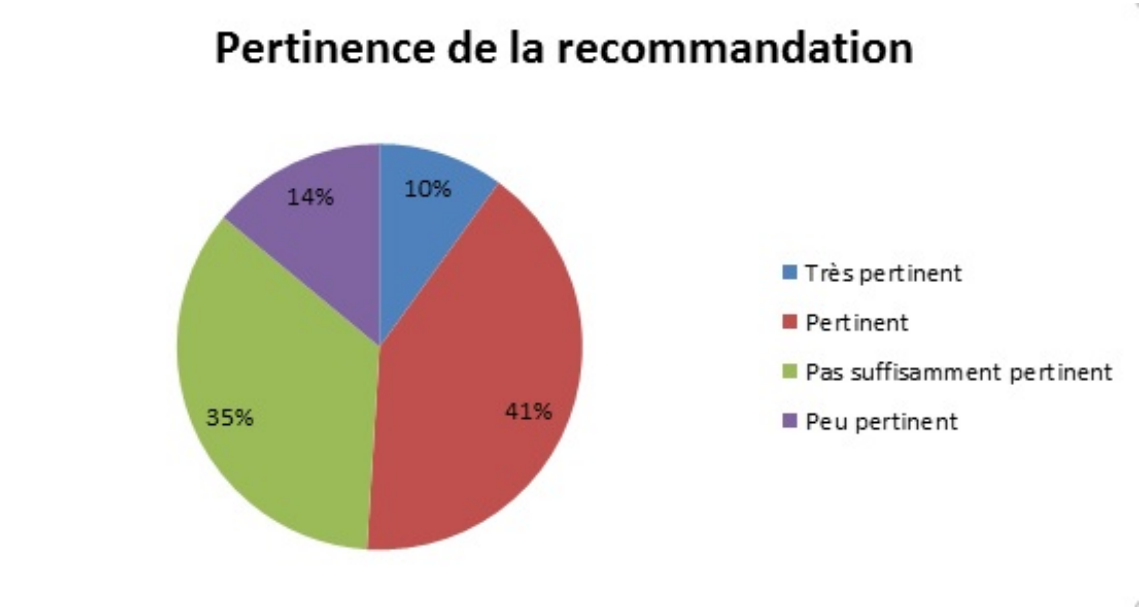


FIGURE F.6 – Réponses à la question : GLOBALEMENT, LE CLASSEMENT DE VOS ARtICLES EST : ? (CHOIX UNIQUE)

Très pertinent : Les articles qui m'intéressent s'affichent tous avant les autres.
Pertinent : La plupart des articles qui m'intéressent s'affichent avant les autres.
Pas suffisamment pertinent : Plusieurs articles qui m'intéressent s'affichent après des articles qui ne m'intéressent pas.
Peu Pertinent : Beaucoup d'articles qui m'intéressent s'affichent après des articles qui ne m'intéressent pas.

FIGURE F.7 – Réponses à la question : QUE PENSEZ VOUS DE L'ASPECT PRATIQUE DE L'APPIICATION ? (CHOIX UNIQUE)

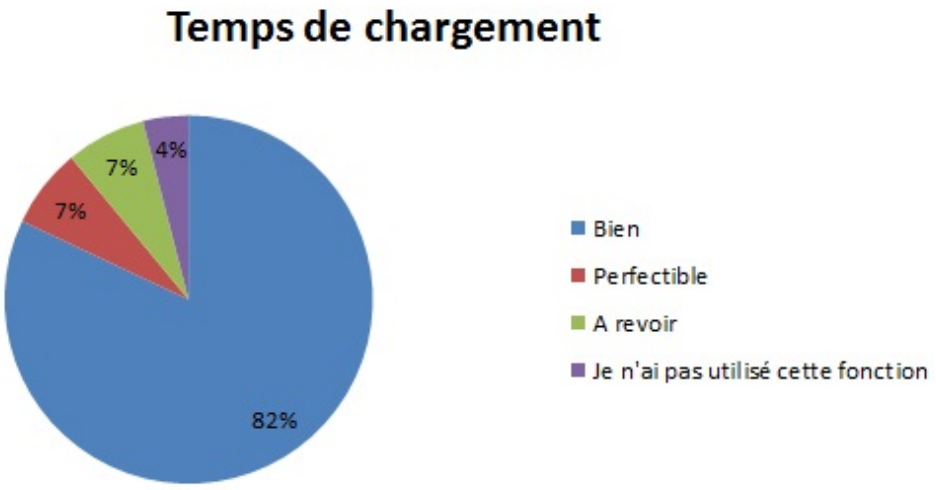


FIGURE F.8 – Réponses à la question : QUE PENSEZ VOUS DU TEMPS DE CHARGEMENT DE L'APPLICATION ? (CHOIX UNIQUE)

F.1.2.2/ PROPOSITION DE CRITÈRES PERMETTANT L'AMÉLIORATION DE LA RECOMMANDATION

Critère de tri	Indispensable	Utile	Inutile
Effectif du site concerné par l'article (ex : moins de 20, 20 à 249, entre 250 et 4999, plus de 5000)	18%	67%	15%
Effectif total de la société (ex : moins de 20, 20 à 249, entre 250 et 4999, plus de 5000)	21%	57%	21%
Etat d'avancement du projet (ex : achevé, en cours, à venir)	68%	32%	0%
Activité du site concerné par l'article (ex : bureaux, locaux industriels, entrepôts, surfaces commerciales)	70%	30%	0%

FIGURE F.9 – Réponses à la question : JUGEZ DE L'INTERET DES CRITERES SUIVANTS

Quels autres critères pourraient améliorer le classement de vos articles ?

- Chiffre d'affaires du site concerné et chiffre d'affaires total
- Code NAF
- Superficie du projet
- Type de projet (construction neuve, extension, réhabilitation, déménagement, etc.)
- Montant de l'investissement

- Nationalité de l'investisseur
- Cibler les déménagements d'entreprises, les constructions de sièges sociaux, les fins de bail, les prises de locations de locaux, les achats de locaux, etc.

Le chiffre d'affaires rejoint la problématique de la taille du site / de la société. Il y a une occurrence pour chacune des réponses listées.

F.1.3/ ENQUÊTE 2015, PHASE QUALITATIVE

Nous ne sommes ici en mesure de ne présenter qu'un faible nombre de résultats. Seule la première phase, la phase qualitative, de l'enquête est achevée au moment de l'écriture de ce mémoire. Une enquête quantitative, qui devrait nous apporter plus d'informations, est en cours.

Les entretiens individuels

Nous avons réalisé 12 entretiens individuels par téléphone dans toute la France.
Chaque entretien a duré entre 22 et 40 minutes.

Période de réalisation

Les entretiens ont eu lieu du 15 décembre au 19 janvier 2015.

Quota	Abonnement	Conversion	Réachat	Degré d'aisance avec l'informatique	Type d'usage
1	First Eco 8 régions	non converti Pro'fil	abonné depuis plusieurs années		
2	First Eco 8 régions	non converti Pro'fil	abonné < 2 ans		
3	First Eco 8 régions	en cours de conversion Pro'fil	abonné depuis plusieurs années		
4	First Eco 8 régions	en cours de conversion Pro'fil	abonné < 2 ans		
5	First Eco Pro'fil	abonné Pro'fil depuis 6 mois		faible	
6	First Eco Pro'fil	abonné Pro'fil depuis 6 mois		élevé ou usage croissant	
7	First Eco Pro'fil	converti Pro'fil < 6 mois		faible	mono-utilisateur
8	First Eco Pro'fil	converti Pro'fil < 6 mois		faible	multi-utilisateurs
9	First Eco Pro'fil	converti Pro'fil < 6 mois		élevé ou usage croissant	mono-utilisateur
10	First Eco Pro'fil	converti Pro'fil < 6 mois		élevé ou usage croissant	multi-utilisateurs
11	First Eco Pro'fil	prospect			mono-utilisateur
12	First Eco Pro'fil	prospect			multi-utilisateurs

FIGURE F.10 – Présentation du panel, partie 1

Les douze participants se répartissent de la manière suivante :

Taille des établissements	Nombre de participants à l'étude
Moins de 10 salariés	2
11 à 50 salariés	5
50 salariés et plus	5

Les secteurs représentés sont les suivants :

Industrie : agitateur industriel pour pétrole AA, Applicateur de revêtements Spéciaux, fournitures industrielles	3
Audit Conseil, expertise comptable, commissariat aux comptes	3
Banque, affacturage et crédit bail	2
chauffage clim dans le bâtiment	1
intérim-recrutement	1
prestataire de services en ingénierie physique pour des entreprises	1
Service de communication extérieure	1

FIGURE F.11 – Présentation du panel, partie 2

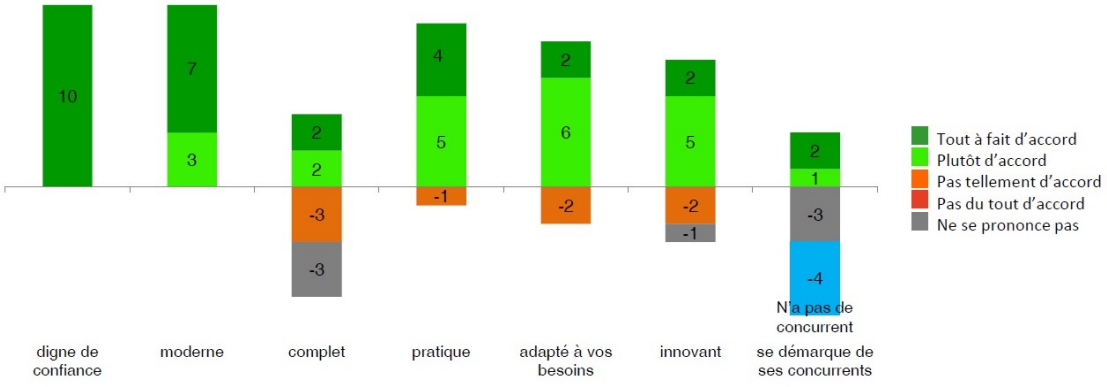


FIGURE F.12 – Degré d'accord avec les affirmations

G

COMPLÉMENT D'INFORMATIONS SUR L'ONTOLOGIE NOYAU

Cette annexe présente plus en détails l'ontologie noyau utilisée dans le chapitre **??** et développe les raisons de sa simplification. La modélisation utilisée permet de limiter la complexité du raisonnement afin d'accélérer les traitements réalisés par les raisonneurs.

G.1/ EXPLICATIONS SUR L'ONTOLOGIE NOYAU

Cette annexe explique la simplification de l'ontologie noyau utilisée dans le chapitre **??**. L'indexation automatique de documents réalisée par un processus d'inférence logique nécessite l'intégration dans l'ontologie noyau d'un modèle prédictif. Afin de permettre son intégration, le modèle prédictif doit être traduit sous forme de contraintes logiques. La façon de créer ces contraintes dépend directement de la structuration des connaissances dans la base de connaissances, c'est-à-dire de la modélisation de l'ontologie noyau. En fonction de cette modélisation, la complexité des traitements devant être réalisés afin de permettre l'indexation automatique des documents ne sera donc pas la même. Cette modélisation influence directement le temps ainsi que la quantité de mémoire nécessaire aux calculs. Cette simplification a donc pour objectif de limiter la complexité du raisonnement afin d'accélérer les traitements réalisés par les raisonneurs.

Nous présentons ci-dessous premièrement la modélisation sémantiquement idéale, puis la modélisation implémentée par souci de performance.

G.1.1/ MODÉLISATION IDÉALE DE L'ONTOLOGIE NOYAU

La figure **??** présente l'ontologie noyau dans sa modélisation idéale. Dans cette modélisation, une instance de concept Item est associée à une et une seule instance du concept Label, via la relation fonctionnelle isAbout. Cette instance est ensuite classée dans différents Concepts spécialisant le concept Label, ce qui permet l'indexation de l'item.

Cette modélisation est sémantiquement correcte. En effet, l'instance classée est une instance de Label et cette instance est classée dans des concepts spécialisant le concept Label. Label, Item et Feature sont des concepts distincts.

Dans notre exemple, le concept Label est spécialisé en LabelEco lui-même spécialisé en deux concepts, Automotive (i.e. Automobile) et Textile. La relation hasFeature permet de faire le lien entre une instance d'Item et une instance de Feature. Les instances de Feature sont par exemple des mots-clés indices (i.e. KeyWord dans l'exemple) apparaissant dans le contenu de l'item. Dans notre exemple, le document Document_1 contient de mot indice "Cotton" (i.e. Coton).

Sur la base de cette modélisation, il est possible de traduire un modèle prédictif pour l'indexation des items à l'aide de règles SWRL. Par exemple :

Item(?y), Label(?x), hasFeature(?y, Cotton), isAbout(?y, ?x) → Textile(?x)

Cette règle permet de classer comme étant une instance du concept Textile, les instances du concept Label en relation avec une instance du concept Item ayant parmi ses mots-clés indices (i.e. instances de Feature) l'instance "Cotton". Ainsi, tout document contenant

le terme indice coton sera indexé comme traitant de l'industrie du textile. Dans notre exemple, l'instance du concept Label, Document_1_Label, en relation avec l'instance Document_1 est une instance du concept Textile. Car, Document_1 possède la Feature Cotton.

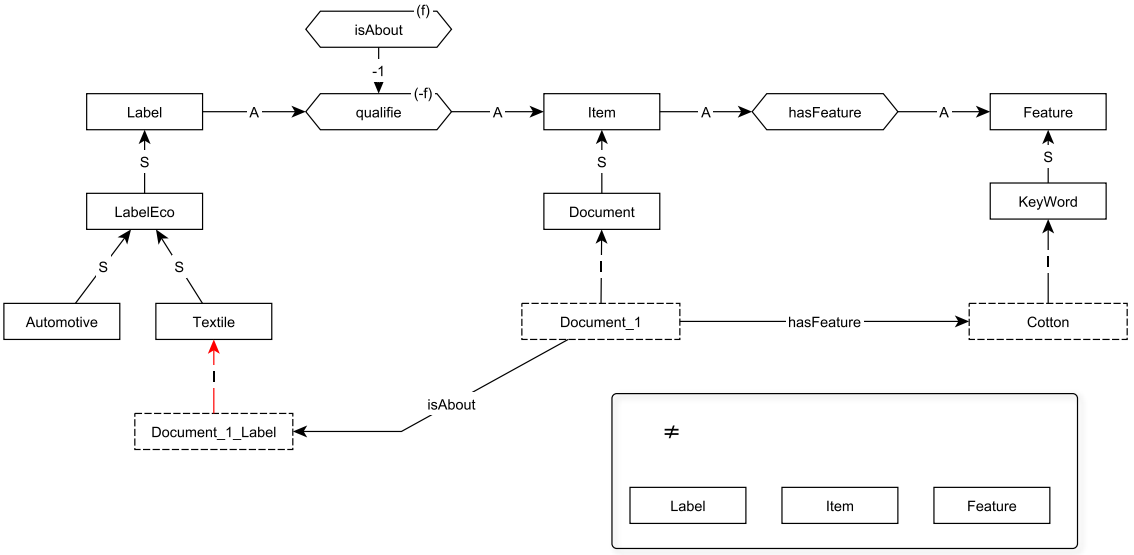


FIGURE G.1 – Schéma de l'ontologie noyau idéal au format G-OWL

Mais notre objectif est l'intégration du modèle prédictif dans l'ontologie sous forme de contraintes logiques, au format OWL. La solution n'est donc pas l'utilisation de règles SWRL, mais l'utilisation de compositions de rôles et de contraintes d'équivalences. Par exemple :

Textile ≡ Label ∧ (qualifie ∘ hasFeature some { Cotton })

Cette contrainte d'équivalence spécifie que toute instance de Label étant en relation (via le rôle "qualifie") avec une instance d'Item, lui-même en relation (via "hasFeature") avec au moins une instance contenue dans la liste d'instances ne contenant que l'instance "Cotton", sera considérée comme une instance de Textile.

La figure **??** présente le résultat du raisonnement. Le résultat est le même avec la règle SWRL qu'avec la contrainte d'équivalence. Ce qui a été inféré par les raisonneurs est en rouge. Cette solution est fonctionnelle, mais le raisonnement est long et complexe pour les raisonneurs. L'expressivité de la description logique utilisée est $\mathcal{ALCOIN}(o)$ ($\mathcal{ALCOI}$(o) sans la restriction de nombre nécessaire aux règles de type B). Cela implique une complexité de raisonnement NExpTime-difficile [1]. Afin de simplifier le raisonnement nous avons implémenté une ontologie noyau simplifiée présentée dans la section suivante.

1. La complexité a été évalué à l'aide de l'outil suivant : http ://www.cs.man.ac.uk/ ezolin/dl/

G.1.2/ MODÉLISATION DE L'ONTOLOGIE NOYAU IMPLÉMENTÉE

La modélisation implémentée est proposée dans la figure **??**. Dans cette modélisation, c'est l'instance du concept Item qui est directement classée en tant qu'instance d'une spécialisation du concept Label.

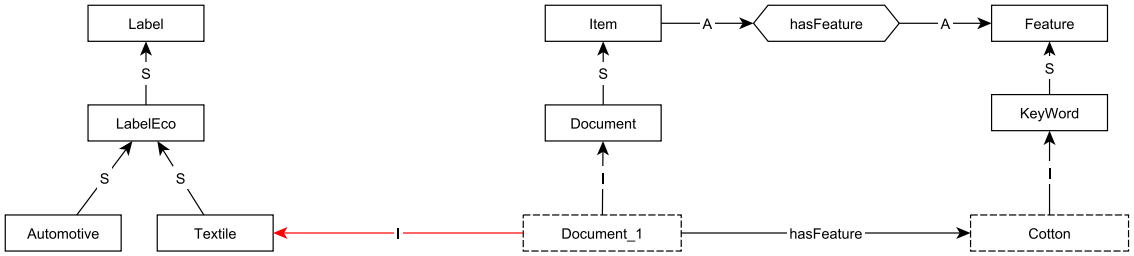


FIGURE G.2 – Schéma de l'ontologie noyau implémenté au format G-OWL

La figure **??** présente le résultat du raisonnement avec la contrainte d'équivalence suivante :

Textile ≡ hasFeature some { Cotton }

Cette contrainte correspond à une règle de type A dans le chapitre **??**. Elle spécifie que toute instance, en relation (via hasFeature) avec au moins une instance de la liste contenant uniquement l'instance Cotton, est une instance du concept Textile. Dans notre exemple, Document_1 est une instance du concept Document, spécialisation du concept Item. Suite au raisonnement, cette instance est classée en tant qu'instance du concept Textile. Textile est une spécialisation du concept LabelEco qui est une spécialisation du concept Label. Textile désigne donc un secteur industriel. Il n'est pas sémantiquement juste de dire qu'un document est un secteur économique, c'est pourquoi nous qualifions ce modèle de non idéal. C'est un modèle simplifié afin d'optimiser le raisonnement.

L'expressivité de la description logique utilisée est $\mathcal{ALCON}$ ($\mathcal{ALCO}$ sans la restriction de nombre nécessaire aux règles de type B). Cela implique une complexité de raisonnement PSpace-complet, c'est-à-dire une complexité plus faible que la complexité NExpTime-difficile de la modélisation idéale. Avec la modélisation implémentée, nous passons d'un problème qui peut être résolu par une machine de Turing non déterministe en un temps exponentiel à un problème qui peut être résolu par une machine de Turing déterministe dans un espace polynomial.

Cette modélisation permet donc de rendre le travail des raisonneurs plus simple et donc plus rapide, sans pour autant amoindrir la qualité du modèle prédictif.

Résumé :

La gestion efficace de grandes quantités d'informations est devenue un défi de plus en plus important pour les systèmes d'information. Tous les jours, de nouvelles sources d'informations émergent sur le web. Un humain peut assez facilement retrouver ce qu'il cherche, lorsqu'il s'agit d'un article, d'une vidéo, d'un artiste précis. En revanche, il devient assez difficile, voire impossible, d'avoir une démarche exploratoire pour découvrir de nouveaux contenus. Les systèmes de recommandation sont des outils logiciels ayant pour objectif d'assister l'humain afin de répondre au problème de surcharge d'informations. Les travaux présentés dans ce document proposent une architecture pour la recommandation efficace d'articles d'actualité. L'approche ontologique utilisée repose sur un modèle permettant une qualification précise des items sur la base d'un vocabulaire contrôlé. Contenu dans une ontologie, ce vocabulaire constitue une modélisation formelle de la vue métier sur le domaine traité. Réalisés en collaboration avec la société Actualis SARL, ces travaux ont permis la commercialisation d'un nouveau produit hautement compétitif, *FristECO Pro'fil*.

Mots-clés : ontologie, base de connaissances, systèmes de recommandation, raisonneur, actualités, économie, sémantique

Abstract:

Effective management of large amounts of information has become a challenge increasingly important for information systems. Everyday, new information sources emerge on the web. Someone can easily find what he wants if (s)he seeks an article, a video or a specific artist. However, it becomes quite difficult, even impossible, to have an exploratory approach to discover new content. Recommender systems are software tools that aim to assist humans to deal with information overload. The work presented in this Phd thesis proposes an architecture for efficient recommendation of news. In this document, we propose an architecture for efficient recommendation of news articles. Our ontological approach relies on a model for precise characterization of items based on a controlled vocabulary. The ontology contains a formal vocabulary modeling a view on the domain knowledge. Carried out in collaboration with the company Actualis SARL, this work has led to the marketing of a new highly competitive product, *FristECO Pro'fil*.

Keywords: ontology, knowledge base, recommender systems, reasoner, news, economy, semantic

SPIM

■ École doctorale SPIM - Université de Bourgogne/UFR ST BP 47870 F - 21078 Dijon cedex

■ tél. +33 (0)3 80 39 59 10 ■ ed-spim@univ-fcomte.fr ■ www.ed-spim.univ-fcomte.fr

UNIVERSITÉ DE BOURGOGNE