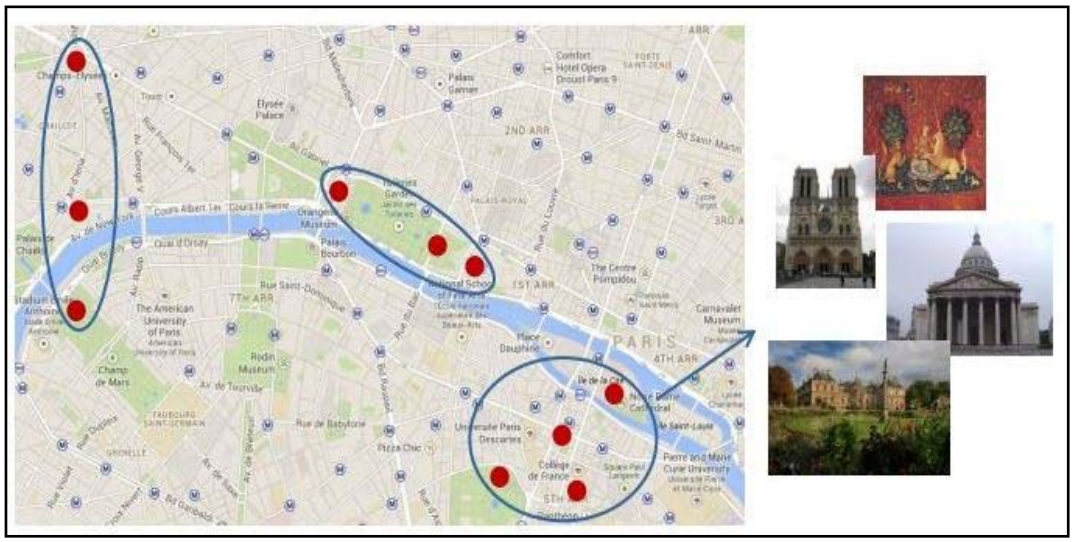
Un système de recommandation contextuel et composite pour la visite personnalisée de sites culturels

Idir Benouaret

Par **Idir BENOUARET**

Un système de recommandation contextuel et composite pour la visite personnalisée de sites culturels

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC



Soutenue le 25 janvier 2017
Spécialité : Technologies de l'Information et des Systèmes :
Unité de recherche Heudyasic (UMR-7253)

D2332

Un système de recommandation contextuel et composite pour la visite personnalisée de sites culturels.

Idir Benouaret

Thèse présentée pour l'obtention du grade
de Docteur de l'UTC

Soutenue le 25 Janvier 2017 devant un jury composé de :

Rapporteurs :

Max Chevalier
Professeur des universités
Université Paul Sabatier Toulouse

Serge Garlatti
Professeur
Telecom Bretagne

Examinateurs :

Elsa Negre
Maître de conférences
Université de Paris-Dauphine

Sebastien Destercke
Chercheur CNRS
Université de Technologie de Compiègne

Philippe Trigano
Professeur des universités
Université de Technologie de Compiègne

Directeur de Thèse :

Dominique Lenne
Professeur des universités
Université de Technologie de Compiègne

Université de Technologie de Compiègne

Laboratoire Heudiasyc UMR CNRS 7253

`Technologies de l'Information et des Systèmes`

Table des matières

Liste des figures

Résumé

Notre travail concerne les systèmes d'aide à la visite de musée et l'accès au patrimoine culturel. L'objectif est de concevoir des systèmes de recommandation, implémentés sur dispositifs mobiles, pour améliorer l'expérience du visiteur, en lui recommandant les items les plus pertinents et en l'aidant à personnaliser son parcours. Nous considérons essentiellement deux terrains d'application : la visite de musées et le tourisme. Nous proposons une approche de recommandation hybride et sensible au contexte qui utilise trois méthodes différentes : démographique, sémantique et collaborative. Chaque méthode est adaptée à une étape spécifique de la visite de musée. L'approche démographique est tout d'abord utilisée afin de résoudre le problème du démarrage à froid. L'approche sémantique est ensuite activée pour recommander à l'utilisateur des œuvres sémantiquement proches de celles qu'il a appréciées. Enfin l'approche collaborative est utilisée pour recommander à l'utilisateur des œuvres que les utilisateurs qui lui sont similaires ont aimées. La prise en compte du contexte de l'utilisateur se fait à l'aide d'un post-filtrage contextuel, qui permet la génération d'un parcours personnalisé dépendant des œuvres qui ont été recommandées et qui prend en compte des informations contextuelles de l'utilisateur à savoir : l'environnement physique, la localisation ainsi que le temps de visite.

Dans le domaine du tourisme, les points d'intérêt à recommander peuvent être de différents types (monument, parc, musée, etc.). La nature hétérogène de ces points d'intérêt nous a poussé à proposer un système de recommandation composite. Chaque recommandation est une liste de points d'intérêt, organisés sous forme de packages, pouvant constituer un parcours de l'utilisateur. L'objectif est alors de recommander les Top-k packages parmi ceux qui satisfont les contraintes de l'utilisateur (temps et coût de visite par exemple). Nous définissons une fonction de score qui évalue la qualité d'un package suivant trois critères : l'appréciation estimée de l'utilisateur, la popularité des points d'intérêt ainsi que la diversité du package et nous proposons un algorithme inspiré de la recherche composite pour construire la liste des packages recommandés. L'évaluation expérimentale du système que nous avons proposé, en utilisant un data-set réel extrait de Tripadvisor démontre sa qualité et sa capacité à améliorer à la fois la précision et la diversité des recommandations.

Mots Clés : *Systèmes de recommandation, web sémantique, similarité, diversité, sensibilité au contexte, musées, tourisme.*

Abstract

Our work concerns systems that help users during museum visits and access to cultural heritage. Our goal is to design recommender systems, implemented in mobile devices to improve the experience of the visitor, by recommending him the most relevant items and helping him to personalize the tour he makes. We consider two mainly domains of application : museum visits and tourism. We propose a context-aware hybrid recommender system which uses three different methods : demographic, semantic and collaborative. Every method is adapted to a specific step of the museum tour. First, the demographic approach is used to solve the problem of the cold start. The semantic approach is then activated to recommend to the user artworks that are semantically related to those that the user appreciated. Finally, the collaborative approach is used to recommend to the user artworks that users with similar preferences have appreciated. We used a contextual post filtering to generate personalized museum routes depending on artworks which were recommended and contextual information of the user namely : the physical environment, the location as well as the duration of the visit. In the tourism field, the items to be recommended can be of various types (monuments, parks, museums, etc.). Because of the heterogeneous nature of these points of interest, we proposed a composite recommender system. Every recommendation is a list of points of interest that are organized in a package, where each package may constitute a tour for the user. The objective is to recommend the Top-k packages among those who satisfy the constraints of the user (time, cost, etc.). We define a scoring function which estimates the quality of a package according to three criteria : the estimated appreciation of the user, the popularity of points of interest as well as the diversity of packages. We propose an algorithm inspired by composite retrieval to build the list of recommended packages. The experimental evaluation of the system we proposed using a real world data set crawled from Tripadvisor demonstrates its quality and its ability to improve both the relevance and the diversity of recommendations.

Keywords : *recommender systems, semantic web, similarity, diversity,context-awareness, museum, tourism.*

Remerciements

En premier lieu, je tiens à remercier mon directeur de thèse Dominique Lenne pour m'avoir donné l'occasion de faire mes premiers pas dans le monde de la recherche. Je le remercie de m'avoir fait confiance et de m'avoir choisi pour effectuer cette thèse. Je le remercie pour sa présence et ses conseils tout au long de ces années. Je remercie ensuite le laboratoire Heudiasyc et plus particulièrement l'équipe ICI de m'avoir accueilli à bras ouverts. Je remercie aussi la région Picardie d'avoir financé mes travaux. La soutenance a été le moment le plus stressant de cette thèse mais aussi le plus agréable : je remercie chacun des membres du jury pour leur présence, leur déplacement, pour l'attention qu'ils ont portée à mon travail et mon manuscrit, pour leurs remarques et leurs questions pertinentes et enrichissantes. En particulier, je remercie Max Chevalier et Serge Garlatti d'avoir rapporté mes travaux. Leurs analyses et commentaires ont été d'une aide précieuse. Je remercie également Elsa Negre et Sebastien Destercke d'avoir accepté d'examiner mes travaux ainsi que pour toutes les questions et remarques qu'ils m'ont adressées. Je remercie Philippe Trigano d'avoir accepté d'être le président de mon jury. Je pense aussi à tous les gens formidables avec qui j'ai pu collaborer, échanger et partager au sein du laboratoire Heudiasyc et au-delà. Je vais commencer par les plus anciens. Merci Kevin d'avoir partagé avec moi le bureau dès le premier jour de ma thèse, tu as été mon premier soutien, tu essayais toujours de me remonter le moral je ne me suis jamais senti seul au bureau. Lucile, tu as été mon coup de cœur pendant cette thèse, j'ai adoré t'écouter râler pendant trois années, on s'est soutenu, on s'est boosté, comme toi j'ai aussi été très content de te retrouver à mon bureau après un an de séparation ! ! Par contre je déteste quand tu tires sur tes oreilles, faut que t'arrêtes de faire ça. Juliette, on ne s'est pas trop vu car tu ne venais pas souvent à Compiègne mais je voulais te dire que tu es très agréable, calme et souriante, j'espère que tu vas revenir une autre fois en Algérie et que j'y serai. Merci aussi à Remy, Taha, Sohaib, Lotfi, Kaci. Maintenant passons aux plus récents : Mélody avec ton rire époustouflant, Azzeddine, Remy le nouveau, Lauriane, Florian, Yann, etc. J'espère n'avoir oublié personne. Merci à tous les potes compiégnois non informaticiens : Khalil, Raouf, Amine. J'aimerais aussi remercier Freddy, je le redis une autre fois ici, tu es l'une des plus belles rencontres que j'ai faite de ma vie, c'est avec un immense plaisir

que j'ai partagé le bureau avec toi. Merci aussi à Ayyoub, Youcef, Rahim et Subeer pour leur bonne humeur et pour toutes les soirées jeux et sorties. Je tiens aussi à remercier mes amis d'enfance Sofiane, Takfa, Amazigh, vous êtes vraiment des amis formidables. Je ne peux terminer mes remerciements sans parler des quatre personnes que j'aime le plus dans ma vie. Tout d'abord celle qui m'a mis au monde. Maman, tu es la lumière de ma vie, ma très chère mère, je ne peux oublier tout ce que tu as fait pour moi, que dieu te garde à mes cotés. A mon père, mon équilibre, mon tout, sans qui chaque pas de ma vie n'aurait été possible, j'adore ton calme, ta gentillesse, je n'oublierai jamais à quel point tu as bossé dur juste pour nous. Mon grand frère adoré, je t'aime tellement, merci pour ton aide dans tous les domaines de la vie et aussi scientifiquement, pour l'amour que tu me portes, souvent on se comprend même sans se parler. Ma grande sœur, je n'ai pas les mots pour décrire à quel point tu es importante dans ma vie, je ne peux pas vivre sans toi, tu es tout pour moi, je t'aime tellement.

Introduction

Motivations et orientations

Le cadre de ce travail concerne l'aide à la visite culturelle, plus particulièrement la visite de musées et le tourisme. Ces domaines sont particulièrement touchés par le problème de la surcharge d'information car ils renferment un volume d'information assez conséquent. A titre d'exemple, le musée du Louvre comprend près de 460000 œuvres différentes à disposition du grand public. Le secteur du tourisme est aussi confronté à cette problématique, le site web Tripadvisor recense plus de 15000 points d'intérêt et lieux d'activité pour la ville de Paris. De ce fait, les visiteurs sont confrontés à plusieurs problèmes : ils sont submergés par le nombre très important de choix possibles dans l'espace qu'ils explorent. L'exploitation de cette longue liste d'options est très complexe pour les visiteurs, qui doivent passer beaucoup de temps pour selectionner les options qui correspondent le plus à leurs intérêts. De plus, les visiteurs ne savent pas forcément ce qu'ils devraient voir ou ce qu'ils pourraient apprécier, le parcours qu'ils font n'est alors en général pas réfléchi, ou bien ils se limitent à voir les items les plus populaires comme dans la plupart des visites guidées. En conséquence, ils peuvent perdre du temps en regardant des œuvres ou en visitant des points d'intérêt qui ne les intéressent pas. Inversement, ils peuvent manquer des œuvres ou des points d'intérêt qui auraient pu les intéresser.

Un des domaines de recherche principaux relatifs à la problématique de la surcharge d'information est le domaine de la recherche d'information. Le principe général est d'élaborer des méthodes et des algorithmes afin de rechercher des ressources (par exemple, des pages web, des films et dans notre cadre d'application des œuvres ou des points d'intérêt) en fonction de requêtes formulées par des utilisateurs. Il n'est cependant pas toujours évident pour un utilisateur de savoir comment exprimer sa demande. De plus, sa requête correspond généralement à une quantité importante de ressources et il est difficile de savoir quels résultats lui présenter en premier, d'autant plus que d'un utilisateur à un autre, l'ordre de priorité peut changer.

Un autre domaine de recherche relatif à cette problématique est le domaine des systèmes de recommandation. Ces systèmes sont capables de fournir des recomman-

dations adaptées aux préférences et aux besoins des utilisateurs. Ils se sont avérés être très satisfaisants pour aider les utilisateurs à accéder aux ressources désirées dans un temps limité. Initialement conçus pour la recommandation de ressources web, films, etc. les systèmes de recommandation sont devenus de plus en plus populaires et sont aujourd'hui un composant principal de beaucoup d'applications dans différents domaines. Un avantage très conséquent des systèmes de recommandation est que l'utilisateur n'a pas besoin de formuler de requêtes. Sa seule requête est implicite, elle peut se traduire par : "Quelles sont les ressources qui correspondent à mes préférences, mes besoins et mes contraintes ? ". Les systèmes de recommandation peuvent être classées en deux types d'approches : les approches basées sur le contenu et les approches basées sur le filtrage collaboratif [Adomavicius and Tuzhilin, 2005]. Les recommandations basées sur le contenu sont effectuées en identifiant les ressources similaires à celles appréciées par un utilisateur en fonction de leur contenu. Les approches basées sur le filtrage collaboratif, quant à elles, permettent de fournir des recommandations à un utilisateur sans forcément considérer les contenu des ressources, mais en se basant sur l'analyse du comportement et/ou des appréciations de l'utilisateur afin de recommander les ressources qui ont été appréciées par d'autres utilisateurs ayant des goûts similaires.

Nos travaux de recherche concernent les systèmes de recommandation dans le but d'offrir aux visiteurs des parcours personnalisés.

Problématique

Notre travail vise à concevoir des systèmes, sur dispositif mobile, afin aider à améliorer l'expérience d'un visiteur. Nous considérons deux types de situation : la visite de musées et la visite touristique (tourisme). Dans les deux cas, il s'agit d'adapter et de personnaliser la visite de l'utilisateur, en fonction de ses préférences, son contexte et ses contraintes. Nos travaux s'orientent donc vers les systèmes de recommandation qui se sont avérés être très efficaces pour aider les utilisateurs à accéder aux ressources par lesquels il seraient potentiellement intéressés.

Pour proposer un système de recommandation dans le cadre de la visite de musées ou du tourisme, il faut faire face à de nombreux problèmes. Certains sont génériques et concernent n'importe quel système de recommandation : démarrage à froid, sur-spécialisation, parcimonie.

- Démarrage à froid : le problème du démarrage à froid est très fréquent dans les systèmes de recommandation. Ce problème est double, il affecte à la fois les utilisateurs mais aussi les items. Il désigne le manque d'information sur un

utilisateur ou sur un item qui vient d'être ajouté. Lorsqu'un nouvel utilisateur entre dans le système, on ne sait pas forcément quelles sont ses préférences et cela rend difficile la tâche de recommandation. Dans notre domaine d'application, le démarrage à froid affecte beaucoup plus les utilisateurs que les items. En effet, il n'est pas très fréquent qu'une nouvelle œuvre soit ajoutée à un musée ou bien qu'un nouveau point d'intérêt (ex. monument) s'ajoute à une ville.

- Sur-spécialisation : lorsque le système ne peut recommander à un utilisateur que des items qui sont en relation avec son profil, l'utilisateur est limité aux recommandations de ressources qui sont similaires à celles qu'il a déjà aimées. En revanche, la diversité des recommandations est souvent une caractéristique souhaitable pour les systèmes de recommandation [Yu et al., 2009].

- Parcimonie : Pour les systèmes de recommandation de type filtrage collaboratif, le nombre de notes déjà obtenues est généralement très faible par rapport au nombre de notes qui doivent être prédites. Un item qui a alors reçu peu d'avis de la part des utilisateurs a moins de chances d'être recommandé par rapport aux autres. Aussi, pour un utilisateur qui a noté des items qui n'ont pas reçu beaucoup d'avis, il est difficile de lui trouver des utilisateurs similaires, du coup il sera difficile de faire des recommandations pertinentes pour cet utilisateur.

D'autres problématiques sont plus spécifiques à notre domaine d'application. En effet, pour la visite de musées d'autres problématiques s'ajoutent. Un visiteur visite un musée donné généralement une seule fois et ne sait pas forcément quelles oeuvres il va aimer ou ne pas aimer dans le musée. Sa visite se passe alors généralement en trois étapes. Premièrement, à son arrivée au musée, le visiteur ne sait pas ce qu'il devrait voir. Ensuite, il commence à découvrir les oeuvres qu'il préfère dans le musée. Finalement, il commence à affiner ses préférences, son profil est plus riche et ses préférences sont plus précises. Une problématique est alors de concevoir un système de recommandation qui prend en compte ces trois étapes de visite et de définir quelle approche est adaptée à quelle étape. Aussi, il est important de pouvoir proposer un parcours de visite à l'utilisateur, c'est-à-dire une liste d'œuvres à contempler successivement dans un ordre défini. Aussi, l'utilisateur dispose généralement d'un temps limité pour faire sa visite, il est alors nécessaire de pouvoir recommander un parcours, qui aide le visiteur à se déplacer de manière efficace et ne soit pas contraint à d'incessants aller-retours, ce qui lui ferait perdre du temps et réduirait sa satisfaction. La prise en compte des informations contextuelles est alors nécessaire dans notre cadre de travail.

Une problématique supplémentaire pour le tourisme est que les points d'intérêt à recommander sont généralement de différents types, par exemple : monument, parc, musée, etc. La nature hétérogène de ces points d'intérêt rend inexploitable les systèmes de recommandation classiques qui fournissent à l'utilisateur des recommandations sous forme de listes triées. Là aussi, d'autres contraintes entrent en jeu. En effet, à chaque point d'intérêt visité, correspond un temps et un budget consommés. L'utilisateur dispose d'un temps de visite et d'un budget limités pour faire sa visite. Il faut donc être en mesure de pouvoir recommander à l'utilisateur des parcours respectant ses contraintes.

Approche proposée

Nous avons proposé une approche de recommandation hybride et sensible au contexte pour la visite de musées. Notre approche combine trois méthodes différentes : démographique, sémantique et collaborative, chaque méthode étant adaptée à une étape spécifique de la visite. Premièrement, l'approche démographique est utilisée pour résoudre le problème du démarrage à froid, des informations démographiques sur l'utilisateur sont récupérées pendant son authentification et sont utilisées pour lui recommander une première liste d'œuvres pour lesquelles il peut exprimer son avis. L'approche sémantique est ensuite activée pour recommander à l'utilisateur des œuvres sémantiquement proches de celles qu'il a appréciées. Cette deuxième méthode utilise le modèle sémantique des œuvres que nous avons proposé, afin de pouvoir mesurer des similarités sémantiques entre les œuvres de notre base de connaissances. Enfin, l'approche collaborative est activée pour recommander à l'utilisateur des œuvres que les utilisateurs qui lui sont similaires ont aimées. Nous avons proposé un calcul de similarité entre deux utilisateurs qui dépend de la similarité sémantique des œuvres qu'ils ont consultées. La prise en compte du contexte se fait *a posteriori* en utilisant un post-filtrage contextuel, pour la génération d'un parcours dépendant de la position des œuvres dans le musée, de la localisation de l'utilisateur ainsi que du temps de visite.

Pour le tourisme, nous proposons un système de recommandation composite pour faire face à la nature hétérogène des points d'intérêt. Nous organisons donc les recommandations sous forme de packages, chaque package étant constitué de plusieurs points d'intérêt. Nous avons défini une fonction de score qui évalue la qualité d'un package suivant trois critères : l'appréciation estimée de l'utilisateur, la popularité ainsi que la diversité du package. Nous avons proposé un algorithme inspiré de la recherche composite pour construire la liste des packages à recommander parmi ceux qui satisfont les contraintes de l'utilisateur (temps et coût de visite).

L'évaluation expérimentale du système que nous avons proposé, en utilisant un jeu de données réel extrait de Tripadvisor démontre sa qualité et sa capacité à améliorer à la fois la précision et la diversité des recommandations.

Liste des publications

Les travaux réalisés pendant cette thèse ont mené à ces publications :

[1] **Idir Benouaret**, Dominique Lenne : A Package Recommendation Framework for Trip Planning Activities. RecSys '16 Proceedings of the 10th ACM Conference on Recommender Systems. [Benouaret and Lenne, 2016b]

[2] **Idir Benouaret**, Dominique Lenne : A Composite Recommendation System for Planning Tourist Visits. IEEE/WIC/ACM International Conference on Web Intelligence 2016. [Benouaret and Lenne, 2016a]

[3] **Idir Benouaret**, Dominique Lenne : Personalizing the Museum Experience through Context-Aware Recommendations. SMC 2015. [Benouaret and Lenne, 2015b]

[4] **Idir Benouaret**, Dominique Lenne : Combining Semantic and Collaborative Recommendations to Generate Personalized Museum Tours. ADBIS 2015 (Short Papers and Workshops). [Benouaret and Lenne, 2015a]

[5] **Idir Benouaret** : Un système de recommandation sensible au contexte pour la visite de musée. CORIA-RJCRI 2015. [Benouaret, 2015]

[6] **Idir Benouaret**, Dominique Lenne : Recommending Diverse and Personalized Travel Packages. En cours de soumission à DEXA 2017. [Benouaret and Lenne, 2017]

Organisation du manuscrit

Le manuscrit est organisé de la façon suivante :

- Chapitre 1 **Les systèmes de recommandation**

  Ce chapitre est consacré à la présentation des concepts de base des systèmes de recommandation. Nous présentons les deux principales approches de recommandation à savoir les approches basées sur le contenu et le filtrage collaboratif. Pour chaque approche, les principales techniques utilisées sont décrites ainsi que leurs avantages et inconvénients. Les différentes approches d'hybridation combinant plusieurs approches de recommandation sont également présentées.

- Chapitre 2 **Représentation des connaissances et similarités sémantiques**

  Ce chapitre décrit la représentation formelle des connaissances ainsi que les différentes techniques qui sont utilisées pour cela. Nous présentons brièvement les langages RDF, RDFS et OWL et les principales mesures de similarités sémantiques.

- Chapitre 3 **Systèmes de recommandation pour l'aide à la visite culturelle**

  Ce chapitre est un état de l'art des travaux qui utilisent des systèmes de recommandation pour la visite de musées et pour le tourisme.

- Chapitre 4 **Approche hybride et contextuelle pour la visite de musée**

  Ce chapitre décrit notre système de recommandation hybride pour la visite de musée, qui combine trois approches de recommandation (démographique, sémantique et collaborative). Chacune des trois approches est adaptée à une étape spécifique de la visite. La génération de parcours de visite se fait grâce à la prise en compte du contexte de l'utilisateur en utilisant un post-filtrage contextuel.

- Chapitre 5 **Un système de recommandation composite pour la planification d'activités touristiques**

  Ce chapitre décrit notre système de recommandation composite pour le tourisme. Nous décrivons ici notre modèle ainsi que notre algorithme pour la recommandation des packages.

- Chapitre 6 **Implémentation et évaluation**

  Ce chapitre décrit l'implémentation que nous avons réalisée ainsi que des éléments de validation de notre approche.

Les Systèmes de recommandation

Sommaire

1.1 Introduction

Fréquemment, nous sommes confrontés à faire des choix. Comment se vêtir ? Quel film regarder ? Quel article acheter ? Que visiter lorsque l'on est en voyage ? La taille de ces domaines de décision est très souvent grande. Par exemple, Netflix disposait en 2007 de plus de 17,000 films dans sa base de données [Bennett and Lanning, 2007], et ce nombre ne cesse de croitre au fil des années. La liste des possibilités qui s'offrent à nous est donc en général de très grande taille, l'évaluation de ces possibilités pour trouver ce qui nous convient le plus est une tâche difficile et peut consommer beaucoup de notre temps. Les systèmes de recommandation sont apparus dans le début des années 1990 pour répondre à ce problème de surcharge d'information et de choix. C'est un besoin similaire à celui des moteurs de recherche (Ex. Google) mais différent dans sa conception. Un moteur de recherche reçoit une requête de la part de l'utilisateur, en général sous forme de texte, et fournit une liste ordonnée d'éléments (pages web, images, vidéos...) dans le but de permettre à l'utilisateur d'accéder rapidement à un contenu considéré comme pertinent par le système par rapport à sa recherche parmi le très grand nombre d'informations disponibles sur Internet.

Figure 1.1 – Interface d'Amazon

A l'inverse, un système de recommandation ne reçoit pas de requête directe de la part de l'utilisateur, mais doit lui proposer de nouvelles possibilités en apprenant ses préférences à partir de son comportement passé. Un système de recommandation doit donc avoir accès à un historique des données qui peut être sous plusieurs formes : des notes, des achats, des clics sur des pages web, des historiques de navigation... A partir de ces informations, le système de recommandation sera en mesure d'adapter la réponse à l'utilisateur.

Les systèmes de recommandation deviennent indispensables dans de nombreux domaines, notamment dans les domaines des industries culturelles. On peut notamment citer :

- Le cinéma (Netflix, Movielens),
- le e-commerce (Amazon.com),
- la musique (lastFM),
- le tourisme (Tripadvisor.com),
- la vidéo à la demande (Youtube.com).

Ces domaines d'application sont différents mais partagent tous un même problème : ils offrent un choix de possibilités trop important aux utilisateurs, chacun d'eux ayant par ailleurs ses propres préférences. Dès lors, un moteur de recommandation qui permet de proposer à un utilisateur donné, de manière personnalisée, le sous-ensemble d'éléments qui l'intéresse, devient très utile. Il réduit de manière considérable l'effort que doit fournir l'utilisateur pour accéder à ce qui l'intéresse et participe ainsi à sa satisfaction et à sa fidélisation.

Un système de recommandation doit mettre en relation deux entités : des utilisateurs et des items. Les items peuvent être de diverses natures (films, vidéos, restaurants, lieux d'activité...). Les informations qui permettent de relier ces deux entités sont elles-mêmes de natures différentes : notes, achats, clics, historiques, etc. Les systèmes de recommandation se focalisent majoritairement sur l'utilisation des "notes" [Adomavicius and Tuzhilin, 2005]. La note est généralement située sur une échelle graduée, et elle permet à l'utilisateur d'exprimer un avis positif ou négatif sur l'item qu'il considère. Au vu des notes qu'il a déjà exprimées sur un ensemble d'items, la tâche d'un système de recommandation a été principalement étudiée de deux façons. La première est la prédiction de notes qui a pour objectif de prédire les notes qu'un utilisateur donnerait aux items qu'il n'a pas encore notés. Le seconde est la recommandation d'une liste ordonnée d'items. Ainsi les items en haut de liste sont donc ceux que le système prédit comme pertinent pour l'utilisateur.

1.2 Généralités

1.2.1 Histoire des Systèmes de Recommandation

La capacité des ordinateurs pour faire des recommandations à des utilisateurs a été reconnue assez tôt dans l'histoire de l'informatique. Grundy [Rich, 1979], un système bibliothécaire, était une première étape vers des systèmes de recommandation automatiques. Ce système était assez primitif. Il classait les utilisateurs en "stéréotypes" en se basant sur une courte interview, et utilisait ces stéréotypes pour produire des recommandations de livres. Ce travail constituait une première tentative intéressante dans le domaine des systèmes de recommandation. Cependant, son utilisation est restée très limitée.

Au début des années 1990, le filtrage collaboratif apparait comme une solution pour faire face à la surcharge d'information. L'année 1992 voit l'apparition du système de recommandation de documents Tapestry [Goldberg et al., 1992], ainsi que la création du laboratoire de recherche GroupLens, qui travaille explicitement sur le problème de la recommandation automatique dans le cadre des forums de news de Usenet. Tapestry avait pour but de recommander à des groupes d'utilisateurs des documents issus des newsgroups susceptibles de les intéresser. L'approche utilisée était de type "plus proches voisins" à partir de l'historique de l'utilisateur. On parle alors de filtrage collaboratif manuel, comme une réponse au besoin d'outils pour le filtrage de l'information énoncé à la même époque. La recommandation résulte d'une action collaborative des utilisateurs qui recommandent à d'autres utilisateurs des documents en leur attribuant des notes d'intérêt selon certains critères. Les systèmes

de filtrage collaboratif automatiques apparaissent ensuite. GroupLens [Resnick et al., 1994] utilise cette technique pour identifier les articles de Usenet susceptibles d'être intéressants pour un utilisateur donné. Les utilisateurs doivent seulement attribuer des notes ou effectuer d'autres opérations observables (par exemple, lire un article) ; le système combine alors ces données avec les notes ou les actions d'autres utilisateurs pour fournir des résultats personnalisés. Avec ces systèmes, les utilisateurs n'ont aucune connaissance directe des opinions des autres utilisateurs, ni des articles présents dans le système.

Au cours de ces dernières années, les systèmes de recommandation deviennent un sujet d'un intérêt croissant dans les domaines de l'interaction homme-machine, de l'apprentissage automatique ainsi que la recherche d'information. En 1995 apparaissent successivement Ringo [Shardanand and Maes, 1995a], un système de recommandation de musique, basé sur les appréciations des utilisateurs et Bellcore [Hill et al., 1995], un système de recommandation de vidéos. La même année, GroupLens crée la société Net Perceptions dont le premier client a été Amazon. De nos jours, les systèmes de recommandation sont devenus des composantes incontournables pour la plupart des sites du e-commerce.

1.2.2 Netflix Challenge

La recherche sur les algorithmes de recommandation a suscité beaucoup d'intérêt en 2006 quand Netflix a lancé le prix Netflix pour améliorer les approches courantes de la recommandation de films. Netflix est alors une entreprise américaine qui offre un service de location de DVD en ligne. Chaque client peut, après avoir visionné un film, donner son avis sur ce dernier. Il peut attribuer une note comprise entre un et cinq au film. Netflix disposait, avant la compétition, d'un système de recommandation, CineMatch, permettant de suggérer aux clients un certain nombre de films qu'ils seraient susceptibles d'aimer. Jugeant qu'une bonne recommandation était un moyen efficace pour, à la fois, fidéliser sa clientèle et augmenter son chiffre d'affaires, Netflix a cherché à améliorer son moteur de recommandation. L'objectif du concours était de construire un algorithme de recommandation qui pourrait surpasser CineMatch de 10 % dans les tests. Le concours a suscité beaucoup d'intérêt, tant dans le milieu de la recherche que dans celui des amateurs de films. Netflix proposait une somme d'un million de dollars au vainqueur du challenge, attestant ainsi de la valeur que les vendeurs et sociétés peuvent placer dans la fourniture de recommandations précises.

Presque trois ans après son lancement, le défi a finalement été remporté par l'équipe "BellKor's Pragmatic Chaos". Ils ont proposé une solution qui consiste en une hybridation de plus de cent modèles, tous relativement simples à mettre en

œuvre. Les solutions proposées sont discutées dans plusieurs articles, notamment [Koren, 2009], [Piotte and Chabbert, 2009] et [Töscher et al., 2009]. Cependant, cet agrégat de modèles s'est avéré difficile à mettre en production car beaucoup trop coûteux en temps de calcul et en mémoire. Ce challenge a permis en revanche de mettre en évidence l'intérêt des méthodes de factorisation pour la résolution de problèmes de recommandation, notamment grâce à l'idée d'introduire des informations complémentaires telles que des évaluations implicites, des effets temporels et des niveaux de confiance [Bell and Koren, 2007].

1.2.3 Définitions, Terminologies et Notations

L'objectif d'un système de recommandation est de fournir à l'utilisateur des objets pertinents selon ses préférences. Il permet de réduire de manière considérable le temps que l'utilisateur met pour chercher les objets les plus intéressants pour lui, et aussi de trouver des objets qu'il est susceptible d'aimer mais auxquels il n'aurait pas forcément fait attention.

Les systèmes de recommandation ont été définis de plusieurs façons. La définition la plus populaire et la plus générale que nous citons ici est celle de *Robin Burke* [Burke, 2002] que nous avons traduite ainsi :

Système de recommandation : *Système capable de fournir des recommandations personnalisées ou permettant de guider l'utilisateur vers des ressources intéressantes ou utiles au sein d'un espace de données important.*

Le domaine d'information pour un système de recommandation de manière générale consiste en une liste d'*utilisateurs* qui ont exprimé leurs préférences pour divers *items*. Comme on l'a vu précédemment, une préférence exprimée par un utilisateur pour un item est appelée *note*, et est souvent représentée par un triplet $(utilisateur, item, note)$. Ces notes peuvent prendre différentes formes. Cependant, la majorité des systèmes utilisent des notes sous formes d'une échelle de 1 à 5, ou bien des notes binaires (j'aime/je n'aime pas). L'ensemble des triplets $(utilisateur, item, note)$ forme ce que l'on appelle la matrice des notes. Les paires $(utilisateur, item)$ où l'utilisateur n'a pas donné de note pour l'item sont des valeurs non connues dans la matrice. Le tableau 1.1 illustre un exemple d'une matrice de notes pour 4 utilisateurs et 4 films. Les valeurs marquées " ? " indiquent que l'utilisateur n'a pas donné d'avis.

Un système de recommandation se focalise sur deux tâches. La première tâche est la prédiction : étant donné un utilisateur et un item, quelle serait la préférence de l'utilisateur pour cet item ? En d'autres termes, le système doit prédire la valeur des notes marquées " ? ". La deuxième tâche est la recommandation : étant donné

	Inception	Batman begins	Titanic	Star wars
User A	4	3	2	4
User B	?	4	5	5
User C	2	2	4	?
User D	3	?	5	2

Tableau 1.1 – Exemple d'une matrice de notes

un utilisateur, quelle liste ordonnée de n recommandations peut-on lui suggérer ? On parle alors de liste Top-n. A noter que la liste des Top-n recommandations n'est pas forcément la liste des n items avec les plus hautes valeurs de prédiction. La prédiction des notes n'est pas le seul critère utilisé pour produire une liste de recommandations. En effet, un algorithme de recommandation peut utiliser d'autres critères, tels que le contexte. Nous présentons en détail les systèmes de recommandation sensibles au contexte dans la section 1.6.

1.2.4 Notations

Dans ce chapitre, nous utiliserons les notations suivantes concernant les différents éléments du modèle d'un système de recommandation : On définit U l'ensemble des utilisateurs du système, et I l'ensemble des items. Les notes des utilisateurs sont stockées dans une matrice $R \in \mathbb{R}^{|U|} \times \mathbb{R}^{|I|}$. I_u est l'ensemble des items notés par l'utilisateur u et U_i est l'ensemble des utilisateurs ayant notés l'item i. On désigne la note que l'utilisateur u a donné à l'item i par $r_{u,i}$.

1.2.5 Classification des systèmes de recommandation

Comme nous l'avons dit, un nombre important de travaux de recherche ont traité la problématique de la recommandation au cours des dernières années. Ces travaux sont issus de plusieurs domaines comme le Machine Learning, les statistiques et surtout la recherche d'information.

Les techniques de recommandation peuvent être classées de différentes manières. Parfois plusieurs termes sont utilisés pour désigner une même méthode ou approche. L'objectif ici est de s'appuyer sur les classifications les plus connues sur lesquelles nous basons notre étude.

La classification la plus utilisée est une classification selon deux approches : les recommandations basées sur le contenu et le filtrage collaboratif [Shahabi et al., 2001, Adomavicius and Tuzhilin, 2005]. En plus de ces deux approches, *Robin Burke* [Burke, 2007] propose de considérer trois autres approches : la recommandation

basée sur les données démographiques, la recommandation basée sur la connaissance (knowledge-based) et la recommandation basée sur l'utilité (utility-based). Mais il note que ces trois approches sont des cas particuliers des approches classiques.

Nous présentons dans la suite les approches basées contenu et le filtrage collaboratif, puis les approches hybrides et enfin, les systèmes de recommandation sensibles au contexte.

1.3 Les approches basées sur le contenu

La recommandation basée sur le contenu consiste à analyser le contenu des items candidats à la recommandation ou les descriptions de ces items. Les méthodes de recommandation basées sur le contenu utilisent des techniques largement inspirées du domaine de la recherche d'information. La différence se trouve essentiellement dans l'absence de requêtes explicites formulées par l'utilisateur. Les approches basées contenu infèrent plutôt les préférences de l'utilisateur et lui recommandent les items dont le contenu est similaire au contenu des items qu'il a aimés auparavant [Balabanović and Shoham, 1997, Adomavicius and Tuzhilin, 2005, Zhang et al., 2002, Pazzani and Billsus, 2007]. Ainsi, quand de nouveaux items sont introduits dans le système, il peuvent être recommandés directement, sans que cela ne nécessite un temps d'intégration comme c'est le cas pour les systèmes de recommandation basés sur une approche de filtrage collaboratif (Cf. section 1.4).

1.3.1 Approche générale

Pour recommander des items en se basant sur le contenu, deux ensembles doivent être constitués : les profils des items et les profils des utilisateurs. La notion de contenu ne se rapporte donc pas uniquement au contenu des items, mais également aux attributs descriptifs des utilisateurs. Une approche basée contenu analyse un ensemble d'items précédemment notés ou consultés par un utilisateur, et construit un modèle ou un profil des intérêts de l'utilisateur sur la base des caractéristiques des items aimés ou détestés par celui-ci. En fonction de ses feedbacks, le profil de l'utilisateur est construit et souvent constitué d'un profil "positif" représentant les items qu'il a aimés et d'un profil "négatif" représentant les items qu'il a détestés.

Le processus de recommandation consiste donc essentiellement à comparer les attributs des items candidats avec les attributs du profil "positif" et "négatif" de l'utilisateur. De ce fait, les items qui seront recommandés à l'utilisateur sont les items qui sont similaires à son profil "positif" et moins similaires à son profil "négatif" . Plus le profil de l'utilisateur construit reflète les préférences de l'utilisateur, plus le

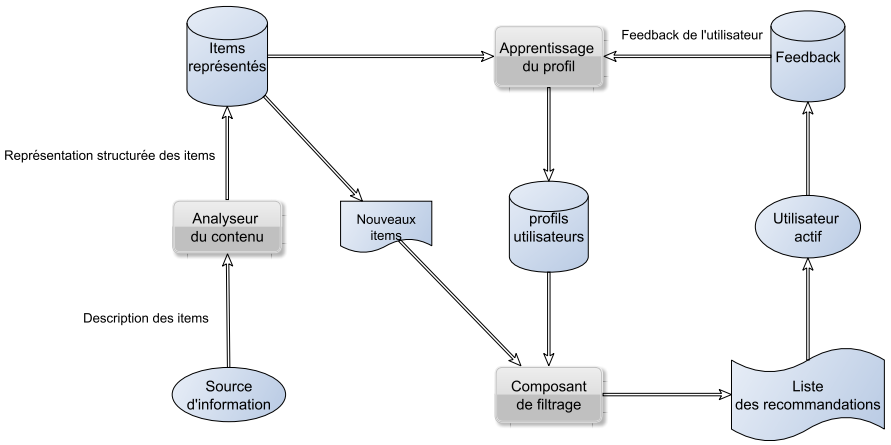


Figure 1.2 – Une architecture haut niveau d'un système de recommandation basé sur le contenu (d'après [Lops et al., 2011]).

système de recommandation peut être efficace.

Un système de recommandation basé sur le contenu a besoin de techniques pour produire une représentation efficace des items et du profil de l'utilisateur pour pouvoir les comparer. Ainsi [Lops et al., 2011] proposent une architecture de haut niveau (Cf. figure 1.2) dans laquelle le processus de recommandation est réalisé en trois étapes, chacune étant gérée par un composant spécifique :

- **Analyseur du contenu** : Lorsque l'information n'est pas structurée (par exemple, un item représenté par un texte), ce module a pour but d'en réaliser le pré-traitement pour extraire l'information pertinente, la structurer et la représenter dans une forme cible appropriée (par exemple un vecteur de mots clés).

- **Apprentissage du profil** : Ce module collecte les données représentatives des préférences de l'utilisateur et généralise ces données, afin d'apprendre et de construire le profil de l'utilisateur. Des techniques d'apprentissage automatique [Michalski et al., 2013] peuvent être utilisées pour cela. On peut citer à titre d'exemple les arbres de décisions, les réseaux de neurones et la classification naïve de Bayes. Ces techniques visent à inférer un profil de l'utilisateur en utilisant l'information sur les items qu'il a aimés ou n'a pas aimés.

- **Composant de filtrage** : Ce module filtre les items pertinents en faisant correspondre la représentation du profil utilisateur aux items candidats à la recommandation. La pertinence de l'item est calculée en utilisant des métriques de similarité entre l'item considéré et le profil de l'utilisateur. Plus la similarité avec le profil "positif" est grande et plus la similarité avec le profil "négatif" est petite, plus l'item a des chances d'être recommandé.

Afin de construire et mettre à jour le profil de l'utilisateur actif, ses réactions aux items (notes) sont recueillies et enregistrées dans le composant Feedback. Ces notes d'intérêt sont exploitées au cours du processus d'apprentissage du modèle utile pour prédire la pertinence *a priori* d'un item que l'utilisateur n'a pas encore noté. Les utilisateurs peuvent aussi définir explicitement leurs domaines d'intérêt au préalable comme profil initial, mais ce cas est assez rare.

1.3.2 Représentation d'un item

Dans la plupart des systèmes basés sur le contenu, la description de l'item est sous forme de texte, extrait de pages web, email ou fiche produit dans un site de e-commerce. Contrairement aux données structurées, il n'y a pas d'attributs avec des valeurs bien définies. Cela rend plus difficile l'apprentissage du profil utilisateur, en raison de l'ambiguïté du langage naturel, et notamment de la polysémie (multiples significations pour un mot) et de la synonymie (plusieurs mots qui ont le même sens). Un pré-traitement peut alors être nécessaire afin d'extraire l'information pertinente et de la structurer sous forme d'un ensemble d'attributs.

1.3.3 Recommandations basées sur les vecteurs de mots-clés

La plupart des systèmes de recommandation basés sur le contenu utilisent le modèle de représentation vectoriel VSM (Vector Space Model) avec la pondération classique TF-IDF (Term Frequency-Inverse Document Frequency). Dans ce modèle, chaque document (qui représente un item) est représenté par un vecteur de dimension n, où une dimension correspond à un terme de l'ensemble du vocabulaire d'une collection de documents. Formellement, tout document est représenté par un vecteur poids sur des termes, où chaque poids indique le degré d'association entre le document et le terme. Soit $D = \{d_1, d_2, ..., d_N\}$ dénotant un ensemble de documents ou corpus, et $T = \{t_1, t_2, ..., t_n\}$ le dictionnaire, c'est-à-dire l'ensemble des mots du corpus. T est généralement obtenu en appliquant des opérations de traitement du langage naturel, comme l'atomisation (tokenization), l'élimination des mots vides de sens, et la troncature (stemming) [Baeza-Yates et al., 1999]. Chaque document d_j est représenté par un vecteur dans un espace vectoriel à n dimensions, tel que $d_j = \{w_{1j}, w_{2j}, ..., w_{nj}\}$, où w_{kj} est le poids du terme t_k dans le document d_j.

La représentation de documents en utilisant le modèle d'espace vectoriel fait apparaitre deux difficultés : la pondération des termes et la mesure de similarité des vecteurs représentant les documents.

La méthode de pondération de termes la plus couramment utilisée est la pondération TF-IDF, qui est basée sur des observations empiriques sur le texte

[Salton, 1989] :

- des occurrences multiples d'un terme dans un document sont souvent plus pertinentes que de simples occurrences (TF) ;
- les termes rares ne sont pas forcément moins discriminants par rapport aux termes fréquents (IDF) ;
- des documents longs ne sont pas préférables à des documents plus courts.

Plus explicitement, les termes qui apparaissent fréquemment dans un document, mais rarement dans le reste du corpus ont plus de chances de représenter le sujet du document [Lops et al., 2011]. De plus, la normalisation des vecteurs résultats empêche les documents longs d'avoir plus de chances d'être retrouvés que les documents courts. Cela est bien pris en compte par la fonction TF-IDF [Sparck Jones, 1972] :

$$TFIDF(t_k, d_j) = TF(t_k, d_j) \times log(\frac{N}{n_k}) \tag{1.1}$$

où N dénote le nombre de documents dans le corpus, et n_k représente le nombre de documents de la collection dans lesquels le terme t_k apparait au moins une fois, avec :

$$TF(t_k, d_j) = \frac{f_{k,j}}{max_z f_{z,j}} \in [0, 1] \tag{1.2}$$

où $f_{k,j}$ représente le nombre d'occurrences du terme t_k dans le document d_j, et $max_z f_{z,j}$ est le maximum des fréquences $f_{z,j}$ des termes t_z apparaissant dans le document d_j.

Afin que tous les poids appartiennent à l'intervalle $[0, 1]$, et que tous les documents soient représentés par des vecteurs de même longueur, les poids obtenus par la fonction TFIDF sont généralement normalisés en utilisant la normalisation cosinus :

$$w_{k,j} = \frac{TFIDF(t_k, d_j)}{\sqrt{\sum_{s=1}^{|T|} TFIDF(t_s, d_j)^2}} \tag{1.3}$$

Une fois que les poids sont calculés et normalisés. Le contenu d'un item d_j est défini par :

$$Content(d_j) = (w_{1j}, w_{2j}, ..., w_{kj}) \tag{1.4}$$

Après cette étape de pondération des termes et de normalisation, il faut définir une mesure de similarité des vecteurs caractéristiques. Cette mesure de similarité est requise pour déterminer la proximité entre deux documents. Il existe de nombreuses mesures de similarité, mais la mesure la plus largement utilisée dans la littérature est la similarité cosinus :

$$sim(d_i, d_j) = \frac{\sum_k w_{ki}.w_{kj}}{\sqrt{\sum_k w_{ki}^2} \times \sqrt{\sum_k w_{kj}^2}} \tag{1.5}$$

Dans les systèmes de recommandation basés sur le contenu s'appuyant sur un modèle d'espace vectoriel, les profils des utilisateurs et les items sont représentés comme des vecteurs de termes pondérés. Notons $ContentBasedProfile(u)$ le profil de l'utilisateur u contenant ses préférences. Ce profil est obtenu en analysant le contenu des items qu'il a notés auparavant, et il est souvent défini par un vecteur de mots clés en utilisant des techniques issues du domaine de la recherche d'information. Plus formellement, $ContentBasedProfile(u)$ est défini comme un vecteur de poids $(w_{1u}, w_{2u}, ..., w_{ku})$, où chaque poids w_{iu} dénote l'importance de l'attribut k_i par rapport aux préférences de l'utilisateur u. Ce vecteur de poids représentant le profil de l'utilisateur peut être obtenu à partir des vecteurs du contenu des items que l'utilisateur a notés en utilisant différentes techniques. Par exemple, l'algorithme de *Rocchio* [Rocchio, 1971] a largement été utilisé pour déterminer $ContentBasedProfile(u)$ comme étant le vecteur moyen à partir des vecteurs des items notés. La prédiction de l'intérêt d'un utilisateur pour un item qu'il n'a pas encore noté peut être effectuée par le calcul de la similarité cosinus entre le vecteur du profil utilisateur et le vecteur de l'item.

Les systèmes de recommandation basés sur le contenu utilisent donc souvent des approches basées sur les mots clés. On trouve des travaux appliquant ces approches dans différents domaines d'application comme la recommandation de pages web : Letizia [Lieberman, 1995], la recommandation pour les news : NewsDude [Billsus and Pazzani, 2000] et YourNews [Ahn et al., 2007], et la recommandation de livres LIBRA [Mooney and Roy, 2000].

1.3.4 Autres modèles

D'autres méthodes ont été utilisées pour modéliser le profil utilisateur dans les systèmes de recommandation basés sur le contenu. Citons à titre d'exemple l'usage des n-grammes pondérés pour modéliser les items dans le système PSUN [Sorensen and McElligott, 1995] et les réseaux associatifs pondérés (weighted associative networks) pour modéliser le profil utilisateur [Riordan and Sorensen, 1995]. Un

réseau associatif pondéré représente les termes, concepts ou mots qui suscitent l'intérêt de l'utilisateur. Un ensemble de relations pondérées établit l'organisation de ces termes en phrases pertinentes.

Par ailleurs, plusieurs systèmes ont modélisé le profil utilisateur à l'aide d'un modèle de classification tel que les arbres de décision, les réseaux de neurones, les règles d'induction ou les réseaux bayésiens [Montaner et al., 2003, Pazzani and Billsus, 2007].

1.3.5 Formes particulières de recommandation basée sur le contenu

1.3.5.1 Recommandation basée sur la connaissance

La recommandation basée sur la connaissance consiste à collecter le maximum d'informations sur un utilisateur pour pouvoir ensuite lui recommander des items [Towle and Quinn, 2000]. Une analogie avec la vie réelle serait par exemple une recommandation faite par un ami qui nous connaît bien et se serait basé sur des informations précises nous concernant, plutôt que sur nos préférences. En d'autres termes les systèmes de recommandation basés sur la connaissance (knowledge-based) se basent sur la connaissance explicite de l'utilisateur. Par exemple, pour un système de recommandation pour la vente d'appartements ou de voitures, l'utilisateur peut spécifier explicitement ses préférences (ex. "le prix maximum de la voiture est x", "la consommation moyenne ne doit pas dépasser y"). Une force majeure des ces systèmes est qu'il ne souffrent pas du problème de démarrage à froid.

1.3.5.2 Recommandation basée sur l'utilité

Les systèmes de recommandation basés sur l'utilité (utility-based) font des suggestions en calculant l'utilité de chaque objet pour l'utilisateur. Bien entendu, le problème central est de créer une fonction d'utilité pour chaque utilisateur [Stolze and Rjaibi, 2001]. Le profil utilisateur est alors la fonction d'utilité que le système obtient de l'utilisateur. Une façon de procéder est de demander aux utilisateurs de remplir des formulaires. Par exemple, dans le cadre de la vente en ligne de micro-ordinateurs, il est possible de demander des renseignements sur l'usage qu'en fera le client.

1.3.6 Avantages et inconvénients des approches basées sur le contenu

Les approches basées sur le contenu présentent plusieurs avantages et inconvénients. Les points forts des approches basées sur le contenu sont :

- **Autonomie de l'utilisateur :** les techniques de recommandation basées sur le contenu traitent chaque utilisateur de façon indépendante. Ainsi, seules les évaluations de l'utilisateur lui-même sont prises en compte pour construire son profil utilisateur et faire la recommandation, ce qui n'est pas le cas pour les approches utilisant le filtrage collaboratif.

- **Prise en compte immédiate d'un nouvel item :** le filtrage basé sur le contenu peut recommander des items nouvellement introduits dans la base avant même qu'ils reçoivent une évaluation de la part d'un utilisateur, au contraire des approches collaboratives qui ne peuvent recommander un item que s'il a été préalablement évalué par un groupe d'utilisateurs. Les auteurs [Ekstrand et al., 2011] ont démontré qu'il faut qu'un item ait reçu au minimum 20 notes pour que le filtrage collaboratif arrive à le recommander de manière pertinente.

Cependant les approches de recommandation basées sur le contenu présentent aussi de nombreux inconvénients :

- **Limite de l'analyse du contenu :** une limite naturelle de la recommandation basée sur le contenu est la nécessité de disposer d'une représentation variée et riche du contenu des items, ce qui n'est pas toujours le cas. La précision des recommandations est liée à la quantité d'informations dont dispose le système pour discriminer les items appréciés de ceux non appréciés par l'utilisateur [Lops et al., 2011]. Contrairement au filtrage collaboratif qui peut traiter tout type d'items sans aucune information sur leur contenu, l'approche basée sur le contenu ne peut traiter que les items disposant d'un contenu pouvant être analysé.

- **Sur-spécialisation (Over-specialization) :** le système ne peut recommander que les items qui sont similaires au profil utilisateur. L'utilisateur ne peut donc recevoir que des recommandations proches des items qu'il a notés ou observés par le passé [Adomavicius and Tuzhilin, 2005]. Or, la diversité des recommandations est souvent appréciée et s'avère être un critère d'évaluation important des systèmes de recommandation [Yu et al., 2009]. Idéalement,

l'utilisateur doit recevoir des recommandations pertinentes et diversifiées. Par exemple, il n'est pas intéressant de recommander toutes les chansons de *Jacques Brel* à un utilisateur qui a aimé l'une de ses chansons.

- **Intégration d'un nouvel utilisateur non immédiate :** un utilisateur doit évaluer un certain nombre d'items avant que le système ne puisse interpréter ses préférences et lui fournir des recommandations pertinentes [Ricci et al., 2011]. Ce problème est connu dans la littérature sous le nom du problème de démarrage à froid pour les utilisateurs (user cold start).

1.4 Les approches basées sur le filtrage collaboratif

Le filtrage collaboratif est une approche basée sur le partage d'opinions entre les utilisateurs. Il reprend le principe du "bouche à oreille" pratiqué depuis toujours par les humains pour se construire une opinion sur un produit ou un service qu'il ne connaissent pas [Schafer et al., 2007]. L'hypothèse fondamentale de cette méthode est que les opinions des autres utilisateurs peuvent être utilisées pour fournir une prédiction raisonnable de la préférence de l'utilisateur actif sur un item qu'il n'a pas encore noté. Ces méthodes supposent que si des utilisateurs ont les mêmes préférences sur un ensemble d'items, alors ils auront probablement les mêmes préférences sur un autre ensemble d'items qu'il n'ont pas encore notés. Supposons par exemple que les voisins de Marie trouvent que le nouveau restaurant qui est ouvert dans son voisinage est un succès, elle peut juger intéressant d'aller l'essayer. Si au contraire, la majorité de ses voisins estiment que c'est un échec, alors il se peut qu'elle décide de s'abstenir d'y aller. Les techniques de filtrage collaboratif recommandent donc, à l'utilisateur courant, les items appréciés par les utilisateurs avec lesquels il partage les mêmes goûts. On parle d'utilisateurs similaires.

On distingue généralement deux sous-familles principales du filtrage collaboratif : les méthodes basées sur la mémoire (memory-based) et les méthodes basées sur un modèle (model-based). Les algorithmes de filtrage collaboratif basés sur la mémoire, appelés aussi *basés sur des heuristiques* selon [Adomavicius and Tuzhilin, 2005] ou plus fréquemment *basés sur les voisins* [Desrosiers and Karypis, 2011] utilisent les notes des utilisateurs stockés en mémoire pour faire de la prédiction. Les algorithmes basés sur un modèle construisent en offline une image réduite de la matrice des notes dans un objectif de réduire la complexité des calculs et/ou de traiter le problème des notes manquantes. Le modèle passe d'abord par une étape d'apprentissage, puis, il est utilisé pour faire de la recommandation. Plusieurs méthodes ont été utilisées pour les algorithmes de recommandation basés sur un modèle. On peut

citer, parmi les plus abouties, les méthodes de réduction de la dimension appelées SVD (décomposition en valeurs singulières) [Koren and Bell, 2011], les approches probabilistes [Breese et al., 1998], les approches basées sur le clustering [Ungar and Foster, 1998] et les approches basées sur les règles d'association [Heckerman et al., 2001].

1.4.1 Filtrage collaboratif basé sur les voisins

Les algorithmes de filtrage collaboratif basés sur les voisins [Nakamura and Abe, 1998, Delgado and Ishii, 1999] utilisent généralement la totalité de la matrice des notes des utilisateurs pour faire la recommandation. On parle d'approche des *k plus proche voisins* (ou *k-Nearest Neighbours - kNN*). Ces approches sont regroupées en deux familles : *basés sur les utilisateurs* (user-user collaborative filtering) ou *basés sur les items* (item-item collaborative filtering). Pour les algorithmes basés sur les utilisateurs tels que GroupLens [Resnick et al., 1994] ou Ringo [Shardanand and Maes, 1995b], l'appréciation estimée d'un utilisateur u pour un item i est prédite en utilisant les notes de ses voisins (ses utilisateurs similaires, avec lesquels il partage les mêmes préférences). De manière analogue, les algorithmes basés sur les items [Linden et al., 2003, Sarwar et al., 2001] déterminent l'appréciation estimée d'un utilisateur u pour un item candidat i à partir des notes de u pour les items voisins de i. Nous détaillons dans la suite ces deux types d'algorithmes.

En plus des notations introduites en section 1.2.4, on note par $\bar{r_u}$ la moyenne des notes données par l'utilisateur u sur les items qu'il a notés (formule 1.6) et par $\bar{r_i}$ la moyenne des notes reçues par l'item i (formule 1.7).

$$\bar{r_u} = \frac{\sum_{i \in I_u} r_{u,i}}{|I_u|} \tag{1.6}$$

$$\bar{r_i} = \frac{\sum_{u \in U_i} r_{u,i}}{|U_i|} \tag{1.7}$$

On note également par $sim(u, v)$ la fonction mesurant la similarité entre les deux utilisateurs u et v, et par $sim(i, j)$ la similarité entre les deux items i et j. On définit $I_{uv} = I_u \cap I_v$ comme étant l'ensemble des items notés à la fois par les utilisateurs u et v, et de façon équivalente $U_{ij} = U_i \cap U_j$ l'ensemble des utilisateurs ayant noté à la fois les items i et j.

1.4.1.1 Filtrage basé sur les utilisateurs

Le filtrage collaboratif basé sur les utilisateurs a été introduit pour la première fois dans le système GroupLens [Resnick et al., 1994], son principe de fonctionnement est

très simple : déterminer les utilisateurs qui sont similaires à l'utilisateur courant, puis calculer une valeur de prédiction pour chaque item candidat à la recommandation en analysant les notes que les voisins de l'utilisateur courant ont exprimées sur cet item.

Calcul de la similarité

La similarité entre deux utilisateurs u et v peut être mesurée en utilisant la similarité Cosinus (voir formule 1.13) ou bien en utilisant le coefficient de corrélation de Pearson (voir formule 1.15). Selon [Schafer et al., 2007], le coefficient de Pearson est le plus utilisé dans la littérature. C'est aussi le plus performant en terme de pertinence des recommandations.

Calcul de la prédiction

Pour le calcul de la prédiction, la méthode la plus simple est de calculer la moyenne des notes de tous les voisins de l'utilisateur courant u comme l'illustre l'équation 1.8 :

$$pred(u,i) = \frac{\sum_{w \in voisins(u) \cap U_i} r_{w,i}}{|voisins(u) \cap U_i|} \tag{1.8}$$

Cette formule a été critiquée parce qu'elle considère tous les voisins sur le même pied d'égalité et ne tient pas compte du fait que certains voisins peuvent être plus similaires que d'autres à l'utilisateur courant u [Schafer et al., 2007]. Afin de tenir compte de cette information, on pondère la note de chaque voisin par la valeur de sa similarité avec l'utilisateur courant. Ainsi, les notes des voisins les plus similaires auront un poids plus important que celui des voisins moins similaires. Vu que la somme des similarités de tous les voisins n'est pas égale à 1, et afin d'avoir une valeur de prédiction normalisée, on divise par la somme des valeurs absolues des similarités de l'utilisateur courant avec ses voisins. L'équation 1.9 donne la formule correspondante pour le calcul de la prédiction.

$$pred(u,i) = \frac{\sum_{w \in voisins(u) \cap U_i} sim(w,u) \times r_{w,i}}{\sum_{w \in voisins(u) \cap U_i} |sim(w,u)|} \tag{1.9}$$

Par ailleurs, tous les utilisateurs sont différents dans leur façon de noter un item. En effet, il existe des utilisateurs qui notent large en affectant la valeur de 5 sur une échelle de 1 à 5 pour un item qu'ils jugent satisfaisant alors que d'autres, qui ont tendance à noter de façon plus stricte, attribueront la valeur 3 à un item qu'ils jugent satisfaisant. Pour compenser la variation dans les jugements des utilisateurs, la note de chaque utilisateur w est ajustée par la moyenne de ses notes $\bar{r_u}$. L'équation 1.10 donne la formule finale qui est adoptée par les auteurs de Gouplens [Resnick et al., 1994] pour le calcul de la prédiction.

$$pred(u,i) = \bar{r_u} + \frac{\sum_{w \in voisins(u) \cap U_i} sim(w,u) \times (r_{w,i} - \bar{r_w})}{\sum_{w \in voisins(u) \cap U_i} |sim(w,u)|} \tag{1.10}$$

Dans le système initial implanté dans Grouplens, tous les voisins d'un utilisateur sont pris en compte lors du processus de prédiction. Il a cependant été démontré par la suite que la restriction aux k plus proches voisins améliore considérablement la qualité des recommandations fournies [Herlocker et al., 1999]. Une analyse des données est nécessaire pour fixer la valeur de k, cette valeur peut dépendre sensiblement du domaine et des données utilisées. Selon [Herlocker et al., 2002] $k = 20$ est généralement une bonne valeur et les valeurs entre 20 et 50 utilisateurs restent raisonnables.

1.4.1.2 Filtrage basé sur les items

Le filtrage collaboratif à base d'items a été introduit par [Sarwar et al., 2001]. La prédiction de la note de l'utilisateur u pour un item candidat i est calculée à partir de ses notes pour les items voisins (similaires) de i. Son principe de fonctionnement est le suivant : pour l'item i candidat à la recommandation, on détermine les voisins les plus proches (les items similaires) en calculant sa similarité avec les autres items disponibles et on calcule ensuite la prédiction de la note de l'utilisateur courant u pour l'item i à partir des notes que u a attribué à aux voisins de i.

Calcul de la similarité : la similarité entre deux items i et j peut être calculée en utilisant soit le Cosinus (voir formule 1.14), soit le coefficient de Pearson (voir formule 1.15), soit le cosinus ajusté (voir formule 1.17). Cependant, une étude expérimentale menée par les auteurs de [Sarwar et al., 2001], comparant les trois mesures, a montré que le Cosinus ajusté est le plus performant en terme de pertinence de prédiction.

Calcul de la prédiction : la prédiction de la note de l'utilisateur courant u pour un item candidat à la recommandation i revient à calculer une moyenne pondérée de ses notes sur l'ensemble des items similaires à i. Chaque note $r_{u,j}$ est pondérée par la similarité de l'item j avec l'item i. Afin d'avoir une prédiction dans le même intervalle de valeurs que les notes, la prédiction est divisée par la somme des similarités. L'ajustement de la note est inutile dans ce cas puisqu'il s'agit du même utilisateur. L'équation 1.11 donne la formule exacte utilisée par [Sarwar et al., 2001].

$$pred(u,i) = \frac{\sum_{i \in I_u} sim(i,j) \times r_{u,j}}{\sum_{i \in I_u} |sim(i,j)|} \tag{1.11}$$

Comme pour le filtrage basé sur les utilisateurs, les auteurs [Sarwar et al., 2001] ont démontré que la pertinence des prédictions est très sensible au nombre de

voisins considérés dans la formule 1.11. Ainsi, parmi les items notés par u seuls les k plus proches voisins de i sont pris en compte pour aboutir à de meilleures recommandations et gagner en temps de calcul.

1.4.1.3 Calcul de la similarité

Le calcul de la similarité a pour objectif de déterminer dans quelle mesure deux utilisateurs ou deux items sont similaires. Il existe plusieurs façons de calculer cette similarité, cependant les méthodes les plus utilisées et qui présentent les meilleurs résultats sont présentées ici :

Cosinus : Cosinus est une mesure de similarité entre deux objets a et b de manière générale, très utilisée en recherche d'informations [Salton, 1989], qui consiste à représenter les deux objets par deux vecteurs $\overrightarrow{x_a}$ et $\overrightarrow{x_b}$ et de mesurer le cosinus de l'angle formé par les deux vecteurs.

$$sim(a,b) = cos(\overrightarrow{x_a}, \overrightarrow{x_b}) = \frac{\overrightarrow{x_a}.\overrightarrow{x_b}}{||\overrightarrow{x_a}||||\overrightarrow{x_b}||} \tag{1.12}$$

Dans le cas du filtrage collaboratif, chaque utilisateur u est représenté par un vecteur x_u, où $x_{ui} = r_{u,i}$. Pour pouvoir calculer la similarité entre deux utilisateurs u et v, le cosinus est calculé sur l'ensemble des items notés par les deux utilisateurs comme l'illustre la formule 1.13.

$$sim(u,v) = cos(\overrightarrow{x_u}, \overrightarrow{x_v}) = \frac{\sum_{i \in I_{uv}} r_{u,i} \times r_{v,i}}{\sqrt{\sum_{i \in I_{uv}} r_{u,i}^2} \cdot \sqrt{\sum_{i \in I_{uv}} r_{v,i}^2}} \tag{1.13}$$

Le cosinus peut aussi s'appliquer pour calculer la similarité entre deux items. En effet, il suffit de remplacer dans l'équation 1.13 les utilisateurs par leurs équivalents en items.

$$sim(i,j) = cos(\overrightarrow{x_i}, \overrightarrow{x_j}) = \frac{\sum_{u \in U_{ij}} r_{u,i} \times r_{u,j}}{\sqrt{\sum_{u \in U_{ij}} r_{u,i}^2} \cdot \sqrt{\sum_{u \in U_{ij}} r_{u,j}^2}} \tag{1.14}$$

Le cosinus varie entre 0 et 1. Une valeur égale à 1 indique que les deux utilisateurs ont des préférences identiques, une valeur égale à 0 indique qu'ils n'ont rien en commun. Un inconvénient majeur de l'utilisation du cosinus dans le filtrage collaboratif est qu'il ne tient pas compte de la variation dans le jugement des utilisateurs.

Coefficient de corrélation de Pearson : ce coefficient a été utilisé notamment par les auteurs du système GroupLens [Resnick et al., 1994] pour calculer la similarité entre deux utilisateurs u et v. Le coefficient de corrélation de Pearson mesure le

rapport entre la covariance et le produit de l'écart-type des notes données par les deux utilisateurs. Il permet ainsi de mesurer la similarité en utilisant les items notés à la fois par u et v. Plus les deux utilisateurs auront tendance à noter les mêmes items de façon équivalente, plus ils seront similaires comme l'illustre la formule 1.15.

$$sim(u,v) = Pearson(u,v) = \frac{\sum_{i \in I_{uv}} (r_{u,i} - \bar{r_u}).(r_{v,i} - \bar{r_v})}{\sqrt{\sum_{i \in I_{uv}} (r_{u,i} - \bar{r_u})^2}.\sqrt{\sum_{i \in I_{uv}} (r_{v,i} - \bar{r_v})^2}} \quad (1.15)$$

Le coefficient de corrélation de Pearson peut également être utilisé pour mesurer la corrélation entre deux items i et j. L'équation 1.16 donne la similarité de Pearson entre deux items.

$$sim(i,j) = Pearson(i,j) = \frac{\sum_{u \in U_{ij}} (r_{u,i} - \bar{r_i}).(r_{u,j} - \bar{r_j})}{\sqrt{\sum_{u \in U_{ij}} (r_{u,i} - \bar{r_i})^2}.\sqrt{\sum_{u \in U_{ij}} (r_{u,j} - \bar{r_j})^2}} \quad (1.16)$$

Le cosinus ajusté (adjusted cosine) : Lorsque la similarité de Pearson est utilisée pour calculer la similarité entre deux items, les notes d'un même utilisateur sont centrées par rapport à la moyenne de ses notes. Or, la variation de notation pour un même utilisateur n'est pas aussi importante que la variation entre les différents utilisateurs. C'est pour cette raison qu'il est plus intéressant, lors du calcul de la similarité entre deux items, d'ajuster les notes par rapport à la moyenne des notes des utilisateurs plutôt que par rapport à la moyenne des notes des items. C'est le rôle du Cosinus ajusté, qui a été introduit par Sarwar et al. [Sarwar et al., 2001]. Selon [Schafer et al., 2007], le cosinus ajusté est considéré comme l'un des moyens les plus efficaces et populaires pour calculer la similarité entre deux items pour les algorithmes de filtrage collaboratif.

$$sim(i,j) = Adjusted\_cosine(i,j) = \frac{\sum_{u \in U_{ij}} (r_{u,i} - \bar{r_u}).(r_{u,j} - \bar{r_u})}{\sqrt{\sum_{u \in U_{ij}} (r_{u,i} - \bar{r_u})^2}.\sqrt{\sum_{u \in U_{ij}} (r_{u,j} - \bar{r_u})^2}} \quad (1.17)$$

1.4.2 Méthode de réduction de la dimension

Les méthodes de réduction de la dimension permettent de projeter les items (et/ou les utilisateurs) dans une dimension réduite définie par des variables latentes afin de traiter le problème de la sensibilité aux données manquantes [Ekstrand et al., 2011] et le problème du passage à l'échelle [Schafer et al., 2007]. Étant donné que les

utilisateurs (ou les items) seront comparés dans cet espace plus dense et plus réduit que l'espace défini par la matrice des notes, de nouvelles relations entre des paires d'utilisateurs (ou d'items) pourront être détectées même s'ils n'ont aucun item en commun. Les méthodes de réduction de dimension ont connu un large succès [Koren and Bell, 2011] surtout après le challenge lancé par Netflix [Bennett and Lanning, 2007]. En effet, des travaux [Takács et al., 2007, Koren et al., 2009] analysant les résultats du challenge ont démontré la supériorité en terme de précision des approches appliquant des techniques de réduction de la dimension par rapport aux algorithmes de filtrage collaboratif basé sur les voisins. Les méthodes de factorisation de matrice sont utilisées pour l'extraction des variables latentes telles que l'Analyse en Composante Principale (ACP) [Goldberg et al., 2001] ou la Décomposition en Valeur Singulière (SVD en anglais Singular Value Decomposition) [Koren, 2008]. Elles ont été essentiellement utilisées, soit pour réduire la dimension de la matrice des notes, soit pour réduire la dimension de la matrice de similarité [Desrosiers and Karypis, 2011].

L'Analyse Latente Sémantique (Latent Semantic Analysis, LSA) appelée également Indexation Sémantique Latente (Latent Semantic Indexing, LSI) est une méthode de réduction de dimension très répandue dans le domaine du filtrage et de la recherche d'informations [Belkin and Croft, 1992]. Elle a également été utilisée pour réduire la dimension de la matrice des notes R dans les algorithmes de filtrage collaboratif [Desrosiers and Karypis, 2011]. LSA approxime la matrice $R_{|U|,|I|}$ de rang r par une matrice $R' = PQ^t$ de rang $k << r$, où $P_{|U|,k}$ est une matrice représentant les utilisateurs dans l'espace réduit, et $Q_{|I|,k}$ est une matrice représentant les items dans l'espace réduit, P et Q étant deux matrices orthonormales. Pour déterminer les matrices P et Q, une décomposition en valeurs singulières (SVD) est appliquée à la matrice des notes R. La SVD décompose la matrice R en trois matrices D , Σ et T.

$$R = D_{|U|,r} \times \Sigma_{r,r} \times T^t_{r,|I|} \tag{1.18}$$

où D et T sont deux matrices orthonormales, r le rang de la matrice R et Σ une matrice diagonale dont les valeurs représentent les valeurs singulières de la matrice R ordonnées par ordre décroissant. LSA applique une SVD tronquée pour approximer la matrice R en ne conservant que les k plus grandes valeurs singulières et leurs vecteurs singuliers correspondants.

$$R \approx R' = D_{|U|,k} \times \Sigma_{k,k} \times T^t_{k,|I|} \tag{1.19}$$

Les matrices modélisant respectivement les utilisateurs et les items dans l'espace

réduit LSA sont alors définies par $P = D_k\Sigma_k^{1/2}$ et $Q = T_k\Sigma_k^{1/2}$. Chaque utilisateur dans $P_{|U|,k}$ est représenté par un ensemble de k variables latentes au lieu de la totalité ses notes. Chaque item dans $Q_{|I|,k}$ est représenté par k variables latentes au lieu des notes des utilisateurs.

Une fois les matrices P et Q obtenues, la prédiction de la note de l'utilisateur u pour l'item i est donné par :

$$pred(u,i) = p_u q_i^t \tag{1.20}$$

Toutefois, le problème principal qui se pose lors de l'application d'une SVD à la matrice des notes R est la présence des données manquantes. Une SVD ne peut pas s'appliquer à une matrice comportant des valeurs manquantes. Une solution consiste à attribuer des valeurs par défaut aux valeurs manquantes ce qui peut biaiser les données [Desrosiers and Karypis, 2011]. La solution la plus fréquente est l'apprentissage de P et Q en se limitant aux notes connues [Takács et al., 2007, Takács et al., 2008, Bell et al., 2007, Koren, 2008]. Pour plus de détails sur les récents travaux appliquant la factorisation de matrice dans les algorithmes de recommandations collaboratives, le lecteur peut se référer à [Koren and Bell, 2011].

1.4.3 Modèles probabilistes

Le principe de ces approches consiste à modéliser le comportement des utilisateurs par un modèle probabiliste pour pouvoir prédire ses comportements futurs. L'idée sur laquelle reposent ces algorithmes consiste à calculer la probabilité $P(v|u,i)$ que l'utilisateur u attribue la note v à l'item i connaissant ses notes antérieures. La prédiction $pred(u,i)$ correspond, soit à la note ayant la plus grande probabilité, soit à la note espérée, telle que définie par la formule 1.21 [Schafer et al., 2007].

$$pred(u,i) = E(v|u,i) = \sum_{v \in V} v.P(v|u,i) \tag{1.21}$$

V étant l'ensemble des valeurs que peut prendre une note (souvent de 1 à 5).

Cross Sel [Kitts et al., 2000] utilise un classifieur bayésien naïf pour faire de la recommandation. A partir de l'historique des achats d'un utilisateur, il estime la probabilité qu'il achète un item j sachant qu'il a acheté l'item i. Breese *et al.* [Breese et al., 1998], dans un des premiers travaux appliquant un modèle probabiliste pour les algorithmes de filtrage collaboratif, proposent un modèle construit à partir des réseaux Bayésiens et utilisant les arbres de décision pour calculer les probabilités.

Les réseaux bayésiens ont également été utilisés par [Breese et al., 1998, Chien and George, 1999, Zigoris and Zhang, 2006]. Enfin, les processus décisionnels de

Markov [Shani et al., 2002] ont aussi été utilisés pour la recommandation.

1.4.4 Avantages et inconvénients du filtrage collaboratif

Les méthodes de filtrage collaboratif présentent plusieurs avantages dont les plus importants sont :

- **Effet de surprise (serendipity) :** l'effet de surprise que peut recevoir l'utilisateur en recevant une recommandation pertinente qu'il n'aurait pas trouvée seul est souvent souhaitable. Les algorithmes basés sur le filtrage collaboratif permettent généralement de faire des recommandations à effet de surprise. Par exemple, si un utilisateur u est proche d'un utilisateur v du fait qu'il ne regarde que des comédies, et si v apprécie un film d'un autre genre, ce film peut être recommandé à u du fait de sa proximité avec v.

- **Non nécessitée de la connaissance du domaine :** les systèmes de recommandation basés sur le filtrage collaboratif ne requièrent aucune connaissance sur les items. Ces méthodes peuvent recommander des items sans avoir besoin de comprendre leurs sens ni disposer de leurs attributs. La recommandation est basée uniquement sur les notes données aux items.

Cependant, l'utilisation des techniques de filtrage collaboratif peut entrainer plusieurs problèmes :

- **Le démarrage à froid :** concerne à la fois les nouveaux utilisateurs et les nouveaux items qui sont introduits dans le système. Un nouvel utilisateur qui n'a noté aucun item ne peut pas recevoir de recommandation puisque le système ne connaît pas ses goûts. Ce problème est connu sous le nom de problème du démarrage à froid pour les utilisateurs (user cold start). Une solution à ce problème est de lui demander explicitement de noter un certain nombres d'items. D'autres solutions consistent à recommander au départ les items les plus populaires ou même des recommandations aléatoires. Ce problème du démarrage à froid se pose aussi lors de l'ajout d'un nouvel item. Celui-ci ne peut pas être recommandé avant d'avoir été noté par un certain nombre d'utilisateurs.

- **La parcimonie (sparsity) :** Le nombre d'items candidats à la recommandation est souvent énorme et les utilisateurs ne notent qu'un petit sous-ensemble des items disponibles. De ce fait, la matrice des notes est une matrice creuse avec un taux de valeurs manquantes pouvant atteindre 95% du total des

valeurs [Papagelis et al., 2005]. Les systèmes de filtrage collaboratif ont des difficultés dans ce cas, le nombre de notes à prédire étant largement supérieur aux nombres de notes déjà connues. Le problème de la parcimonie peut être réduit en utilisant les approches par modèles qui réduisent la dimension de la matrice des notes.

- **Le problème du mouton gris (gray sheep) :** Les utilisateurs qui ont des goûts étranges (qui varient de la norme ou qui sortent du commun) n'auront pas beaucoup d'utilisateurs voisins. Il sera donc difficile de faire des recommandations pertinentes pour ce genre d'utilisateurs [Ghazanfar and Prügel-Bennett, 2014].

1.5 Les Approches Hybrides

Un système de recommandation est dit hybride quand il combine deux ou plusieurs approches de recommandation différentes. La recommandation basée sur le contenu et la recommandation collaborative ont souvent été considérées comme complémentaires [Adomavicius and Tuzhilin, 2005]. Les approches basées sur le contenu ont l'avantage de pouvoir recommander les nouveaux items non encore évalués par un utilisateur, alors que le filtrage collaboratif ne peut recommander un item que s'il a été noté par un certain nombre d'utilisateurs auparavant. Les approches basées sur le contenu nécessitent de disposer des attributs des items, en plus d'une étape d'analyse pour pouvoir les extraire et les représenter, alors que le filtrage collaboratif ne requiert pas d'accès au contenu des items pour pouvoir faire de la recommandation. Il s'appuie uniquement sur la matrice des notes d'utilisateurs pour les différents items. L'hybridation de ces deux techniques, afin de traiter les insuffisances de chaque technique utilisée seule et profiter de leurs points forts, a fait l'objet de plusieurs travaux de recherche. Le système FAB [Balabanović and Shoham, 1997] est un des premiers systèmes de recommandation hybrides. Il combine le filtrage collaboratif et une approche basée sur le contenu afin de traiter à la fois le problème du démarrage à froid pour les items et la sur-spécialisation. Dans ce système, deux critères doivent être satisfaits pour recommander un item : son contenu doit être similaire au profil de l'utilisateur, et il doit être apprécié par les voisins les plus proches.

Il existe plusieurs manières de faire de l'hybridation et aucun consensus n'a été défini par la communauté des chercheurs. Toutefois, Burke [Burke, 2002] a identifié sept manières différentes de faire l'hybridation :

- **Pondérée** (Weighted) : le score ou la prédiction obtenu par chacune des deux techniques est combiné en un seul résultat.

- **Par sélection** (Switching) : le système bascule entre les deux techniques de recommandation en fonction de la situation.
- **Mixte** (Mixed) : les listes des recommandations issues des deux techniques sont fusionnées en une seule liste.
- **Par combinaison des propriétés** (Feature combination) : les données issues des deux techniques sont combinées et transmises à un seul algorithme de recommandation.
- **Par augmentation de propriétés** (Feature augmentation) : le résultat d'une technique est utilisé comme entrée de l'autre technique.
- **En cascade** : Dans ce type d'hybridation, une technique de recommandation est utilisée pour produire un premier classement des items candidats et une deuxième technique affine ensuite la liste des recommandations.
- **En définissant un niveau méta** : Cette méthode est analogue à la méthode par augmentation de propriétés mais c'est le modèle appris qui est utilisé en entrée de la deuxième technique et non la liste résultat des recommandations.

1.6 Systèmes de recommandation sensibles au contexte

L'importance des informations contextuelles a été reconnue par les chercheurs dans plusieurs domaines, y compris la recherche d'information, l'informatique ubiquitaire, le marketing et management, etc. Cependant, les travaux de recherche sur les systèmes de recommandation ont assez peu exploité les informations contextuelles. Des informations telles que le temps, la localisation, la compagnie d'autres personnes peuvent pourtant améliorer le processus de recommandation dans certains domaines. Les systèmes de recommandation traditionnels traitent seulement de deux types d'entités, les utilisateurs et les items. Cependant, pour de nombreuses applications, par exemple les systèmes de recommandation dédiés au tourisme, il peut ne pas être suffisant de ne considérer que les utilisateurs et les items. Il est souvent important d'intégrer des informations sur le contexte. Par exemple, un système de recommandation de séjours de vacances doit tenir compte de la saison pour fournir une recommandation adaptée. De même, un système de recommandation pour le tourisme implémenté sur dispositif mobile peut privilégier la recommandation de lieux d'activités proches de la position de l'utilisateur.

1.6.1 Définition du contexte

Le contexte est une notion vaste pour laquelle il est particulièrement difficile de donner une définition générale et opérationnelle. [Abowd et al., 1999] proposent la définition suivante que nous citons en Anglais et qui est l'une des plus largement acceptées :

Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.

Cette définition a été cependant critiquée par [Zimmermann et al., 2007] comme étant trop générale et non opérationnelle. Les auteurs proposent un modèle du contexte inspiré de celui de [Abowd et al., 1999], en spécifiant cinq catégories d'informations contextuelles : individualité, activité, relation, temporalité et localisation.

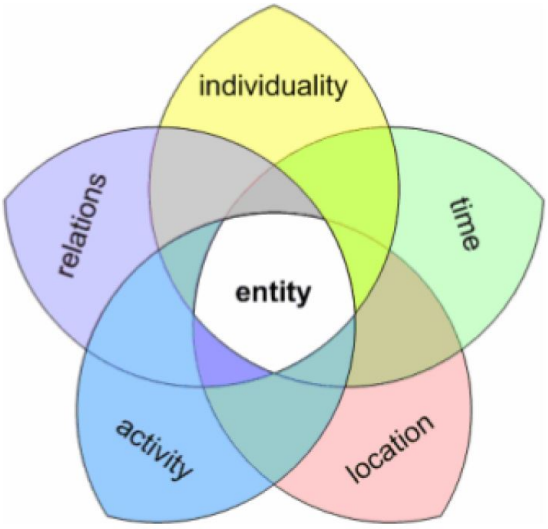


Figure 1.3 – Les différents éléments du contexte d'après [Zimmermann et al., 2007]

[Zimmermann et al., 2007] définissent le contexte comme ceci : *Context is any information that can be used to characterize the situation of an entity. Elements for the description of this context information fall into five categories : individuality, activity, location, time, and relations.*

La figure 1.3 représente une entité ainsi que ses différentes catégories contextuelles suivant l'approche de [Zimmermann et al., 2007].

1.6.2 Sources d'informations contextuelles

On dénote trois grands types de sources d'informations contextuelles : explicite, implicite ou inférée.

Explicite : l'information sur le contexte est déjà incluse dans les données ou directement demandée à l'utilisateur. Par exemple, sur des plateformes d'achats en ligne, on peut demander à l'utilisateur s'il effectue un achat pour des raisons personnelles ou professionnelles ou encore pour offrir un cadeau au moment de l'achat. On peut aussi, lors de l'inscription de l'utilisateur sur le système, lui demander de remplir un formulaire contenant des informations personnelles le concernant, telles que son âge, sa profession, etc.

Implicite : l'information est obtenue à partir des données ou de l'environnement dans lequel se trouve un utilisateur sans la lui demander explicitement. Pour des plateformes de recommandation de sites touristiques à partir des smartphones, on peut par exemple connaître la situation géographique exacte d'un individu au moment d'effectuer une recommandation.

Inférée : l'information est obtenue à l'aide de méthodes d'exploitation et d'exploration des données. Par exemple, l'identité d'une personne parcourant les chaînes de télévision peut ne pas être explicitement connue pour une société de télévision par câble. Cependant, le système peut arriver à apprendre le moment de la journée, la chaine et le type de programme regardés par les différents utilisateur d'un même foyer (père, mère, enfants...) et ceci avec une précision acceptable en utilisant des techniques de data mining.

1.6.3 Modélisation du contexte pour les systèmes de recommandation

Les systèmes de recommandation traditionnels sont à deux dimensions (2D) car ils ne considèrent que les dimensions de l'utilisateur et de l'item. Dans le cadre de la recommandation intégrant le contexte, celui-ci est vu comme une information additionnelle. En plus de l'utilisateur et de l'item, on ajoute la dimension du contexte qui va contribuer à améliorer la recommandation fournie par le système. Un système de recommandation sensible au contexte va donc considérer des fonctions de score sous la forme :

$$R : User \times Item \times Contexte \rightarrow Rating$$

Pour illustrer ce concept prenons un exemple de recommandation de films dans des salles de cinéma, où les utilisateurs et les items sont décrits respectivement par les attributs :

- *Films :* (ID_film, Titre, Durée, Année, Genre)
- *Utilisateurs :* (ID_utilisateur, Nom, Adresse, Age)

Et les informations contextuelles peuvent être modélisées par les trois types suivants, qui sont eux-mêmes décrits par les attributs correspondants :

- Salle : (ID_salle, Nom_salle, Adresse_salle)
- Temps : (JourDeSemaine, Weekend)
- Compagnon : (Type_compagnon : prend une des valeurs " seul ", " amis ", " famille ", " collègues ")

Du coup, la note attribuée à un film pour un utilisateur ne dépend plus seulement de ses préférences mais aussi de la salle où le film a été vu, avec qui et à quel moment. En effet, il est évident que le genre de film à recommander pour un utilisateur qui est seul peut être totalement différent du genre de film à recommander s'il est accompagné.

Comme on peut le constater, l'information contextuelle peut être de différents aspects, tel que le temps ou la localisation. De plus, chaque aspect contextuel peut avoir une structure complexe reflétant la nature de l'information contextuelle(par exemple, sous forme de texte). Pour faire face à cette complexité, l'information contextuelle est souvent hiérarchisée et représentée sous forme d'arbre.

Ainsi [Adomavicius et al., 2005] et [Palmisano et al., 2008], représentent le contexte par un ensemble de dimensions contextuelles K, chaque dimension contextuelle $k \in K$ étant définie par un ensemble de q attributs : $k = \{k^1, k^2, ..., k^q\}$ ayant une structure hiérarchique et capturant un aspect particulier du contexte.

1.6.4 Méthodes d'incorporation du contexte

L'incorporation de l'information contextuelle peut se faire à différents étapes du processus dans un système de recommandation. [Adomavicius and Tuzhilin, 2011] définissent trois grandes approches de contextualisation suivant le moment où le contexte est injecté. Ces approches sont les suivantes :

- pré-filtrage (contextual pre-filtering)
- post-filtrage (contextual post-filtering)
- modélisation contextuelle (contextual modeling)

Nous présentons brièvement ces trois approches dans la suite.

1.6.4.1 Pré-filtrage contextuel

L'incorporation du contexte par pré-filtrage ou prétraitement consiste à sélectionner un sous-ensemble de données significatif pour le contexte dans lequel on se situe et de restreindre le processus de recommandation à ce sous-ensemble. Ceci implique donc de construire un modèle pour chaque contexte. Pour illustrer cette approche prenons l'exemple d'un système de recommandation de films qui utilise le contexte temporel : si un utilisateur souhaite regarder un film pendant le weekend, seuls les films disponibles pendant le weekend sont candidats à la recommandation et seules les notes des utilisateurs ayant vu les films pendant le weekend sont utilisées pour la prédiction de notes. L'utilisation de ce filtrage *a priori* a été critiquée, car l'ensemble des données est réduit et peut créer des problèmes pour la prédiction de notes si le système ne dispose pas d'assez de données.

1.6.4.2 Post-filtrage contextuel

Dans une approche d'incorporation du contexte par post-filtrage contextuel, le système de recommandation ne prend pas en compte les données contextuelles lors du processus de recommandation. Les sorties des algorithmes de recommandation sont modifiées *a posteriori* pour réordonner la liste des items recommandés en fonction du contexte. Par exemple, un système recommandation de lieux touristiques utilisera la situation géographique de l'utilisateur (contexte de localisation), et peut décider d'éliminer *a posteriori* les recommandations de lieux trop éloignés de la position de l'utilisateur.

1.6.4.3 Modélisation directe du contexte

L'approche de modélisation du contexte consiste à intégrer directement les informations contextuelles dans le processus de recommandation pour la prédiction de notes pour les items. Pour intégrer le contexte, [Karatzoglou et al., 2010] proposent des méthodes de factorisation tensorielle. Pour ces méthodes, en plus des deux premières dimensions traditionnellement utilisées pour les items et les utilisateurs, chaque type de contexte est considéré comme une nouvelle dimension. La note n'est plus considérée comme une fonction avec les deux paramètres item et utilisateur, mais une fonction avec comme paramètres l'item, l'utilisateur et les aspects du contexte.

1.7 Conclusion

Dans ce chapitre, nous avons présenté les systèmes de recommandation qui sont devenus omniprésents ces dernières années dans de nombreux domaines. Ces

systèmes sont conçus pour aider les utilisateurs à trouver des ressources qui les intéressent et qui sont adaptées à leurs préférences, parmi le nombre important des choix qui s'offrent à eux.

Nous avons suivi la classification des systèmes de recommandation en deux approches principales : les approches de recommandation basées sur le contenu et les approches de filtrage collaboratif. Cette classification simple nous a permis le regroupement d'autres approches présentées parfois séparément. Les deux approches présentent néanmoins des caractéristiques complémentaires. Par conséquent un grand nombre de travaux se sont intéressé aux approches hybrides qui combinent plusieurs approches, et qui permettent de profiter des avantages, tout en limitant les inconvénients, de chaque méthode utilisée séparément. Ensuite, nous avons discuté l'importance du contexte dans le domaine des systèmes de recommandation en présentant les différentes sources d'informations contextuelles ainsi que les différents paradigmes d'incorporation du contexte pour les techniques de recommandation.

Pour le domaine de la visite de musées, nous avons constaté que la visite s'effectue généralement en trois étapes en fonction de la progression de la visite. Pour chaque étape une technique différente de recommandation doit être utilisée en fonction de l'enrichissement du profil de l'utilisateur pendant la visite. Nous nous sommes alors focalisés sur les approches hybrides qui combinent plusieurs approches de recommandation. Après avoir étudié les différentes techniques d'hybridation, nous avons constaté que celle par sélection est la technique la plus appropriée pour notre cas. En effet, cette technique est bien adaptée de manière générale à un système où l'utilisateur peut se trouver dans différentes situations.

En ce qui concerne le domaine du tourisme, l'objectif est de recommander à l'utilisateur des points d'intérêt qui sont susceptibles de l'intéresser. Cependant, ces points d'intérêt peuvent être de différents types (monument, musée, parc). La nature hétérogène des points d'intérêt rend inappropriée l'utilisation des systèmes de recommandation qui fournissent à l'utilisateur des recommandations sous forme de listes triées. Nous nous sommes alors focalisés sur les systèmes de recommandation composite, où la liste des recommandations est une liste de packages, chaque package étant constitué de plusieurs items.

Représentation des connaissances et similarités sémantiques

Sommaire

2.1 Introduction

Nous détaillons dans ce chapitre des éléments sur la représentation des connaissances et sur les mesures de similarité qu'il est possible de définir sur ce type de représentations. Ces éléments vont nous permettre dans la suite de notre travail de construire notre représentation du domaine, à savoir la représentation sémantique d'une œuvre lorsqu'il s'agit de la visite de musées ou bien la représentation d'un lieu d'activité (Point d'intérêt) lors d'une visite touristique, ainsi que les mesures de similarités qui seront utilisées pour pourvoir prédire dans quelle mesure une œuvre ou un Point d'intérêt peut être intéressant pour un utilisateur.

L'ingénierie des connaissances a pour objet l'acquisition, la modélisation et le traitement de connaissances dans des environnements informatiques. Cette discipline se distingue donc de l'informatique classique, centrée sur le traitement de données (structures les plus élémentaires) et d'informations (données structurées). En ingénierie des connaissances, les objets informatiques sont considérés sont des connaissances formalisées. Cette discipline se base sur les langages formels, permettant ainsi d'exprimer de manière formelle et computationnelle des savoirs du langage naturel [Bachimont, 2000]. La nature computationnelle des langages formels facilite considérablement la réalisation automatique d'inférences sur des problèmes

décrits. L'inférence est un outil puissant qui permet de déduire des connaissances nouvelles à partir de la base de connaissances et de la description initiale du domaine.

La manipulation directe de ces représentations de connaissances peut cependant s'avérer une tâche difficile, la représentation informatique donnant lieu à des objets complexes (ontologies, bases de connaissances, etc.). L'utilisation de similarités et proximités sémantiques permet de limiter en grande partie la complexité de la manipulation des connaissances formelles par l'utilisateur, ces similarités définissent en effet des métriques qui permettent d'associer et de grouper les connaissances entre elles.

Nous présentons dans ce chapitre les paradigmes dominants pour la représentation informatique des connaissances. Nous détaillons particulièrement les langages RDF, RDF(S) et OWL. Nous nous pencherons ensuite sur les différents types de mesures de similarité sémantiques.

2.2 La représentation des connaissances

2.2.1 Fondements logiques de la représentation des connaissances

Définir ce qu'est une connaissance n'est pas chose facile. Une recherche sur différents dictionnaires français donne plusieurs définitions pour " connaissance " dont les plus intéressantes :

- Fait, manière de connaître. La connaissance d'un objet : conscience ; compréhension, représentation.
- (Avoir connaissance de) : être informé de.
- Faculté de connaître propre à un être vivant : intelligence.
- Fait de sentir, de percevoir : conscience, sentiment.
- (Les connaissances) : ce qui est connu ; ce que l'on sait pour l'avoir appris.

La définition de la " connaissance " relève de domaines tels que la philosophie ou l'épistémologie. [Bachimont, 2004] propose la définition suivante "Une connaissance est la capacité d'exercer une action pour atteindre un but" ; Cette définition met en avant le caractère idéal de la connaissance, et l'importance de la finalité de la connaissance.

Définir les connaissances n'est pas chose facile. Dire ce qu'est une représentation de ces connaissances n'est donc pas évident non plus. Plusieurs définitions ont été

proposées pour la représentation informatique des connaissances. Selon [Guarino, 1995], une représentation des connaissances doit permettre de dénoter des objets et de décrire les relations entre eux. Pour [Levesque, 1986] il s'agit d'écrire une représentation d'une partie du monde de telle façon qu'une machine puisse parvenir à de nouvelles conclusions sur l'environnement réel en manipulant cette représentation. Suivant [Bachimont, 2000], les connaissances s'appréhendent à travers leurs inscriptions (c'est-à-dire leurs représentations), ces inscriptions étant de nature documentaire (destinées à une interprétation par l'humain) ou formelle (destinée à une interprétation par une machine). Représenter des connaissances d'un domaine a alors pour but de refléter ce domaine et permettre la réalisation d'opérations de raisonnement sur la représentation pour en déduire des conséquences sur le domaine. Prenons par exemple le domaine des musées, la représentation des connaissances muséales va nous permettre d'établir des faits du type, l'œuvre x est proche de l'œuvre y, ou bien l'œuvre x a été réalisée par l'artiste a. Pour cela, les représentations de connaissances portent en elles un ensemble d'engagements ontologiques, définissant les objets d'intérêts du domaine, leurs relations, les règles qui les lient entre eux, etc. Enfin, une représentation des connaissances définit de manière formelle et computationnelle les mécanismes d'inférence permis sur les objets du domaine [Davis et al., 1993].

Les modèles de représentation des connaissances reposent essentiellement sur des théories issues de la logique. En effet pour manipuler des connaissances explicites, un système doit utiliser un langage formel de représentation, le plus efficace possible. Ces langages formels de description des connaissances permettent la construction de concepts, de relations entre concepts, d'individus, de règles, ainsi que la description des mécanismes d'inférence permis. On distingue généralement quatre approches principales pour les langages de description des connaissances : les approches provenant de la logique du premier ordre, les approches provenant de la Frame Logic, les approches issues des logiques de description et enfin les approches issues du Web sémantique.

Les approches issues de la logique du premier ordre utilisent une formalisation du langage des mathématiques proposée par les logiciens à la fin du 19^{me} siècle. Ils se basent sur un ensemble de symboles appelés *variables* et un ensemble de symboles désignant des *prédicats*. Il est ainsi possible d'écrire des expressions du langage naturel en utilisant les connecteurs logiques (et, ou, etc.) et la quantification universelle ($\forall$) ou existentielle ($\exists$). La description du domaine (du monde) permise par ces langages consiste donc en un ensemble de "phrases" exprimées en logique du premier ordre. Cependant, ce type de représentation a été abandonné, en effet les langages purement logiques s'avèrent difficiles à manipuler pour la description des

connaissances de haut niveau. De plus la logique du premier ordre peut donner lieu à des bases de connaissances indécidables et donc à construire des systèmes qui ne pourront conclure sur la véracité d'une assertion.

[Minsky, 1974] introduit les langages basés sur la F-Logic (frame logic). Ces langages possèdent plusieurs points communs avec la modélisation orientée objet. Ils se basent sur les notions de frames (classes), sur la définition de liens de généralisation/spécialisation entre classes, ainsi que la définition des attributs des classes. On peut également définir des individus, ce qui équivaut en langage objet aux instances de classes. Les langages basés sur les frames sont particulièrement intéressants pour les informaticiens étant donné qu'ils reprennent les principes de la modélisation orientée objet. Néanmoins, ces langages basés sur la frame logic sont peu expressifs comparé aux langages utilisant les logiques de description [Baader et al., 2005]. Ainsi, l'utilisation de ces langages rend nécessaire l'écriture de procédures pour implémenter des mécanismes déclaratifs en logique de description [Gruber, 1993].

Les logiques de description aussi appelées logiques descriptives (LD) sont une famille de langages de représentation de connaissance qui peuvent être utilisés pour représenter la connaissance terminologique d'un domaine d'application d'une manière formelle et structurée. Le nom de logique de description se rapporte, d'une part à la description de concepts utilisée pour décrire un domaine et d'autre part à la sémantique basée sur la logique qui peut être donnée par une transcription en logique des prédicats du premier ordre. Les langages issus des logiques de description offrent une grande expressivité ainsi qu'une garantie de décidabilité. En particulier, ils permettent de définir des concepts en termes de contraintes à satisfaire pour qu'un objet soit une instance d'un concept (description intentionnelle). Nous citons ici l'exemple introduit par [Baader et al., 2005], qui décrit le concept d'un "homme heureux" par la formule suivante :

$$Human \sqcap \neg Female \sqcap \exists married.Doctor \sqcap (\geqslant 5 hasChild) \forall hasChild.Professor$$

L'interprétation est ainsi : un homme heureux est un humain de sexe masculin marié à un docteur et dont les enfants, au nombre d'au moins 5 sont tous des professeurs. Les logiques de description permettent ainsi d'exprimer de manière concise des concepts complexes. Par ailleurs, les logiques de description introduisent la distinction instance/concept via les notions de T-Box (terminological box), permettant la construction des concepts et de A-Box (assertional box) pour la description des individus. Une limite des logiques de description est liée à cette expressivité importante, l'utilisation de constructeurs et de quantificateurs logiques demandant une certaine expertise et pouvant s'avérer très complexe pour des connaissances de haut niveau.

Enfin, les langages du web sémantique reprennent les avantages des deux derniers, ils utilisent les éléments des langages issus des logiques de description pour la sémantique et des éléments issus de la F-Logic pour la syntaxe. La combinaison donne un langage à la fois relativement simple sur le plan syntaxique en plus de permettre d'exprimer les connaissances dans un formalisme proche de la modélisation orientée objet, et ils sont également particulièrement puissants sur le plan de l'expressivité. Nous décrivons dans la partie suivante les langages de description des connaissances basés sur les recommandations du World Wide Web Consortium (W3C[1]).

2.3 Ontologies et Web sémantique

2.3.1 Architecture du web sémantique

Le web sémantique est une idée relativement récente qui remonte à 2001. T. Berners-Lee caractérise le web sémantique comme étant *une extension du web actuel dans lequel on donne à une information un sens bien défini pour permettre aux ordinateurs et aux individus de travailler en coopération* [Berners-Lee et al., 2001]. Le W3C met en place les recommandations suivantes :

- description de web par des classifications précises, à l'aide d'ontologies exploitables par les machines et compréhensibles par les humains ;
- utilisation d'un langage commun pour exprimer les ontologies et décrire des annotations utilisant leurs termes ;
- création de moteurs de raisonnement permettant d'inférer sur les annotations d'après les axiomes déclarés dans les ontologies.

La notion de web sémantique fait donc référence à la vision du web dans lequel les utilisateurs devraient être déchargés d'une bonne partie de leurs tâches de recherche et ainsi d'exploitation des résultats, grâce aux capacités accrues des machines à accéder aux contenus des ressources et à effectuer des raisonnements sur ceux-ci [Laublet et al., 2002]. Ici, nous ne donnons qu'une vision assez générale de l'architecture du web sémantique. Pour une présentation en détail, les lecteurs peuvent se rapporter à [Gandon et al., 2012].

L'architecture du web sémantique s'appuie sur une pyramide de langages proposée par Tim Berners-Lee pour représenter des connaissances sur le web

[1] http ://www.w3c.org

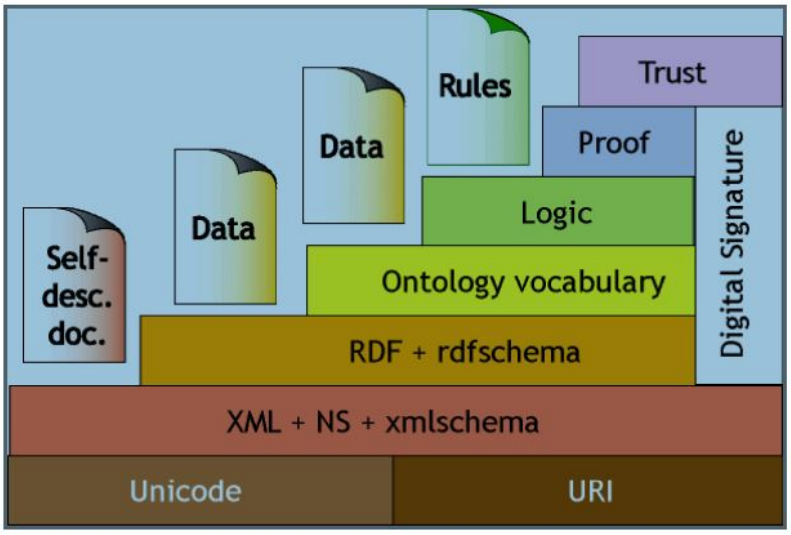


Figure 2.1 – Les couches du web sémantique [Berners-Lee et al., 2001]

en satisfaisant les critères de standardisation, interopérabilité et flexibilité. Cette architecture en couches communément appelée " Semantic web cake" (figure 2.1). Les niveaux inférieurs définissent l'alphabet (UNICODE) et la syntaxe (XML) utilisés pour les échanges, les niveaux supérieurs introduisent des représentations formelles de plus haut niveau (RDFS et OWL). Nous décrivons dans la partie suivante les langages basés sur RDF.

2.3.2 Ontologie

La définition d'une ontologie est tout d'abord apparue dans le domaine de la philosophie. Dans ce contexte, une Ontologie est une théorie à propos de la nature et de l'existant, des types de choses qui existent et des liens entre eux. Les chercheurs en informatique, plus précisément, dans le domaine de l'intelligence artificielle ont repris ce terme et l'ont adapté à leur propre jargon. La définition "informatique" la plus rencontrée d'une ontologie est certainement celle de Gruber [Gruber, 1993] : *une ontologie est la spécification explicite d'une conceptualisation partagée.* La conceptualisation partagée désigne la représentation des connaissances du domaine (conceptualisation) pour laquelle il existe un accord entre experts de ce domaine (partagée). Une ontologie est une spécification explicite de cette conceptualisation partagée, le terme spécification explicite renvoie au caractère formel de cette représentation, qui est donc écrite dans un langage formel. Une ontologie est donc une manière formelle de décrire un domaine de connaissance.

L'utilisation des ontologies s'est révélée être utile dans de nombreux domaines de l'informatique : représentation d'informations et de connaissances, intégration des systèmes d'informations et plus récemment les ontologies ont trouvées leur place

dans les domaines de la recherche d'information et des systèmes de recommandation pour pouvoir représenter les items à recommander de manière formelle et précise. Selon [Jasper et al., 1999], les objectifs de l'utilisation d'ontologies sont les suivants :

- *communication* (entre humains et/ou organisations) : le bénéfice d'utiliser une ontologie est l'absence d'ambiguïté. Dans une ontologie, il n'existe pas deux termes ayant la même signification sémantique. Cette situation est souvent rencontrée lors de l'utilisation du langage naturel pour la communication.
- *interopérabilité* : l'ontologie peut être utilisée comme un modèle intermédiaire pour la traduction entre les modélisations de différentes collections d'objets. l'ontologie définira alors le format d'échange entre les systèmes.

Dans le web sémantique, l'ontologie joue un rôle important pour faire communiquer les personnes et les machines, par échange de sémantiques et pas seulement de syntaxes. En effet, d'une part, une fois construite et acceptée par une communauté particulière, une ontologie doit traduire un certain consensus explicite et un certain niveau de partage qui sont essentiels pour permettre l'exploitation des ressources du web par différentes applications ou agents logiciels. D'autre part, la formalisation, autre facette importante des ontologies, qui est nécessaire pour que ces outils puissent être muni de capacités de raisonnement permettant de décharger les différents utilisateurs d'une partie de leurs tâche d'exploitation et de combinaison des ressources web.

Cependant, l'exploitation des ontologies présentent plusieurs inconvénients. On peut citer par exemple les problèmes liés à l'évolution et à la maintenance des ontologies. En effet, les ontologies de domaine doivent être maintenues pour faire face aux incomplétudes et aux erreurs, ou encore s'adapter aux innovations dans le domaine [Xuan et al., 2006]. Il est nécessaire en particulier de propager les changements au niveau des artefacts dépendants, c'est-à-dire au niveau des objets référencés par l'ontologie ainsi que des ontologies et des applications qui lui sont liées [Noy and Klein, 2004].

2.3.3 Les langages ontologiques du web sémantique

Notre description se limitera ici aux langages basés sur RDF (Resource Description Framework). Un langage ontologique définit une syntaxe et une sémantique pour la description des ontologies et des bases de connaissances. La syntaxe du langage définit les notations admissibles et la sémantique du langage permet la réalisation d'inférences, c'est-à-dire la dérivation automatique de connaissances nouvelles à partir des assertions d'une ontologie ou d'une base de connaissances.

Figure 2.2 – Triplet RDF

2.3.3.1 RDF (Ressource Description Framework)

RDF est un dialecte XML, ce qui signifie que RDF peut s'écrire en XML mais aussi avec d'autres syntaxes telles que *N3*, *N-triples* ou encore *Turtle*. RDF est un modèle conceptuel normalisé par le W3C permettant de décrire des ressources de manière très simple et sans ambiguïté. L'écriture est sous la forme de triplets qui consiste en des déclarations <*Sujet*> <*Prédicat*> <*Objet*> (on remarque la proximité avec le langage naturel et le triplet sujet-verbe-complément) [Gandon et al., 2012]. RDF et XML sont complémentaires, RDF comme méta-modèle de données basées sur XML, spécifie leurs sémantiques de manière standardisée et inter-opérable. La syntaxe XML permet donc l'encodage, le transport et le stockage de fichiers RDF. Ce système repose sur trois piliers : **RDF**, description de ressources web (méta-données) ; puis **RDF Schéma**, vocabulaire de description (ontologies) ; et enfin une **syntaxe** (XML) pour l'échange des méta-données et des schémas.

Comme le montre la figure 2.2, la composition fondamentale de toute expression en RDF est une collection de triplets sous la forme <*Sujet*> <*Prédicat*> <*Objet*>. Chaque triplet est représenté par un arc prédicat orienté du nœud source sujet vers le nœud destination objet. L'ensemble de tout les triplets RDF forme un graphe orienté, appelé *graphe* RDF.

RDF travaille avec des données élémentaires : les ressources, les propriétés et les valeurs littérales. Précisément, le sujet en RDF (ce sur quoi porte la déclaration) est nécessairement un objet de type ressource, entendons par là toute chose pouvant être référencée par une URI (Uniform Resource Identifier). Quant au prédicat, il doit être de type propriété. Il est lui-même identifié par une URI. Chaque propriété possède une signification bien précise qui donnera la sémantique de description. Enfin l'objet peut être soit une autre ressource mais aussi simplement une chaîne de caractères appelée "littéral".

Prenons un exemple pour illustrer tout ça. Supposons qu'on a un graphe de deux triplets correspondants aux assertions suivantes :

(http ://example.com/jean , family : hasSon , http ://example.com/jacque)
(http ://example.com/jacque , family : hasWebPage , "http ://jacque.com")

Ces deux triplets expriment la description en langage naturel suivante : jean

a comme fils jacque et jacque a une page web "http :jacque.com", les différentes représentations de ces deux triplets utilisant RDF/XML, N3 et la représentation graphique sont décrites dans la figure 2.3.

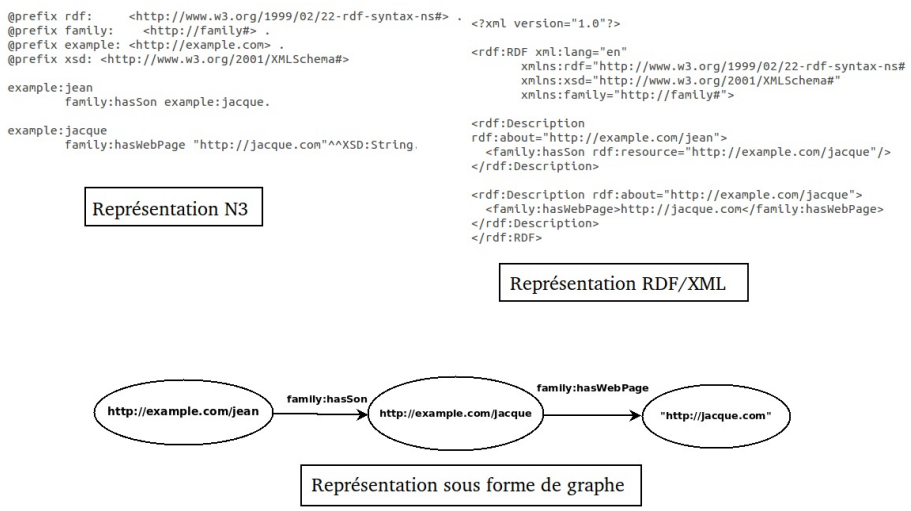


Figure 2.3 – Différentes représentations des mêmes assertions RDF

On peut alors synthétiser RDF par les règles suivantes :

- RDF peut être utilisé pour représenter tout objet ;
- RDF peut être traité par une machine ;
- RDF est composé de triplet (sujet,prédicat,objet) ;
- le sujet et le prédicat sont toujours identifiés par une URI ;
- l'objet est soit une URI, soit une valeur explicite (littéral) ;
- RDF peut être représenté en XML.

2.3.3.2 RDFs (Resource Description Framework Schema)

RDFs est la première extension de RDF. Il propose un modèle de description de vocabulaires RDF, à bases de classes et de propriétés. Parmi ces classes et propriétés nous trouvons :

- la classe *Class* : un ensemble de plusieurs objets (par exemple la classe des véhicules) ;
- la propriété *subClassOf* : permet de définir qu'une classe est sous-ensemble d'une autre classe (par exemple la classe voiture est sous-classe de véhicule) ;

- la classe *Resource* : qui est la classe parente de toute chose, en sachant que tout est ressource sauf la classe *Literal* (valeur typée) ;
- la propriété *range* : permet d'indiquer le champs d'application d'une propriété ;
- la propriété *domain* : permet de spécifier les classes auxquelles nous pouvons affecter une propriété.

RDFS décrit alors les ressources en termes de *classe, propriétés et valeurs*. Ce système est donc très similaire aux classes des langages de programmation orienté objet ; ce qui permet aux ressources d'être définies comme des instances. Notons que deux notions fondamentales de RDFS qui sont la subsomption et la restriction des valeurs que prennent les propriétés sont très importantes dans le rôle de la modélisation et la réalisation des inférences sur une base de connaissances.

2.3.3.3 OWL (Web Ontology Language)

OWL est, tout comme RDF, un langage XML profitant de l'universalité syntaxique de XML. Fondé sur la syntaxe de RDF/XML, OWL offre un moyen d'écrire des ontologies web. OWL se différencie du couple RDF/RDFS en ceci que, contrairement à RDF, il est justement un langage d'ontologies.

Il existe trois versions de OWL, qui sont, par expressivité croissante : OWL-Lite, OWL-DL et OWL-Full. En pratique on utilise essentiellement OWL-DL car l'expressivité de OWL-Lite est faible et OWL-Full est indécidable. La différence essentielle entre RDFS et OWL tient au fait que la sémantique de OWL est basée sur une logique de description. Ce langage permet donc une expression beaucoup plus précise des représentations de connaissances, avec en particulier les contraintes de cardinalité sur les propriétés, l'existence du quantificateur existentiel pour les propriétés d'un concept (c'est-à-dire qu'un concept doit posséder cette propriété), la définition de propriétés transitives, inverses et fonctionnelles ainsi que la définition intentionnelle des classes (ex. définir la classe A comme la restriction de la classe B pour une valeur donnée d'une propriété). Ainsi, OWL offre aux machines une plus grande capacité d'interprétation du contenu web que RDF et RDFS, grace à un vocabulaire plus large et à une vraie sémantique formelle.

2.3.4 Conclusion sur la représentation des connaissances

Nous avons présenté dans la première moitié de ce chapitre la représentation informatique des connaissances ainsi que différentes techniques qui sont utilisées pour ces représentations. Nous avons étudié les recommandations du W3C : RDF,

RDFS, OWL. Ces langages de représentation des connaissances utilisent une syntaxe basée sur la F-Logic et la représentation des ressources par leurs URI, permettant ainsi une grande flexibilité dans les descriptions.

Le choix des formalismes RDFS et OWL dans la suite de notre travail n'est donc pas simplement dû à des questions techniques, mais également au fait que les langages issus du web sémantique sont conçus pour le traitement et la réutilisation des connaissances déjà existantes et formalisées.

2.4 Mesures de similarité sémantiques

Avec l'avènement du web des données, lié à l'essor considérable des capacités de stockage et de traitement des données par les outils informatique, la question de la comparaison d'entités a connu un nouveau regain d'intérêt. Les mesures de similarité sémantiques sur lesquelles s'appuient les comparaisons doivent non seulement s'adapter aux nouveaux usages, comme la recherche d'information ou les systèmes de recommandation, mais aussi pouvoir supporter un passage à l'échelle sur des volumes de données de l'ordre de centaines de milliers de concepts. Les similarités sémantiques sont donc très utilisées dans un large panel de domaines de recherche : recherche d'information, data-mining et systèmes de recommandation. Cependant, la manipulation directe de représentations formelles de connaissances pour déterminer les similarités est relativement difficile. En particulier, la détermination de concepts ou d'instances en lien ou proches d'autres ressources nécessite des requêtes complexes.

Plusieurs méthodes ont été proposées dans la littérature pour faciliter la détermination automatique des similarités, tant au niveau des concepts qu'au niveau des instances et d'individus. Nous définissons tout d'abord formellement les notions de similarités sémantiques. Nous proposons ensuite une classification des similarités sémantiques suivant la formalisation utilisée pour décrire les concepts. On distingue en général trois catégories :

- les mesures de type structurel ;
- les mesures de type intentionnel ;
- les mesures de type expressionnel.

Pour définir ces similarités dans la suite de ce chapitre, nous utiliserons les notations suivantes :

- $depth(c_i)$: la profondeur du concept c_i dans la hiérarchie de concepts ;

- $lcs(c_i, c_j), leastcommomsubsummer$: le plus petit concept père commun aux concepts c_i et c_j ;
- max : la profondeur maximale de la hiérarchie ;
- $dist_{edge}(c_i, c_j)$: la longueur du plus court chemin entre les concepts c_i et c_j ;

2.4.1 Définition

Une mesure de similarité est, en général, une fonction qui quantifie le rapport entre deux objets comparés en fonction de leurs points de ressemblance et de dissemblance. Les deux objets comparés doivent être de même type. Il faut noter que toutes les mesures de similarité ne sont pas des métriques. En mathématiques, une métrique ou distance est une fonction à valeurs dans l'ensemble $\mathbb{R}$ des nombres réels qui définit la distance entre les éléments d'un ensemble X, tel que $d : X \times X \to \mathbb{R}$.

Pour être une métrique, une mesure d doit satisfaire les 4 conditions suivantes :

Soient x, y et z, trois éléments d'un ensemble, et soit d(x, y) la distance entre x et y.

- Positivité : $d(x, y) \geq 0$.
- Principe d'identité des indiscernables : $d(x, y) = 0 \equiv x = y$.
- Symétrie : $d(x, y) = d(y, x)$.
- Inégalité triangulaire : $d(x, z) \leq d(x, y) + d(y, z)$.

2.4.2 Similarité ou proximité ?

Deux catégories de mesures qui respectent la définition donnée, permettent de capturer la notion de distance ou d'éloignement entre des ressources d'une ontologie : les similarités et les proximités. La similarité entre deux objets quantifie de manière numérique la ressemblance entre ces deux objets. Prenons un exemple pour illustrer cette notion, le domaine des véhicules. Une mesure de similarité doit établir que les concepts de *camion* et de *voiture* sont similaires, ce sont tous les deux des véhicules avec quatre roues qui peuvent être utilisés pour se déplacer et transporter ce dont on a besoin. La similarité sémantique est donc une similarité dans la structure des concepts, deux concepts similaires pouvant être, dans une certaine limite, échangeables dans une phrase sans changer complètement le sens de la phrase.

Les mesures de proximité quantifient quant à elles les associations entre concepts. De ce fait, deux concepts de nature différente peuvent avoir une forte proximité.

Toujours avec notre exemple sur les véhicules, nous remarquons que les deux concepts de *voiture* et *chauffeur* sont proches sémantiquement, ils sont cependant différents pour la notion de similarité. On ne pourra généralement pas dans une phrase remplacer un concept par un concept proche sans changer fortement le sens de la phrase. Les concepts de proximité et similarité sémantique sont donc proches mais ne sont pas similaires [Aimé et al., 2011]. On peut cependant noter que dans beaucoup de cas des concepts ayant une forte similarité sémantique auront une forte proximité sémantique, mais ce n'est pas toujours vrai. Dans la suite de notre travail, on parlera de similarité sémantique lorsque nous comparons deux entités de même nature, par exemple deux œuvres, ou bien deux points d'intérêt, et nous parlerons de proximité sémantique lorsque nous comparons deux entités de concepts différents, par exemple, nous parlerons de proximité entre un artiste et une oeuvre, ou bien une oeuvre et une époque.

2.4.3 Mesures de similarités sémantiques

2.4.3.1 Mesures de type structurel (basées sur les hiérarchies de concepts)

A. Collins et M. Quillian, un psychologue et un informaticien, ont élaboré les premiers réseaux sémantiques sur la base du temps de réponse à des questions du type "*Un canari est-il un oiseau ?*" ou encore *"Un canari est-il un animal ?"* [Collins and Quillian, 1969]. Intuitivement, ils ont supposé que plus le temps de réponse était long pour la réponse, plus les deux concepts étaient plus distants l'un de l'autre. En terme de combinatoire ils ont modélisé cette interprétation de leurs tests par un nombre d'arcs plus élevé dans une hiérarchie entre concepts distants qu'entre concepts similaires.

Mesure de Rada

Les premières approches de similarité sémantique, basées sur des hiérarchies de concepts sont issues des travaux de [Rada et al., 1989]. Le principe de cette approche est de considérer que deux concepts sémantiquement proches sont également proches dans l'ontologie, en terme de parcours d'arcs. De ce fait, la similarité entre deux concepts est décrite comme étant la longueur du plus court chemin qui les sépare. Le mesure de similarité de Rada [Rada et al., 1989] se calcule donc en mesurant les arcs séparant les deux concepts. Plus précisément, la distance de Rada entre deux concepts C1 et C2 est la somme de ces arcs :

$$sim_{Rada}(c_1, c_2) = dist_{edge}(c_1, c_2) \tag{2.1}$$

La distance de Rada entre deux concepts pose cependant un problème. En effet, elle ne tient pas compte de la différence de spécificité que peuvent prendre les liens is-a. En effet, suivant sa position dans l'ontologie un lien is-a porte plus ou moins d'information, typiquement les liens les plus spécifiques (situés en bas de la hiérarchie de concepts) sont plus porteurs d'information que les liens les plus génériques. Par exemple, l'application de la mesure de Rada pour une taxonomie de la nature, donne une même similarité pour plante et animal que pour zèbre et cheval, alors qu'il est clair que zèbre est sémantiquement plus proche de cheval que plante d'animal.

D'autres types d'approches, basées sur une pondération des liens hiérarchiques, ont alors été proposées. Les premières approches [Whan Kim and Kim, 1990, Ho lee et al., 1993], se sont basées sur une annotation manuelle des arcs de l'ontologie à l'aide de poids, les chercheurs donnant une estimation de la force du lien. Ces approches sont appropriées pour des hiérarchies de petites tailles. Cependant ces approches deviennent très contraignantes lorsque la hiérarchie atteint une taille importante.

Mesure de Richardson

[Richardson et al., 1994] ont proposé un calcul automatique appliqué à la hiérarchie de concepts Wordnet (une taxonomie comportant l'essentiel des termes de langue anglaise) et prenant en compte la densité locale de concepts et la profondeur des concepts comparés pour assigner des poids aux liens de l'ontologie. Le principe de cette méthode est que les distances associées aux arcs sont plus faibles dans les parties denses de la hiérarchie de concepts (les parties où les concepts possèdent de nombreux frères) et de la même manière les distances associées aux arcs sont plus faibles lorsque les concepts sont profonds dans l'ontologie. La méthode de calcul proposée prend également en compte le contenu informationnel des concepts. L'idée étant qu'un concept ayant un contenu informationnel important est un concept "rare", que l'on a peu de chance de rencontrer, et donc que les liens sémantiques sont plus forts entre les concepts ayant des contenus informationnels moins importants. Ceci est illustré en figure 2.4 où le concept Life_form est plus fortement lié aux concepts Animal, Person et Plant qu'aux concepts Aerobe et Plankton, ces derniers étant relativement rares.

On notera que la similarité proposée par [Richardson et al., 1994] n'est pas juste limitée à une hiérarchie de concepts car elle fait intervenir le contenu informationnel.

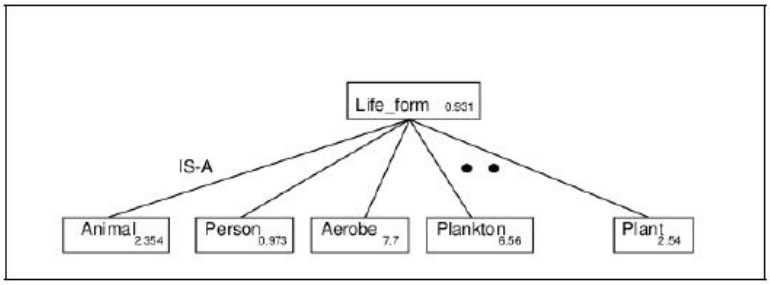


Figure 2.4 – Les contenus informationnels de quelques concepts de Wordnet

Nous avons cependant choisi de la faire figurer dans cette section car elle se situe dans la continuité directe des mesures basées sur les calculs d'arcs dans les hiérarchies. On notera aussi que les auteurs ne donnent pas d'expression formelle et claire de la similarité qu'ils proposent, ce qui limite l'applicabilité de leur proposition. Aussi, l'assignation des poids pour les différents concepts peut s'avérer trop contraignant pour des hiérarchies de très grande taille.

Mesure de Resnik

[Resnik, 1995] complète la mesure de Rada pour prendre en compte la profondeur maximale de la hiérarchie (max). la similarité entre les concepts c_1 et c_2 est égale au ratio entre la profondeur maximale et le plus court chemin entre ces concepts. la similarité de Resnik entre deux concepts c_1 et c_2 est définie par :

$$sim_{Resnik}(c_1, c_2) = \frac{2 \times max}{dist_{edge}(c_1, c_2)} \tag{2.2}$$

Mesure de Leacock

[Leacock et al., 1998] normalise la mesure de Resnik au moyen de la fonction log de manière à obtenir des similarités dans l'intervalle $[0, 1]$. avec une valeur proche de 0 pour des concepts distincts, et une valeur de 1 pour des concepts totalement similaires. La similarité entre les concepts c_1 et c_2 est calculée comme le ratio entre le plus court chemin entre ces concepts et la profondeur maximale de la hiérarchie. Plus formellement :

$$sim_{Leacock}(c_1, c_2) = -log(\frac{dist_{edge}(c_1, c_2)}{2 \times max}) \tag{2.3}$$

Mesure de Wu et Palmer

[Wu and Palmer, 1994] proposent une mesure de similarité plus complète prenant en compte à la fois (1) la profondeur des concepts comparés dans la hiérarchie et (2) la structure de cette hiérarchie par la proximité relative de leur père commun. Formellement la similarité entre les concepts c_1 et c_2 est :

$$sim_{Wu}(c_1, c_2) = \frac{2 \times depth(lcs(c_1, c_2))}{depth(c_1) + depth(c_2)} \tag{2.4}$$

Cette formule se comprend assez facilement. Plus les concepts ont un subsumant commun profond dans la hiérarchie, plus la similarité sera importante. De plus, plus les concepts sont proches de ce subsumant commun, donc plus $depth(c_1) + depth(c_2)$ est faible, plus les concepts sont proches.

Conclusion sur les mesures basées sur les hiérarchies

Nous avons présenté les principales méthodes de calcul de similarité basées sur des hiérarchies de concepts. Ces mesures seront utiles dans la suite de notre travail pour comparer certaines données issues soit du domaine des musées ou du tourisme, exprimées sous forme de hiérarchies (ex. hiérarchie de styles, hiérarchie de thèmes). Ces mesures ne permettent cependant pas de capturer tous les éléments des similarités sémantiques. Nous poursuivons donc dans la section suivante notre description des similarités sémantiques avec les approches basées sur les propriétés.

2.4.3.2 Mesures de type intentionnel (basées sur les propriétés des concepts)

Dans les approches basées sur les propriétés, l'espace D dans lequel sont comparés les concepts est plus complexe que dans les approches basées sur les hiérarchies. Outre les relations de généricité/spécificité entre concepts (relation is-a), des propriétés sont définies qui associent les concepts entre eux (objectProperty) ou associent des concepts à des classes de littéraux (datatypeProperty).

Le principe général des approches basées sur les propriétés des concepts est que deux concepts qui partagent de nombreuses propriétés sont similaires. Prenons par exemple l'ontologie des transports, le concept de voiture et celui de camionnette sont similaires, car ces concepts partagent les propriétés d'avoir quatre roues, un conducteur unique, une valeur de plaque d'immatriculation, etc. Les concepts liés entre eux par des relations sont quant à eux proches ; par exemple il existe une relation fonctionnelle entre le concept de voiture et le concept de conducteur, ces concepts sont donc proches. L'avantage des approches basées sur les propriétés est alors de pouvoir à la fois calculer des mesures de similarité ainsi que des mesures de

proximités.

La mesure la plus simple a été proposé par [Tversky, 1977]. Il utilise pour ça les travaux sur les ensembles. D'un point de vue ensembliste, deux entités sont similaires si le cardinal de l'intersection des ensembles de leurs caractéristiques est plus grand que celui des sous-ensembles restants. Il définit ainsi la similarité comme suit :

$$sim_{tversky}(A, B) = \alpha.comm(A, B) - \beta.diff(A, B) - \gamma.diff(B, A) \tag{2.5}$$

avec $comm(A, B)$ le cardinal de l'intersection des propriétés de A et B (propriétés communes) et diff(A,B) le cardinal de l'ensemble des propriétés de A privé des propriétés de B (propriétés uniques à A) (figure 2.5).

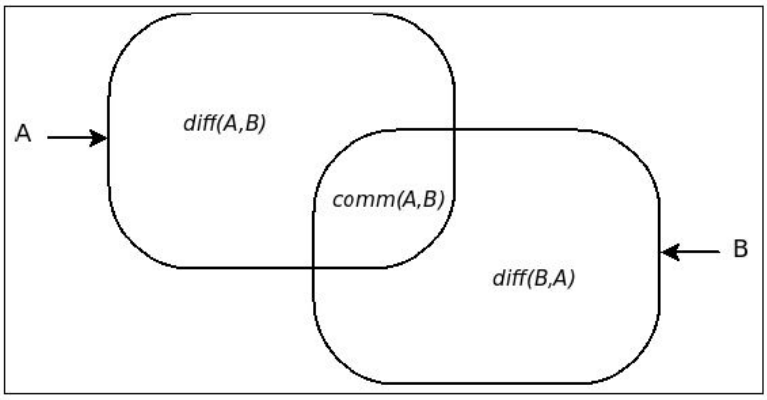


Figure 2.5 – Représentation des similarités dans le modèle de Tversky

On notera que le modèle de similarité de Tversky ne respecte pas la contrainte de symétrie. Ce modèle a pour objectif de représenter la perception humaine de la similarité et, suivant les travaux de Tversky cette perception n'est pas symétrique.

La similarité de Trevsky a été reprise par [Poitrenaud, 1998] pour définir une distance sémantique entre deux catégories par le nombre de propriétés partagées par ces deux catégories au sein d'une hiérarchie conceptuelle. En adoptant le principe de différentiabilité (c'est à dire, héritage des propriétés du concept père auxquels sont ajoutés ses propres caractéristiques), il faut remonter dans la hiérarchie pour obtenir le plus petit concept père partageant les propriétés communes aux deux concepts comparés, de ce fait plus la distance entre les deux concepts est grande plus la différence conceptuelle est grande.

[Rodríguez and Egenhofer, 2003] proposent une interprétation du modèle de similarité de Tversky pour la mesure de similarité entre concepts de différentes ontologies. Les concepts sont comparés suivant différents aspects, en particulier suivant les *features* (propriétés) communes ou distinctes de ces concepts. Cette

mesure se base sur une distinction structurelle entre catégories de propriétés : propriétés de composition, propriétés fonctionnelles et attributs. Les propriétés de type composition correspondent aux méronymies, par exemple le concept de voiture est en relation de composition avec le concept de roue. Les propriétés fonctionnelles relient les concepts à des fonctions (typiquement représentées par des verbes), par exemple le concept de voiture est en relation fonctionnelle avec le concept conducteur. Enfin, les attributs correspondent aux datatype Properties. La formule générale de la similarité par propriété est alors :

$$S_{features}(a,b) = w_p.S_p(a,b) + w_f.S_f(a,b) + w_a.S_a(a,b) \tag{2.6}$$

Avec S_p, S_f et S_a les fonctions de similarité respectivement pour les propriétés de composition, fonctionnelles et d'attributs, et w_p, w_f et w_a leur poids respectifs. Pour chacune des similarités $S_i(a,b)$ ils proposent l'utilisation d'une forme normalisée de la distance de Tversky.

L'avantage de cette formulation de la similarité est qu'elle peut facilement s'adapter à plusieurs types d'ontologies. Par exemple, si on veut considérer une ontologie ne présentant que des relations de type **part-of**, il suffira dans ce cas de mettre les poids des similarités d'attributs et fonctionnelles a une valeur nulle. Cependant, son utilisation présente deux inconvénients, il n'est pas chose facile de distinguer de manière automatique les relations de type part-of des relations fonctionnelles et ensuite quelle méthode utiliser pour attribuer les poids w_i ?

La similarité précédente peut s'interpréter comme l'agrégation d'un vecteur de similarités suivant trois dimensions : une similarité suivant les propriétés, une similarité suivant les relations et une similarité selon l'axe fonctionnel, la fonction d'agrégation étant une moyenne pondérée. Ce type de mesure de similarité par agrégation d'un vecteur de similarité a été proposé par [Bergmann and Stahl, 1998] pour le calcul de similarités sur différents objets informatiques. Le principe de cette similarité est de considérer le *least common subsummer* des deux classes et ainsi comparer ces classes en fonction des propriétés de leur *least common subsummer*, qui ne comporte que des propriétés communes aux deux classes. Formellement, la mesure générale proposée par les auteurs est la suivante :

$$sim(q,c) = \Phi(sim_{A_1}(q.A_1, c.A_1), sim_{A_2}(q.A_2, c.A_2)...sim_{A_N}(q.A_N, c.A_N)) \tag{2.7}$$

Avec q et c deux classes, les $\{A_1...A_N\}$ les attributs du *least common subsummer* de q et c et les $\{sim_{A_1}...sim_{A_N}\}$ des fonctions de similarités propres à ces attributs, $\Phi()$ étant une fonction d'agrégation.

Cependant, [Bergmann and Stahl, 1998] ne proposent pas de critères généraux pour les calculs des sim_{A_i}. [Aimé, 2011] a proposé ainsi une approche similaire qui se base sur la définitions de vecteurs prototypes. La formule générale proposée par [Aimé, 2011] est la suivante :

$$sim_{Aime}(c_1, c_2) = 1 - dist(\overrightarrow{p_{c_1}}, \overrightarrow{p_{c_2}}) \tag{2.8}$$

$\overrightarrow{p_{c_i}}$ étant le vecteur prototype du concept C_i, en d'autres termes le vecteur ayant pour valeur de composante k l'importance de la k^{ime} propriété pour ce concept.

Il est à noter que les calculs basés sur les propriétés permettent également de déterminer la proximité entre concepts. [Aimé, 2011] propose donc une mesure qui permet de quantifier la proximité entre deux concepts C_1 et C_2, son idée se base sur le fait de considérer le nombre de liens existants dans l'ontologie entre les concepts C_1 et C_2. Intuitivement, le calcul de la proximité correspond alors à la fraction du nombre de concepts liants C_1 et C_2 et C_2 et C_1 sur le nombre total de relations impliquant C_1 et C_2, ainsi deux concepts ayant de nombreuses autres relations auront une faible proximité alors que deux concepts ayant uniquement des relations l'un vers l'autre auront une forte proximité.

Nous pouvons constater que l'utilisation de propriétés dans le calcul des similarités apporte une grande richesse par rapport à l'utilisation unique de la hiérarchie de concepts, au prix de l'introduction de plusieurs paramètres à fixer, comme la pondération des propriétés. Ces mesures sont par ailleurs plus adaptées au cadre actuel du web sémantique, dans lequel les objets possèdent quasiment toujours différents types de propriétés.

2.4.3.3 Mesures de type expressionnel (basées sur les corpus)

Les concepts peuvent également être comparés sur un plan expressionnel, avec les termes qui les dénotent, à l'aide notamment du contenu en information (CI) qui a été défini par [Resnik, 1993]. C'est une mesure issue de la théorie de l'information qui permet de mesurer de manière numérique la quantité en information portée par un concept. Le contenu informationnel de l'événement A est défini comme suit : $IC(A) = -log(P(A))$.

Dans le cadre de ces approches, les concepts des ontologies ont des représentants dans différents documents (corpus). Le principe fondamental de ces approches est de déduire la similarité entre deux concepts en analysant les co-occurrences de leurs représentants dans les documents.

Mesure de Resnik

Les premières approches de l'utilisation du contenu informationnel étaient basées sur la hiérarchie wordnet. [Resnik, 1995] calcule alors le contenu en information d'un concept c en se fondant sur la probabilité $p(c)$ d'avoir ce concept dans un corpus donné. Ce contenu en information est défini par :

$$\Psi(c) = -log(p(c)) \tag{2.9}$$

avec

$$p(c) = \frac{\sum_{t \in monde(c)} count(t)}{N} \tag{2.10}$$

Où :

- N représente le nombre total d'occurrences des termes de tous les concepts dans le corpus ;
- $monde(c)$ représente l'ensemble des termes possibles pour le concept c mais également pour l'ensemble de ses descendants dans la hiérarchie.

Plus le concept est générique plus son contenu en information est faible, c'est à dire qu'il apporte peu d'informations. Inversement plus il est spécifique plus son contenu en information est fort.

Pour établir la similarité entre deux concepts, [Resnik, 1995] propose de calculer la valeur du contenu informationnel qu'ils partagent, en d'autres termes le contenu informationnel de leur *least common subsummer*. Formellement, la similarité entre deux concepts c_1 et c_2 est définie par :

$$sim_{res} = \psi(lcs(c_1, c_2)) \tag{2.11}$$

Mesure de Lin

Lin [Lin, 1998] a proposé une approche dans la lignée des travaux de [Resnik, 1995]. Son idée est de tenir compte à la fois du contenu informationnel commun des deux concepts mais également des caractéristiques propres à chacun d'eux. La similarité entre deux concepts c_1 et c_2 est définie par :

$$sim_{Lin} = \frac{2 \times \psi(lcs(c_1, c_2))}{\psi(c_1) + \psi(c_2)} \tag{2.12}$$

Les approches basées sur les corpus sont intéressantes car elles permettent de faire intervenir d'autres sources de connaissances (les corpus) dans le calcul de la

similarité. De nombreux corpus existants peuvent ainsi être facilement réutilisés. Une limite de ces méthodes est qu'il sera plus difficile de justifier de la similarité de deux concepts en utilisant une similarité par corpus. En effet, dans le cas des similarités sur des hiérarchies ou sur des propriétés, il est possible de justifier les similarités en considérant par exemple les propriétés communes. Les méthodes par corpus, plus calculatoires, rendent plus difficile une justification directe.

2.4.4 Conclusion

Nous avons décrit les différentes méthodes existantes pour le calcul de similarités sémantiques, suivant l'espace de définition des concepts. L'une des constatations que l'on peut faire est que les mesures que nous avons étudiées sont complémentaires mais pas concurrentes. En effet, les mesures de similarité utilisant la structure hiérarchique permettent de comparer la position de deux concepts dans une ontologie, les mesures de similarités utilisant les propriétés permettent de comparer la structure de ces concepts, enfin, les mesures de type expressionnel qui se basent sur des corpus, permettent d'intégrer des sources extérieures à l'ontologie étudiée. Plusieurs travaux ont essayé de comparer ces différentes mesures pour savoir quelle était la meilleure, on peut par exemple citer [Warin et al., 2005], qui considère que la mesure de Leacock est la plus proche du jugement humain. Cependant ces tentatives de comparaison se limitent en général aux mesures basées sur les hiérarchies de concepts, les autres mesures comportant trop de paramètres pour pouvoir être comparées de manière simple.

Dans le cadre de notre travail, l'utilisation de similarités sémantiques a pour objectif de pouvoir quantifier la ressemblance entre deux entités, par exemple la similarité entre deux œuvres, deux artistes, deux styles dans le cadre de la visite de musée, ou bien deux points d'intérêt dans le cadre de la visite touristique. Ce type de similarités se retrouve essentiellement dans les calculs basés sur la hiérarchie de concepts ou les calculs basés sur les propriétés. Nous utiliserons donc dans le chapitre 4 et 5 ces types de similarités. Le but est pouvoir faire des recommandations pour un utilisateur en se basant sur la similarité entre les items candidats que l'utilisateur n'a pas encore vus et les items qu'il a aimés.

Systèmes de recommandation pour l'aide à la visite culturelle

Sommaire

3.1 Introduction

La recherche sur les systèmes d'aide à la visite de musées et de sites culturels a été particulièrement importante ces dernières années. En effet, le musée n'est plus seulement un lieu pour la conservation et l'affichage des œuvres, mais aussi une institution pour l'éducation et le divertissement des visiteurs. Avec l'avènement du web, les conservateurs et les médiateurs culturels ont décidé de se mettre à la culture du numérique dans le but d'attirer l'attention d'un public plus large. Aujourd'hui, presque chaque musée possède un site web mais en général il ne contient pas la totalité des œuvres disponibles dans l'espace réel. Le site web permet d'aider le visiteur à prendre connaissance des œuvres disponibles dans le musée avant de faire sa visite. Il peut ainsi avoir une première idée sur le parcours qu'il veut effectuer.

De nombreux travaux de recherche ont également vu le jour dans le but d'assister le visiteur pendant sa visite. On trouve trois types de travaux, tant d'abord des travaux centrés sur des " tâches " à effectuer dans le musée, qui sont typiquement destinés à des groupes scolaires dans le but d'informatiser les activités de visite. On trouve ensuite des travaux centrés sur la navigation dans l'espace physique du musée, qui sont destinés en général à un visiteur adulte, dans le but de fournir un accès alternatif aux collections et de proposer des informations d'intérêt pour le

visiteur en temps réel. Enfin, d'autres travaux visent à personnaliser la visite de musée, c'est-à-dire à suggérer au visiteur des œuvres qui sont le plus susceptibles de l'intéresser. Les systèmes de recommandation peuvent être très utiles pour cela. Notre travail se situe dans cette dernière catégorie.

Dans le domaine du tourisme, les systèmes informatiques d'aide à la visite se développent aussi de plus en plus. En effet, pendant la dernière décennie, la façon dont les touristes préparent leur visite à considérablement évolué. En effet, plus de 74 % des visiteurs utilisent Internet comme une source d'information pour la planification de leur visite [Borras et al., 2014]. Les visiteurs passent beaucoup de temps en ligne à explorer des alternatives sur ce qu'il faut voir ou ne pas voir pendant la visite touristique d'une ville. Une autre étude (TripAdvisor 2015) montre que 67 % des visiteurs utilisent le site web Tripadvisor qui fournit un détail des différents points d'intérêt disponibles dans une ville, ainsi que l'avis des visiteurs sur ceux-ci. Cependant, l'utilisation d'Internet de manière générale, ou bien des sites spécialisés sur le tourisme, ne répond pas complètement aux besoins des visiteurs. Il est nécessaire de personnaliser les visites en fonction du profil de l'utilisateur. Un nombre important de travaux de recherche s'est développé dans ce but. Ces travaux sont souvent basés sur des systèmes de recommandation.

Dans ce chapitre, nous présentons un état des travaux concernant les systèmes d'aide à la visite de manière générale, en nous intéressant plus particulièrement aux travaux qui utilisent des systèmes de recommandation pour personnaliser les visites.

3.2 Systèmes pour l'assistance à la visite de musées

3.2.1 Systèmes orientés tâche

Les approches orientées tâches sont les systèmes de visite visant à proposer au visiteur des activités sur dispositif mobile. Ce dernier est plus qu'un support informatique, il joue le rôle d'un prescripteur d'activités (il remplace à un certain niveau un conservateur de musée). Il propose en général une série d'activités ou de tâches à effectuer, impliquant une étude sur les œuvres. Ces systèmes sont généralement destinés à un groupe de visiteurs et plus particulièrement à un groupe scolaire. Ils mettent souvent en œuvre les trois étapes de visite : l'avant visite, la visite en elle-même et le retour sur l'expérience de visite. Les phases de préparation et de retour d'expérience sont typiquement mises en œuvre *via* des sites Web.

Museum detective guide [Thom-Santelli et al., 2005] reprend parfaitement l'idée de l'approche orientée tâches. Ce système reprend des exercices existant sur papier en les implémentant sur un dispositif mobile (tablette). Les visiteurs sont répartis

en petit groupes et le parcours est imposé et prédéfini. Devant chaque œuvre, une série de questions à choix multiples encourage chaque groupe à examiner l'œuvre en question (ex. What do you think this object is made of ?). Les bonnes réponses donnent lieu à un complément d'information et les mauvaises réponses des indices pour déterminer la bonne réponse. Le but étant de donner un attrait pédagogique à la visite.

Le système *MYST* [Laine et al., 2010] est lui-aussi centré sur la réalisation d'un ensemble de tâches par l'utilisateur. Ces tâches sont introduites au travers d'une scénarisation pédagogique qui a pour but de guider le visiteur dans les tâches à réaliser suivant des dialogues prédéfinis. *MYST* propose deux types d'interaction. Il offre au visiteur la possibilité d'enregistrer de manière vocale ses impressions sur les œuvres, afin de les réécouter ensuite. Il propose aussi des jeux sous forme de quizz (*battles*) où chaque visiteur se voit attribuer un score qui correspond au nombre de réponses correctes. On peut se demander cependant, dans quelle mesure ces types d'interaction augmentent vraiment la satisfaction des visiteurs.

Ces systèmes semblent donner de bons résultats concernant l'expérience de la visite en la rendant plus interactive. Cependant, ils présentent plusieurs problèmes. Le premier problème est que l'attention de l'utilisateur est en grande partie focalisée sur le dispositif mobile. Ce problème est général pour tous les systèmes mobiles d'aide à la visite, mais il est plus important dans les systèmes orientés tâches. L'utilisateur est en effet sollicité de manière régulière pour effectuer des tâches utilisant le dispositif au point qu'il en oublie le but principal qui est la visite en elle-même. Ainsi [Thom-Santelli et al., 2005], dans le cadre du système *Museum detective guide*, notait que certains utilisateurs passaient clairement plus de temps à interagir avec le dispositif qu'à regarder les œuvres du musée. Le deuxième problème réside dans le fait que les systèmes orientés tâches sont prévus pour un musée en particulier, et qu'ils sont difficilement généralisables à d'autres musées. Enfin, le dernier problème est qu'il n'y a aucune personnalisation, le parcours ainsi que toutes les tâches à accomplir sont imposés et sont les mêmes pour tous les visiteurs.

3.2.2 Systèmes orientés navigation

Les approches orientées navigation ont pour objectif d'offrir aux visiteurs des parcours plus libres, tout en apportant des informations adaptées à chaque situation. Elles se distinguent des approches orientées tâches par le public ciblé, étant conçues généralement pour un visiteur seul, adulte ou non. Les interactions proposées sont plus implicites que dans les systèmes orientés tâches, offrant au visiteur une liberté plus importante. Nous décrivons brièvement les systèmes orientés navigation les plus

aboutis dans la suite.

Le système *LISTEN*, proposé par [Zimmermann et al., 2003], illustre parfaitement l'approche basée navigation. Dans ce système, l'utilisateur est équipé d'un casque audio, comme dans les audio-guides classiques. Les informations sonores que reçoit l'utilisateur du système dépendent de la position du visiteur dans le musée ainsi que de ses mouvements et de l'orientation de sa tête. Le musée est partitionné en plusieurs zones. Chaque zone est associée à plusieurs œuvres. Lorsque le visiteur approche d'une œuvre donnée, des informations sonores sur cette derrière sont déclenchées. Afin de proposer des informations pertinentes et intéressantes pour l'utilisateur, un modèle dynamique de ses intérêts est alimenté au cours de ses déplacements. Les seules formes d'interaction entre le visiteur et le système étant des interactions implicites (déplacements et arrêts du visiteur devant une œuvre), le système utilise les informations sur le temps resté devant chaque œuvre et le temps passé à écouter les commentaires pour construire un modèle des intérêts du visiteur.

Le projet PEACH [Zancanaro et al., 2003] se base aussi sur la localisation du visiteur dans l'espace physique et propose une interaction multimodale via un PDA. Ce système propose de générer automatiquement des clips vidéos suivant l'œuvre vers laquelle l'utilisateur pointe son PDA. Ces vidéos permettent de souligner des points de détails de l'œuvre ainsi que de raconter une petite " histoire " à propos de l'œuvre, en se basant sur le texte qui la décrit. Le projet PIL (Personal experience with active cultural heritage-IsraeL) proposé par [Rocchi et al., 2004] est la suite du projet PEACH. Il propose une modélisation plus fine des intérêts de l'utilisateur, grâce à un questionnaire initial permettant de choisir des thèmes ou des artefacts a priori intéressants. Cependant, l'utilisation d'un long questionnaire avant de commencer la visite n'est en général pas apprécié par les visiteurs.

De nombreux autres systèmes pour l'assistance à la visite se basant sur une approche orientée sur la navigation ont été proposés. Pour une revue détaillée, on pourra se référer à [Kuflik et al., 2011].

3.2.3 Systèmes de recommandation pour la visite de musées

Les musées disposent en général de collections de très grande taille mises à disposition du grand public. Cependant, les visiteurs ne peuvent pas accéder à l'intégralité des œuvres présentées dans le musée. Il existe en effet un problème de surcharge d'information lié au nombre important d'œuvres présentes dans les musées et à leurs différents styles, thèmes et artistes. Le visiteur d'un musée est de plus confronté à un certain nombre de problèmes. Tout d'abord, il dispose en général d'un temps limité à passer dans le musée et ne sait pas forcément ce qu'il devrait voir

en priorité ou ce qu'il va aimer. Le parcours qu'il effectue n'est donc généralement pas très réfléchi [Kuflik et al., 2011]. En conséquence, il se peut que le visiteur perde du temps en regardant des œuvres qui ne l'intéressent pas beaucoup. Inversement, il se peut qu'il ait manqué des œuvres qui auraient pu l'intéresser. Un autre problème auquel le visiteur est confronté est l'ordre dans lequel il découvre les œuvres. En effet, si cet ordre est mal choisi, il peut être amené à faire un nombre important d'aller-retour, ce qui peut lui faire perdre du temps et l'empêcher de découvrir de nouvelles œuvres.

Pour répondre à la diversité des préférences des visiteurs, les conservateurs de musées proposent des visites sur différents thèmes. Cependant, ces thèmes sont généralement choisis en fonction des œuvres les plus connues de la collection et les visites comprennent une séquence fixe et prédéfinie d'œuvres d'art à voir, qui est la même pour tous les visiteurs. Or, imposer un parcours est une contrainte forte pour les visiteurs, qui aiment à faire usage de leur liberté de visiter [Gob and Drouguet, 2014]. Une visite à l'aide d'un audio guide offre une plus grande liberté dans la détermination d'une séquence personnelle d'œuvres lors de la visite. Mais là encore, l'ensemble des œuvres à choisir est prédéfini et il est toujours le même pour tous les visiteurs. Les travaux effectués par [Jeong and Lee, 2006] montrent que les visiteurs d'un musée souffrent de la fatigue physique et/ou psychologique en raison de l'effort de marche et de la surcharge d'information. Il est ainsi dit que 28.9% des visiteurs quittent leur visite à mi-parcours, et parmi ceux-là 21% à cause de la fatigue physique et 20% à cause de l'ennui engendré en regardant des œuvres qui ne les intéressent pas.

Pour faire face aux problèmes cités précédemment et aller au-delà des fonctionnalités traditionnelles qu'offre l'audio-guide, l'utilisation de systèmes de recommandation pour améliorer la visite de musées commence à devenir une solution très répandue. Nous citons par la suite les travaux les plus influents qui se focalisent sur cet aspect de personnalisation de la visite de musées.

Le projet Hippie [Oppermann and Specht, 2000] a été l'un des premiers systèmes de recommandation basé contenu dans le cadre de la visite de musée. Hippie utilise la taxonomie ICONCLASS, une classification exhaustive des différents thèmes de l'art occidental. Les visiteurs sont caractérisés par des scores d'intérêt pour les différents thèmes de la taxonomie. Quand un visiteur se déplace dans le musée, le système détecte sa position en utilisant une technique de localisation radio. Par conséquent, Hippie est en mesure d'informer le visiteur sur les œuvres qui peuvent l'intéresser et qui se situent autour de lui. Cependant, le modèle des œuvres dans le projet HIPPIE est basé uniquement sur ICONCLASS, ce qui en limite l'intérêt. On peut noter également que l'avis d'autres visiteurs n'est pas pris en compte et que le

système ne propose pas de parcours au visiteur.

Le projet Mymuseum [Bright et al., 2005] utilise un prototype web qui offre des visites virtuelles à l'utilisateur. Dans ce système, un utilisateur doit tout d'abord entrer ses paramètres (données personnelles) ainsi que ses préférences générales, par exemple les thèmes qu'il apprécie. Ces informations sont ensuite utilisées pour lui recommander une visite virtuelle comportant des œuvres qui correspondent à ses préférences initiales. Le système adapte aussi l'information présentée à l'utilisateur, par exemple l'utilisateur peut choisir s'il veut des descriptions détaillées ou simples pour les œuvres.

Le projet CHIP [Wang et al., 2009] est l'un des projets les plus aboutis en ce qui concerne les approches basées contenu pour l'aide à la visite de musées. Le système recommande des œuvres et des thèmes qui correspondent au profil de l'utilisateur. Le profil est basé sur les notes d'intérêt données par l'utilisateur aux différentes œuvres ou à leurs caractéristiques (artiste, style,..) comme le montre la figure 3.1. À partir des notes positives données, le système peut alors proposer à l'utilisateur des œuvres ou des informations en lien avec celles qu'il a aimées ou qui l'ont intéressé. L'implémentation de ce système a donné lieu à une application mobile en musée, SPACE-CHIP [Van Hage et al., 2010], qui permet de construire dynamiquement des parcours en fonction des goûts de l'utilisateur. Le modèle de l'œuvre est plus riche dans le projet CHIP comparé au modèle proposé dans HIPPIE. Cependant, il ne permet la comparaison qu'entre les œuvres d'art (et non pas entre artistes par exemple). De plus, le nombre de recommandations est trop élevé (toutes les recommandations ne sont pas pertinentes pour l'utilisateur). De ce fait, le problème de surcharge d'information n'est pas complètement résolu. Le contexte ainsi que l'avis des autres utilisateurs n'est pas pris en compte.

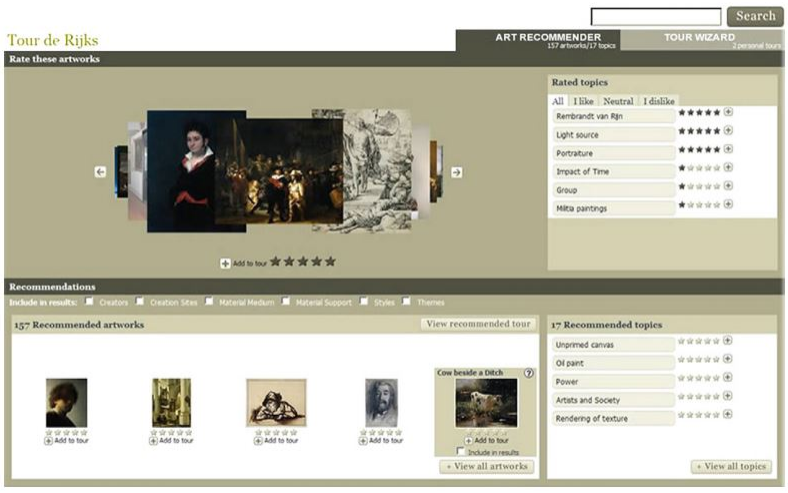
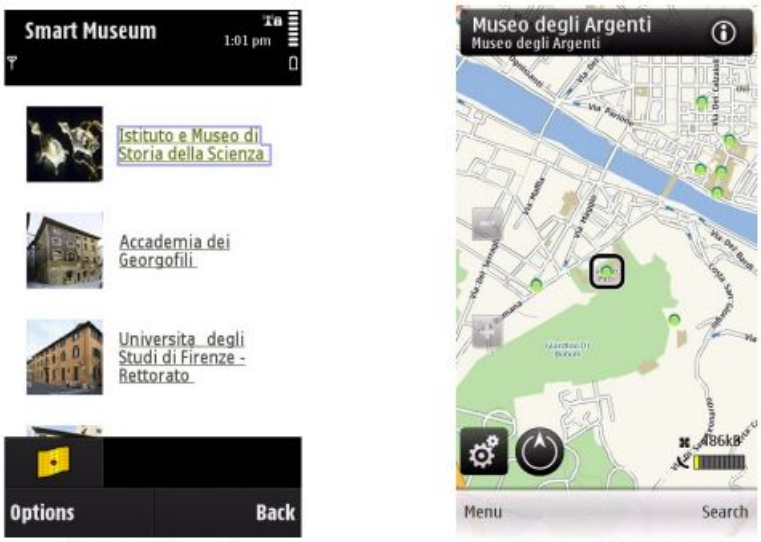


Figure 3.1 – Système de recommandation du projet CHIP

SMARTMUSEUM [Ruotsalo et al., 2013] utilise quant à lui un système de recommandation plus complexe. Le système détecte automatiquement si l'utilisateur est à l'extérieur ou à l'intérieur, en se basant sur sa localisation (utilisant la technologie GPS ou RFID). Pour le premier cas, le système utilise la position de l'utilisateur, le temps qu'il souhaite accorder à la visite ainsi que son profil. Ces informations sont entrées par l'utilisateur sur l'interface graphique du système. Le système est alors capable de recommander les items qui sont le plus proches de l'utilisateur et qui correspondent et à son profil. La visualisation des recommandations peut se faire de deux manières, sous forme de liste ou bien à l'aide d'une map (figure 3.2).



(a) Recommendations as a list

(b) Recommendations on a map

Figure 3.2 – Recommandations pour le scénario outdoor

Concernant le scénario indoor, c'est-à-dire, dans le musée, le système recommande la liste des œuvres les plus pertinentes pour l'utilisateur (figure 3.3) en se basant sur un modèle défini *a priori* par l'utilisateur. Ce modèle contient ses préférences et son contexte (temps de visite, but de la visite, etc.).

3.3 Systèmes de recommandation pour le tourisme

La manière dont les touristes préparent et organisent leur voyage a considérablement changé ces dernières années. Il y a peu de temps, ils utilisaient des guides de voyages et des cartes géographiques dans le but de trouver des points d'intérêt à visiter. De nos jours, ils peuvent trouver n'importe quelle information sur n'importe quel lieu d'activité à l'aide d'un simple clic sur le web. Les informations sur les destinations de voyage et leurs ressources associées (hôtels, attractions, musées, événements,etc.) sont souvent recherchées par les touristes afin de planifier leur voyage. Cependant,

(a) Context input screen (b) Recommendations for the indoor scenario

Figure 3.3 – Recommandations pour le scénario indoor

la liste des possibilités offertes par les moteurs de recherche des sites touristiques spécialisés peut être trop importante pour les utilisateurs. L'évaluation de cette liste d'options est très complexe et peut demander beaucoup de temps aux touristes. Les systèmes de recommandation peuvent donc aider les visiteurs à trouver ce qui les intéresse et leur proposer des parcours de visite personnalisés.

Dans cette partie de chapitre, nous faisons un résumé des travaux qui ont été réalisés sur les systèmes de recommandation pour le tourisme

3.3.1 Fonctionnalités offertes

Dans cette section, nous décrivons les fonctionnalités générales des systèmes de recommandation touristiques présents dans la littérature. [Borras et al., 2014] distinguent quatre fonctionnalités : suggestion d'une destination et construction d'un parcours de visite, recommandation d'attractions, planification d'un itinéraire et aspects sociaux. La figure 3.4 donne une estimation visuelle du pourcentage de systèmes implémentant ces fonctionnalités.

3.3.1.1 Suggestion de destination et parcours de visite

Une petite partie des systèmes de recommandation pour le tourisme se concentre sur la suggestion de la destination à choisir pour le visiteur, en fonction de ses préférences. Ce genre de système est cependant assez rare, étant donné que la plupart du temps c'est le visiteur qui décide lui même de sa destination et qu'il n'utilise donc pas un système pour cette tâche.

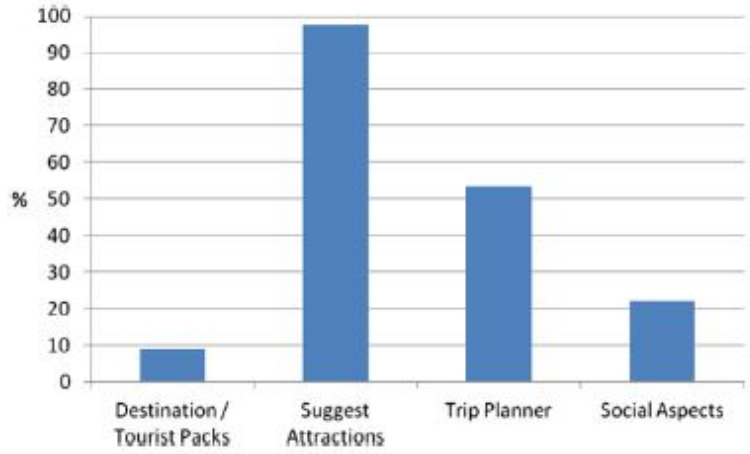


Figure 3.4 – Fonctionnalités offertes dans les approches étudiées ([Borras et al., 2014])

Dans cette catégorie de systèmes, on peut citer *PersonalTour* [Lorenzi et al., 2011], *Itchy Feet* [Seidel et al., 2009] et *MyTravelPal* [Koceski and Petrevska, 2012]. *PersonalTour* est un système utilisé par les agences de voyages afin d'aider leurs clients à trouver le meilleur pack voyage en prenant en compte leurs préférences. Une fois le processus de recommandation terminé, une liste triée d'options est fournie au client. La figure 3.5 montre un exemple du service de recommandation d'hôtels. Une fois les recommandations faites, le client peut évaluer chaque item de chaque service proposé.

Id	Hotel name	City	Hotel category	Room category	Room type	Swimming Pool	WiFi
1	Libertel	Paris	Economic	Standard	Double	No	Yes
2	Palladium	Punta Cana	Resort	Luxe	Double	Yes	Yes
3	Amadeus	Milan	Economic	Standard	Single	No	Yes
4	Riu Palace	Cancun	First	Luxe	Double	Yes	Yes
5	WestIn	Aruba	Economic	Luxe	Double	Yes	Yes

Figure 3.5 – Exemple de la recommandation d'hôtels dans le système PersonalTour

Itchy Feet ne fait pas que recommander des destinations mais fournit aussi un service d'achat pour la réservation du voyage. Les utilisateurs peuvent formuler des requêtes, qui sont traitées par des agents autonomes qui se chargent de la recherche d'information dans la base de données interne ainsi que des sources de données externes. Les résultats sont fournis à l'utilisateur au travers d'une interface dans laquelle les items recommandés (vols et hôtels) peuvent être sélectionnés et achetés.

MyTravelPal recommande au visiteur, en premier lieu, les zones d'intérêt au moyen de cercles sur une région géographique (voir figure 3.6). La taille du cercle indique le niveau d'appréciation de l'utilisateur. Une fois que l'utilisateur se focalise sur une région d'intérêt en particulier, les attractions qui constituent cette région forment à leurs tours des cercles en fonction du degré de ressemblance avec le profil de l'utilisateur.

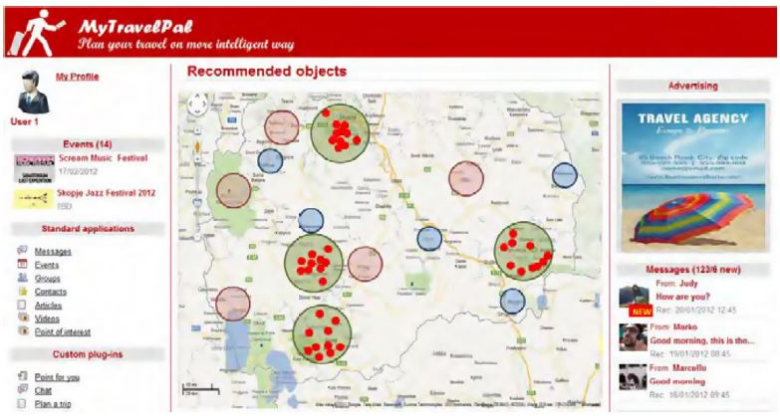


Figure 3.6 – MyTravelPal-Recommandation de régions d'intérêt

3.3.1.2 Recommandation de points d'intérêt

La plupart des systèmes de recommandation touristiques se focalisent sur la recommandation de points d'intérêt (POIs). Ces systèmes sont conçus généralement pour un visiteur qui a déjà choisi sa destination ou bien qui est sur place.

Ces systèmes sont plus complexes, car ils gèrent un nombre important de points d'intérêt. Un système de recommandation peut donc être d'une aide très précieuse pour l'utilisateur. Les utilisateurs peuvent ainsi trouver rapidement des lieux d'activités qui vont probablement les intéresser, et ceci de manière efficace. Ils peuvent même découvrir de nouveaux lieux qu'ils ne connaissaient pas ou qu'ils n'auraient pas penser visiter sans l'utilisation de ces systèmes. Les points d'intérêt qui sont candidats à la recommandation sont généralement stockés dans une base de données statique. Cependant quelques systèmes (par exemple *Otium* [Montejo-Ráez et al., 2011]) préfèrent extraire automatiquement l'information sur les points d'intérêt depuis le Web afin de garantir qu'ils ont toujours l'information la plus à jour.

Ce genre de systèmes de recommandation de points d'intérêt dédiés au tourisme [Borràs et al., 2011, Fenza et al., 2011, Sebastia et al., 2009] fournit généralement une liste des points d'intérêt qui correspondent le plus au profil de l'utilisateur, qui ont été visités ou positivement évalués par des utilisateurs similaires dans le passé, ou bien qui sont similaires à ceux que l'utilisateur a déjà visité. Ainsi, ces systèmes utilisent des mécanismes pour comparer les préférences de l'utilisateur courant avec les caractéristiques d'un point d'intérêt ou calculer la similarité entre deux utilisateurs ou deux points d'intérêt. La liste de points d'intérêt recommandée peut aussi changer si le processus de recommandation prend en compte les informations contextuelles, comme par exemple la localisation de l'utilisateur, son budget, le prix des activités, etc. Quelques projets se focalisent aussi sur le fait de pouvoir justifier

les recommandations qui sont faites [Jannach et al., 2009].

3.3.1.3 Planification de la visite

Certains projets fournissent non seulement une liste de points d'intérêt en fonction des préférences de l'utilisateur, mais offrent aussi la possibilité d'aider les touristes à créer leur parcours de visite.

Le système *CT-Planner* [Kurata, 2011, Kurata and Hara, 2013] illustre parfaitement l'approche de la planification du parcours. Il offre à l'utilisateur des recommandations de parcours planifiés comme le montre la figure 3.7. Ces parcours sont progressivement mis à jour au fur et à mesure que l'utilisateur exprime ses préférences et ses contraintes (durée de la visite, vitesse de marche, réticence à marcher, etc.). Le système affiche un radar graphique qui représente les préférences de l'utilisateur sur différents thèmes possibles. Il est alors en mesure de produire une liste de recommandations, présentée à la fois sur une carte de la ville avec un itinéraire à gauche de l'interface, et à l'aide d'une description des lieux avec une image pour chaque lieu à droite de l'interface.

Figure 3.7 – L'interface utilisateur du système CT-Planner [Kurata, 2011]

Il existe de nombreux systèmes qui fournissent une liste de points d'intérêt initiale (ou un plan de visite initial), avec lequel l'utilisateur peut directement interagir pour ajouter des lieux d'activités de son choix, en supprimer, sélectionner un lieu pour avoir plus d'informations, changer l'ordre de la visite, etc. La composante "planification" du système de recommandation prend en compte des facteurs importants tels que la durée prévue de la visite, le temps de déplacement entre les points d'intérêts, les horaires d'ouverture et fermeture des attractions, etc. Parmi les systèmes les plus pertinents, on peut citer *EnoSigTur* [Borràs et al., 2011], *City*

Trip Planner [Vansteenwegen et al., 2011a], Smart City [Luberg et al., 2011], Otium [Montejo-Ráez et al., 2011] et e-tourism [Sebastia et al., 2009].

Quelques projets comme *SAMAP* [Castillo et al., 2008] et *PaTac* [Ceccaroni et al., 2009] sont capables d'analyser les connexions possibles entre points d'intérêt en proposant différents moyens de transport (marche, vélo, voiture ou transports en commun) pour se déplacer d'un lieu à un autre.

En général pour la majorité des systèmes, une fois que le plan de la visite a été complètement construit, l'utilisateur peut souhaiter récupérer le programme complet de la visite ainsi que les itinéraires. Ceci peut prendre plusieurs formes. Des systèmes comme *SAMAP* [Castillo et al., 2008]ou *EnoSigTur* [Borràs et al., 2011] permettent à l'utilisateur de télécharger un fichier PDF contenant une carte avec un explication détaillée du plan. Dans d'autres systèmes, tels que *City Trip Planner* [Vansteenwegen et al., 2011a] et *Otiumtium* [Montejo-Ráez et al., 2011], l'utilisateur peut télécharger directement son parcours sur son smartphone.

3.3.1.4 Aspects sociaux de la visite

Plusieurs projets tels que [Ceccaroni et al., 2009], [Garcia et al., 2013], [Umanets et al., 2014] et [Vansteenwegen et al., 2011a] ont accordé une attention plus particulière à l'inclusion de fonctionnalités sociales pour la visite. Ces fonctionnalités sont intéressantes par le fait qu'elles autorisent les visiteurs à partager leurs visites (images, commentaires, préférences, etc.) avec d'autres touristes. Les aspects sociaux sont ainsi très intéressants pour donner plus de possibilités aux visiteurs en leur permettant d'échanger avec les autres et d'apprendre des autres visiteurs.

Des systèmes comme *moreTourism* [Rey-López et al., 2011] et *Itchy Feet* [Seidel et al., 2009] permettent aux utilisateurs de ne pas seulement interagir avec des réseaux sociaux populaires, mais aussi de créer des groupes d'activités qui peuvent être employés pour poster des commentaires, rejoindre des groupes pour faire des activités communes ou interagir avec d'autres utilisateurs. Le système *e-Tourism* [Garcia et al., 2011] permet la recommandation de visites qui satisferaient les préférences d'un groupe de visiteurs au lieu d'un visiteur unique.

Dans le système *iTravel* [Yang and Hwang, 2013], les utilisateurs communiquent entre eux à l'aide d'un système de communication mobile peer-to-peer afin de partager leurs notes (leur avis sur les différents points d'intérêt). Leur carte de navigation montre non seulement l'emplacement des points d'intérêts recommandés, mais aussi la position des visiteurs proches avec qui il est possible de communiquer. La figure 3.8 illustre une carte avec des attractions recommandées (marqueurs verts) et des utilisateurs voisins (marqueurs bleus).

Figure 3.8 – carte de navigation de *iTravel* [Yang and Hwang, 2013]

Le système *VISIT* [Meehan et al., 2013] utilise des techniques d'analyse de sentiments sur les commentaires relatifs à des points d'intérêt dans Twitter et Facebook et détermine si ces commentaires sont positifs ou négatifs. Cette information est indiquée par des couleurs vertes et rouges dans l'interface du système, pour que l'utilisateur puisse facilement identifier les points d'intérêt que les autres visiteurs aiment ou n'aiment pas.

3.3.2 Techniques de recommandation en e-Tourisme

La figure 3.9 montre la distribution des techniques de recommandation pour le tourisme utilisées dans les travaux cités dans la revue détaillée [Borras et al., 2014]. Un peu plus de la moitié des travaux utilisent une approche hybride (53%). Les systèmes hybrides peuvent combiner les approches classiques de différentes manières. Les plus utilisées pour le tourisme sont :

- Sélection de la méthode : le système incorpore les méthodes de recommandation classiques DM (démographic), CB (content-based) et CF (collaborative filtering), mais une seule méthode est appliquée, en fonction de la situation de l'utilisateur. Par exemple, quand l'utilisateur arrive dans le système, la méthode basée sur les données démographiques est utilisée. Ensuite, si des utilisateurs similaires peuvent être trouvés, une approche de filtrage collaboratif est activée, sinon l'approche basée contenu est utilisée. c'est ce

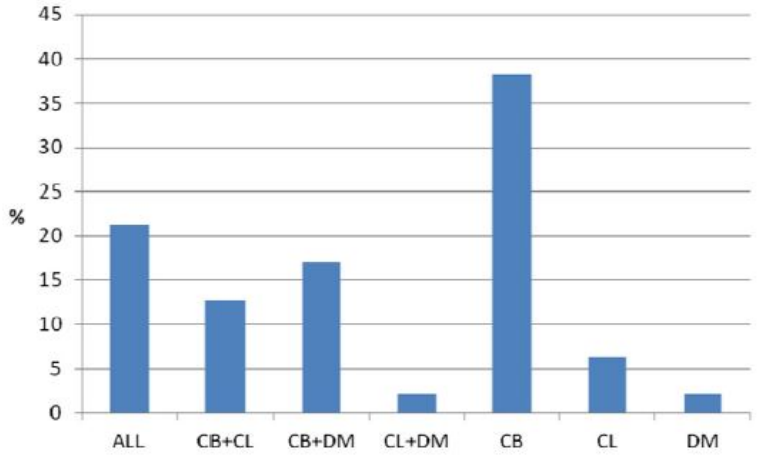


Figure 3.9 – Techniques de recommandation utilisées pour le tourisme [Borras et al., 2014]

qui est appliqué dans les travaux suivants : [Huang and Bian, 2009], [Martinez et al., 2009] et [Noguera et al., 2012].

- Utilisation séquentielle : chaque technique de recommandation classique est utilisée dans différentes étapes du processus. Par exemple, le système *SPETA* [García-Crespo et al., 2009] utilise quatre étapes : tout d'abord, l'information contextuelle telle que la localisation ou le temps est utilisée pour faire une première sélection des points d'intérêt. Ensuite, une liste de recommandations plus précise est obtenue en utilisant une technique de filtrage basée sur la connaissance, ceci en calculant la similarité sémantique entre les préférences de l'utilisateur et les points d'intérêt candidats. Puis, les préférences de l'utilisateur et l'utilisation d'un filtrage collaboratif permettent d'obtenir des résultats plus raffinés. Enfin, un vecteur de préférences est utilisé pour faire la recommandation finale. Dans [Braunhofer et al., 2013] une hybridation du filtrage collaboratif et du filtrage démographique est utilisée dans une phase d'apprentissage pour construire un modèle de prédiction de différents contextes. Une fois que le modèle a été appris, une approche basée contenu génère la liste des recommandations en calculant un score pour chaque point d'intérêt en utilisant les valeurs prédites précédemment.

- Utilisation intégrée : les deux techniques de filtrage collaboratif et filtrage sur le contenu sont combinées durant le processus de recommandation. Par exemple, dans *SugTur* [Borràs et al., 2011] des scores différents sont calculés pour estimer l'appréciation *a priori* d'un point d'intérêt pour un utilisateur cible. Ces scores sont obtenus en utilisant un filtrage collaboratif, un filtrage démographique ainsi que des similarités basées sur le contenu. Ensuite, ces

scores sont fusionnés pour obtenir un score de qualité global pour chaque point d'intérêt candidat à la recommandation, afin de suggérer à l'utilisateur cible les meilleurs points d'intérêt possibles. Dans [Lucas et al., 2009], les utilisateurs sont classifiés aux sein de groupes en utilisant simultanément différentes techniques de recommandation classiques (DM+CF+CB). Ensuite, un ensemble de règles basées sur la logique floue est automatiquement généré afin que les nouveaux utilisateurs puissent être automatiquement classifiés dans ces différents groupes (avec différents degrés d'appartenance à un groupe). La liste des items recommandés est finalement dérivée d'un score de prédiction basé sur les groupes auxquels l'utilisateur appartient.

Le tableau 3.1 regroupe les différents systèmes en fonction des techniques de recommandation qu'ils utilisent et/ou qu'ils combinent.

Méthode	Références
CB+CF+DM	[Lucas et al., 2009, Ruiz-Montiel and Aldana-Montes, 2009] [Borràs et al., 2011, Batet et al., 2012, Koceski and Petrevska, 2012][Braunhofer et al., 2013, Garcia et al., 2013, Lucas et al., 2013][Meehan et al., 2013]
CB+CF	[Castillo et al., 2008, García-Crespo et al., 2009, Fenza et al., 2011, Rey-López et al., 2011, Noguera et al., 2012, Rojas and Uribe, 2013]
CB+DM	[Collins and Quillian, 1969, Ceccaroni et al., 2009, Lamsfus et al., 2009, Niaraki and Kim, 2009, Mínguez et al., 2009, Martin et al., 2011, Sebastia et al., 2009, Garcia et al., 2011]
CF+DM	[Gavalas and Kenteris, 2011]
CB	[Huang and Bian, 2009, Lee et al., 2009, Seidel et al., 2009, Yu and Chang, 2009, Jannach et al., 2009, Ricci et al., 2009, Sebastia et al., 2010, Vansteenwegen and Souffriau, 2010, García-Crespo et al., 2011, Kurata, 2011, Linaza et al., 2011, Lorenzi et al., 2011, Luberg et al., 2011, Montejo-Ráez et al., 2011, Santiago et al., 2012]
CF	[Savir et al., 2013, Umanets et al., 2014, Yang and Hwang, 2013]
DM	[Wang et al., 2011]

Tableau 3.1 – Revue des techniques de recommandation pour le tourisme [Borras et al., 2014]

3.4 Conclusion

Dans ce chapitre, nous avons présentés en détail les approches d'aide à la visite culturelle, à la fois pour la visite de musées et pour le tourisme. Nous nous sommes focalisé principalement sur les systèmes de recommandation dans ce cadre-là.

Concernant la visite de musées, nous avons classifié les systèmes selon trois catégories différentes, les approches orientées tâches, les approches orientées navigation et les approches de personnalisation utilisant des systèmes de recommandation. Notre système de recommandation pour la visite de musées diffère de ceux que nous avons présentés dans ce chapitre sur deux aspects principaux. Tout d'abord, dans les travaux que nous avons présentés, seules les préférences de l'utilisateur sont prises en compte. Les opinions des autres utilisateurs sur les œuvres sont souvent négligées. L'historique des notes d'autres utilisateurs peut être bénéfique pour affiner les recommandations et introduire de la diversité en utilisant des techniques de filtrage collaboratif. Ensuite, les travaux cités se focalisent sur la recommandation d'œuvres seulement. Cependant, il pourrait être très utile de recommander à l'utilisateur un parcours incluant des œuvres intéressantes à découvrir dans un ordre efficace et en un temps limité spécifié par l'utilisateur.

Pour le tourisme, nous avons classifié les travaux existants suivant les fonctionnalités que ces systèmes offrent à l'utilisateur, à savoir, choix de la destination, planification du parcours de visite, recommandation de points d'intérêt et intégration d'aspects sociaux. Nous avons enfin présenté les méthodes de recommandation et leur utilisation dans le domaine du tourisme. La majorité des systèmes se focalisent sur la recommandation de points d'intérêt soit sous forme de liste triée soit à l'aide d'une map. Cependant, la recommandation sous cette forme n'est pas adaptée à un domaine comme le tourisme, ou l'on traite de plusieurs types d'items (monument, parc, musée, etc). Nous nous sommes ainsi focalisé sur les systèmes de recommandation composites qui organisent les recommandations sous forme de packages, chaque package étant constitué de plusieurs points d'intérêt.

Une approche hybride et contextuelle pour la visite de musée

Sommaire

4.1 Introduction

Dans ce chapitre, nous présentons notre proposition de système de recommandation pour système mobile, adaptable au profil de l'utilisateur et sensible à son contexte. Notre objectif est d'améliorer l'expérience du visiteur en lui proposant un parcours de visite sur mesure, comportant des œuvres qui correspondent à ses préférences et à son contexte de visite (temps de visite, localisation, etc.). Nous proposons une approche hybride et sensible au contexte qui combine trois approches différentes : démographique, sémantique et collaborative. Chaque approche étant adaptée à une étape spécifique de la visite. Premièrement, l'approche démographique est utilisée pour résoudre le problème du démarrage à froid. Ensuite, l'approche sémantique est activée pour recommander à l'utilisateur des œuvres sémantiquement proches de celles qu'il a appréciées. Enfin, l'approche collaborative est activée

pour lui recommander des œuvres que ses utilisateurs similaires ont aimées, afin d'affiner la liste des recommandations et introduire de la diversité. Un post-filtrage contextuel est utilisé pour la génération de parcours dépendant des informations du contexte : l'environnement physique, la localisation ainsi que le temps de visite. Le choix d'utiliser le contexte uniquement en post-filtrage est motivé par le nombre relativement faible d'items dans le domaine de la visite de musée, par rapport à d'autres domaines d'application comme les films, la musique ou bien le tourisme.

4.2 Modélisation sémantique du domaine

La taille des collections et la grande variabilité des objets exposés ont amené les musées à définir de nombreux systèmes d'organisation permettant l'indexation et la recherche d'artefacts muséaux. Ces systèmes d'organisation ont longtemps été constitués de fiches physiques. Avec l'avènement du numérique, les musées font de plus en plus appel aux technologies de l'information pour l'indexation et la recherche d'informations dans leurs collections [Bowen et al., 2008].

Pour concevoir notre système de recommandation, nous devons construire des mesures de similarité entre les concepts liés au musée (œuvres, artistes, styles, etc). L'utilisation de connaissances formalisées s'avère être pour cela la solution la plus appropriée. De nombreuses propositions ont été faites pour la modélisation du domaine des musées et du patrimoine culturel : Europeana Data Model (EDM) [Doerr et al., 2010], LIDO [Autere and Vakkari, 2011] et Art Museum Image Consortium (AMICO[1]).

Les collections numériques dont disposent la plupart des musées utilisent des bases de données pour stocker les informations sur les œuvres exposées ou non dans le musée. Ces informations comportent des propriétés pour chacune des œuvres telles que : le style, le thème, l'artiste, l'époque, etc. L'utilisation d'une simple base de données pour décrire les œuvres est insuffisante, aucune relation sémantique n'étant définie entre les œuvres et les concepts. Nous ne pouvons pas, par exemple, faire des comparaisons entre deux styles, deux thèmes, deux artistes, etc. Nous ne pouvons pas alors dire dans quelle mesure deux œuvres sont similaires, afin de pouvoir suggérer des recommandations pertinentes à l'utilisateur. Nous avons alors besoin d'une description sémantique riche et précise des œuvres, ce qui nous permettra de mettre en évidence les différents aspects et propriétés sur lesquels les œuvres sont comparées et de nous appuyer sur les mesures de similarité sémantiques décrites dans le chapitre 2 pour construire notre modèle de recommandation. Après avoir défini de manière précise les mesures de similarité entre les œuvres, les artistes, les

[1]http ://www.amico.org/home.html

styles, etc. de la base de connaissance, nous pourrons recommander au visiteur des œuvres qui correspondent à ses préférences.

La première étape de notre proposition est alors de définir et construire un modèle sémantique de connaissances pour le domaine des musées.

4.2.1 Sources de connaissances utilisées

Le modèle de connaissances que nous proposons repose essentiellement sur le modèle défini par [Gicquel et al., 2013], qui utilise l'ontologie CIDOC-CRM (Center for Intercultural Documentation-Conceptual Reference Model) [Doerr et al., 2007], qui est l'ontologie de référence pour la description sémantique du patrimoine culturel, la taxonomie ICONCLASS (ICONography CLASSification) qui permet de caractériser les thèmes des œuvres, ainsi que le thesaurus AAT (Art & Architecture Thesaurus) pour la description des modes de production des œuvres. Dans le but de proposer un modèle de connaissances plus riche, nous proposons d'ajouter les deux thesaurus ULAN (Union List of Artist Names) et TGN (Thesaurus of Geographic Names). En effet, avec ULAN, il est possible pour un artiste d'avoir des informations plus riches : sa nationalité, son rôle, les artistes qu'il a influencés, ses lieu de naissance et de décès. Concernant TGN, il s'agit du thesaurus le plus abouti regroupant les noms de différentes places géographiques. TGN nous permet de définir les lieux où les œuvres ont été réalisées, les lieux de décès des artistes, etc.

Les différentes sources de connaissances utilisées ainsi que leurs propriétés sont regroupées dans le tableau 4.1. Nous discuterons des propriétés de chacune des sources dans les sous-sections suivantes.

Nom	Modèle	Relations	Exemples
CIDOC-CRM	Ontologie (RDFS)	Hiérarchique Propriété Sous-Propriétés	Concept d'oeuvre cidoc :Man-made-thing
ICONCLASS	Taxonomie	Hiérarchique	Portrait
AAT	Thésaurus	Hiérarchique Association	Concept de style aat :Style
ULAN	Thésaurus	Hiérarchique Association	Concept d'artiste ulan :Person
TGN	Thésaurus	Hiérarchique	Concept de lieu tgn :Place

Tableau 4.1 – Sources de connaissances utilisées pour notre modèle de l'œuvre

4.2.1.1 CIDOC-CRM

L'ontologie CIDOC-CRM est l'ontologie de référence pour la description du patrimoine culturel. Elle définit les concepts d'événement, d'œuvre, d'époque, etc, en utilisant le formalisme RDFS. Une caractéristique essentielle de CIDOC-CRM est la possibilité de réutiliser et importer des connaissances issues d'autres sources avec d'autres formalismes de représentation (qu'elles soient basées sur des ontologies ou non). En effet, CIDOC-CRM est devenue un standard (ISO 21127) et des alignements (mappings) ont déjà été définis pour permettre l'import, dans une base de connaissance basée sur CIDOC-CRM, de connaissances décrites avec d'autres formalismes. La généricité, ainsi que la stabilité de CIDOC-CRM (Standard ISO depuis 2006) ont poussé [Gicquel et al., 2013] à choisir cette ontologie comme base de leur modèle sémantique, que nous avons nous-même repris. Cependant, le caractère très générique de l'ontologie impose de la compléter à l'aide de vocabulaires contrôlés (metadata vocabularies). En effet, l'utilisation seule de CIDOC-CRM ne permettrait pas de comparer les œuvres suivant leur style ou thème, car l'ontologie ne comprend pas de vocabulaire spécifique à l'art (thème d'une œuvre par exemple). Il faut alors étendre CIDOC-CRM avec d'autres sources de connaissances, que nous avons présenté dans le tableau 4.1 et qui sont décrites dans la suite.

4.2.1.2 ICONCLASS

La taxonomie ICONCLASS[2] permet de caractériser les thèmes des œuvres les uns par rapport aux autres. C'est une classification exhaustive des thèmes artistiques de l'art occidental. Les termes de cette taxonomie sont organisés sous une forme strictement hiérarchique, c'est à dire que l'unique relation existante est la relation Parent → Enfant. Cette taxonomie comporte 10 niveaux. La figure 4.1 représente les concepts les plus généraux (racines de la taxonomie à partir desquels sont dérivés les autres concepts).

A titre d'exemple la figure 4.2 représente la description du thème " Lois Juives sur le blasphème ". Ce thème est décrit suivant trois axes : un ensemble de termes pouvant le qualifier en plusieurs sujets (via la relation *dc :subject*), un ensemble de termes préférés (via la relation *skos :preflabel*) et un ensemble de relations de type *broader*, établissant des relations de type générique/spécifique avec d'autres entités.

4.2.1.3 AAT

Le vocabulaire Art and Architecture Thesaurus (AAT) est développé par le Getty Institute. Il s'agit un thésaurus à facettes qui fournit un vocabulaire contrôlé

[2]http ://www.iconclass.nl/home

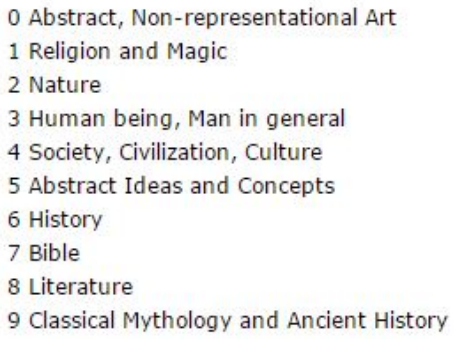


Figure 4.1 – Les dix niveaux de la taxonomie ICONCLASS

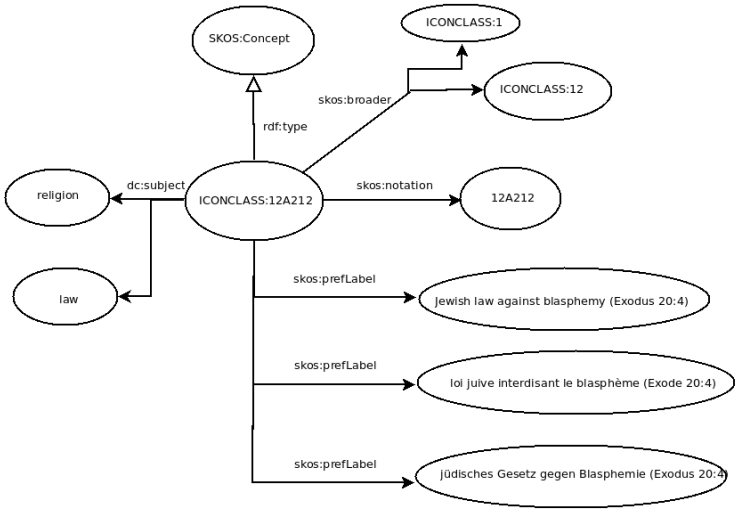


Figure 4.2 – Description d'un élément de la taxonomie ICONCLASS ([Gicquel, 2013])

principalement pour la description des styles des œuvres ainsi que leur technique de production (ex. peinture à l'huile). AAT se compose de sept facettes organisées sous forme de hiérarchie, une facette correspondant à une ou plusieurs hiérarchies. Des relations d'association et d'équivalence établissent des liens entre les hiérarchies. Les descriptions de Getty-AAT sont plus complexes que les descriptions de concepts d'ICONCLASS. La figure 4.3 illustre la description du terme *graffiti*.

Chaque élément appartient à deux hiérarchies, la hiérarchie principale, *hierarchical position*, et la hiérarchie secondaire, *additional parents*. les élements de Getty-AAT ont un *Record Type*, caractérisant le type des éléments de la hiérarchie. Dans l'exemple de la figure 4.3, le *Record Type* est de type concept, mais il existe trois autres types : *Guide Term, Hierarchie Name* et *Facet*. Dans notre système, l'utilisation de AAT a pour but de capturer les styles des œuvres, nous n'utilisons que la facette *Style and Period*. La figure 4.4 offre une illustration de cette facette

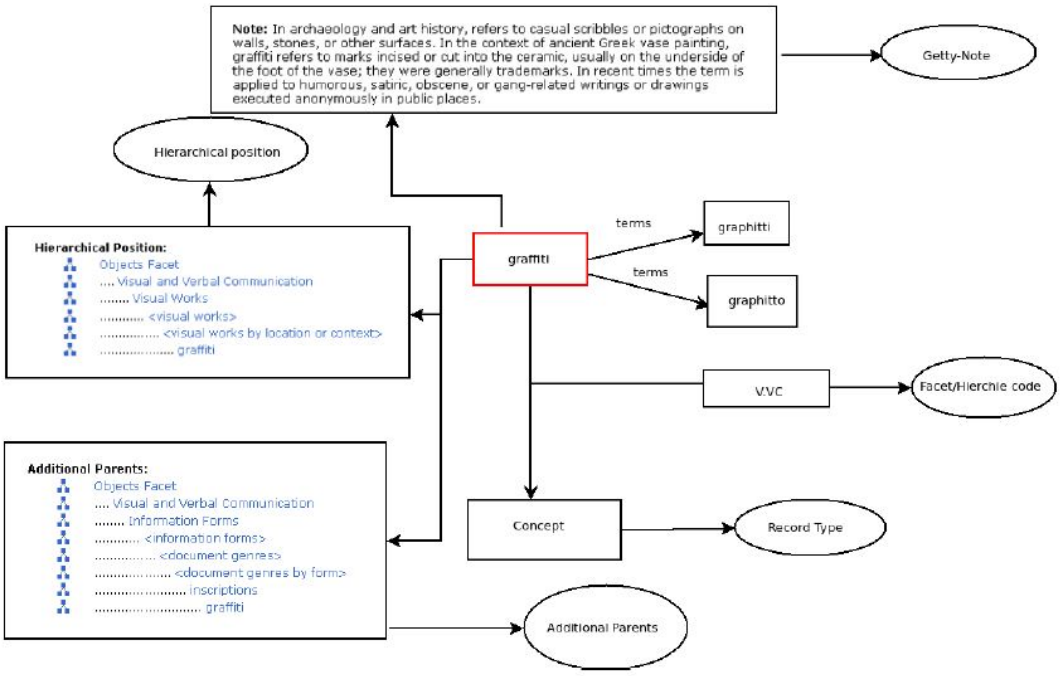


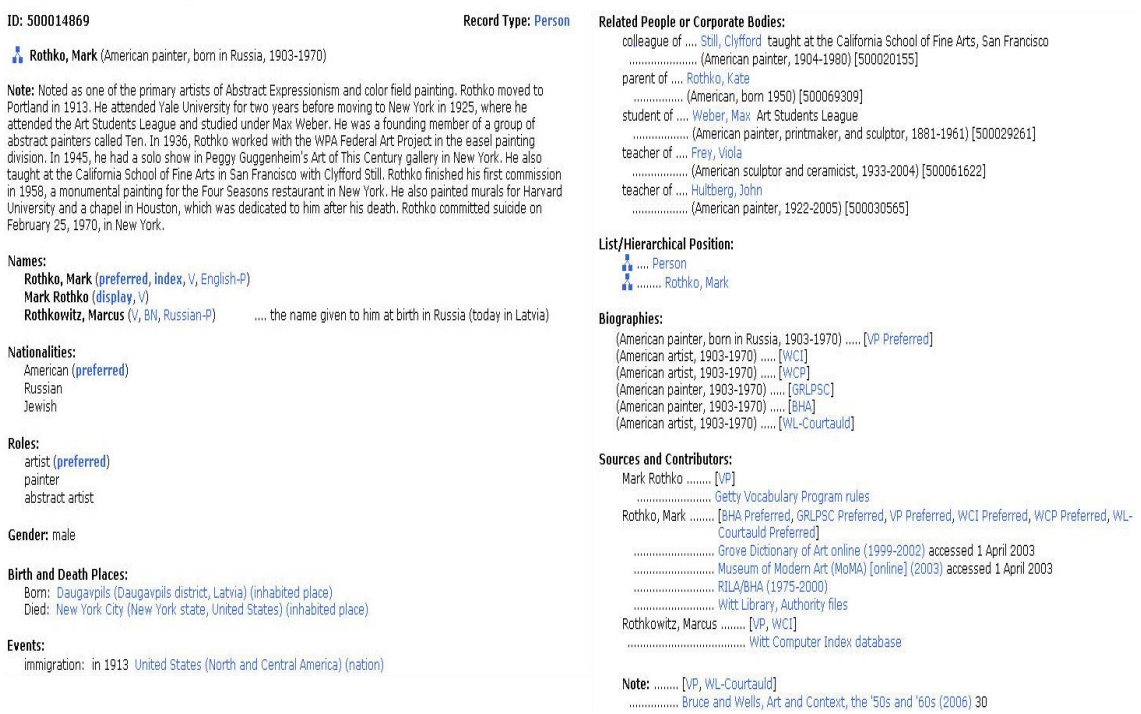
Figure 4.3 – Représentation d'un exemple de concept de AAT ([Gicquel, 2013])

Figure 4.4 – Position du concept de *Style Louis XIV* dans le thésaurus AAT

en représentant la position du concept *Style Louis XIV* dans la hiérarchie du Getty-AAT.

4.2.1.4 ULAN

Le vocabulaire Union List of Artist Names (ULAN) est aussi développé par le Getty Institute. Il s'agit d'un thésaurus à facettes qui fournit un vocabulaire structuré contenant les noms d'artistes (peintres, sculpteurs, etc.) ainsi que d'autres informations sur les artistes telles que : la nationalité, la date de naissance ou de mort, les artistes qu'il a influencés et ceux par qui il a été influencé, etc. Le thésaurus contient environ 293000 noms ainsi que d'autres informations à propos des artistes. Il contient des relations d'équivalence, hiérarchiques et associatives entre les concepts.


ID: 500014869 Record Type: Person

Rothko, Mark (American painter, born in Russia, 1903-1970)

Note: Noted as one of the primary artists of Abstract Expressionism and color field painting. Rothko moved to Portland in 1913. He attended Yale University for two years before moving to New York in 1925, where he attended the Art Students League and studied under Max Weber. He was a founding member of a group of abstract painters called Ten. In 1936, Rothko worked with the WPA Federal Art Project in the easel painting division. In 1945, he had a solo show in Peggy Guggenheim's Art of This Century gallery in New York. He also taught at the California School of Fine Arts in San Francisco with Clyfford Still. Rothko finished his first commission in 1958, a monumental painting for the Four Seasons restaurant in New York. He also painted murals for Harvard University and a chapel in Houston, which was dedicated to him after his death. Rothko committed suicide on February 25, 1970, in New York.

Names:
Rothko, Mark (preferred, index, V, English-P)
Mark Rothko (display, V)
Rothkowitz, Marcus (V, BN, Russian-P) .... the name given to him at birth in Russia (today in Latvia)

Nationalities:
American (preferred)
Russian
Jewish

Roles:
artist (preferred)
painter
abstract artist

Gender: male

Birth and Death Places:
Born: Daugavpils (Daugavpils district, Latvia) (inhabited place)
Died: New York City (New York state, United States) (inhabited place)

Events:
immigration: in 1913 United States (North and Central America) (nation)

Related People or Corporate Bodies:
colleague of .... Still, Clyfford taught at the California School of Fine Arts, San Francisco
..... (American painter, 1904-1980) [500020155]
parent of .... Rothko, Kate
..... (American, born 1950) [500069309]
student of .... Weber, Max Art Students League
..... (American painter, printmaker, and sculptor, 1881-1961) [500029261]
teacher of .... Frey, Viola
..... (American sculptor and ceramicist, 1933-2004) [500061622]
teacher of .... Hultberg, John
..... (American painter, 1922-2005) [500030565]

List/Hierarchical Position:
.... Person
..... Rothko, Mark

Biographies:
(American painter, born in Russia, 1903-1970) ..... [VP Preferred]
(American artist, 1903-1970) ..... [WCI]
(American artist, 1903-1970) ..... [WCP]
(American painter, 1903-1970) ..... [GRLPSC]
(American painter, 1903-1970) ..... [BHA]
(American artist, 1903-1970) ..... [WL-Courtauld]

Sources and Contributors:
Mark Rothko ..... [VP]
..... Getty Vocabulary Program rules
Rothko, Mark ..... [BHA Preferred, GRLPSC Preferred, VP Preferred, WCI Preferred, WCP Preferred, WL-Courtauld Preferred]
..... Grove Dictionary of Art online (1999-2002) accessed 1 April 2003
..... Museum of Modern Art (MoMA) [online] (2003) accessed 1 April 2003
..... RILA/BHA (1975-2000)
..... Witt Library, Authority files
Rothkowitz, Marcus ..... [VP, WCI]
..... Witt Computer Index database

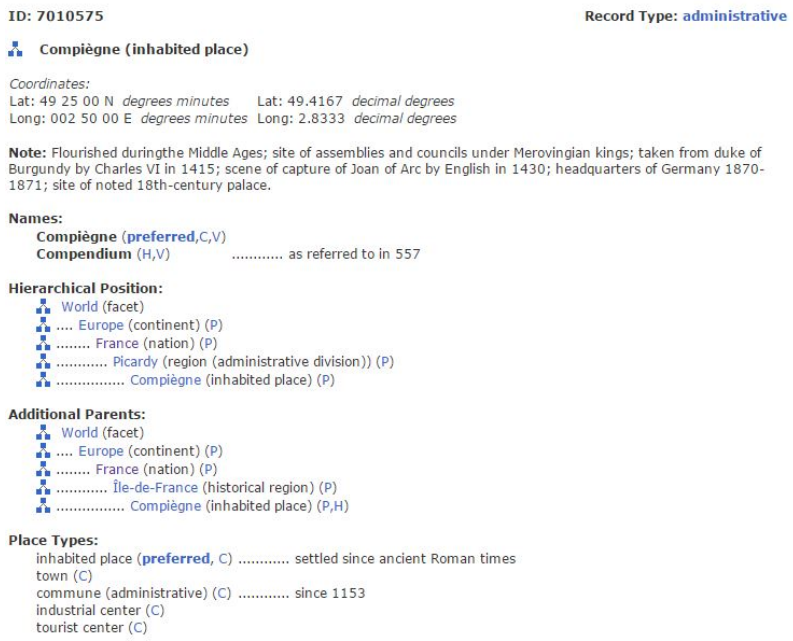
Note: ..... [VP, WL-Courtauld]
..... Bruce and Wells, Art and Context, the '50s and '60s (2006) 30

Figure 4.5 – Exemple de la représentation d'un artiste dans ULAN

Chaque élément de ULAN se focalise sur un artiste, et chaque artiste est identifié par un ID unique. La figure 4.5 illustre la description de l'artiste *Rothko Mark*, qui comporte une note correspondant à la biographie de l'artiste, les différents noms de l'artiste, ses nationalités, son rôle, son sexe, les lieux de sa naissance et mort, les artistes qui sont en liens avec lui (professeur, étudiant, collègue, etc.). Toutes ses informations seront utiles lorsque nous serons amené à calculer la similarité entre deux artistes.

4.2.1.5 TGN

Le dernier thésaurus que nous utilisons est TGN (Thesaurus of Geographic Names). Développé aussi par le Getty Institute, ce thésaurus contient des informations associées à des places (i.e. des lieux, par exemple Compiègne). Ces informations peuvent être le nom, l'histoire, la population, la culture, etc. TGN est un vocabulaire structuré qui contient actuellement des informations sur plus d'un million de lieux. Chaque lieu est identifié par un ID unique. Cet ID est relié à un des noms du lieu, la position de ce lieu dans la hiérarchie, ses coordonnées géographiques et son type ("admistrative", "inhabited place", "capital", etc.). Dans notre système, l'utilisation de TGN a pour but de capturer les lieux en liens avec des œuvres ou des artistes, par exemple le lieu de création d'une œuvre ou bien le lieu de naissance ou de mort d'un artiste. Nous utiliserons alors la position des lieux dans la hiérarchie TGN. La


ID: 7010575 Record Type: administrative

Compiègne (inhabited place)

Coordinates:
Lat: 49 25 00 N *degrees minutes* Lat: 49.4167 *decimal degrees*
Long: 002 50 00 E *degrees minutes* Long: 2.8333 *decimal degrees*

Note: Flourished duringthe Middle Ages; site of assemblies and councils under Merovingian kings; taken from duke of Burgundy by Charles VI in 1415; scene of capture of Joan of Arc by English in 1430; headquarters of Germany 1870-1871; site of noted 18th-century palace.

Names:
Compiègne (preferred,C,V)
Compendium (H,V) ..... as referred to in 557

Hierarchical Position:
World (facet)
.... Europe (continent) (P)
..... France (nation) (P)
..... Picardy (region (administrative division)) (P)
..... Compiègne (inhabited place) (P)

Additional Parents:
World (facet)
.... Europe (continent) (P)
..... France (nation) (P)
..... Île-de-France (historical region) (P)
..... Compiègne (inhabited place) (P,H)

Place Types:
inhabited place (**preferred**, C) ..... settled since ancient Roman times
town (C)
commune (administrative) (C) ..... since 1153
industrial center (C)
tourist center (C)

Figure 4.6 – Exemple de la représentation d'un lieu dans TGN

figure 4.6 illustre la description de la ville de Compiègne dans TGN

4.2.2 Modèle sémantique de l'œuvre

Dans les travaux de [Gicquel et al., 2013], sur lequel notre modèle sémantique se base, quelques modifications de l'ontologie CIDOC-CRM sont apportées. En effet, le modèle original de CIDOC-CRM est riche et donc relativement difficile à mettre en œuvre. Le peuplement de la base de connaissance par des sources externes peut s'avérer complexe si l'on suit le modèle original. Il est donc nécessaire de créer un certain nombre d'instances "artificielles" qui vont assurer le lien entre les différentes propriétés de l'œuvre, comme par exemple un lien direct entre l'œuvre et sa date de création, qui n'existe pas dans le modèle original. Par ailleurs, comme nous l'avons précisé plus tôt, l'objectif principal de la modélisation sémantique du domaine est de permettre des calculs de similarités sémantiques qui sont essentielles pour notre système de recommandation. L'objectif des simplifications du modèle est d'avoir des similarités sémantiques qui se basent sur des propriétés plus "expressives" pour la comparaison entre les œuvres. Par exemple, comparer deux œuvres en comparant leur artiste, leur style ou leur date de production.

Afin de faciliter le peuplement de la base de connaissances et de rendre les similarités sémantiques plus "naturelles", un certain nombre de compositions de propriétés sont réalisées afin de simplifier le modèle. C'est à dire, que nous remplaçons certains triplets (aP_ib) et (bP_jc) par un nouveau triplet (aP_kc). La figure

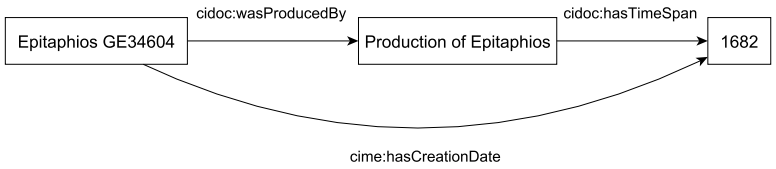


Figure 4.7 – Exemple de composition de propriétés

4.7 illustre ce processus.

L'intégration des vocabulaires contrôlés AAT, ULAN et TGN ainsi que de la taxonomie ICONCLASS dans le modèle final s'appuie également sur les travaux de [Gicquel et al., 2013]. Cette intégration dans CIDOC-CRM pose deux problèmes : tout d'abord l'expression des taxonomies sous forme de hiérarchies intégrables dans CIDOC-CRM, ensuite l'établissement des nouveaux liens entre ces taxonomies et CIDOC-CRM.

Concernant la taxonomie ICONCLASS, il existe déjà une version de cette taxonomie exprimée en SKOS (Simple Knowledge Organization System). Nous avons donc directement utilisé cette taxonomie. SKOS est une recommandation du W3C qui a pour but de représenter des thésaurus, classifications ou d'autres types de vocabulaires contrôlés ou de langages documentaires et de les intégrer dans des modèles de connaissances sous formes d'ontologies. Concernant les vocabulaires Getty, un mapping vers SKOS est réalisé en transformant leurs éléments en instances *skos :concept*. Nous avons reproduit la hiérarchie définie par les relations hiérarchiques via les relations *skos :broader* et *skos :narrower*.

La deuxième étape est maintenant d'intégrer ces taxonomies exprimées en SKOS dans CIDOC-CRM, l'objectif de cette intégration est de pouvoir réaliser des assertions du type : *cidoc : LaJoconde cime :hasTheme iconclass :Portrait*, ce qui donne en langage naturel, La Joconde a pour thème portrait. Un mécanisme prévu par les concepteurs de CIDOC-CRM nous permet de réaliser cette tâche de manière assez simple. En effet, la classe *cidoc :E55.Type* est spécialement conçue pour servir d'interface entre des vocabulaires métier et les autres classes de CIDOC-CRM. Nous avons donc déclaré les concepts issus du mapping SKOS des vocabulaires comme étant des instances de E55.Type et défini des sous-propriétés telles que : cime :hasTheme, cime : hasStyle, etc. qui relient la classe de l'œuvre et ses propriétés.

Après intégration des vocabulaires, nous aboutissons à la représentation des œuvres (instances de *cidoc :Man-made-Thing*), dont le modèle simplifié est représenté dans la figure 4.8. Une œuvre est produite par un artiste (ULAN) ayant éventuellement un maître ou un élève, elle possède un style (AAT), un matériau de production (AAT) et un thème (ICONCLASS), elle est produite dans un lieu

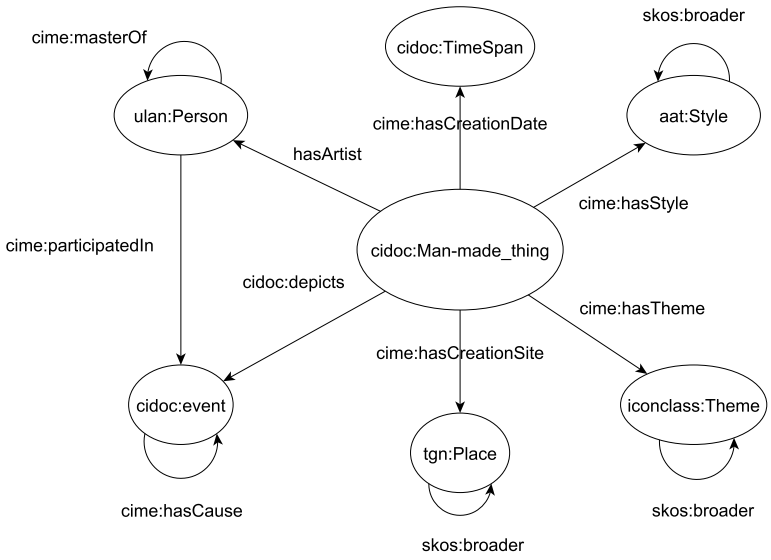


Figure 4.8 – Modèle sémantique des œuvres

(TGN) donné et à une période (cidoc :TimeSpan).

4.3 Modélisation du contexte

Nous proposons d'utiliser un modèle du contexte qui s'inspire du modèle du contexte défini par [Zimmermann et al., 2007] tout en l'adaptant à notre cas de la visite de musée. Ainsi, nous définissons six classes, une classe principale " *User*" et cinq autres classes *"Individuality", "Activity", "Time", "Location" et "Relation"* pour décrire l'utilisateur et les informations de contexte de sa visite. Notre modèle du contexte joue deux rôles importants dans notre framework :

- Il est utilisé dans le processus de recommandation. La classe *"Individuality"* contient les informations sur les données démographiques de l'utilisateur à savoir : l'âge, le sexe, le pays, la profession ainsi que le niveau d'expertise en art. Toutes ces informations sont entrées par l'utilisateur au moment où il commence à utiliser le système. Ces informations sont précieuses pour la phase de recommandation démographique (section 4.5). La classe *"Activity"* capture les différentes activités de l'utilisateur, entre autres donner une note à une œuvre, contempler une œuvre, etc. C'est dans cette classe que nous stockons aussi l'ensemble des notes données par l'utilisateur aux différentes œuvres du musée. Ces notes d'intérêt qui reflètent ses préférences sont nécessaires pour le système de recommandation (section 4.6 et 4.7). La classe "relation" est utilisée pour stocker l'ensemble des utilisateurs similaires à l'utilisateur cible. De ce fait, le filtrage collaboratif pourra être utilisé directement lorsqu'un

utilisateur visite plusieurs fois le musée. Cela évite de recalculer les similarités entre utilisateurs.

- Il est utilisé pour la génération de parcours en utilisant un post-filtrage contextuel. Les deux classes *"Time" et "Location"* capturent le contexte de la visite en termes de localisation du visiteur, le temps qu'il souhaite passer dans le musée (spécifiée manuellement par l'utilisateur au début de la visite) et le temps qu'il passe devant chacune des œuvres. La méthode de génération du parcours de visite suivant le contexte de l'utilisateur est décrite dans la section 4.8.

4.4 Architecture de recommandation contextuelle

Le but de notre système est d'améliorer l'expérience des visiteurs de musées en leur recommandant les œuvres qui correspondent à leur préférences et qui sont susceptibles de les intéresser. Pour atteindre cet objectif nous proposons l'architecture décrite dans la figure 4.9. Notre travail est globalement différent des autres approches étudiées dans l'état de l'art (chapitre 3) principalement sur deux aspects. Premièrement, dans les approches dédiées à ce domaine, seules les notes d'intérêt propres de l'utilisateur sont utilisées dans le processus de recommandation, les opinions précédentes du public sur les œuvres sont souvent négligées. Or les avis d'autres utilisateurs peuvent être utilisés et peuvent être très utiles pour affiner la liste des recommandations. Deuxièmement, la plupart des approches étudiées dans l'état de l'art ne se focalisent que sur la tache de recommander une liste d'œuvres à l'utilisateur. Notre objectif est aussi de recommander à l'utilisateur un parcours constituant une liste d'œuvres à visiter de manière à ce que le déplacement dans l'espace physique se fasse de manière efficace et cela dans un temps limité (spécifié par l'utilisateur).

Notre processus de recommandation est de type hybride, c'est à dire qu'il combine plusieurs approches de recommandation. Nous mettons en œuvre trois approches différentes : démographique, sémantique et collaborative. Chaque méthode est adaptée à une étape de visite spécifique (figure 4.9). Nous distinguons trois étapes différentes de visite. Tout d'abord, quand l'utilisateur commence sa visite et entre dans le système pour la première fois, comme aucune information sur ses préférences n'est disponible puisqu'il n'a encore évalué aucune œuvre, l'approche démographique est utilisée. Il s'agit de recommander à l'utilisateur les items que les utilisateurs qui ont des attributs démographiques similaires ont appréciés. Ensuite, une fois que l'utilisateur a exprimé son avis et donné des notes aux différentes œuvres

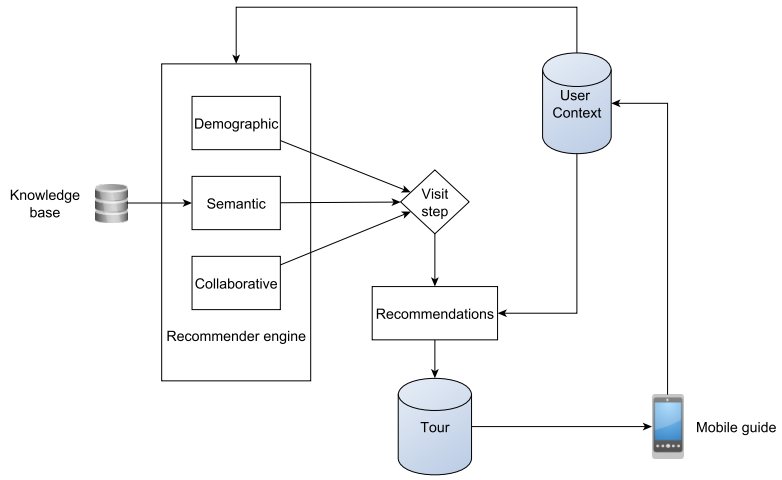


Figure 4.9 – Architecture de notre système

recommandées par l'approche démographique, l'approche sémantique est activée. Il s'agit dans cette étape de recommander à l'utilisateur des œuvres sémantiquement proches de celles qu'il a aimées, ceci à l'aide des mesures de similarités définies à partir de notre modèle de connaissances (Cf. 4.2.2). Enfin, quand le système "connait" mieux l'utilisateur et qu'il a un historique de notes suffisant, c'est l'approche collaborative qui est utilisée. L'utilisateur peut par ailleurs avoir un temps limite à passer dans le musée, il est alors important qu'il puisse découvrir les œuvres qui lui sont recommandées dans un ordre efficace afin d'éviter des allers-retours susceptibles d'engendrer de la fatigue [Jeong and Lee, 2006]. Pour cela, un post-filtrage contextuel est utilisé pour la génération de parcours. Celui-ci prend en compte la localisation du visiteur, l'environnement physique ainsi que le temps que souhaite passer le visiteur dans le musée.

4.4.1 Principe de fonctionnement

Notre objectif est de réaliser un système qui permet de recommander des œuvres et un parcours à un visiteur en fonction de son profil ainsi que ses préférences et pour cela nous nous sommes fixé un scénario d'utilisation, qui doit être à la fois cohérent avec une visite réelle de musée et aussi avec notre architecture de recommandation hybride.

A titre d'exemple, supposons qu'un utilisateur prévoit de visiter le musée du second Empire à Compiègne pour la première fois. Il ne connait donc pas grand-chose sur les collections du musée et dispose d'un temps limité pour sa visite. Il souhaite donc être efficace et voir les œuvres qui l'intéressent le plus. Il est cependant confronté à une collection assez grande et à un nombre d'informations

important : les styles, les thèmes, les artistes, etc. Il souhaiterait alors bénéficier de recommandations pertinentes. Avant de commencer sa visite, l'utilisateur saisit ses informations personnelles : âge, sexe, pays et niveau d'expertise. Il peut aussi créer un compte utilisateur et spécifier la durée de sa visite.

Notre système de recommandation est hybride et utilise trois méthodes différentes de recommandation, démographique, sémantique et collaborative de manière séquentielle en fonction de la progression de la visite. L'approche démographique est la première qui est utilisée, elle exploite directement les informations sur l'utilisateur afin de calculer une première liste de recommandations et de résoudre ainsi le problème du démarrage à froid (non connaissance des préférences de l'utilisateur). L'utilisateur peut consulter la liste des œuvres recommandées par l'approche démographique, il est alors invité à exprimer ses préférences sur les œuvres qui lui sont suggérées afin d'affiner ses préférences. Les notes données par l'utilisateur sont ensuite utilisées pour l'activation de l'approche sémantique, puis de l'approche collaborative. Une problématique est alors de définir un critère d'activation qui permet de passer d'une approche de recommandation à une autre. En d'autres termes, au bout de combien de notes bascule-t-on sur l'approche sémantique puis collaborative ? Ce critère est difficile à fixer, une analyse de la performance de la prochaine approche de recommandation à activer serait nécessaire. Ceci permettrait de savoir s'il est pertinent de basculer vers la prochaine approche de recommandation ou bien de rester sur l'approche courante. Toutefois cette approche n'est envisageable qu'en offline. Elle est donc difficile à appliquer dans le cadre d'une visite de musée réelle. Nous avons de ce fait décidé dans notre implémentation (Cf. Chapitre 6) d'utiliser un critère de déclenchement avec un pas de 10 notes. C'est à dire qu'on passe séquentiellement vers l'approche sémantique puis collaborative lorsque l'utilisateur a exprimé ses préférences sur 10 nouvelles œuvres.

Dans les sections suivantes, nous décrivons plus en détails les approches de recommandation proposées ainsi que la méthode de génération du parcours.

4.5 Approche démographique

Cette approche est la première de notre système de recommandation, c'est une alternative pour surmonter le problème du démarrage à froid au début de la visite d'un utilisateur. Comme les préférences de l'utilisateur ne sont alors pas connues, les approches classiques telles que les approches basées contenu ou collaboratives ne peuvent pas s'appliquer (problème du "nouvel utilisateur"). Nous utilisons alors les données démographiques de l'utilisateur cible telles que, l'âge, le sexe, la nationalité, le niveau d'expertise dans le domaine, la profession, etc afin de

construire un stéréotype de l'utilisateur. Nous nous basons sur l'hypothèse que, si deux utilisateurs ont des données démographiques similaires, ils auront tendance à avoir des préférences qui seront proches. A noter que d'autres approches peuvent être utilisées pour le problème du démarrage à froid, par exemple des recommandations aléatoires pour obtenir quelques notes de l'utilisateur mais cette approche potentiellement diminue considérablement la qualité des premières recommandations. On peut aussi recommander les œuvres les plus populaires, mais cette approche limite les recommandations à ces œuvres et néglige toutes les autres. Enfin, il est également possible de soumettre un questionnaire avant le début, sur les préférences de l'utilisateur, mais cette pratique est souvent mal ressentie pas la plupart des utilisateurs.

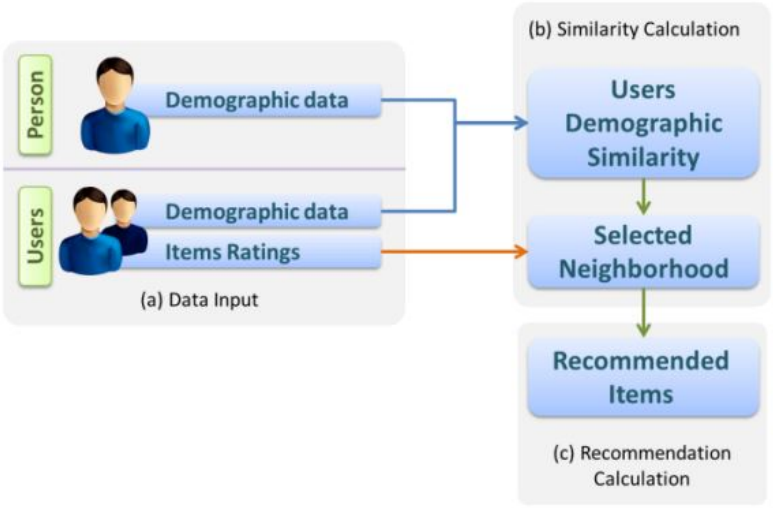


Figure 4.10 – Approche démographique pour un nouvel utilisateur [Safoury and Salah, 2013]

L'approche de recommandation démographique se fait en trois étapes : récupération des données, calcul de similarités et recommandations, comme le montre la figure 4.10. La première étape consiste à récupérer les données démographiques de l'utilisateur. Ceci se fait rapidement lors du début de la visite. En effet, lorsqu'un utilisateur s'enregistre, on récupère quelques informations utiles telles que l'âge, le sexe, le pays, le niveau d'expertise, la profession, etc. Nous supposons que nous disposons d'un historique de notes des utilisateurs du système ainsi que leurs données démographiques.

L'étape de calcul de similarité consiste à calculer la similarité entre les utilisateurs par rapport à leurs données démographiques pour construire des classes d'utilisateurs. Le but est d'identifier les utilisateurs qui ont plus ou moins les mêmes données démographiques que l'utilisateur cible. Le tableau 4.2 illustre les données

démographiques de quatre utilisateurs (*John, Jean, Sarah et Sonia*). Supposons que *Sonia* soit l'utilisateur cible auquel nous devons fournir des recommandations. Notre approche doit calculer la similarité de l'utilisateur *Sonia* avec les autres utilisateurs du système. Dans l'exemple, l'utilisateur *Sarah* est le plus similaire avec notre utilisateur cible : ils ont un âge proche, le même sexe, le même pays et la même profession.

Noms	Âge	Sexe	Pays	Profession	Niveau d'expertise
John	23	M	USA	vendeur	avancé
Jean	54	M	France	professeur	expérimenté
Sarah	26	F	France	étudiant	débutant
Sonia	28	F	France	étudiant	novice

Tableau 4.2 – Exemple de données démographiques de quatre utilisateurs

La plupart des données démographiques sont nominales, mais l'âge est une information numérique. Nous pouvons partitionner les données numériques telles que l'âge en intervalles (5 ans par exemple). La formule suivante calcule la similarité entre les âges des utilisateurs a et b :

$$S_{age}(a,b) = \begin{cases} 1 - \frac{|age(a)-age(b)|}{5}, & \text{si } |age(a) - age(b)| < 5 \\ 0 & sinon \end{cases}$$

Pour les autres données démographiques qui sont nominales telles que le sexe ou le pays, nous pouvons simplement dire que la similarité est égale à 1 si les deux utilisateurs ont le même attribut, 0 sinon.

La mesure suivante représente la similarité entre deux utilisateurs par rapport à leurs attributs démographiques.

$$sim_{dem}(a,b) = \sum_{k} \alpha_k \times S_k(a,b) \tag{4.1}$$

ou, $S_k(a,b)$ est la similarité entre les deux utilisateurs par rapport au k^{ieme} attribut démographique. α_k représente le facteur d'importance du k^{ieme} attribut démographique.

Finalement, la dernière étape de l'approche démographique est la recommandation. En effet, une fois que nous avons trouvé l'ensemble des k utilisateurs les plus proches de notre utilisateur cible, nous recommandons les œuvres que ces utilisateurs proches ont notées positivement. Cette liste d'œuvres est la première liste de recommandations de notre système hybride. L'utilisateur peut alors commencer à donner son avis sur ces recommandations par le biais de notes, qui seront alors

utilisées pour activer la deuxième étape de visite et en conséquence notre approche de recommandation sémantique que nous détaillons dans la section qui suit.

4.6 Approche sémantique

Cette approche a pour objectif de recommander à l'utilisateur les œuvres qui sont proches sémantiquement de celle qu'il a déjà aimées. Par exemple, si un utilisateur aime l'œuvre *Mona Lisa* de *Leonardo da Vinci*, d'autres œuvres du même artiste peuvent l'intéresser. Nous utilisons pour cela les notes positives qu'il a données aux différentes œuvres et le modèle sémantique pour le processus de recommandation. Une œuvre qui a une forte similarité avec des œuvres que l'utilisateur a aimées a de forte chances d'être recommandée.

Nous avons suivi l'approche générale des similarités par propriétés proposée par [Pirró and Euzenat, 2010], en l'adaptant à notre cadre d'application. Rappelons que le principe de cette approche est de déterminer la similarité entre deux objets par la similarité de leurs propriétés. Plus les objets possèdent des propriétés en commun et plus ils peuvent être considérés comme similaires. Cette méthode de calcul de la similarité est bien adaptée aux concepts d'une ontologie. Mais dans notre système, nous ne cherchons pas à savoir si un concept est proche d'un autre concept, par exemple "peinture" et "sculpture", nous comparons plutôt une "peinture" x avec une autre "peinture" y en cherchant à déterminer ce qu'elles ont de commun. Nous nous intéressons donc aux instances des concepts et non aux concepts eux-mêmes. Plutôt que de comparer les propriétés des concepts, nous comparons donc les valeurs que prennent les propriétés des instances de ces concepts. De ce fait, la similarité entre deux œuvres est calculée en comparant les styles, les thèmes, les artistes, etc. de ces deux œuvres.

Il s'agit donc, pour disposer d'une mesure de similarité sémantique, de comparer les instances de la base de connaissances suivant notre modèle sémantique, en fonction des valeurs que prennent leurs propriétés. Plus les valeurs prises par les propriétés sont similaires, plus les instances sont similaires. Le problème auquel nous sommes confronté pour la construction de cette mesure de similarité réside dans la variété des types de valeurs que peuvent prendre les propriétés. En effet, les propriétés peuvent avoir pour codomaine des instances organisées sous forme de hiérarchies (Ex. Les styles des œuvres : *cime :hasStyle*), des instances non organisées hiérarchiquement, (Ex. Les maîtres d'un artiste : *cime :masterof*) ou encore des *datatype*, (Ex. Les dates de création des œuvres : *cime :hasCreationDate*). Les méthodes de calculs de similarité seront alors différentes suivant ces cas. Par exemple, comparer deux œuvres suivant leurs styles ne se fera pas de la même

façon que comparer deux œuvres suivant leurs dates de création. Notre calcul de similarité doit donc permettre d'intégrer ces informations hétérogènes pour en tirer une valeur numérique entre 0 et 1. Enfin, pour pouvoir comparer deux instances de la base de connaissance suivant les valeurs que prennent leurs propriétés, il est évidemment nécessaire que ces instances possèdent des propriétés communes. Dans notre approche, c'est toujours le cas car nous comparons toujours des instances issues du même concept. En effet, nous nous intéressons pas à comparer par exemple une œuvre de *Van Gogh* avec l'artiste *Leonardo da Vinci* mais plutôt *Van Gogh* avec *Leonardo da Vinci*, nous comparons alors toujours deux instances du même concept, c'est à dire, deux styles, deux thèmes, deux artistes, etc.

4.6.1 Formalisation

Étant donné I l'ensemble des œuvres candidates à la recommandation, U l'ensemble des utilisateurs, $u \in U$ un utilisateur actif, nous cherchons à recommander à l'utilisateur u un sous-ensemble de la liste des œuvres I, qui soient similaires à celles qu'il a aimées. Pour cela, comme nous l'avons expliqué précédemment nous avons besoin de savoir quantifier de manière numérique la similarité entre deux œuvres o_i et o_j, $o_i, o_j \in I$.

Nous définissons pour cela, la fonction de similarité $SIM : I \times I \rightarrow [0, 1]$ par :

$$SIM(o_i, o_j) = \sum_k W_k * SIM_k(o_i, o_j) \tag{4.2}$$

La similarité entre deux œuvres o_i et o_j est ainsi la somme pondérée des similarités entre les valeurs que prennent les propriétés des deux œuvres. En d'autres termes, la similarité entre les deux œuvres est fonction de la similarité de leurs artistes, de leurs styles, de leurs thèmes, etc.

W_k est un poids représentant le facteur d'importance de la propriété k, avec $\sum_k W_k = 1$. Pour fixer la valeur de ces poids, nous pouvons consulter l'avis d'un expert en muséologie pour déterminer l'importance qu'a une propriété par rapport à l'appréciation d'une œuvre.

SIM_k représente la similarité entre les deux œuvres o_i et o_j par rapport à la propriété k. Par exemple, si k représente la propriété *hasStyle* alors $SIM_k(o_i, o_j)$ représente la similarité entre les styles des deux œuvres o_i et o_j.

Nous avons donc besoin d'un calcul générique permettant de comparer les valeurs que prennent deux instances x et y du même concept c de la base de connaissance suivant une propriété donnée k, c'est à dire comparer deux ensembles d'instances ou de valeurs littérales. Cette comparaison nous permettra de calculer $SIM_k(x, y)$ qui

est de manière générale la similarité entre deux instances x et y du même concept suivant la propriété k.

Nous détaillons maintenant les différentes approches de calcul de cette similarité, correspondant aux différentes catégories de valeurs que peuvent prendre les propriétés. Ces types de valeurs sont les instances organisées hiérarchiquement, les instances non organisées hiérarchiquement et les littéraux.

Soient alors x et y deux instances d'un même concept, nous notons $P_k(x) = \{x_1, x_2, ..., x_n\}$ et $P_k(y) = \{y_1, y_2, ..., y_m\}$ l'ensemble des instances qui sont liées à x, respectivement y, par rapport à la propriété k. Nous détaillons dans les sous-sections suivantes la façon de calculer la similarité par rapport à la propriété k pour les trois catégories différentes.

4.6.1.1 Cas où les valeurs de la propriété sont des instances organisées hiérarchiquement

Dans le cas où les valeurs de la propriété k sont des instances hiérarchisées, par exemple des instances issues d'ICONCLASS (thèmes) ou bien de Getty-AAT (styles), la valeur de la similarité doit dépendre de la similarité des instances dans ces hiérarchies.

Tout d'abord supposons qu'il n'y ait que deux instances à comparer, c'est-à-dire que les instances x et y n'aient chacune qu'une valeur, respectivement x_1 et y_1 pour la propriété k, i.e $P_k(x) = \{x_1\}$ et $P_k(y) = \{y_1\}$. Dans ce ca, nous utilisons la similarité définie par [Wu and Palmer, 1994] (cf. chapitre 2). Cette mesure est en effet celle qui est la mieux adaptée pour déterminer la similarité entre deux éléments d'une hiérarchie. Formellement, la similarité est calculée ainsi :

$$SIM_k(x, y) = SIM_{wu}(x_1, y_1) = \frac{2 \times depth(lcs(x_1, y_1))}{depth(x_1) + depth(y_1)} \tag{4.3}$$

Elle peut être calculée avec la seule information de la structure de la hiérarchie sous forme d'arbre. Elle prend en compte la spécificité des éléments pour la comparaison entre les instances, c'est-à-dire que deux éléments situés profondément dans la hiérarchie seront plus similaires que deux éléments situés en haut de la hiérarchie, même si le même nombre d'arcs les séparent. De plus, cette similarité est normalisée. Nous rappelons que $depth(x_1)$ et $depth(y_1)$ représentent respectivement la profondeur de x_1, respectivement y_1 dans la hiérarchie (le nombre d'arcs les séparant de la racine de la hiérarchie). $lcs(x_1, y_1)$ est le *least common subsummer* de x_1 et y_1.

Si x et y ont plusieurs valeurs pour la propriété k, c'est-à-dire $P_k(x) = \{x_1, x_2, ..., x_n\}$ et $P_k(y) = \{y_1, y_2, ..., y_m\}$, nous calculons la similarité moyenne entre

les instances de x et de y vis-à-vis de la propriété k. Pour chaque instance liée à x via la propriété k, nous déterminons l'instance la plus proche dans les valeurs de y pour la propriété k selon la similarité de *Wu & Palmer*. Nous effectuons la moyenne de ces différentes similarités. La formule suivante présente le cas général de calcul de similarité dans le cas ou la propriété k prend ses valeurs parmi un ensemble d'instances hiérarchisées.

$$SIM_k(x,y) = \frac{\sum_{i \in P_k(x)} Max_{j \in P_k(y)} SIM_{wu}(i,j)}{2 \times |P_k(x)|} + \frac{\sum_{j \in P_k(y)} Max_{i \in P_k(x)} SIM_{wu}(j,i)}{2 \times |P_k(y)|} \tag{4.4}$$

$P_k(x)$ et $P_k(y)$ étant respectivement l'ensemble des valeurs prises par x et y pour la propriété k. Nous notons que cette similarité est bien symétrique et comprise entre 0 et 1. Elle augmente si les groupes d'instances associées aux instances x et y sont proches pour atteindre son maximum (1) si ces groupes sont identiques.

4.6.1.2 Cas où les valeurs de la propriété sont des instances non hiérarchisées

Nous décrivons maintenant le calcul de la similarité pour les propriétés ayant pour valeur des instances de la base de connaissances qui ne font pas partie d'une hiérarchie. Par exemple, c'est le cas si nous considérons la propriété *cime :participatedIn*, cette propriété lie des instances de type *ulan/Person* (artiste) à des instances de type *cidoc :event* (événements). Les événements auxquels a participé un artiste ne sont pas organisés de manière hiérarchique.

Supposons que nous souhaitions calculer la similarité entre deux artistes x_1 et y_1, qui sont des instances du concept *ulan :person*, et que l'on cherche à déterminer la similarité entre x_1 et y_1 vis-à-vis d'une propriété k. Dans ce cas, plus le nombre d'instances communes à x_1 et y_1 vis-à-vis d'une propriété k est important, plus la similarité sera élevée. Dans le cas de la propriété *cime :participatedIn*, par exemple, cela revient à dire que deux artistes sont proches si ils ont tous les deux été impliqués dans les mêmes événements.

Cette assertion en langage naturel correspond l'indice de Jaccard. Nous utilisons donc cet indice entre les valeurs des propriétés pour le calcul de la similarité. L'indice de Jaccard est en effet bien adapté pour déterminer la similarité entre deux ensembles, l'ensemble des valeurs de l'instance x pour la propriété k et l'ensemble des valeurs de l'instance y pour la propriété k. Formellement, si x et y sont deux instances qui ont pour valeur respectivement $P_k(x) = \{x_1, x_2, ..., x_n\}$ et $P_k(y) = \{y_1, y_2, ..., y_m\}$

pour la propriété k, la valeur de similarité est :

$$\begin{aligned} SIM_k(x,y) &= SIM_{jaccard}(P_k(x), P_k(y)) \\ &= \frac{|P_k(x) \cap P_k(y)|}{|P_k(x) \cup P_k(y)|} \\ &= \frac{|\{x_1, x_2, ..., x_n\} \cap \{y_1, y_2, ..., y_n\}|}{|\{x_1, x_2, ..., x_n\} \cup \{y_1, y_2, ..., y_n\}|} \end{aligned} \tag{4.5}$$

On constate aisément que la similarité entre deux instances est symétrique et toujours comprise entre 0 et 1, avec un maximum à 1 lorsque les ensembles comparés sont identiques.

4.6.1.3 Cas où les valeurs de la propriété sont de type littéral

Finalement, une propriété peut également prendre comme valeur des littéraux. Ce cas est plus délicat à analyser en terme de similarité, car il est indispensable de disposer d'une fonction de comparaison entre littéraux, une telle fonction ne pouvant pas, par nature, être générique.

Mais, dans notre modèle sémantique, le seul cas possible de comparaison de deux littéraux intervient lorsque nous avons à faire avec le type *xsd :date*, c'est-à-dire que les littéraux considérés sont des dates. Cela correspond aux dates de création des œuvres, de naissance ou mort d'un artiste et à des événements. Nous avons défini, pour ces différents cas, une fonction permettant la comparaison de dates. La fonction donnant la valeur de la similarité dépend de la durée, exprimée en années : $d = |d_1 - d_2|$. On a :

$$Sim_{date}(d_1, d_2) = \begin{cases} \frac{100-d}{100} & \text{, si } 100 - d < 0 \\ 0 & sinon \end{cases}$$

4.6.2 Cas des liens directs entre instances

Il nous reste un dernier cas à étudier et que nous avons pas abordé précédemment, ce cas correspond aux liens directs entre instances. Cela peut arriver par exemple dans le cas ou on compare deux artistes, un artiste peut être le maître d'un autre artiste ou bien son élève. Si les instances comparées sont x et y du même concept, ces liens directs prennent la forme xP_ky ou bien yP_kx, où k est une propriété reliant un concept à lui même. Dans ce cas, nous nous inspirons de la méthode de similarité proposée par [Aimé, 2011], en définissant la similarité entre x et y, instances d'un concept C, telles que $x \neq y$, comme le rapport entre le nombre de propriétés existant

dans la base de connaissance entre x et y et l'ensemble des propriétés liant le concept c à lui même. Notons $P(A, B)$ l'ensemble de toutes les propriétés reliant le concept A au concept B. La similarité est alors calculée ainsi :

$$SIM_{link}(x, y) = \frac{|(xP_iy) \cup (yP_jx)|}{|P(C, C)|} \tag{4.6}$$

$P(C, C)$ étant l'ensemble des propriétés existantes entre instances de C. Nous notons que cette expression est nécessairement inférieure ou égale à 1. Par ailleurs, afin d'obtenir une similarité maximale lors de la comparaison d'instances identiques, nous posons quelque soit x instance d'un concept C de la base de connaissances :

$$SIM_{link}(x, x) = 1 \tag{4.7}$$

4.6.3 Prédictions et recommandations

Maintenant que nous savons quantifier de manière numérique la similarité entre deux œuvres de la base de connaissances, nous pouvons estimer le score de prédiction pour une œuvre o non encore consultée par l'utilisateur u. Pour cela nous utilisons la liste des œuvres aimées par l'utilisateur u : plus une œuvre candidate à la recommandation est similaire aux œuvres que l'utilisateur a aimées, plus elle a de chances d'être recommandée. La formule suivante calcule le score de prédiction :

$$preferences(u) = \{o_i / u \ aime \ o_i\} \tag{4.8}$$

$$Pred(u, o) = \frac{\sum_{o_i \in preferences(u)} SIM(o_i, o)}{|preferences(u)|} \tag{4.9}$$

$preferences(u)$ est l'ensemble des œuvres que l'utilisateur a appréciées. Une œuvre o est ajoutée à l'ensemble des préférences de l'utilisateur u si et seulement si la note qu'il lui a attribuée est supérieure à une seuil défini. Ainsi, le score de prédiction d'une œuvre non encore notée par l'utilisateur est calculé en fonction de sa similarité sémantique avec les œuvres présentes dans l'ensemble des préférences de l'utilisateur.

Une fois les scores de prédiction calculés pour toutes les œuvres non consultées par l'utilisateur considéré u, nous trions la liste de ces œuvres en fonction de leur score respectif pour aboutir à une liste de recommandations sémantiques.

4.7 Approche collaborative

L'approche collaborative est activée, lorsque le profil des préférences de l'utilisateur est un peu plus riche et qu'il a notées un nombre d'œuvres suffisant. Le but de notre approche collaborative est de recommander à l'utilisateur des œuvres parmi celles candidates à la recommandation, en se basant sur l'appréciation des utilisateurs similaires sur les œuvres qu'il n'a pas encore notées. L'idée est que si un utilisateur appartient à un groupe d'utilisateurs similaires, les œuvres aimées par ce groupe pourraient intéresser l'utilisateur. Pour ce faire, des mesures de similarité entre utilisateurs sont calculées, elles utilisent les notes des utilisateurs sur les différents items.

4.7.1 Similarité entre utilisateurs

Pour définir une mesure de similarité entre deux utilisateurs, une approche basique est de compter la proportion des œuvres en commun dans leurs historiques de notes en utilisant par exemple l'indice de Jaccard. La méthode PCC (Pearson's Correlation Coefficient) reste cependant la mesure la plus utilisée dans la littérature [Schafer et al., 2007]. Pour calculer la similarité entre deux utilisateurs u et v, le coefficient de corrélation de Pearson mesure le rapport entre leur covariance et leur écart-type. Plus les deux utilisateurs auront tendance à noter les mêmes œuvres de façon équivalente plus ils seront similaires, comme le montre l'équation suivante :

$$sim(u,v) = Pearson(u,v) = \frac{\sum_{i \in I_{uv}} (r_{u,i} - \bar{r_u}).(r_{v,i} - \bar{r_v})}{\sqrt{\sum_{i \in I_{uv}} (r_{u,i} - \bar{r_u})^2}.\sqrt{\sum_{i \in I_{uv}} (r_{v,i} - \bar{r_v})^2}} \quad (4.10)$$

où $\overline{r_u}$ et $\overline{r_v}$ sont la moyenne des notes données par l'utilisateur u et v respectivement et $r_{u,i}$, $r_{v,i}$ les notes données pour l'œuvre i par u et v respectivement.

La limite de cette formule est que la similarité entre deux utilisateurs est calculée en se basant sur les œuvres qui ont été notées par les deux utilisateurs à la fois. En effet, cette similarité n'est significative que si les deux utilisateurs ont notés un nombre important d'œuvres en commun. Cela apparait notamment quand la matrice utilisateur-item est très "parcimonieuse" (sparse), l'ensemble des œuvres notées en commun par les deux utilisateurs étant très petit ou parfois nul.

La figure 4.11 illustre ces limites. Dans cet exemple, nous avons une matrice de dix items et six utilisateurs. Pour prédire la préférence de l'utilisateur u_3 sur l'item i_2, seuls les utilisateurs u_2 et u_5 ont notés l'item considéré i_2. Pour calculer la similarité entre ces utilisateurs (u_2 et u_5) et l'utilisateur cible (u_3), l'ensemble

d'items communs dans leurs historiques est seulement l'item i_5. Il est évident que la calcul de similarité utilisant juste cette information n'est pas suffisante pour statuer sur la similarité entre ces différents utilisateurs.

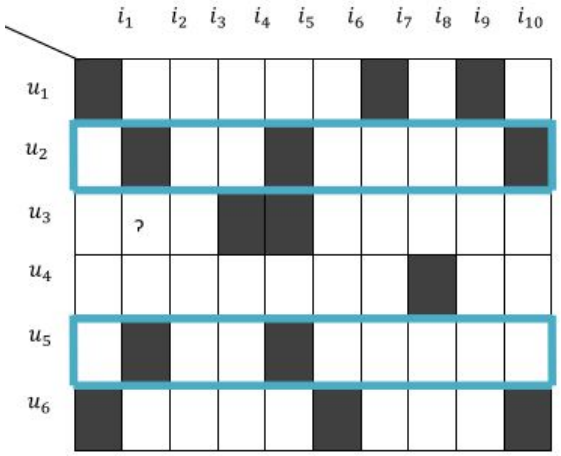


Figure 4.11 – Explicitation du problème de parcimonie

Pour faire face à ce problème, nous proposons d'intégrer l'information sémantique existante entre les œuvres dans le processus de calcul de la similarité entre deux utilisateurs. Par exemple : considérons deux utilisateurs, le premier ayant aimé l'œuvre "la Joconde" ainsi que l'œuvre "Saint Jean Baptiste", le deuxième ayant aimé l'œuvre "la dame à l'hermine" ainsi que "la vierge aux rochers". Dans ce cas, comme nous venons de le voir, la mesure de Pearson donnera une similarité égale à 0 entre ces deux utilisateurs car aucune œuvre n'est en commun entre les deux utilisateurs. Cependant, la similarité ne doit pas être nulle car les quatre œuvres citées ci-dessus sont proches sémantiquement puisqu'elles sont toutes du même artiste "Léonard de Vinci".

Intégrer l'information sémantique entre les œuvres peut alors être très bénéfique. Il s'agit de dire que, même si deux utilisateurs n'ont pas beaucoup d'œuvres en commun dans leurs historiques, si les œuvres aimées sont proches sémantiquement, la similarité entre ces deux utilisateurs est forte. Nous définissons alors la similarité entre deux utilisateurs comme suit :

$$sim(u, v) = max(sim(u * v), sim(v * u)) \tag{4.11}$$

$$sim(u * v) = \frac{\sum_{k_1} \max_{k2}(SIM(o_{k1}, o'_{k2}))}{k_1} \tag{4.12}$$

$$sim(v * u) = \frac{\sum_{k_2} \max_{k1}(SIM(o'_{k2}, o_{k1}))}{k_2} \tag{4.13}$$

où l'utilisateur u a aimé les œuvres o_1 jusqu'à o_{k1}, et l'utilisateur u a aimé les œuvres o'_1 à o'_{k2}, SIM représente la mesure de similarité définie dans l'approche sémantique.

Il s'agit alors de déterminer la similarité entre deux utilisateurs en fonction de la similarité des œuvres qu'ils ont aimées. Cette mesure nous permet alors de faire face au problème de parcimonie (sparsity).

Même si un utilisateur cible n'a pas noté beaucoup d'œuvres en commun avec d'autres utilisateurs, nous utiliserons les utilisateurs qui ont noté des œuvres similaires à celles qu'il a aimé.

4.7.2 Prédictions et recommandations

Une fois la mesure de similarité entre utilisateurs définie, on peut maintenant estimer si une nouvelle œuvre o non encore consultée par l'utilisateur u peut être intéressante pour lui. Pour cela, nous calculons un score de prédiction qui est la note prédite de l'utilisateur u pour l'œuvre o. Nous utilisons les notes que les k utilisateurs les plus similaires (voisins) à l'utilisateur u ont donné à cette œuvre o :

$$pred(u,o) = \bar{r_u} + \frac{\sum_{w \in voisins(u) \cap U_i} sim(w,u) \times (r_{w,o} - \bar{r_w})}{\sum_{w \in voisins(u) \cap U_i} sim(w,u)} \tag{4.14}$$

Une fois les scores de prédiction calculés pour toutes les œuvres non consultées par l'utilisateur considéré u, nous trions la liste de ces œuvres en fonction de leur score respectif.

4.8 Génération de parcours de visite

Nous recommandons au visiteur un parcours de visite qui est basé sur les résultats du système de recommandation, la liste de recommandations ainsi que les œuvres que l'utilisateur a aimées sont triées par ordre de pertinence en fonction de leur score. Notons que le parcours n'est établi que lorsque l'utilisateur a exprimé ses préférences sur les œuvres et qu'il a utilisé le système de recommandation à l'aide d'un smartphone. Les œuvres qui sont alors sélectionnées pour le parcours final sont les œuvres que l'utilisateur a notées positivement ainsi que celles qui lui sont recommandées.

Notons cependant que les œuvres qui sont présentes dans cette liste peuvent être exposées n'importe où dans l'espace réel, ce qui peut engendrer une visite avec un nombre important d'aller-retours. De plus, le visiteur ne sait pas à l'avance où se

trouvent les œuvres qu'il souhaite voir. Enfin, il peut avoir un temps limité a passer dans le musée.

Nous proposons alors d'utiliser un post-filtrage contextuel qui va trier à nouveau la liste des œuvres à visiter, de manière à ce que le parcours effectué soit optimal. Les informations contextuelles sur l'environnement physique (l'emplacement des œuvres), la position du visiteur ainsi que le temps de la visite sont utilisées pour proposer un parcours.

Nous définissons le parcours en musée comme étant un graphe orienté, où les nœuds du graphe représentent les œuvres et les arcs entre deux nœuds le temps nécessaire au déplacement entre les deux œuvres. Avec l'hypothèse qu'un utilisateur ne visite qu'une seule fois une même œuvre, notre problème se ramène alors au problème d'optimisation "Orienteering problem" (OP) [Vansteenwegen et al., 2011b].

Formellement, nous avons un ensemble de nœuds correspondant aux différentes œuvres, un score S_i correspond au score de pertinence d'une œuvre. Le parcours doit commencer au nœud 1 (l'entrée par exemple) et se finir au nœud N (la sortie par exemple). t_{ij} est l'arc reliant les nœuds i et j et représente le temps nécessaire pour se déplacer de l'œuvre i à l'œuvre j. Notons t_i le temps passé par l'utilisateur devant l'œuvre i.Ce temps est difficile à évaluer de manière précise. Nous utilisons pour chacune des œuvres du musée un temps de visite qui est une valeur estimée. Cette estimation peut se faire en fonction de ses préférences, en supposant que l'utilisateur passe plus de temps devant les œuvres qu'il apprécie le plus que devant les œuvres qu'il apprécie moins. L'utilisateur dispose d'un temps limité pour la visite que nous notons : T_{max}. Notre objectif est alors de déterminer un chemin qui maximise le score total des œuvres visitées avec la contrainte sur le temps T_{max}.

En faisant usage de la notation indiquée ci-dessus, le problème peut être formulé comme un programme linéaire en nombres entiers. Les variables de décision suivantes sont utilisées : $x_{ij} = 1$ si le visiteur se déplace de l'œuvre i vers l'œuvre j, 0 sinon.

$$Max \sum_{i=2}^{N-1} \sum_{j=2}^{N} S_i x_{ij} \tag{4.15}$$

$$\sum_{j=2}^{N} x_{1j} = \sum_{i=1}^{N-1} x_{iN} = 1 \tag{4.16}$$

$$\sum_{i=1}^{N-1} x_{ik} = \sum_{j=2}^{N} x_{kj} \leq 1, \forall k = 2, ..., N-1 \tag{4.17}$$

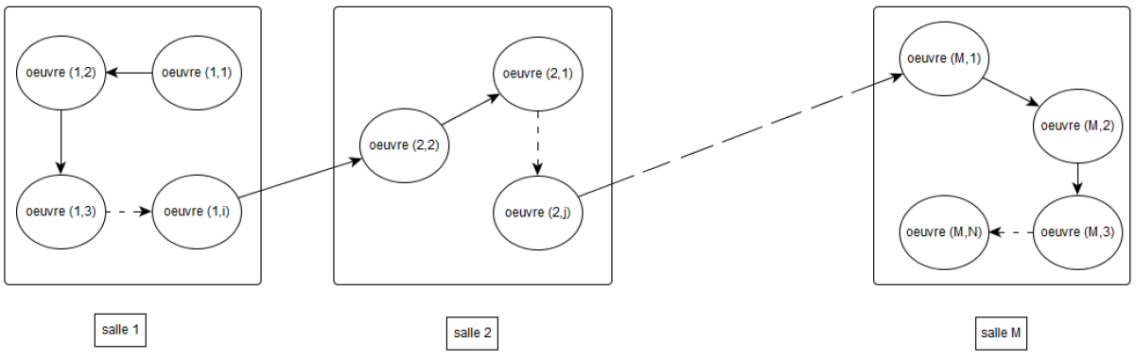


Figure 4.12 – Composition du parcours en fonction des salles.

$$\sum_{i=1}^{N-1}\sum_{j=2}^{N} t_{ij}x_{ij} + t_j \leq Tmax \tag{4.18}$$

la fonction objective à maximiser (4.15) est le score total engendré par la visite des œuvres les plus intéressantes. La contrainte (4.16) garantit que le parcours commence au nœud 1 et se termine au nœud N. La contrainte (4.17) assure la connectivité du parcours et que chaque œuvre est visitée au plus une fois. La contrainte (4.18) assure que le temps limite de la visite ne soit pas dépassé.

Ce problème d'optimisation est connu pour être NP-difficile, à ce jour des chercheurs dans le domaine de l'optimisation travaillent sur le développement d'algorithmes et d'heuristiques pour résoudre ce problème, qui est donc hors de notre champ de compétences. Nous proposons alors une méthode assez simple pour la génération du parcours.

Généralement, le musée est organisé en salles. Dans chaque salle sont exposées un ensemble d'œuvres, et une salle peut être adjacente à une ou plusieurs salles. En supposant que l'utilisateur ne voit qu'une seule fois la même œuvre, nous pouvons proposer une approche de type "diviser pour régner". La première étape de cette approche consiste à rechercher des séquences de salles où se trouvent les œuvres qui sont candidates au parcours pour réorganiser la liste des œuvres dans chaque salle. La deuxième étape consiste à résoudre le problème OP localement pour chaque salle, en utilisant un algorithme "Greedy". C'est-à-dire qu'on sélectionne systématiquement à chaque étape l'œuvre qui a le plus grand score et qui est la plus proche de la position actuelle de l'utilisateur, ce qui conduit généralement à une solution raisonnable qui est proche de la solution optimale. Et enfin, une étape de composition (figure 4.12) permet de relier les œuvres entre les différentes salles du musée.

4.9 Conclusion

Nous avons présenté dans ce chapitre la visite de musée et la possibilité d'améliorer l'expérience du visiteur au moyen des systèmes de recommandation. Nous avons présenté notre architecture de recommandation qui se base à la fois sur la modélisation sémantique du domaine et sur des techniques de recommandation sémantiquement améliorées. Notre approche hybride est constituée de trois approches de recommandation : démographique, sémantique et collaborative, chacune des méthodes étant activée à une étape de visite spécifique. Les résultats de la recommandation sont ensuite utilisés dans un post-filtrage contextuel pour permettre la génération de parcours optimaux.

Dans ce chapitre, nous nous sommes intéressé à la recommandation d'un seul type d'item, en l'occurrence les tableaux d'un musée. Dans le cadre de la visite de musée, il pourrait être intéressant de considérer plusieurs types d'items (sculptures, arts décoratifs, arts graphiques, etc). Le chapitre suivant aborde la recommandation composite en considérant cette fois le domaine du tourisme.

Un système de recommandation composite pour la planification d'activités touristiques

Sommaire

5.1 Introduction et motivation

La quantité d'informations disponibles dans le domaine du tourisme sur le Web a connu une très forte augmentation durant la dernière décennie et a changé la manière dont les touristes planifient leurs voyages. Ces informations concernent les destinations de voyage et leurs ressources associées, telles que les hébergements, les attractions, les musées ou les événements.

Toutes ces informations peuvent être d'une aide considérable pour les utilisateurs qui prévoient de faire un voyage et de visiter une ville pour la première fois. Cependant, la liste des possibilités offertes par les moteurs de recherche, ou des sites spécialisés de tourisme comme Tripadvisor[1] peut être trop large pour les utilisateurs. L'exploitation de cette longue liste d'options est très complexe, les touristes peuvent

[1]https ://www.tripadvisor.fr/

passer beaucoup de temps pour selectionner les options qui correspondent le plus à leurs intérêts.

Là encore, les systèmes de recommandation peuvent être utiles pour résoudre ce problème. Les systèmes de recommandation ont d'ailleurs trouvé leur place dans le contexte du tourisme durant la dernière décennie [Ricci, 2002]. Ils visent à faire correspondre les caractéristiques des points d'intérêt ou des attractions aux besoins et aux préférences de l'utilisateur. Ces systèmes sont particulièrement utiles lorsqu'ils peuvent automatiquement apprendre les préférences de l'utilisateur par l'analyse de ses retours d'information explicites ou implicites sur les différents point d'intérêt [Sieg et al., 2007].

Les systèmes de recommandation classiques fournissent à l'utilisateur cible des recommandations sous la forme d'une liste triée par ordre de pertinence (score). Ces listes triées sont constituées d'un seul type d'item (films, livres, œuvres, etc).

Dans le domaine du tourisme et de la planification de voyages, l'utilisateur est susceptible d'être intéressé par des suggestions de points d'intérêt (POI) qui peuvent être très hétérogènes : musées, parcs, restaurants, monuments, etc. Un système de recommandation classique n'est donc pas très adapté à ce type de domaine. Considérons par exemple, un touriste qui souhaite visiter une ville. Dans ce scénario le visiteur ne serait probablement pas intéressé par un système qui ne lui recommanderait que des musées par exemple. Il souhaiterait aussi avoir des suggestions d'hôtels, de monuments, de restaurants, etc. Un système de recommandation pour le tourisme doit donc pouvoir recommander à l'utilisateur des points d'intérêt organisés sous forme de packages (bundles) hétérogènes au lieu de listes triées homogènes. Ce sont donc les packages qui correspondent le mieux aux préférences de l'utilisateur qui doivent lui être proposés, chaque package étant constitué d'un ensemble de points d'intérêt (POI).

De plus, il peut y avoir un coût et un temps nécessaire pour visiter chaque point d'intérêt, que l'utilisateur peut vouloir contraindre avec un budget à ne pas dépasser. Le budget peut aussi simplement être le nombre d'items dans chaque package. Optionnellement, l'utilisateur peut spécifier des contraintes au sein d'un même package, par exemple : "pas plus de trois musées dans un package", "pas plus de deux restaurants" ou bien "la distance totale pour visiter tous les POIs d'un package doit être inférieure à 20 kilomètres", etc. Certains sites web dits de *troisième génération* spécialisés dans la planification de voyages, comme par exemple "YourTour" visent à aider l'utilisateur avec des suggestions de points d'intérêt intégrant ce type de contraintes, mais les recommandations sont souvent basées seulement sur les POIs les plus populaires et négligent l'aspect de personnalisation des parcours de visite pour l'utilisateur. De ce fait, l'utilisation de ce genre de sites

reste très limitée.

Étant donnée une collection de points d'intérêt, où chaque POI a un coût et un temps de visite moyen, et un utilisateur cible spécifiant une valeur maximale pour le coût et le temps (budgets), notre but est de recommander les packages les plus intéressants pour l'utilisateur (Top-k packages), chaque package devant respecter des contraintes éventuelles sur les budgets.

Notre contribution réside dans la conception et la mise en œuvre d'un modèle de recommandation promouvant la diversité et inspirée de la recherche composite [Amer-Yahia et al., 2014]. L'ensemble des packages recommandés couvrira alors une large diversité de thèmes. Pour chaque package recommandé, les contraintes de l'utilisateur (budgets) doivent être satisfaites. Les POIs dans chaque package sont choisis en fonction d'un score qui prend en compte les préférences de l'utilisateur, la diversité des items incluant les packages ainsi que la popularité des POIs.

5.2 Systèmes de recommandation composites

Avant de présenter notre approche, nous évoquons dans cette section un état de l'art sur les systèmes de recommandation composites.

Dans les travaux de [Angel et al., 2009], les auteurs se sont intéressés à la recherche des Top-k tuples d'entités. Des exemples d'entités peuvent être des villes, des hôtels, ou des vols d'avions. Les packages sont des tuples d'entités, et des requêtes sont exécutées sur des documents en utilisant des mots-clés pour déterminer les scores des entités. Dans leur framework, un package est toujours d'une taille fixe, par exemple, une ville, un hôtel, et une compagnie aérienne. En revanche, dans notre travail, nous recommandons des packages (recommandations composites) qui peuvent être de taille variable, soumis à une contrainte de budget spécifiée par l'utilisateur.

CARD [Brodsky et al., 2008] est un framework dont le but est de trouver les Top-k recommandations composites de produits et services. Un langage similaire à SQL est proposé afin de prendre en compte les exigences de l'utilisateur sous forme de requêtes et de définir comment les coûts atomiques de chaque item sont combinés. Cependant, comme dans les travaux de [Angel et al., 2009], les packages ont toujours une taille fixe, ce qui simplifie largement le problème.

CourseRank [Parameswaran et al., 2009] est un projet centré sur la recommandation de cours, dans le but d'aider les étudiants à planifier leur programme académique à l'université de Stanford. L'ensemble des cours recommandés doit aussi satisfaire des contraintes par exemple : prendre deux sessions d'un ensemble de cinq cours de mathématiques. De façon analogue à notre travail, à chaque cours est associé un

score calculé utilisant un moteur de recommandation sous-jacent. Plus précisément, la popularité des cours au sein de l'université ainsi que les cours choisis par des utilisateurs similaires (qui ont le même cursus par exemple) sont utilisés. Étant donné un certain nombre de contraintes, qui peuvent être spécifiées par l'utilisateur ou par l'université, le système trouve un ensemble de cours qui satisfont ces exigences, et qui a le score plus élevé possible.

Les mêmes auteurs dans [Parameswaran et al., 2011] étendent le système CourseRank avec des contraintes au préalable, où il existe un ordre important entre les cours à recommander, exprimées sous forme de contraintes de pré-requis (par exemple, le cours "Analyse réelle" est un pré-requis du cours "Analyse complexe"). Le problème est NP-difficile et les auteurs proposent plusieurs algorithmes d'approximation qui retournent des recommandations de cours de haute qualité, et qui satisfont toutes les contraintes de pré-requis. Les packages recommandés sont cette fois de taille variable. Cependant, les auteurs de [Parameswaran et al., 2009, Parameswaran et al., 2011] ne considèrent pas de contraintes pour les items (i.e les cours), tandis que nous prenons en compte le coût et le temps engendré pour la visite de chaque POI, que l'utilisateur peut contraindre avec un budget, ces aspects sont essentiels dans des applications du domaine du tourisme et de la planification de voyages que nous considérons.

Un autre travail proche du nôtre est celui de [De Choudhury et al., 2010], qui propose un framework pour recommander automatiquement des parcours de voyage à partir de données produites par les utilisateurs en ligne, comme par exemple des téléchargements d'images en utilisant des sites web sociaux comme Flickr. Les auteurs formulent le problème de recommandation d'itinéraires. Le temps de visite est contraint par un budget temps, c'est-à-dire, le visiteur spécifie le temps dont il dispose pour effectuer un parcours. Cependant, dans ce travail, la valeur (score) de chaque POI est déterminée seulement par le nombre de fois où il a été mentionné par d'autres visiteurs dans le réseau social. Dans notre travail, le score d'un POI est déterminé non seulement en utilisant la popularité des POIs mais aussi en tenant compte des préférences de l'utilisateur et des évaluations qu'il a déjà faites pour d'autres POIs.

Enfin, le travail le plus proche du nôtre est [Xie et al., 2010]. Les auteurs explorent des solutions approximatives pour le problème de la recommandation composite. Le point central de leur travail est l'utilisation d'un algorithme du style Fagin pour la construction de packages de taille variable. Ils prouvent aussi l'optimalité de l'algorithme. les mêmes auteurs développent une extension de leur travail pour un prototype de système de recommandation pour le tourisme (CompRec) [Xie et al., 2011]. Cependant, on peut noter que pour eux le score d'un POI dépend seulement d'une estimation *a priori* de l'appréciation de l'utilisateur. Nous croyons

qu'il est nécessaire aussi d'utiliser la popularité des POIs afin d'améliorer de manière conséquente la qualité des packages recommandés.

Notons enfin qu'aucun des travaux que nous avons cités ne prend en compte la diversité des recommandations dans les packages, ce qui amène généralement à une meilleure satisfaction de l'utilisateur.

5.3 Diversité

La diversification est un problème qui a été largement étudié dans le domaine de la recherche d'information. La diversification des résultats associés à une requête de recherche aide à couvrir différentes interprétations de la requête et ainsi des résultats de cette requête [Clarke et al., 2008]. A titre d'exemple, un utilisateur qui soumet la requête "Jaguar" à un moteur de recherche peut rechercher des informations liées à l'animal ou bien à la marque de voiture. L'intégration de documents traitant des deux aspects à la liste de résultats mène alors à une augmentation des chances de satisfaire les besoins en information de l'utilisateur. Cependant, cela peut aussi introduire du bruit dans les résultats de la recherche.

La recherche dans le domaine des systèmes de recommandation a aussi prouvé que les utilisateurs ont tendance à préférer des recommandations diversifiées [Ziegler et al., 2005, Bradley and Smyth, 2001]. La diversité des résultats de la recommandation permet de couvrir au mieux les intérêts de l'utilisateur et peut lui permettre de découvrir de nouveaux items. Plusieurs travaux prennent en compte cette dimension, qui est devenue de plus en plus importante dans le domaine des systèmes de recommandation.

Ainsi, dans [Ziegler et al., 2005], la diversification est basée sur une mesure de similarité intra-liste pour les items. Cette mesure est obtenue en réalisant un mapping des items avec une taxonomie afin de déterminer les thèmes des items. Par exemple, dans un système de recommandation de films, on peut utiliser des attributs tels que le genre, le producteur, l'année, etc. Cette méthode se base sur un algorithme de post-traitement exhaustif, qui opère sur une liste Top-N déjà établie, afin de trouver la liste Top-K (N>K) des recommandations qui satisfont au mieux la diversité des résultats de recommandation, en se basant sur la similarité intra-liste ainsi définie.

Dans les travaux de [Di Noia et al., 2014], une liste de recommandations triée par ordre de pertinence (le score des items) est établie, en estimant la note que l'utilisateur donnerait à des items qu'il n'a pas encore notés. Dans une deuxième étape, la liste des items déjà recommandés est réorganisée en utilisant cette fois-ci une fonction de score intégrant à la fois les deux critères de, pertinence (préférences

de l'utilisateur) et de diversité.

[Vargas and Castells, 2013] modélisent chaque utilisateur du système en utilisant un ensemble de sous-profils, chaque sous-profil représentant une partition des intérêts ou préférences de l'utilisateur. Ensuite, ils utilisent une technique de filtrage collaboratif avec chaque sous-profil afin de générer une liste de recommandations. Les listes de recommandations provenant des différents sous-profils sont ensuite fusionnées pour obtenir une liste finale de recommandations. Les items sont alors classés en fonction de leur score, qui est une combinaison entre la pertinence des recommandations et la diversité des sous-profils représentés dans la liste finale.

5.4 Architecture et formalisation du problème

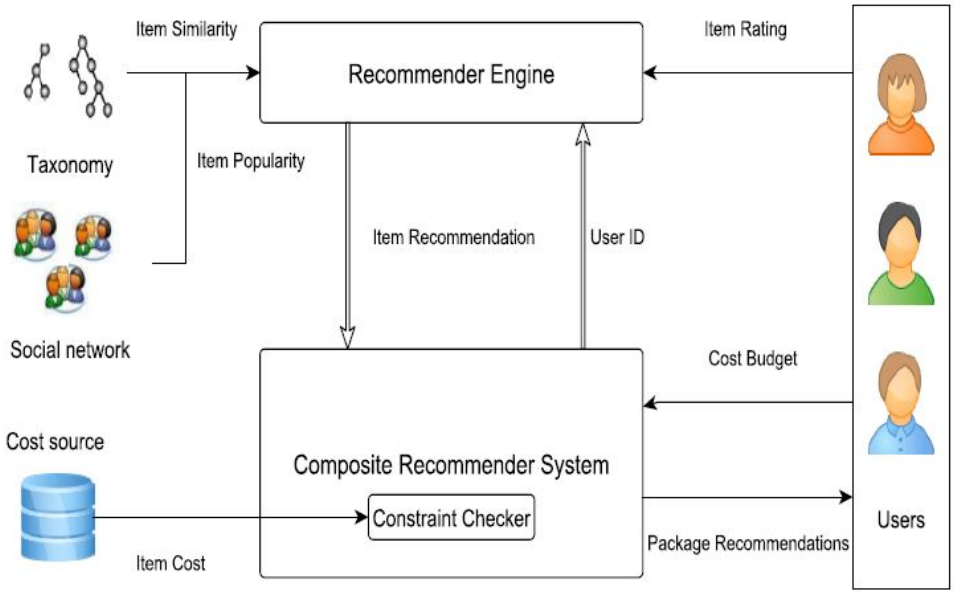


Figure 5.1 – Architecture de notre système de recommandation composite

Comme le montre la figure 5.1, notre système comprend deux composantes principales, le moteur de recommandation et le système de recommandation composite. Le moteur de recommandation capture les notes de l'utilisateur pour les différents POIs. Ces notes sont alors utilisées par le système pour calculer l'appréciation estimée des POIs que l'utilisateur n'a pas encore notés en tenant compte de ses préférences. L'appréciation estimée d'un POI est calculée en utilisant la similarité entre POIs qui est définie dans la suite et qui se base sur une taxonomie modélisant le domaine du tourisme 5.2.

La popularité d'un POI peut être estimée en utilisant le nombre d'avis donnés

par les utilisateurs, le nombre de visites par période, ou bien le nombre de "like" en utilisant les réseaux sociaux, etc. Le système de recommandation composite reçoit alors l'appréciation des POIs et leur popularité. Il recommande à l'utilisateur les meilleurs packages en fonction de ses préférences, de la popularité et de la diversité des POIs constituant chaque package. Chaque package peut constituer un parcours de visite pour l'utilisateur qui doit être compatible avec les budgets de l'utilisateur (coût et temps). Le système de recommandation composite comprend aussi un vérificateur de contrainte (*constraint checker*). Son rôle est de vérifier si un package satisfait les contraintes ou pas.

L'utilisateur spécifie non seulement son budget et la durée de sa visite mais aussi un entier k représentant le nombre de packages recommandés qu'il souhaite avoir. Notre système détermine alors les k meilleurs packages de POIs et les recommande à l'utilisateur.

5.4.1 Formalisation du problème

Étant donné un ensemble de POIs I, un ensemble U d'utilisateurs, un utilisateur actif $u \in U$ et un POI $i \in I$, nous notons $c(i)$ le coût engendré par la visite du POI i et $t(i)$ le temps moyen nécessaire pour sa visite.

Soit $P \subset I$ un ensemble de POIs, nous définissons :

- $score(P)$ le score d'un package P estimant sa qualité.
- $c(P) = \sum_{i \in P} c(i)$ est le coût engendré par la visite de tous les POIs du package P.
- $t(P) = \sum_{i \in P} t(i)$ le temps total nécessaire pour la visite de tous les POIs du package P.

Étant donné un budget sur le coût B_c et un budget sur le temps B_t, tous les deux fixés par l'utilisateur, un package P est dit *valide* si et seulement si :

$c(P) \leq B_c$ et $t(P) \leq B_t$.

Problème 1. ***Top-k composite recommendations***

Étant donné un ensemble de points d'intérêt (POIs) I, un utilisateur actif u avec son historique de notes, un budget sur le coût B_c, un budget sur le temps B_t et un entier k, un système de recommandation composite Top-k doit déterminer les Top-k packages $P_1, P_2, ..., P_k$ de manière à ce que chaque package P_i ait $c(P_i) \leq B_c$, $t(P_i) \leq B_t$, et que parmi tous les packages valides, $P_1, P_2, ..., P_k$ aient les k meilleurs scores, i.e $Score(P) \leq Score(P_i)$ pour tous les packages valides $P \notin \{P_1, P_2, ..., P_k\}$

Le problème de la recommandation composite est un problème NP-difficile [Amer-Yahia et al., 2013]. Plus précisément, il a été montré qu'il peut être réduit au problème *Sous-graphe Edge Maximum*, très connu dans le domaine de l'optimisation et la théorie des graphes, qui est un problème NP-complet. Il est donc nécessaire de concevoir des algorithmes efficaces pour la création des Top-k packages.

5.5 Modèle

5.5.1 Distance thématique et similarité entre POIs

Nous avons basé notre distance entre les POIs sur une taxonomie de catégories thématiques (par exemple, "musée d'art", "quartier gothique", etc.), organisées dans une hiérarchie sous forme d'une structure d'arbre.

Plus précisément, nous avons utilisé une taxonomie de domaine *SAMAP*, développée par [Castillo et al., 2008] afin de représenter l'ensemble de ces catégories thématiques. La figure 5.2 illustre une partie de cette ontologie, qui s'apparente davantage à une taxonomie.

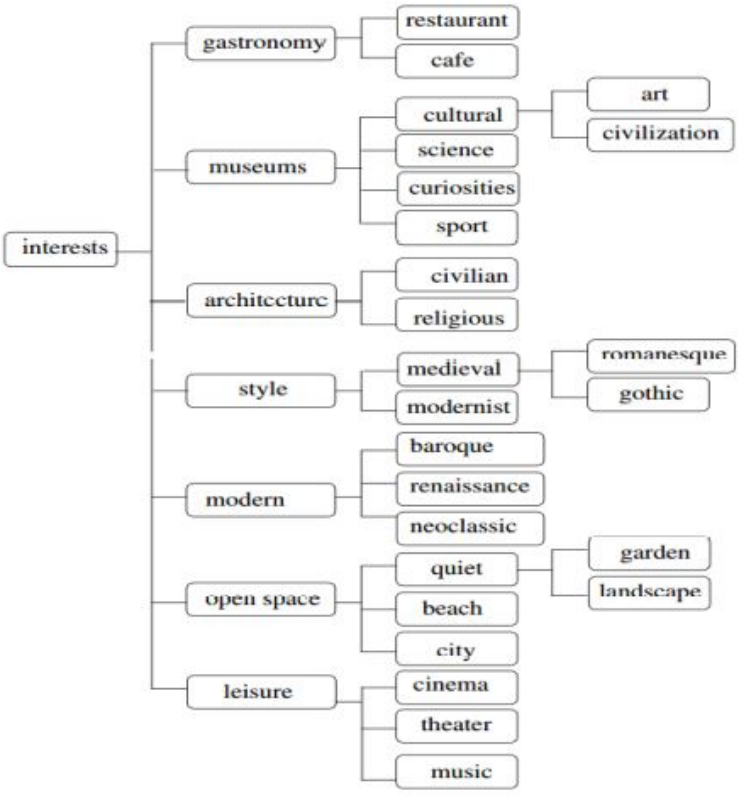


Figure 5.2 – Une portion de la taxonomie représentant les thématiques des POIs

Soit I l'ensemble de tous les POIs disponibles pour de potentielles recommanda-

tions. Chaque POI $i \in I$ est associé à une ou plusieurs catégories dans la taxonomie, une catégorie peut être par exemple : musée, parc, restaurant, etc. Nous définissons la distance thématique $dist_t : I \times I \rightarrow \mathbb{N}$ entre deux POIs i et j comme la longueur du plus court chemin dans la hiérarchie des catégories reliant les deux plus proches catégories de i et j. Formellement, en notant C_i et C_j l'ensemble des catégories thématiques des POIs i et j respectivement, la distance entre deux POIs est donnée par la formule suivante :

$$dist_t(i,j) = min_{c_i \in C_i, c_j \in C_j} ppc(c_i, c_j) \tag{5.1}$$

où *ppc* est la fonction calculant le plus court chemin entre deux catégories de la hiérarchie.

La similarité entre deux POIs i et j peut alors se calculer en fonction de leur distance thématique. Cette similarité évalue dans quelle mesure deux POIs i et j traitent d'une même thématique. Plus la distance thématique définie précédemment entre deux POIs est grande plus la similarité entre ces deux POIs est petite et inversement. Formellement, la similarité entre deux POIs i et j est calculée ainsi :

$$sim(i,j) = \frac{1}{1 + dist_t(i,j)} \tag{5.2}$$

5.5.2 Critères de qualité des packages

Afin de déterminer les k meilleurs packages pour un utilisateur et de construire un algorithme qui les calcule, il est indispensable de définir les critères de qualité d'un package. Le but étant d'estimer dans quelle mesure un package P est intéressant pour un utilisateur u. Nous dénotons le score d'un package P par $Score_u(P)$.

Le premier critère de qualité est l'appréciation estimée, nous avons besoin de savoir si un nouveau POI non encore visité pas l'utilisateur serait intéressant pour lui, en comparant la similarité entre les POIs que l'utilisateur a déjà aimés au POI candidat.

Le deuxième critère de qualité est la popularité des POIs le constituant. En effet la popularité est un facteur important de l'appréciation générale d'un utilisateur (par exemple un utilisateur visitant Paris ne pourrait pas rater *la tour Eiffel*).

De plus, nous supposons que l'utilisateur n'est pas forcément intéressé que par la visite de tous les POIs qui ont une forte similarité avec ses préférences, mais aussi qu'il souhaiterait visiter la liste des points d'intérêt qui couvrent au mieux toutes ses préférences et ses intérêts. Par exemple, supposons qu'un visiteur soit intéressé par les musées de manière générale, il ne serait surement pas intéressé par un package de POIs ne contenant que des musées. La diversité des POIs au sein d'un package

est donc un critère à prendre en considération pour une meilleure satisfaction de l'utilisateur.

Nous détaillons dans ce qui suit, les méthodes que nous utilisons pour prendre en compte ces critères de qualité.

5.5.2.1 Popularité globale

La popularité globale (overall popularity) mesure la popularité d'un point d'intérêt i. La popularité globale d'un POI $i \in I$ est définie ainsi :

$$opop(i) = \frac{pop(i)}{max_{j \in I} pop(j)} \in [0, 1] \tag{5.3}$$

où les j désignent les POIs de I et $pop : I \rightarrow \mathbb{N}$ représente un indicateur de popularité pour un POI, par exemple le nombre de "*like*" dans les réseaux sociaux ou le nombre de visiteurs par période. Dans notre système, nous avons utilisé le nombre de notes reçues par les utilisateurs du système.

Par extension, la popularité d'un package P est définie comme étant la moyenne de la popularité globale de tous les POIs constituant le package P. Formellement la popularité d'un package P est calculée ainsi :

$$opop(P) = \frac{\sum_{i \in P} opop(i)}{|P|} \in [0, 1] \tag{5.4}$$

5.5.2.2 Appréciation estimée

L'appréciation estimée (ou la prédiction) correspond à l'appréciation qu'un utilisateur u pourrait donner sur un point d'intérêt i qu'il n'a pas encore visité. Cette estimation est basée sur les préférences de l'utilisateur et les notes qu'il a exprimées sur d'autres POIs. Plus précisément, l'appréciation estimée pour un POI i de la part d'un utilisateur u est calculée en utilisant les notes que cet utilisateur a données à un échantillon de POIs similaires au POI i candidat à la recommandation. Ces notes sont pondérées par la similarité entre le POI à estimer et les autres POIs de l'échantillon. Formellement, l'appréciation estimée d'un utilisateur $u \in U$ pour un POI candidat $i \in I$ est définie par la formule suivante :

$$eapp_u(i) = \frac{\sum_{j \in S_i} rating_u(j) \times sim(i, j)}{\sum_{j \in S_i} sim(i, j)} \tag{5.5}$$

où S_i est l'ensemble des POIs similaires à i notés par l'utilisateur u, et $rating_u : j \rightarrow [0, 1]$ une fonction qui associe à chaque POI j la note donnée par l'utilisateur u divisée par la note maximale possible (généralement la note est comprise entre 1 et 5).

Notons que notre algorithme pour la recommandation des Top-k packages qui est décrit un peu plus tard en section 5.6.1, ne dépend pas d'une méthode de recommandation spécifique. Nous avons utilisé un simple filtrage collaboratif item-item pour prédire les valeurs des appréciations estimées (prédictions). Cette méthode est choisie pour sa simplicité et son faible coût de calcul. A noter que les prédictions de chaque POI pour un utilisateur donné peuvent être calculées au préalable.

Par extension, l'appréciation estimée (prédiction) de l'utilisateur u pour un package P est définie comme étant la moyenne des appréciations estimées de chaque POI j inclus dans le package. Formellement, l'appréciation estimée d'un utilisateur $u \in U$ pour un package $P \subset I$ est définie par la formule suivante :

$$eapp_u(P) = \frac{\sum_{i \in P} eapp_u(i)}{|P|} \in [0, 1] \tag{5.6}$$

Où les i désigne les POIs constituant le package P.

5.5.2.3 Diversité

La majorité des travaux sur les systèmes de recommandation dédiés au domaine du tourisme se focalisent sur la modélisation des préférences de l'utilisateur ainsi que sur la représentation des points d'intérêt à recommander, dans le but est de pouvoir recommander les POIs les plus intéressants pour l'utilisateur en les classant par ordre de pertinence en utilisant les prédictions obtenues sur les POIs. La diversité des recommandations dans le domaine des systèmes de recommandation pour le tourisme a rarement été prise en compte. Néanmoins, il a été montré que la diversité des recommandations a un effet positif sur la satisfaction de l'utilisateur [Ziegler et al., 2005].

Dans le but de prendre en compte la diversité des points d'intérêt dans chaque package, nous utilisons la diversité intra-liste introduite par [Ziegler et al., 2005] que nous adaptons à notre cas pour capter la diversité d'un package. Formellement, pour un package P, nous définissons la diversité intra-package (intra package diversity) $idp(P)$ avec la formule suivante :

$$ipd(P) = \frac{\sum_{i,j \in P} 1 - sim(i, j)}{(|P| - 1) \times |P|} \tag{5.7}$$

5.5.3 Score d'un package

Après avoir défini les critères de qualité d'un package ainsi que la façon de les estimer, nous définissons le score d'un package en fonction de la qualité des POIs qui le composent. Pour un utilisateur u, le score d'un package P est défini comme

suit :

$$Score_u(P) = C_{eapp} \times eapp_u(P) + C_{opop} \times opop(P) + C_{div} \times ipd(P) \tag{5.8}$$

où $C_{eapp}, C_{opop}, C_{div}$ sont des coefficients positifs modulant respectivement l'importance de l'appréciation estimée, la popularité et la diversité dans la fonction de score, avec $C_{eapp} + C_{opop} + C_{div} = 1$. Nous avons affecté à ces coefficients des valeurs différentes lors de nos expérimentations (présentées dans le chapitre 6) pour évaluer l'importance de ces trois critères vis-à-vis de la qualité des recommandations produites.

5.6 Recommandation des Top-k packages

Maintenant que nous avons défini une fonction pour calculer le score d'un package P pour un utilisateur cible u, il nous faut développer un algorithme qui recommande les Top-k packages pour cet utilisateur tels que les k packages aient les scores les plus élevés possible et que chacun des packages recommandés satisfasse les contraintes spécifiées par l'utilisateur telles que le coût de chaque package ainsi que le temps de visite de chaque package.

Comme nous l'avons mentionné auparavant, le problème de la recommandation composite sous contraintes que nous traitons est un problème NP-difficile. Il peut être résolu en générant un nombre assez grand de packages candidats avec des scores relativement grands et en sélectionnant ultérieurement le meilleur sous-ensemble possible de tous les packages construits. Ce type d'approche est appelée *Produire puis choisir* (*Produce-and-choose approach*).

Nous avons choisi cette approche comme paradigme fondamental pour résoudre notre problème. De ce fait, la construction des Top-k packages qui seront recommandés à l'utilisateur est faite en deux étapes :

- Premièrement, un ensemble de packages valides (qui respectent toutes les contraintes spécifiées par l'utilisateur) sont produits avec une cardinalité c tel que $c >> k$ (c étant le nombre de packages à créer et k le nombre de package à recommander). Les packages sont formés par agrégation autour d'un point d'intérêt pivot, à partir duquel on construit un package en prenant en compte les critères de qualité défini précédemment (popularité globale, appréciation estimée et diversité).

- Deuxièmement, quand un nombre suffisant de packages a été construit, les

packages sont triés en fonction de leur score respectif et il suffit alors de sélectionner les k meilleurs packages et de les recommander à l'utilisateur actif.

5.6.1 Création de packages candidats

Notre approche pour la formation d'un ensemble de packages valides et avec un bon score est inspirée de l'algorithme BOBO (Bundles One By One) introduit par [Amer-Yahia et al., 2014]. Nous avons étudié et adapté cet algorithme pour prendre en compte les critères de qualité des packages, à savoir l'appréciation estimée, la popularité globale ainsi que la diversité qui sont définis en section 5.5.2. Le but de l'algorithme est de créer un nombre c de packages valides respectant les contraintes spécifiées par l'utilisateur avec de bons scores. Inspirée du $k - nn$ clustering, l'algorithme détermine à chaque étape un POI qui sert de pivot, et construit un package valide avec un score maximal autour de ce pivot. Le pseudo code est décrit dans l'algorithme *1*.

Algorithm 1: BOBO

Input: I, B_c, B_t, number of packages c
Output: a set c of packages

1 $Packages \leftarrow \emptyset$
2 $Pivots \leftarrow Descending\_sort(I, opop)$
3 **while** $(Pivots \neq \emptyset)$ **and** $|Packages| < c$ **do**
4 $\quad w \leftarrow Pivots[0]$
5 $\quad Pivots \leftarrow Pivots - \{w\}$
6 $\quad P \leftarrow$Pick_bundle(w, I, B_c, B_t)
7 $\quad Pivots \leftarrow Pivots - P$
8 $\quad Packages \leftarrow Packages \cup P$
9 **end**
10 **return** *Packages*

Notre algorithme BOBO commence avec un ensemble vide de packages (ligne 1). Après cela, une liste de pivots candidats est construite (ligne 2), en triant tous les POIs candidats (I) par ordre décroissant de leur popularité globale ($opop$).

Ensuite, tant que le nombre de packages formés est inférieur au nombre de packages requis c, à chaque itération, le premier POI est choisi parmi l'ensemble des POIs *Pivots* (ligne 4) et un package est construit autour de ce pivot (ligne 5). Cette étape est réalisée par la fonction *Pick_Bundle* qui est décrite dans l'algorithme*2*.

L'algorithme de cette fonction est de type Greedy, il choisit à chaque itération le POI qui maximise le score du package en cours de formation autour du pivot (ligne 7), ceci tant que les contraintes de budget (coût et temps) spécifiées par l'utilisateur sont satisfaites (ligne 8). Si le POI sélectionné, maximisant le score du package,

respecte les contraintes imposées alors il est ajouté au package (ligne 9). Son coût est ajouté au coût du package (ligne 10) et son temps de visite est ajouté au temps de visite du package (ligne 11). Ce POI est alors supprimé de la liste *active* (la liste courante des POIs candidats à la recommandation) pour ne plus être recommandé dans un autre package (ligne 16). Rappelons en effet qu'un POI ne peut pas faire partie de deux packages différents, et que, sans perte de généralité, nous supposons que tous les POIs ont un coût inférieur au budget sur le coût B_c et ont un temps de visite inférieur au budget sur le temps de visite B_t, tous deux spécifiées par l'utilisateur.

Algorithm 2: PICK_BUNDLE

Input: pivot w, I, B_c, B_t
Output: a package S

1 $S \leftarrow w$
2 $active \leftarrow I - \{w\}$
3 $cost \leftarrow c(w)$
4 $time \leftarrow t(w)$
5 $finish \leftarrow false$
6 **while** *(not* $finish$*)* **do**
7 $\quad i \leftarrow argmax_{j \in active} Score_u(S \cup \{i\})$
8 $\quad$ **if** $(cost + c(i) \leq B_c)$ *and* $(time + t(i) \leq B_t)$ **then**
9 $\quad\quad S \leftarrow S \cup \{i\}$
10 $\quad\quad cost \leftarrow cost + c(i)$
11 $\quad\quad time \leftarrow time + t(i)$
12 $\quad$ **end**
13 $\quad$ **else**
14 $\quad\quad finish \leftarrow true$
15 $\quad$ **end**
16 $\quad active \leftarrow active - i$
17 **end**
18 **return** S

Revenons maintenant à la boucle principale de notre algorithme BOBO, une fois qu'un package candidat est créé, il est ajouté à la liste des packages construits *Packages* (ligne 8). Tous les POIs le constituant sont supprimés de la liste *Pivots* (ligne 7) pour qu'ils n'apparaissent plus dans un autre package.

5.6.2 Sélection des Top-k packages

Une fois que le nombre de packages nécessaire est construit, ceux-ci sont triés en fonction de leur score respectifs. Les k packages ayant eu les meilleurs scores sont sélectionnés et la liste de ces packages est recommandée à l'utilisateur, chaque

package pouvant constituer un parcours de visite différent.

5.7 Conclusion

Nous nous sommes intéressé dans ce chapitre à la personnalisation des visites touristiques. Nous avons étudié le problème de la recommandation composite, un type de recommandation qui consiste à dépasser les limites des suggestions à base de listes triées, et de les remplacer par des suggestions de packages, chaque package étant constitué de plusieurs items. Cette approche se révèle être très satisfaisante pour des domaines comme le tourisme où les items sont hétérogènes. Nous nous sommes alors intéressé à déterminer les k meilleurs packages pour un utilisateur donné, chaque package étant constitué d'un ensemble de points d'intérêt.

Notre algorithme de recommandation composite consiste à trier la liste des packages en fonction de leur score, le score d'un package étant déterminé en fonction de l'appréciation estimée, de la popularité ainsi que de la diversité des POIs le constituant. Nous avons formalisé le problème de la génération des Top-k packages avec des contraintes de budgets (coût et temps) sur chaque package, où la visite d'un POI engendre un coût et consomme un temps de visite. Nous avons ensuite proposé un algorithme pour la détermination des Top-k packages avec les meilleurs scores, notre algorithme est basé sur des principes de la recherche d'information composite.

L'évaluation de notre système en utilisant un data-set réel crawlé du site web Tripadvisor démontre sa qualité et son capacité à améliorer à la fois la précision des recommandations ainsi que leur diversité. L'implémentation de notre système ainsi que les évaluations que nous avons réalisées sont présentées dans le chapitre suivant.

Implémentation et évaluation

Sommaire

6.1 Introduction

Nous décrivons dans ce chapitre, dans un premier temps les travaux applicatifs ainsi que les détails d'implémentation réalisés pendant cette thèse, à savoir la conception et le développement de nos deux plateformes de recommandation : Cime-Musée et Cime-Tourisme. La première est une application mobile destinée à être utilisée pour la visite de musée et la deuxième consiste aussi en une application mobile destinée cette fois au tourisme. Chacune des deux applications est constituée de deux parties principales, une application client fonctionnant sous Android et destinée aux utilisateurs finaux, et une partie serveur chargée notamment des calculs des recommandations et des parcours.

Nous décrirons aussi dans ce chapitre les évaluations que nous avons réalisées de CIME-tourisme, notre système de recommandation composite destiné au tourisme grâce à un jeu de données réel extrait de Tripadvisor.

6.2 Implémentation de CIME-Musée

6.2.1 Les cas d'utilisation

Le diagramme des cas d'utilisation représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. Les principales fonctionnalités de notre

système sont :

- Créer un compte utilisateur,
- S'authentifier,
- Consulter les œuvres disponibles,
- Noter les œuvres,
- Démarrer une nouvelle visite,
- Consulter le parcours recommandé,
- Ajouter/supprimer une œuvre au parcours.

La figure 6.1 représente le diagramme des cas d'utilisation.

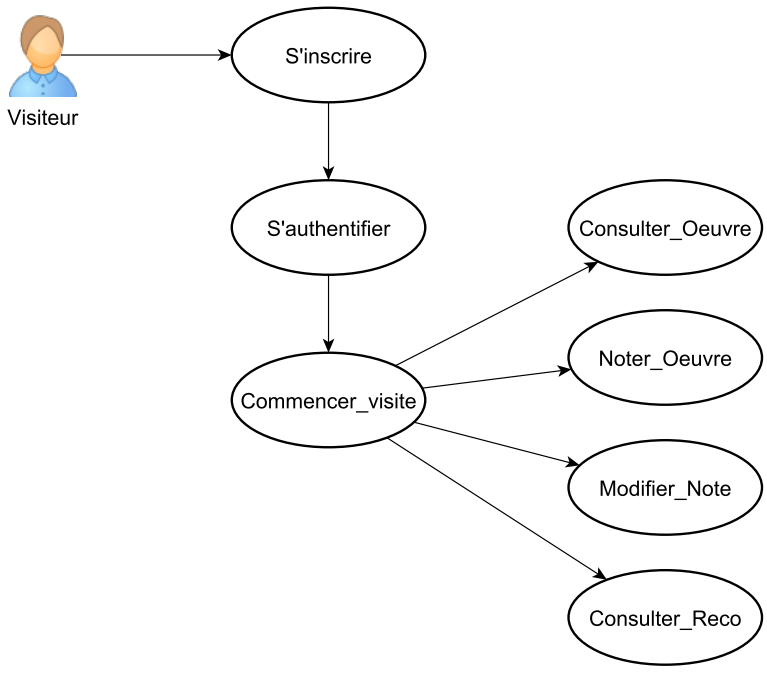


Figure 6.1 – Digramme des cas d'utilisation

6.2.2 Architecture du prototype

Le prototype mis en place (figure 6.2), est constitué de plusieurs éléments : une base de connaissance RDF (avec un module de requêtes SPARQL), une base de données SQL (avec un module de requêtes SQL), un module de recommandation, un web service hébergé sur un serveur et une application mobile comme client.

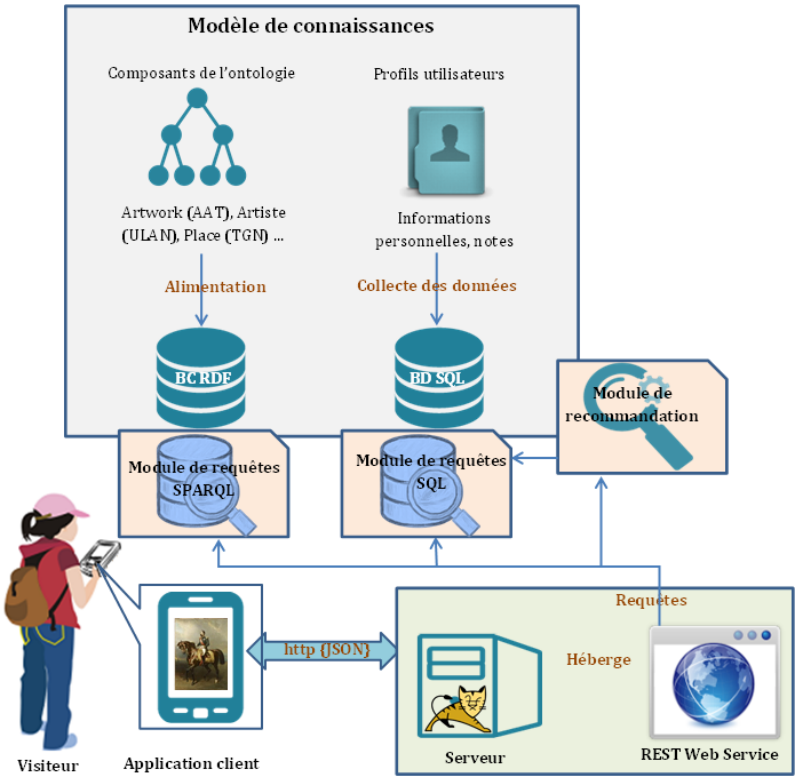


Figure 6.2 – Architecture générale du prototype CIME-Musée

Le rôle de la base de connaissance est de permettre le stockage de notre modèle de connaissances développé dans le chapitre 4, ainsi que son peuplement avec les informations correspondant aux différentes œuvres. Elle contient donc les œuvres, les artistes, les styles, les thèmes, etc, ainsi que leurs caractéristiques et leurs relations. Afin d'interroger la base de connaissance et d'extraire les connaissances sur les œuvres sous forme d'objets manipulables par notre application, un module de requêtes SPARQL est implémenté.

Pour permettre le stockage des profils utilisateurs, nous avons développé et maintenu une base de données SQL. Cette base contient ainsi les informations personnelles sur les utilisateurs, leurs visites, leurs préférences (les notes qu'ils ont exprimées pour les différentes œuvres), etc. Le module de requêtes SQL a pour but d'interroger la base de données afin d'insérer et/ou de récupérer des données exploitables par notre système, particulièrement par le module de recommandation. De telles données peuvent par exemple être les notes données par un utilisateur.

Le moteur de recommandation est la partie principale de notre architecture, c'est dans cette partie qu'est implémenté notre système de recommandation. Ce module utilise les modules des requêtes SPARQL ainsi que SQL pour consulter les données sur les utilisateurs et sur les œuvres pour pouvoir produire des recommandations

personnalisées à un visiteur cible. Les calculs des recommandations se font sur le serveur, l'application mobile est juste utilisée comme une interface pour présenter les œuvres recommandées à l'utilisateur et lui permettre d'interagir avec le système. La communication entre la partie serveur et la partie cliente se fait à l'aide du web service REST.

6.2.3 Implémentation

6.2.3.1 Base de données

Le modèle relationnel, représenté par la figure 6.3, contient la liste des tables qui constituent la base de données de notre système de recommandation. Les tables utilisées sont les suivantes :

- **User** : cette table représente un visiteur du musée (utilisateur du système). Elle contient notamment les informations démographiques de l'utilisateur tels que : l'âge, le sexe, la nationalité, etc.
- **Visit** : cette table représente une visite d'un utilisateur, elle contient des informations sur une visite donnée : temps de début, durée de la visite, etc.
- **Artwork** : contient l'identifiant d'une œuvre "idArtwork".
- **Rating** : cette table représente la matrice des notes, chaque entrée de cette table nous donne la note qu'a donné un utilisateur donné à une œuvre donnée.
- **SimArtWork** : cette table stocke la matrice des similarités sémantiques entre chaque paire d'œuvres. Notons que les similarités sont calculées au préalable par le module de recommandation sémantique et sont stockées dans la table SimArtWork.

6.2.3.2 Base de connaissances et sources utilisées

Le modèle sémantique que nous proposons n'est qu'une structure abstraite pour représenter les œuvres, il est cependant nécessaire de disposer d'une base de connaissances muséales qui soit instanciée pour développer notre application.

La construction d'une telle base de connaissance est assez difficile, en raison l'absence de ressources informatisées et exploitables. Suite à des visites au musée du palais impérial de Compiègne, nous avons pris des photos des œuvres qui sont exposées, ainsi que leurs cartels qui contiennent des informations générales sur les œuvres. Cependant, toutes les propriétés de la base de connaissances ne sont pas

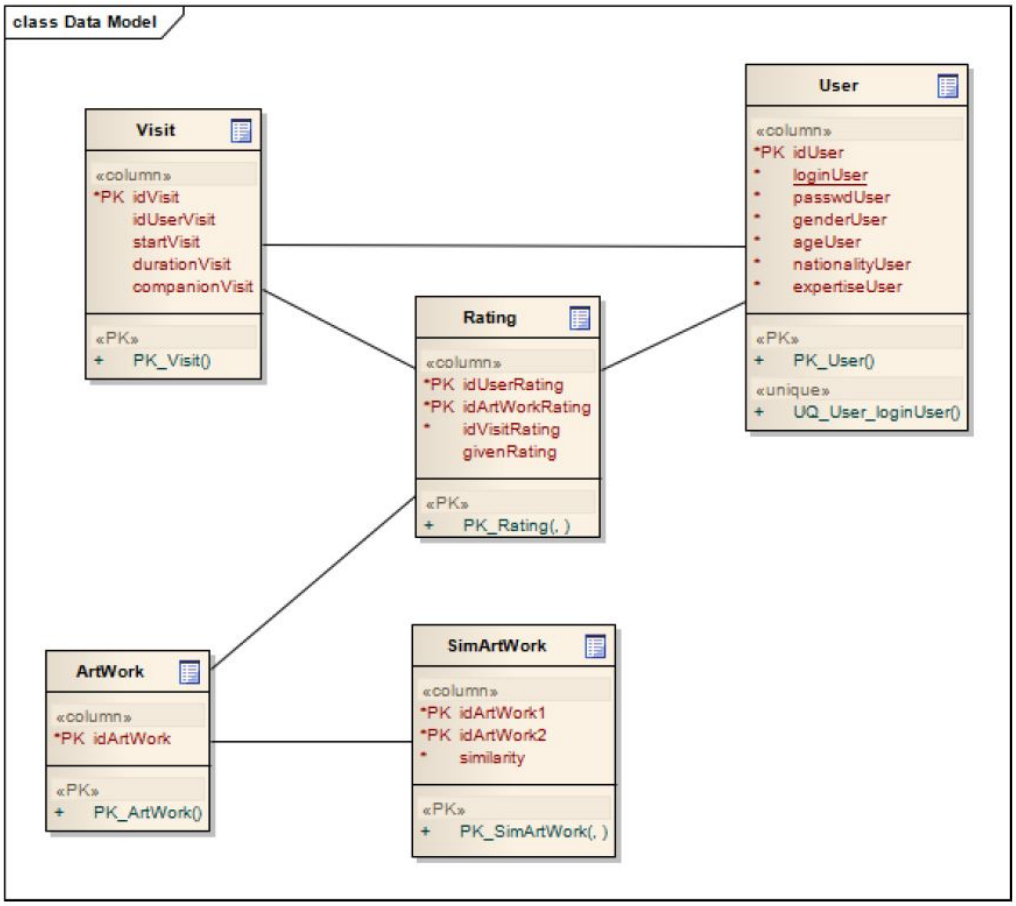


Figure 6.3 – Schéma de la base de données

disponibles. Nous avons donc utilisé les ressources du web de données pour les compléter, un ensemble de base de connaissances liées entre elles et en accès libre. Nous avons identifié les plus importantes, qui offrent des informations d'intérêt sur les œuvres, styles, artistes, thèmes, etc. : The Getty, DBPedia et Freebase.

- **The Getty**[1] **:** Le J. Paul Getty Trust est une institution culturelle et philanthropique dédiée à la représentation, la conservation et l'interprétation de l'héritage artistique du monde. Getty est conçu pour le grand public et aussi pour les communautés professionnelles et académiques, dans le but d'offrir des connaissances sur les arts visuels.

- **DBPedia**[2] **:** est la version sémantique de Wikipédia. Elle est au cœur de l'organisation du web de données. DBPedia est une base de connaissances de taille très importante, les entités qui la composent étant directement issues des pages Wikipédia. Cependant, il s'est avéré que DBPedia n'était pas une solution viable pour alimenter notre base de connaissances. En effet, la

[1]http ://www.getty.edu/
[2]http ://dbpedia.org/

structure hiérarchique des concepts de DBPedia est déduite de l'arborescence des pages Wikipédia et ces hiérarchies sont beaucoup moins riches que celles offertes par le Getty. De plus il existe souvent des incohérences dans les hiérarchies de concepts de Wikipédia.

- **Freebase**[3] **:** est une base de données sémantique dont l'originalité est d'être peuplée directement par les utilisateurs. Les liens sémantiques entre ressources sont donc spécifiés par les utilisateurs. Freebase offre ainsi des données plus complètes et mieux structurées que DBPedia. Par ailleurs, Freebase est particulièrement riche dans la description sémantique des œuvres, artistes, thèmes, etc. Nous avons donc choisi d'utiliser Freebase pour compléter les informations manquantes de notre base de connaissances.

Notre modèle sémantique (Cf. chapitre 4) a été implémenté à l'aide de l'éditeur Protégé. La figure 6.4 représente une copie d'écran des principales classes de ce modèle.

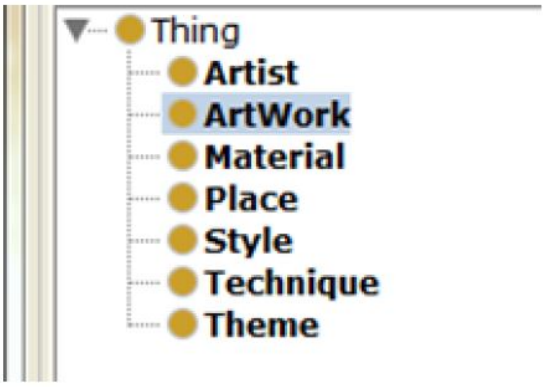


Figure 6.4 – Les principales classes de notre modèle sémantique

Les différents classes sont :

- ArtWork :
  Cette classe représente l'œuvre, et elle est décrite par les propriétés suivantes : *artWorkId, artWorkDesignation, artWorkSummary, artWorkPicture, artWorkCreationDate, createdBy, hasMaterial, hasStyle, hasTechnique, hasTheme, hasCreationPlace.* Ces propriétés sont respectivement l'identifiant de l'œuvre, son titre, sa description, un lien vers l'image stockée en dur, sa date de création, son auteur, son matériau de production, son style, sa technique, son thème et son lieu de création. La figure 6.5 représente les propriétés de l'œuvre *L'impératrice Eugénie et le prince impérial dans le parc de Camden place.*

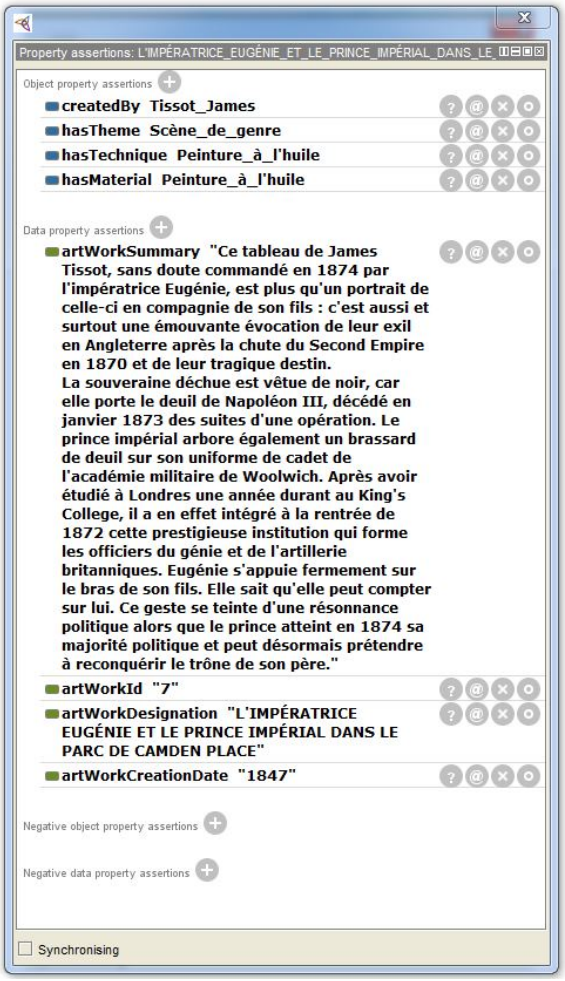


Figure 6.5 – Représentation et propriétés d'un exemple d'œuvre

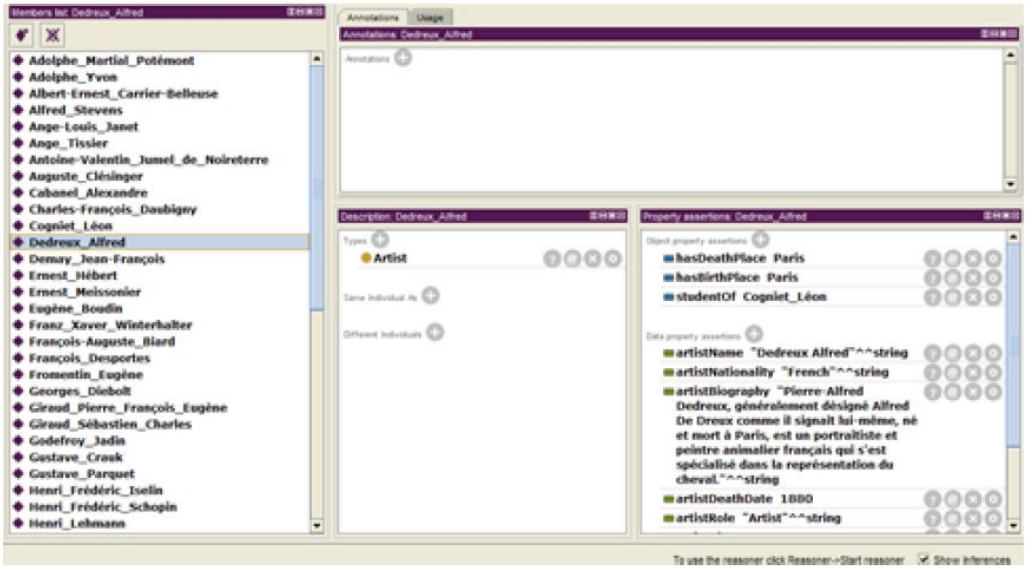


Figure 6.6 – Propriétés d'un artiste

- **Artist :**
  Cette classe décrit les artistes, à l'aide des propriétés suivantes : *artistId, artistName, artistBiography, artistNationality, artistRole, artistBirthDate, artistDeathDate, artistBirthPlace, artistDeathPlace, studentOf.* Ces propriétés sont respectivement l'identifiant de l'artiste, son nom, sa biographie, sa date de naissance, sa date de mort, son lieu de naissance, son lieu de mort et son maître. La figure 6.6 représente les propriétés de l'artiste *Alfred Dedreux.*

- **Material, Place, Style, Technique, Theme :**
  Ces classes sont toutes décrites par leur désignation ainsi que la propriété "broader/narrower", cette propriété montre la position de l'élément considéré dans la hiérarchie correspondante.

6.2.3.3 L'application CIME-Musée

CIME-Musée est composée d'une partie cliente qui est une application mobile et d'un serveur. Les principaux rôles de l'application mobile sont l'affichage des informations ainsi que l'interaction avec l'utilisateur. L'application mobile permet à un utilisateur de créer un compte de s'authentifier, de créer une visite et de visualiser et naviguer parmi les œuvres disponibles sur un dispositif mobile (smartphone ou tablette).

L'application mobile fait appel au serveur à chaque fois qu'elle a besoin d'une donnée ou d'une information. Le serveur effectue la tâche demandée et renvoie le résultat à l'application mobile. Par exemple, lorsqu'un utilisateur consulte et note un ensemble d'œuvres, et que ces notes sont insérées dans la base de données, le moteur de recommandation calcule la liste des recommandations et les envoie à l'application mobile. La communication entre le client et le serveur est réalisée en utilisant le protocole REST [Fielding, 2000].

6.2.3.3.1 Application mobile : L'utilisateur est tout d'abord invité à s'enregistrer et à introduire ses informations personnelles, ou à s'identifier. Une fois que l'utilisateur s'est connecté, l'application serveur utilise les informations personnelles de celui-ci pour produire une liste de recommandations en utilisant l'approche démographique. La figure 6.7 montre un exemple d'une liste de recommandations pour un utilisateur donné.

L'utilisateur peut alors consulter les œuvres recommandées pour avoir plus d'informations sur celles-ci. Il peut aussi exprimer ses préférences pour ces œuvres en donnant une note (figure 6.8).

[3]https ://www.freebase.com/

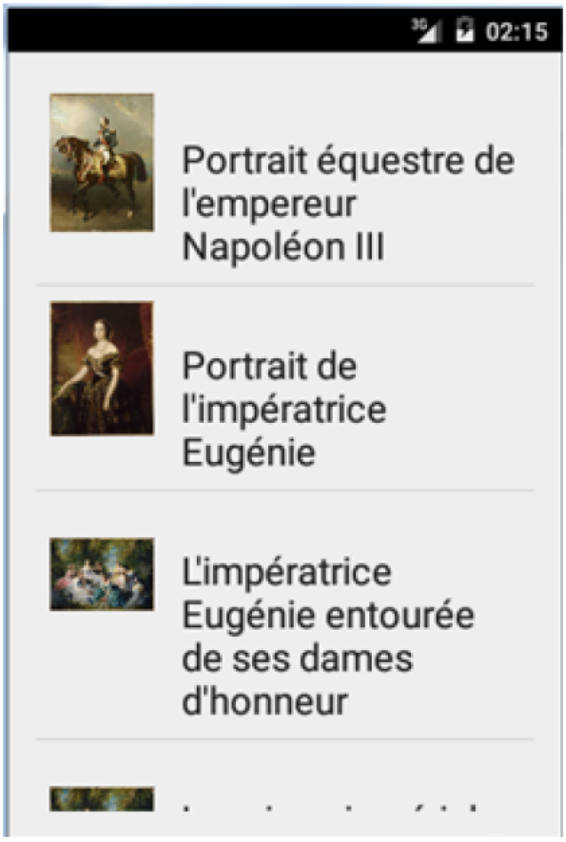


Figure 6.7 – Interface : liste de recommandations

6.2.3.4 Application serveur

L'application serveur réalise le calcul des similarités sémantiques en utilisant la base de connaissance, produit les recommandations et gère la base de données (ex. mise à jour de la matrice des notes).

La figure 6.9 présente les classes du web service. Les paquets sparql et sql exploitent respectivement la base de connaissances et la base de données. Le paquet "Recommender" avec ses différentes classes va permettre de faire le calcul de similarités ainsi que d'exécuter les différentes méthodes de recommandation que nous avons développées. Le paquet ressource est le paquet principal pour effectuer les requêtes au serveur. Il définit les chemins d'accès aux ressources pour pourvoir les appeler depuis l'application mobile.

La liste des œuvres recommandées, les données d'enregistrement des utilisateurs et les notes données aux différentes œuvres, sont communiquées via le protocole HTTP entre l'application mobile et le serveur grâce à une URL qui va retourner un résultat en format JSON.

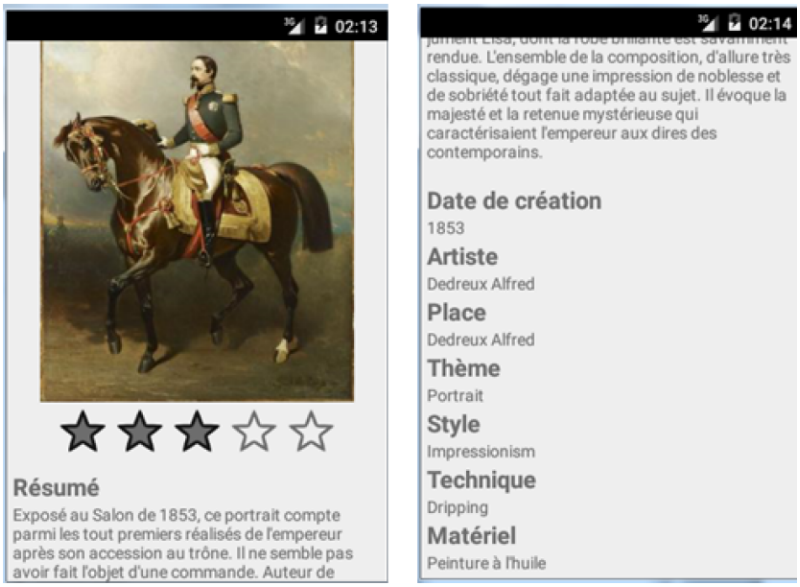


Figure 6.8 – Interface : Consulter et noter une œuvre

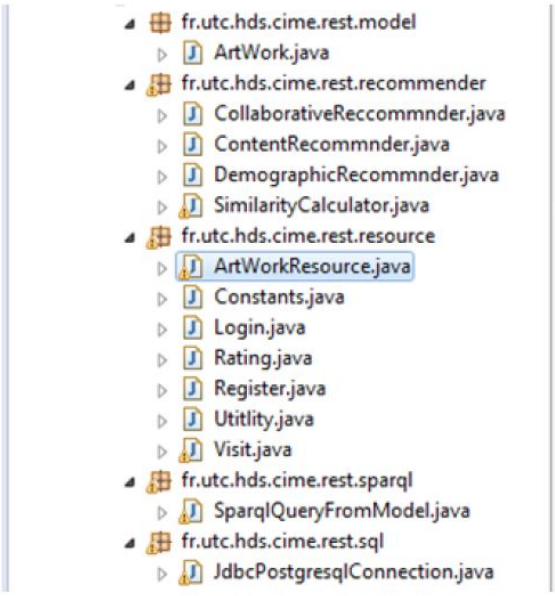


Figure 6.9 – Classes du web service

6.3 Implémentation de CIME-Tourisme

L'objectif est de créer une application mobile dédiée au domaine du tourisme en implémentant l'approche de recommandation composite que nous avons proposée au chapitre 5. Étant donné les préférences de l'utilisateur et son budget, l'application exécute notre algorithme de recommandation pour offrir à l'utilisateur une recommandation sous forme de parcours (chaque package peut être vu comme un parcours de visite). Quand l'utilisateur sélectionne ou choisit un parcours il est en mesure d'effectuer les actions suivantes :

- Observer la position des différents points d'intérêt d'un des packages recommandé,
- Observer le meilleur trajet à effectuer pour visiter ces points d'intérêt,
- Obtenir des informations sur le parcours, notamment la distance à parcourir et le coût total engendrés,
- Ajouter ou supprimer un point d'intérêt au parcours (l'utilisateur doit rester maître) et consulter les mises à jour sur le parcours,
- Recevoir un aperçu (image) d'un point d'intérêt sélectionné,
- Noter un point d'intérêt pour enrichir son profil ou bien consulter la note qu'il a déjà attribuée.

CIME-Tourisme se décompose en deux activités principales. La première s'exécute dès le lancement de l'application, c'est une activité de paramétrage qui consiste à inviter l'utilisateur à entrer diverses informations utiles au bon fonctionnement de l'application : son budget, son temps de visite et le nombre de packages qu'il souhaite recevoir.

La deuxième activité est lancée ensuite et permet principalement l'affichage :

- d'une carte au travers d'un fragment généré par l'API Google Maps,
- d'un menu latéral contenant les parcours générés par notre système de recommandation,
- d'un menu contextuel contenant des boutons déclenchant les fonctions relatives à l'application (sélection d'un POI, affichage d'images, informations sur le parcours, etc.)

Une fois que l'utilisateur s'est authentifié, qu'il a entré ses contraintes de budget et de temps de visite, ainsi que le nombre de packages (parcours) k qu'il souhaite considérer, le moteur de recommandation est appelé.

Les parcours recommandés sont alors présentés à l'utilisateur pour qu'il puisse les selectionner. Quand l'utilisateur sélectionne un parcours, les points d'intérêt le constituant sont affichés dans une liste et sur une carte.(figure 6.10). Chaque POI est signalé à l'aide d'un marqueur. Pour un parcours donné, l'utilisateur peut choisir les POIs qu'il souhaite garder ou supprimer. Il peut aussi en rajouter manuellement sans qu'ils aient été recommandés. L'utilisateur reste ainsi maître de ses choix. La distance totale à parcourir pour visiter tous les POIs du parcours est affichée, et pour chaque

parcours l'API Google maps affiche le trajet optimal. L'utilisateur peut sélectionner un POI pour exprimer son avis en lui attribuant une note ou bien changer sa note si il avait déjà noté ce POI. Il peut aussi afficher une image du lieu qu'il est entrain de consulter. Ces différentes fonctionnalités sont présentées dans la figure 6.11.

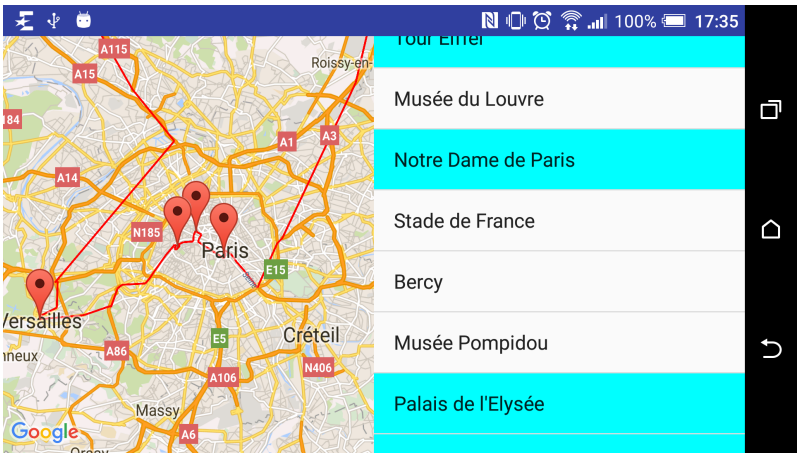


Figure 6.10 – Exemple d'un parcours recommandé

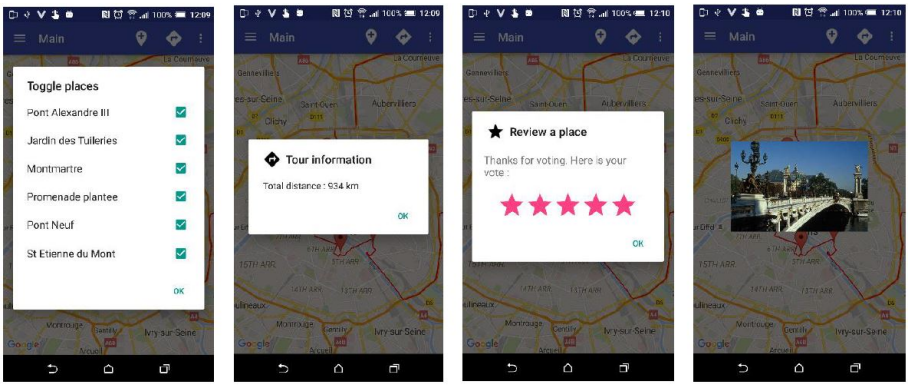


Figure 6.11 – Les différentes fonctionnalités de l'application

6.4 Évaluation

6.4.1 Jeu de données

Le but de nos expérimentations était d'évaluer la pertinence des packages recommandés par notre approche ainsi que leur diversité. Afin de disposer d'un ensemble de points d'intérêt constituant des recommandations potentielles, nous avons exploité le site web Tripadvisor bien connu dans le domaine du tourisme. Sur Tripadvisor, les utilisateurs peuvent partager et découvrir des informations sur les points d'intérêt disponibles dans la ville qu'il souhaitent visiter. Les utilisateurs notent aussi les

POIs qu'ils ont visités et peuvent laisser un commentaire. Tripadvisor fournit pour chaque ville, un ensemble de points d'intérêt appartenant chacun à une ou plusieurs catégories thématiques organisées sous forme de taxonomie. Cette structure nous a permis de construire notre mesure de similarité entre POIs. De plus, Tripadvisor fournit pour chaque POI un ensemble d'indicateurs sociaux, comme la note moyenne de tous les utilisateurs et le nombre d'utilisateurs qui l'ont visité. Ces informations constituent un indice de popularité important. En particulier, nous avons utilisé le nombre de visiteurs ayant noté un POI comme un indicateur de popularité.

Utilisateurs	Nombre de POIs notés	Note moyenne
Utilisateur 1	54	3
Utilisateur 2	52	4
Utilisateur 3	49	3.5
Utilisateur 4	49	4
Utilisateur 5	46	4.5
Utilisateur 6	44	2.5
Utilisateur 7	42	4
Utilisateur 8	41	4
Utilisateur 9	40	4.5
Utilisateur 10	39	3.5
Utilisateur 11	36	4
Utilisateur 12	35	4
Utilisateur 13	35	3.5
Utilisateur 14	34	4.5
Utilisateur 15	33	4
Utilisateur 16	32	4
Utilisateur 17	32	4
Utilisateur 18	32	3
Utilisateur 19	31	3.5
Utilisateur 20	31	4

Tableau 6.1 – Statistiques sur l'échantillon de test

Pour nos expérimentations, nous avons développé un crawler pour récupérer les notes d'utilisateurs sur les différents POIs disponibles dans la ville de Paris. Nous avons exclu les POIs qui n'avaient pas reçu de notes ou bien qui en avaient reçu trop peu. Le jeu de données contient 40635 notes pour 1183 POIs données par 18227 utilisateurs. Les données sont donc très parcimonieuses. Nous avons associé à chaque POI son coût et son temps moyen de visite que nous avons aussi récupérés de Tripadvisor. Le coût moyen de tous les POIs est proche de 7 €et le temps moyen pour visiter un POI varie entre 30 minutes et 3 heures. En raison de la large parcimonie de la matrice utilisateurs/POIs sous-jacente, nous avons sélectionné les 20 utilisateurs

les plus actifs (ayant donné le plus de notes) comme échantillon pour tester notre approche. Le tableau 6.1 récapitule des statistiques concernant les 20 utilisateurs de notre échantillon de test. Notamment le nombre de POIs notés ainsi que la note moyenne qu'ils ont donnés pour les différents POIs qu'ils ont notés.

6.4.2 Mesures d'évaluation

Lorsqu'un système de recommandation est développé, il est important d'être en mesure d'évaluer son fonctionnement et sa capacité à répondre aux objectifs qui lui ont été fixés.

Le but de nos expérimentations est de tester la pertinence ainsi que la diversité des points d'intérêt qui sont recommandés. Pour cela nous avons utilisé deux mesures. La première est la précision qui est la mesure la plus utilisée pour évaluer la pertinence d'une liste de recommandations. La précision est définie par le ratio des items qui sont recommandés et qui sont pertinents sur le nombre total des recommandations (formule 6.1). Dans nos paramètres d'évaluation, nous considérons qu'un point d'intérêt est pertinent si l'utilisateur lui a attribué une note de 4 ou 5.

$$precision = \frac{|relevant\ recommended\ POIs|}{|recommended\ POIs|} \tag{6.1}$$

Le seconde mesure a pour but d'évaluer la diversité des packages recommandés. Pour cela, nous étendons la similarité intraliste introduite par *Ziegler et al* [Ziegler et al., 2005] pour un ensemble de k packages $\{P_1, ..., P_k\}$. La diversité intraliste moyenne (Mean Intralist Diversity, MILD) est définie comme suit (formule 6.2) :

$$MILD(\{P_1, ..., P_k\}) = \frac{\sum_{i=1}^{k} ILD(P_i)}{k} \tag{6.2}$$

où ILD est la diversité intraliste pour un seul package, qui est définie ainsi :

$$ILD(P) = \frac{\sum_{i,j \in P} 1 - sim(i,j)}{|P|^2} \tag{6.3}$$

Nous utilisons une troisième mesure d'évaluation afin de comparer les différentes approches qui aboutissent à un meilleur compromis entre précision et diversité. Cette mesure est la moyenne harmonique entre la précision et la diversité F_{PD}, définie par :

$$F_{PD} = \frac{2 \times precision \times diversity}{precision + diversity} \tag{6.4}$$

6.4.3 Protocole expérimental

Notre but était de tester l'impact de la personnalisation, de la popularité ainsi que de la diversité vis-à-vis de la qualité des recommandations selon les critères de précision et de diversité ainsi que du compromis entre précision et diversité. Pour cela, nous avons comparé plusieurs versions de notre système correspondant aux différentes combinaisons possibles des facteurs de personnalisation "per" (influencé par C_{eapp}), de popularité "pop" (influencé par C_{opop}) et de diversité "div" influencé par C_{div}. Chaque version correspond donc à une combinaison différente des paramètres C_{eapp}, C_{opop} et C_{div} dans la fonction de score d'un package P : $Score_u(P) = C_{eapp} \times eapp_u(P) + C_{opop} \times opop(P) + C_{div} \times ipd(P)$.

Les différentes versions que nous avons évaluées sont regroupées dans la table 6.2. Le nom de chaque version indique l'utilisation ou non des différents aspects lors de la construction des Top-k packages. Par exemple, la version per + div considère seulement la personnalisation et la diversité dans la fonction de score d'un package, et ignore la popularité lors du processus de recommandation.

Notons que dans nos expérimentations, nous ne cherchons pas à obtenir la combinaison optimale des poids associés aux facteur de personnalisation, popularité et diversité. Notre but n'est pas de trouver la meilleure combinaison des poids pour chaque critère mais plutôt de tester l'impact des différentes critères vis-à-vis de la qualités des recommandations. Cela justifie les valeurs simples que nous avons attribuées aux différents poids C_{eapp}, C_{opop} et C_{div} pour chacune des versions. Nous voulions en effet tester l'impact de la présence ou non de chacun des critères sur les résultats des recommandations.

Versions de notre approche	C_{eapp}	C_{opop}	C_{div}
per	1	0	0
pop	0	1	0
div	0	0	1
per+pop	1/2	1/2	0
per+div	1/2	0	1/2
pop+div	0	1/2	1/2
per+pop+div	1/3	1/3	1/3

Tableau 6.2 – Différentes versions de notre système

Nous avons testé notre système en faisant varier le nombre k de packages recommandés, suivant quatre valeurs : $5, 10, 15$ et 20. Le budget sur le coût est

fixé à 60 €et le budget sur le temps de visite à 8 heures. Nous avons testé notre système avec d'autres budgets pour des résultats très similaires. En effet, nous avons constaté que la variation des budgets n'a pas d'impact important et n'affecte pas vraiment ni la précision ni la diversité des recommandations. C'est plutôt la nature et le nombre de points d'intérêt qui vont être sélectionnés dans chaque package qui change sensiblement.

En effet, si le budget de l'utilisateur sur le coût est faible, nous avons remarqué d'après nos tests que les recommandations concerneront principalement des points d'intérêt gratuits. Ceci s'explique par le fait que le principe de notre algorithme est d'ajouter à chaque étape le POI maximisant le score du package qui est en train d'être formé. De ce fait, si le budget de l'utilisateur sur le coût est très faible, seuls les POIs ayant un coût plus faible que ce budget seront ajoutés au package, ceci jusqu'à atteindre la limite du temps de visite.

De manière similaire, si le budget sur le temps de visite de l'utilisateur est très faible, notre système aura tendance à recommander des packages qui contiennent très peu de points d'intérêt ou même des packages vides si le temps disponible est trop court, puisque notre algorithme ajoute au package, à chaque étape, le POI maximisant le score du package tant que les contraintes sur le budget et sur le temps sont respectées. Si le temps de visite disponible est court cela aboutira à des packages contenant trop peu de POIs. De plus, si le budget sur le temps de visite de l'utilisateur est inférieur au temps moyen de visite de chaque POI, aucun package ne peut être construit.

Pour évaluer l'efficacité du système que nous avons proposé, nous comparons nos résultats avec le système de recommandation composite proposé par *Xie et al* [Xie et al., 2010]. Ce travail est le plus proche du nôtre. Les auteurs ont aussi utilisé le filtrage collaboratif item-item pour calculer les appréciations estimées pour chaque point d'intérêt, mais ne prennent en considération ni la popularité ni la diversité des recommandations.

Pour expérimenter les différentes versions de notre système et comparer nos résultats avec l'approche de [Xie et al., 2010]. Nous avons sélectionné les 20 utilisateurs les plus actifs de notre jeux de données. Nous avons ensuite utilisée l'approche standard qui consiste à diviser les données en deux catégories de manière aléatoire : approximativement 80 % des notes du profil utilisateur sont assignées pour le "training set" et les 20 % qui restent sont assignées pour le "test set".

Les données sont traduites en données binaires pour pouvoir utiliser la mesure de précision, très populaire dans le domaine de la recherche d'information. les notes des POIs comprises dans l'ensemble {1,2,3} sont transformées en "0" (POI non pertinent) et les notes des POIs comprises dans l'ensemble {4,5} sont transformées

en "1" (POI pertinent). Pour chaque combinaison des poids C_{eapp}, C_{opop} et C_{div} ainsi que pour chaque utilisateur notre algorithme utilise le "training set" et produit une liste de k packages recommandés. la pertinence des POIs regroupés dans les k packages est calculée en utilisant la formule 6.1. Pour cela, nous comparons les recommandations avec les valeurs de vérité du "test set". Une recommandation d'un POI est pertinente si et seulement si la valeur de ce POI dans le "test set" est 1. De plus, pour les packages recommandés nous calculons la diversité intraliste moyenne selon la formule 6.2, ainsi que la moyenne harmonique entre précision et diversité selon la formule 6.4. Notons que nous utilisons à chaque fois le même processus pour les 20 utilisateurs de notre échantillon. Les résultats finaux sont alors obtenus en calculant la moyenne des résultats pour les 20 utilisateurs. Ces résultats d'évaluation sont présentés et discutés dans la section qui suit.

6.4.4 Résultats et discussions

Les résultats des différentes versions de notre système comparés à l'approche compétitive selon la précision P, la diversité D et la moyenne harmonique F_{PD} sont rapportés dans les tables 6.3 et 6.4.

	k=5			k=10		
	P	D	F_{PD}	P	D	F_{PD}
per	0.4973	0.4002	0.4435	0.5019	0.4112	0.4520
pop	0.5775	0.4789	0.5236	0.535	0.5133	0.5239
div	0.4101	**0.6196**	0.4935	0.4301	**0.6039**	0.5025
pop+div	0.537	0.5573	0.5469	0.5336	0.5804	0.5560
per+div	0.4785	0.5503	0.5119	0.4988	0.5784	0.5356
per+pop	**0.5938**	0.4325	0.5004	**0.5453**	0.5057	0.5247
per+pop+div	0.5509	0.5845	**0.5672**	0.5366	0.5988	**0.5659**
Xie et al	0.5724	0.4358	0.4948	0.5332	0.4811	0.5058

Tableau 6.3 – Résultats pour $k = 5$ et $k = 10$

Dans toutes les versions, nous constatons une influence significative de la popularité des POIs sur la précision des recommandations. Il faut souligner que la popularité est un facteur important au même titre que le facteur de personnalisation. En effet, dans la plupart des cas, la version "pop" obtient une meilleure précision que la version "per", et la version "pop+div" est meilleure que la version "per+div" concernant toujours la précision. Ces résultats sont en accord avec [Steck, 2011] qui met en évidence l'importance de la popularité et son effet sur la précision des recommandations.

	k=15			k=20		
	P	D	F_{PD}	P	D	F_{PD}
per	0.5113	0.4274	0.4656	0.4888	0.4375	0.4617
pop	0.5184	0.4943	0.5059	0.5018	0.4861	0.4938
div	0.4268	**0.5921**	0.4960	0.3808	**0.5781**	0.4591
pop+div	0.5192	0.5614	0.5394	0.4967	0.5581	0.5256
per+div	0.481	0.5718	0.5224	0.4822	0.5346	0.5070
per+pop	**0.5303**	0.5037	0.5166	**0.5111**	0.4925	0.5016
per+pop+div	0.5175	0.5733	**0.5439**	0.5039	0.5584	**0.5297**
Xie et al	0.5165	0.508	0.5122	0.5045	0.5285	0.5162

Tableau 6.4 – résultats pour $k = 15$ et $k = 20$

La version "div" est celle qui aboutit à la précision la plus faible car elle ignore les aspects de personnalisation et de popularité lors de la construction des packages. Sa fonction de score n'utilise que le seul critère de diversité. Les packages qui sont crées sont alors trop divers et peuvent s'éloigner des préférences de l'utilisateur. En variant le nombre de packages, la version "per+pop" aboutit toujours à la précision la plus grande et surpasse l'algorithme de *Xie et al*, en raison de la combinaison de la personnalisation et la popularité dans la fonction de score des packages. Concernant donc la précision, la meilleure approche est de combiner à la fois la précision et la popularité des POIs lors de la construction des packages qui seront recommandés.

Concernant la diversité des recommandations, sans surprise la version "div" est celle qui aboutit à une meilleure diversité en comparaison avec toutes les autres approches, puisqu'elle ne prend en compte que la diversité des POIs dans chaque package lors du processus de recommandation. Cependant, cette méthode est aussi celle qui offre la plus faible précision. Aussi, dans tous les cas, la version "pop+div" obtient une meilleure diversité que la version "pop" et la version "per+div" obtient une meilleure diversité que la version "per". Cependant, il faut souligner que les versions "pop+div" et "per+div" ont une précision qui est assez éloignée de la précision la plus haute, celle-ci étant obtenue par la version "per+pop".

Étudions maintenant le cas de la F_{PD} pour analyser le comportement des différentes versions vis-à-vis du compromis entre précision et diversité. Nous remarquons que la version "per+pop+div" réalise le meilleur compromis avec la plus grande valeur de F_{PD} et surpasse l'algorithme de *Xie et al*. Cette version a tendance à promouvoir une large diversité, et surtout, elle n'est pas significativement différente en précision de l'approche "per+pop". Ainsi l'approche "per+pop+div" prenant en compte la personnalisation, la popularité et la diversité est la meilleure approche en ce qui concerne le compromis entre précision et diversité.

6.5 Conclusion

Nous avons présenté dans ce chapitre les scénarios d'utilisation de nos deux applications mobiles pour l'aide à la visite de musée et pour le tourisme. Nous avons présenté les différentes implémentations que nous avons réalisées et les technologies que nous avons utilisées pour la réalisation des applications. Nous avons ensuite présenté les évaluations que nous avons réalisées de notre système de recommandation composite avec un jeu de données réelles issu de Tripadvisor. Nous avons testé et comparé notre approche en termes de précision, de diversité et de compromis entre précision et diversité. Nous avons montré que notre approche est très compétitive, elle permet de promouvoir une large diversité tout en gardant une précision élevée. Le dernier chapitre sera consacré aux conclusions, limites de nos travaux ainsi qu'aux perspectives pour d'éventuelles améliorations.

Conclusion et Perspectives

Sommaire

7.1 Conclusion

Dans cette thèse, nous nous sommes intéressé aux systèmes d'aide à la visite de musée et aux visites touristiques. Notre but était de concevoir des systèmes sur dispositifs mobiles, afin d'améliorer l'expérience du visiteur lors de sa visite. Nous nous sommes focalisé sur les systèmes de recommandation dans le but de personnaliser les visites, afin d'offrir une visite sur mesure pour chaque visiteur.

Nous avons proposé une approche de recommandation hybride et sensible au contexte qui utilise trois méthodes de recommandation : démographique, sémantique et collaborative. Chaque méthode est adaptée à une étape spécifique de la visite. L'approche démographique est tout d'abord utilisée afin de résoudre le problème du démarrage à froid. L'approche sémantique est ensuite activée pour recommander à l'utilisateur des œuvres sémantiquement proches de celles qu'il a appréciées. L'approche collaborative est finalement utilisée, pour recommander à l'utilisateur des œuvres que les utilisateurs qui lui sont similaires ont aimées. Nous prenons en compte le contexte de l'utilisateur à l'aide d'un post-filtrage contextuel, qui permet la génération d'un parcours, dépendant des œuvres qui ont été recommandées et aimées par l'utilisateur et qui prend en compte des informations contextuelles à savoir : l'environnement physique, la localisation ainsi que le temps de visite. L'implémentation de ce système a abouti à l'application mobile CIME-Musée.

Concernant notre deuxième domaine d'application à savoir le tourisme, nous avons constaté que les items à recommander sont des points d'intérêt qui peuvent être de différents types (monument, musée, parc, etc). La forme de recommandation

classique sous forme de listes triées n'est pas adaptée à ce cas de figure. La nature hétérogène de ces points d'intérêt nous a poussé à proposer un système de recommandation composite. Dans l'approche que nous proposons, chaque recommandation est une liste de points d'intérêt, c'est-à-dire que nous organisons les recommandations sous forme de packages, chaque package pouvant éventuellement être un parcours. Notre objectif est alors de recommander à l'utilisateur les Top-k packages parmi ceux qui satisfont les contraintes de l'utilisateur (temps et coût de visite). Nous avons formellement défini le problème et nous avons proposé un algorithme inspiré de la recherche d'information composite. Pour construire la liste des packages recommandés, notre algorithme utilise une fonction de score qui évalue la qualité d'un package suivant trois critères : l'appréciation estimée de l'utilisateur, la popularité des points d'intérêt ainsi que la diversité du package. L'implémentation de notre système de recommandation composite a abouti à l'application CIME-Tourisme. Une évaluation en utilisant un jeu de données réel extrait du site web Tripadvisor a démontré la qualité de notre approche et sa capacité à améliorer à la fois la précision et la diversité des recommandations.

7.2 Perspectives

Une des limites du travail que nous avons présenté est l'absence d'évaluation du système de recommandation proposé pour la visite de musées. En effet, l'évaluation d'un système de recommandation en utilisant des critères de performances adéquats est une étape importante pour montrer la qualité du système proposé. Dans le domaine des musées, il n'existe pas de jeu de données public comme dans le domaine des films (MovieLens, Netflix, etc.), ni de site web spécialisé où des utilisateurs donnent leurs avis sur des œuvres d'un musée que nous pourrions extraire comme nous l'avons fait pour Tripadvisor. Nous avons alors décidé de créer notre propre jeu de données qui contient des évaluations réalisées par des internautes sur les œuvres de notre base de connaissances (œuvres exposées au palais impérial de Compiègne). Nous avons utilisé Google forms pour la récupération de ces données, mais le nombre d'utilisateurs ainsi que le nombre d'œuvres notées que nous avons pu obtenir était trop faible pour pouvoir réaliser une expérimentation. Nous n'avons donc pas pu tester la qualité des recommandations et comparer notre approche hybride avec les approches classiques de la littérature en calculant l'erreur sur les prédictions de notes. L'utilisation de données synthétiques (automatiquement générées) n'est pas envisageable car elles ne refléteraient pas les préférences réelles des utilisateurs. Nous souhaitons ensuite réaliser une expérimentation avec des visiteurs réels en musée et en situation de mobilité, afin d'évaluer la satisfaction des utilisateurs de

notre système en comparaison avec un parcours prédéfini dans le cadre d'une visite guidée. Mais les systèmes de localisation Indoor sont actuellement trop limités et cela rend cette tâche compliquée.

Une autre perspective est l'amélioration de l'agrégation des similarités sémantiques. En effet, notre mesure de similarité entre deux œuvres est définie comme la somme pondérée des similarités sémantiques entre les valeurs que prennent les propriétés de ces deux œuvres. Cette solution permet d'associer des poids aux propriétés en fonction de leur importance (pondération des critères d'agrégation). Ces poids peuvent être fixés par exemple par un expert en muséologie. L'utilisation de la moyenne pondérée présente une limite, elle ne permet pas de prendre en compte un aspect important qui est la dépendance entre critères d'agrégation. C'est pour cela que nous souhaitons étudier l'utilisation de l'intégrale de Choquet. Cet opérateur d'agrégation est particulièrement utilisé en aide multicritère à la décision (AMD), la particularité de cet opérateur est de prendre en compte les interactions qui peuvent exister entre les critères. Plusieurs critères peuvent en effet être complémentaires ou bien redondants. L'idée fondamentale de l'intégrale de Choquet est d'associer des poids non seulement à chaque critère d'agrégation mais également à chaque ensemble de critères d'agrégation.

Une autre limite de notre travail est la prise en compte du contexte de localisation de notre système de recommandation composite pour le tourisme. En effet, en l'état actuel, nous supposons que le temps de visite d'un package (liste de POIs) est égal à la somme des temps de visite de chaque POI. Or, dans le cadre d'une visite touristique réelle, le temps de déplacement est aussi à prendre en compte, d'où l'importance de la localisation des points d'intérêt ainsi que de la localisation du visiteur dans la ville. Nous prévoyons d'étendre notre approche de recommandation composite pour prendre en compte cette dimension. Nous comptons intégrer dans la fonction de score d'un package une mesure qui calculera la distance totale à parcourir pour visiter tous les POIs du package et notre but sera de la minimiser. Nous réaliserons une expérimentation avec des utilisateurs réels en situation de mobilité, afin de comparer leur satisfaction avec notre système de recommandation composite et avec un système de recommandation classique qui fournirait des recommandations sous forme de listes triées.

Une autre perspective particulièrement intéressante dans le domaine du tourisme et de la visite de musées, serait de pouvoir faire des recommandations non pas pour un utilisateur individuel, mais pour un groupe d'utilisateurs. Nous souhaitons ainsi étendre notre approche pour pouvoir suggérer de "bonnes" recommandations à un groupe d'utilisateurs en essayant de satisfaire, autant que possible, les préférences individuelles de chaque membre du groupe. Nous étudierons deux méthodes de

recommandation pour les groupes. La première méthode est basée sur la génération d'un profil utilisateur abstrait représentant l'ensemble des profils des utilisateurs dans le groupe, et sur l'application d'un algorithme de recommandation pour cet utilisateur. La deuxième méthode consiste à produire une liste de recommandations pour chaque utilisateur du groupe, et ensuite à agréger les différentes listes en une seule liste finale.

Bibliographie

[Abowd et al., 1999] Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., and Steggles, P. (1999). Towards a better understanding of context and context-awareness. In *Handheld and ubiquitous computing*, pages 304–307. Springer.

[Adomavicius et al., 2005] Adomavicius, G., Sankaranarayanan, R., Sen, S., and Tuzhilin, A. (2005). Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems (TOIS)*, 23(1) :103–145.

[Adomavicius and Tuzhilin, 2005] Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems : A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6) :734–749.

[Adomavicius and Tuzhilin, 2011] Adomavicius, G. and Tuzhilin, A. (2011). Context-aware recommender systems. In *Recommender systems handbook*, pages 217–253. Springer.

[Ahn et al., 2007] Ahn, J.-w., Brusilovsky, P., Grady, J., He, D., and Syn, S. Y. (2007). Open user profiles for adaptive news systems : help or harm ? In *Proceedings of the 16th international conference on World Wide Web*, pages 11–20. ACM.

[Aimé, 2011] Aimé, X. (2011). *Gradients de prototypicalité, mesures de similarité et de proximité sémantique : une contribution à l'Ingénierie des Ontologies.* PhD thesis, Université de Nantes.

[Amer-Yahia et al., 2013] Amer-Yahia, S., Bonchi, F., Castillo, C., Feuerstein, E., Méndez-Díaz, I., and Zabala, P. (2013). Complexity and algorithms for composite retrieval. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 79–80. ACM.

[Amer-Yahia et al., 2014] Amer-Yahia, S., Bonchi, F., Castillo, C., Feuerstein, E., Mendez-Diaz, I., and Zabala, P. (2014). Composite retrieval of diverse and complementary bundles. *IEEE Transactions on Knowledge and Data Engineering*, 26(11) :2662–2675.

[Angel et al., 2009] Angel, A., Chaudhuri, S., Das, G., and Koudas, N. (2009). Ranking objects based on relationships and fixed associations. In *Proceedings of the 12th International Conference on Extending Database Technology : Advances in Database Technology*, pages 910–921. ACM.

[Autere and Vakkari, 2011] Autere, R. and Vakkari, M. (2011). Towards cross-organizational interoperability : The lido xml schema as a national level integration tool for the national digital library of finland. In *International Conference on Theory and Practice of Digital Libraries*, pages 62–68. Springer.

[Baader et al., 2005] Baader, F., Horrocks, I., and Sattler, U. (2005). Description logics as ontology languages for the semantic web. In *Mechanizing Mathematical Reasoning*, pages 228–248. Springer.

[Bachimont, 2000] Bachimont, B. (2000). Engagement sémantique et engagement ontologique : conception et réalisation d ?ontologies en ingénierie des connaissances. *Ingénierie des connaissances : évolutions récentes et nouveaux défis*, pages 305–323.

[Bachimont, 2004] Bachimont, B. (2004). Arts et sciences du numérique : ingénierie des connaissances et critique de la raison computationnelle. *Mémoire de HDR*.

[Baeza-Yates et al., 1999] Baeza-Yates, R., Ribeiro-Neto, B., et al. (1999). *Modern information retrieval*, volume 463. ACM press New York.

[Balabanović and Shoham, 1997] Balabanović, M. and Shoham, Y. (1997). Fab : content-based, collaborative recommendation. *Communications of the ACM*, 40(3) :66–72.

[Batet et al., 2012] Batet, M., Moreno, A., Sánchez, D., Isern, D., and Valls, A. (2012). Turist@ : Agent-based personalised recommendation of tourist activities. *Expert Systems with Applications*, 39(8) :7319–7329.

[Belkin and Croft, 1992] Belkin, N. J. and Croft, W. B. (1992). Information filtering and information retrieval : Two sides of the same coin ? *Communications of the ACM*, 35(12) :29–38.

[Bell et al., 2007] Bell, R., Koren, Y., and Volinsky, C. (2007). Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 95–104. ACM.

[Bell and Koren, 2007] Bell, R. M. and Koren, Y. (2007). Lessons from the netflix prize challenge. *ACM SIGKDD Explorations Newsletter*, 9(2) :75–79.

[Bennett and Lanning, 2007] Bennett, J. and Lanning, S. (2007). The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35.

[Benouaret, 2015] Benouaret, I. (2015). Un système de recommandation sensible au contexte pour la visite de musée. In *CORIA*, pages 515–524.

[Benouaret and Lenne, 2015a] Benouaret, I. and Lenne, D. (2015a). Combining semantic and collaborative recommendations to generate personalized museum tours. In *East European Conference on Advances in Databases and Information Systems*, pages 477–487. Springer.

[Benouaret and Lenne, 2015b] Benouaret, I. and Lenne, D. (2015b). Personalizing the museum experience through context-aware recommendations. In *Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on*, pages 743–748. IEEE.

[Benouaret and Lenne, 2016a] Benouaret, I. and Lenne, D. (2016a). A composite recommendation system for planning tourist visits. In *Web Intelligence (WI), 2016 IEEE/WIC/ACM International Conference on*, pages 626–631. IEEE.

[Benouaret and Lenne, 2016b] Benouaret, I. and Lenne, D. (2016b). A package recommendation framework for trip planning activities. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 203–206. ACM.

[Benouaret and Lenne, 2017] Benouaret, I. and Lenne, D. (2017). Recommending diverse and personalized travel packages. In *International Conference on Database and Expert Systems Applications*, pages 325–339. Springer.

[Bergmann and Stahl, 1998] Bergmann, R. and Stahl, A. (1998). *Similarity measures for object-oriented case representations*. Springer.

[Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., Lassila, O., et al. (2001). The semantic web. *Scientific american*, 284(5) :28–37.

[Billsus and Pazzani, 2000] Billsus, D. and Pazzani, M. J. (2000). User modeling for adaptive news access. *User modeling and user-adapted interaction*, 10(2-3) :147–180.

[Borràs et al., 2011] Borràs, J., de la Flor, J., Pérez, Y., Moreno, A., Valls, A., Isern, D., Orellana, A., Russo, A., and Anton-Clavé, S. (2011). Sigtur/e-destination : a system for the management of complex tourist regions. In *Information and Communication Technologies in Tourism 2011*, pages 39–50. Springer.

[Borras et al., 2014] Borras, J., Moreno, A., and Valls, A. (2014). Intelligent tourism recommender systems : A survey. *Expert Systems with Applications*, 41(16) :7370–7389.

[Bowen et al., 2008] Bowen, J., Bradburne, J., Burch, A., Dierking, L., Falk, J., Fantoni, S. F., Gammon, B., Giusti, E., Gottlieb, H., Hsi, S., et al. (2008). *Digital technologies and the museum experience : Handheld guides and other media*. Rowman Altamira.

[Bradley and Smyth, 2001] Bradley, K. and Smyth, B. (2001). Improving recommendation diversity. In *Proceedings of the Twelfth Irish Conference on Artificial Intelligence and Cognitive Science, Maynooth, Ireland*, pages 85–94. Citeseer.

[Braunhofer et al., 2013] Braunhofer, M., Elahi, M., Ricci, F., and Schievenin, T. (2013). Context-aware points of interest suggestion with dynamic weather data management. In *Information and communication technologies in tourism 2014*, pages 87–100. Springer.

[Breese et al., 1998] Breese, J. S., Heckerman, D., and Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52. Morgan Kaufmann Publishers Inc.

[Bright et al., 2005] Bright, A., Kay, J., Ler, D., Ngo, K., Niu, W., and Nuguid, A. (2005). Adaptively recommending museum tours. In *Workshop on Smart*

Environments and Their Applications to Cultural Heritage at UbiComp, pages 29–32.

[Brodsky et al., 2008] Brodsky, A., Morgan Henshaw, S., and Whittle, J. (2008). Card : a decision-guidance framework and application for recommending composite alternatives. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 171–178. ACM.

[Burke, 2002] Burke, R. (2002). Hybrid recommender systems : Survey and experiments. *User modeling and user-adapted interaction*, 12(4) :331–370.

[Burke, 2007] Burke, R. (2007). *The Adaptive Web : Methods and Strategies of Web Personalization*, chapter Hybrid Web Recommender Systems, pages 377–408. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Castillo et al., 2008] Castillo, L., Armengol, E., Onaindía, E., Sebastiá, L., González-Boticario, J., Rodríguez, A., Fernández, S., Arias, J. D., and Borrajo, D. (2008). Samap : An user-oriented adaptive system for planning tourist visits. *Expert Systems with Applications*, 34(2) :1318–1332.

[Ceccaroni et al., 2009] Ceccaroni, L., Codina, V., Palau, M., and Pous, M. (2009). Patac : Urban, ubiquitous, personalized services for citizens and tourists. In *Digital Society, 2009. ICDS'09. Third International Conference on*, pages 7–12. IEEE.

[Chien and George, 1999] Chien, Y.-H. and George, E. I. (1999). A bayesian model for collaborative filtering. In *AISTATS*.

[Clarke et al., 2008] Clarke, C. L., Kolla, M., Cormack, G. V., Vechtomova, O., Ashkan, A., Büttcher, S., and MacKinnon, I. (2008). Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 659–666. ACM.

[Collins and Quillian, 1969] Collins, A. M. and Quillian, M. R. (1969). Retrieval time from semantic memory. *Journal of Memory and Language*, 8(2) :240.

[Davis et al., 1993] Davis, R., Shrobe, H., and Szolovits, P. (1993). What is a knowledge representation ? *AI magazine*, 14(1) :17.

[De Choudhury et al., 2010] De Choudhury, M., Feldman, M., Amer-Yahia, S., Golbandi, N., Lempel, R., and Yu, C. (2010). Automatic construction of travel itineraries using social breadcrumbs. In *Proceedings of the 21st ACM conference on Hypertext and hypermedia*, pages 35–44. ACM.

[Delgado and Ishii, 1999] Delgado, J. and Ishii, N. (1999). Memory-based weighted majority prediction. In *SIGIR Workshop Recomm. Syst. Citeseer*. Citeseer.

[Desrosiers and Karypis, 2011] Desrosiers, C. and Karypis, G. (2011). A comprehensive survey of neighborhood-based recommendation methods. In *Recommender systems handbook*, pages 107–144. Springer.

[Di Noia et al., 2014] Di Noia, T., Ostuni, V. C., Rosati, J., Tomeo, P., and Di Sciascio, E. (2014). An analysis of users' propensity toward diversity in recommendations. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 285–288. ACM.

[Doerr et al., 2010] Doerr, M., Gradmann, S., Hennicke, S., Isaac, A., Meghini, C., and van de Sompel, H. (2010). The europeana data model (edm). In *World Library and Information Congress : 76th IFLA general conference and assembly*, pages 10–15.

[Doerr et al., 2007] Doerr, M., Ore, C.-E., and Stead, S. (2007). The cidoc conceptual reference model : a new standard for knowledge sharing. In *Tutorials, posters, panels and industrial contributions at the 26th international conference on Conceptual modeling-Volume 83*, pages 51–56. Australian Computer Society, Inc.

[Ekstrand et al., 2011] Ekstrand, M. D., Riedl, J. T., and Konstan, J. A. (2011). Collaborative filtering recommender systems. *Foundations and Trends in Human-Computer Interaction*, 4(2) :81–173.

[Fenza et al., 2011] Fenza, G., Fischetti, E., Furno, D., and Loia, V. (2011). A hybrid context aware system for tourist guidance based on collaborative filtering. In *Fuzzy Systems (FUZZ), 2011 IEEE International Conference on*, pages 131–138. IEEE.

[Fielding, 2000] Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures*. PhD thesis, University of California, Irvine.

[Gandon et al., 2012] Gandon, F., Corby, O., and Faron-Zucker, C. (2012). *Le web sémantique : Comment lier les données et les schémas sur le web ?* Dunod.

[Garcia et al., 2013] Garcia, A., Torre, I., and Linaza, M. T. (2013). Mobile social travel recommender system. In *Information and communication technologies in tourism 2014*, pages 3–16. Springer.

[Garcia et al., 2011] Garcia, I., Sebastia, L., and Onaindia, E. (2011). On the design of individual and group recommender systems for tourism. *Expert systems with applications*, 38(6) :7683–7692.

[García-Crespo et al., 2009] García-Crespo, A., Chamizo, J., Rivera, I., Mencke, M., Colomo-Palacios, R., and Gómez-Berbís, J. M. (2009). Speta : Social pervasive e-tourism advisor. *Telematics and Informatics*, 26(3) :306–315.

[García-Crespo et al., 2011] García-Crespo, Á., López-Cuadrado, J. L., Colomo-Palacios, R., González-Carrasco, I., and Ruiz-Mezcua, B. (2011). Sem-fit : A semantic based expert system to provide recommendations in the tourism domain. *Expert systems with applications*, 38(10) :13310–13319.

[Gavalas and Kenteris, 2011] Gavalas, D. and Kenteris, M. (2011). A web-based pervasive recommendation system for mobile tourist guides. *Personal and Ubiquitous Computing*, 15(7) :759–770.

[Ghazanfar and Prügel-Bennett, 2014] Ghazanfar, M. A. and Prügel-Bennett, A. (2014). Leveraging clustering approaches to solve the gray-sheep users problem in recommender systems. *Expert Systems with Applications*, 41(7) :3261–3275.

[Gicquel, 2013] Gicquel, P.-Y. (2013). *Proximités sémantiques et contextuelles pour l'apprentissage en mobilité : application à la visite de musée.* PhD thesis, Université de Technologie de Compiègne.

[Gicquel et al., 2013] Gicquel, P.-Y., Lenne, D., and Moulin, C. (2013). Using semantic proximities to control contextualized activities during museum visits. In *International Conference on Artificial Intelligence in Education*, pages 864–867. Springer.

[Gob and Drouguet, 2014] Gob, A. and Drouguet, N. (2014). *La muséologie-4e éd. : Histoire, développements, enjeux actuels*, volume 1. Armand Colin.

[Goldberg et al., 1992] Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12) :61–70.

[Goldberg et al., 2001] Goldberg, K., Roeder, T., Gupta, D., and Perkins, C. (2001). Eigentaste : A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2) :133–151.

[Gruber, 1993] Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2) :199–220.

[Guarino, 1995] Guarino, N. (1995). Formal ontology, conceptual analysis and knowledge representation. *International journal of human-computer studies*, 43(5) :625–640.

[Heckerman et al., 2001] Heckerman, D., Chickering, D. M., Meek, C., Rounthwaite, R., and Kadie, C. (2001). Dependency networks for inference, collaborative filtering, and data visualization. *The Journal of Machine Learning Research*, 1 :49–75.

[Herlocker et al., 2002] Herlocker, J., Konstan, J. A., and Riedl, J. (2002). An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information retrieval*, 5(4) :287–310.

[Herlocker et al., 1999] Herlocker, J. L., Konstan, J. A., Borchers, A., and Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237. ACM.

[Hill et al., 1995] Hill, W., Stead, L., Rosenstein, M., and Furnas, G. (1995). Recommending and evaluating choices in a virtual community of use. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 194–201. ACM Press/Addison-Wesley Publishing Co.

[Ho lee et al., 1993] Ho lee, J., Ho kim, M., and Joon lee, Y. (1993). Information retrieval based on conceptual distance in is-a hierarchies. *Journal of documentation*, 49(2) :188–207.

[Huang and Bian, 2009] Huang, Y. and Bian, L. (2009). A bayesian network and analytic hierarchy process based personalized recommendations for tourist attractions over the internet. *Expert Systems with Applications*, 36(1) :933–943.

[Jannach et al., 2009] Jannach, D., Zanker, M., and Jessenitschnig, M. (2009). Developing knowledge-based travel advisor systems : a case study. *Tourism informatics : Visual travel recommender systems, social communities, and user interface design*, pages 38–53.

[Jasper et al., 1999] Jasper, R., Uschold, M., et al. (1999). A framework for understanding and classifying ontology applications. In *Proceedings 12th Int. Workshop on Knowledge Acquisition, Modelling, and Management KAW*, volume 99, pages 16–21.

[Jeong and Lee, 2006] Jeong, J.-H. and Lee, K.-H. (2006). The physical environment in museums and its effects on visitors ? satisfaction. *Building and Environment*, 41(7) :963–969.

[Karatzoglou et al., 2010] Karatzoglou, A., Amatriain, X., Baltrunas, L., and Oliver, N. (2010). Multiverse recommendation : n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 79–86. ACM.

[Kitts et al., 2000] Kitts, B., Freed, D., and Vrieze, M. (2000). Cross-sell : a fast promotion-tunable customer-item recommendation method based on conditionally independent probabilities. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 437–446. ACM.

[Koceski and Petrevska, 2012] Koceski, S. and Petrevska, B. (2012). Empirical evidence of contribution to e-tourism by application of personalized tourism recommendation system. *Annals of the Alexandru Ioan Cuza University-Economics*, 59(1) :363–374.

[Koren, 2008] Koren, Y. (2008). Factorization meets the neighborhood : a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434. ACM.

[Koren, 2009] Koren, Y. (2009). The bellkor solution to the netflix grand prize. *Netflix prize documentation*, 81.

[Koren and Bell, 2011] Koren, Y. and Bell, R. (2011). Advances in collaborative filtering. In *Recommender systems handbook*, pages 145–186. Springer.

[Koren et al., 2009] Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, (8) :30–37.

[Kuflik et al., 2011] Kuflik, T., Stock, O., Zancanaro, M., Gorfinkel, A., Jbara, S., Kats, S., Sheidin, J., and Kashtan, N. (2011). A visitor's guide in an active museum : Presentations, communications, and reflection. *Journal on Computing and Cultural Heritage (JOCCH)*, 3(3) :11.

[Kurata, 2011] Kurata, Y. (2011). Ct-planner2 : more flexible and interactive assistance for day tour planning. In *ENTER*, volume 2011, pages 25–37.

[Kurata and Hara, 2013] Kurata, Y. and Hara, T. (2013). Ct-planner4 : Toward a more user-friendly interactive day-tour planner. In *Information and communication technologies in tourism 2014*, pages 73–86. Springer.

[Laine et al., 2010] Laine, T. H., Vinni, M., Sedano, C. I., and Joy, M. (2010). On designing a pervasive mobile learning platform. *Research in Learning Technology*, 18(1).

[Lamsfus et al., 2009] Lamsfus, C., Alzua-Sorzabal, A., Martin, D., Salvador, Z., and Usandizaga, A. (2009). Human-centric ontology-based context modelling in tourism. In *KEOD*, pages 424–434.

[Laublet et al., 2002] Laublet, P., Reynaud, C., and Charlet, J. (2002). Sur quelques aspects du web sémantique. *Assises du GDR I3*, page 46.

[Leacock et al., 1998] Leacock, C., Miller, G. A., and Chodorow, M. (1998). Using corpus statistics and wordnet relations for sense identification. *Computational Linguistics*, 24(1) :147–165.

[Lee et al., 2009] Lee, C.-S., Chang, Y.-C., and Wang, M.-H. (2009). Ontological recommendation multi-agent for tainan city travel. *Expert Systems with Applications*, 36(3) :6740–6753.

[Levesque, 1986] Levesque, H. J. (1986). Knowledge representation and reasoning. *Annual review of computer science*, 1(1) :255–287.

[Lieberman, 1995] Lieberman, H. (1995). Letizia : An agent that assists web browsing. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, pages 924–929, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Lin, 1998] Lin, D. (1998). An information-theoretic definition of similarity. In *ICML*, volume 98, pages 296–304.

[Linaza et al., 2011] Linaza, M. T., Agirregoikoa, A., Garcia, A., Torres, J. I., and Aranburu, K. (2011). *Image-based travel recommender system for small tourist destinations*. Springer.

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

[Lops et al., 2011] Lops, P., De Gemmis, M., and Semeraro, G. (2011). Content-based recommender systems : State of the art and trends. In *Recommender systems handbook*, pages 73–105. Springer.

[Lorenzi et al., 2011] Lorenzi, F., Loh, S., and Abel, M. (2011). Personaltour : A recommender system for travel packages. In *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Volume 02*, pages 333–336. IEEE Computer Society.

[Luberg et al., 2011] Luberg, A., Tammet, T., and Järv, P. (2011). Smart city : a rule-based tourist recommendation system. In *Information and Communication Technologies in Tourism 2011*, pages 51–62. Springer.

[Lucas et al., 2009] Lucas, J. P., Laurent, A., Moreno, M., and Teisseire, M. (2009). Fuzz-cba : Classification à base de règles d'association floues et systèmes de recommandation. In *LFA'09 : Rencontres Francophones sur la Logique Floue et ses Applications*, pages 283–290. Cepadues.

[Lucas et al., 2013] Lucas, J. P., Luz, N., Moreno, M. N., Anacleto, R., Figueiredo, A. A., and Martins, C. (2013). A hybrid recommendation approach for a tourism system. *Expert Systems with Applications*, 40(9) :3532–3550.

[Martin et al., 2011] Martin, D., Alzua, A., and Lamsfus, C. (2011). A contextual geofencing mobile tourism service. In *ENTER*, pages 191–202.

[Martinez et al., 2009] Martinez, L., Rodriguez, R. M., and Espinilla, M. (2009). Reja : a georeferenced hybrid recommender system for restaurants. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 03*, pages 187–190. IEEE Computer Society.

[Meehan et al., 2013] Meehan, K., Lunney, T., Curran, K., and McCaughey, A. (2013). Context-aware intelligent recommendation system for tourism. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on*, pages 328–331. IEEE.

[Michalski et al., 2013] Michalski, R. S., Carbonell, J. G., and Mitchell, T. M. (2013). *Machine learning : An artificial intelligence approach*. Springer Science & Business Media.

[Mínguez et al., 2009] Mínguez, I., Berrueta, D., and Polo, L. (2009). Cruzar : An application of semantic. *Cases on Semantic Interoperability for Information Systems Integration : Practices and Applications : Practices and Applications*, page 255.

[Minsky, 1974] Minsky, M. (1974). A framework for representing knowledge.

[Montaner et al., 2003] Montaner, M., López, B., and De La Rosa, J. L. (2003). A taxonomy of recommender agents on the internet. *Artificial intelligence review*, 19(4) :285–330.

[Montejo-Ráez et al., 2011] Montejo-Ráez, A., Perea-Ortega, J. M., García-Cumbreras, M. Á., and Martínez-Santiago, F. (2011). Otiŭm : A web based planner for tourism and leisure. *Expert Systems with Applications*, 38(8) :10085–10093.

[Mooney and Roy, 2000] Mooney, R. J. and Roy, L. (2000). Content-based book recommending using learning for text categorization. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 195–204. ACM.

[Nakamura and Abe, 1998] Nakamura, A. and Abe, N. (1998). Collaborative filtering using weighted majority prediction algorithms. In *ICML*, volume 98, pages 395–403.

[Niaraki and Kim, 2009] Niaraki, A. S. and Kim, K. (2009). Ontology based personalized route planning system using a multi-criteria decision making approach. *Expert Systems with Applications*, 36(2) :2250–2259.

[Noguera et al., 2012] Noguera, J. M., Barranco, M. J., Segura, R. J., and MartíNez, L. (2012). A mobile 3d-gis hybrid recommender system for tourism. *Information Sciences*, 215 :37–52.

[Noy and Klein, 2004] Noy, N. F. and Klein, M. (2004). Ontology evolution : Not the same as schema evolution. *Knowledge and information systems*, 6(4) :428–440.

[Oppermann and Specht, 2000] Oppermann, R. and Specht, M. (2000). A context-sensitive nomadic exhibition guide. In *International Symposium on Handheld and Ubiquitous Computing*, pages 127–142. Springer.

[Palmisano et al., 2008] Palmisano, C., Tuzhilin, A., and Gorgoglione, M. (2008). Using context to improve predictive modeling of customers in personalization applications. *Knowledge and Data Engineering, IEEE Transactions on*, 20(11) :1535–1549.

[Papagelis et al., 2005] Papagelis, M., Plexousakis, D., and Kutsuras, T. (2005). Alleviating the sparsity problem of collaborative filtering using trust inferences. In *International Conference on Trust Management*, pages 224–239. Springer.

[Parameswaran et al., 2009] Parameswaran, A., Venetis, P., and Garcia-Molina, H. (2009). Recommendation systems with complex constraints : A course recommendation perspective. *Technical report*.

[Parameswaran et al., 2011] Parameswaran, A., Venetis, P., and Garcia-Molina, H. (2011). Recommendation systems with complex constraints : A courserank perspective. *Transactions on Information Systems (TOIS)–To Appear*.

[Pazzani and Billsus, 2007] Pazzani, M. J. and Billsus, D. (2007). Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer.

[Piotte and Chabbert, 2009] Piotte, M. and Chabbert, M. (2009). The pragmatic theory solution to the netflix grand prize. *Netflix prize documentation.*

[Pirró and Euzenat, 2010] Pirró, G. and Euzenat, J. (2010). A feature and information theoretic framework for semantic similarity and relatedness. In *The Semantic Web–ISWC 2010*, pages 615–630. Springer.

[Poitrenaud, 1998] Poitrenaud, S. (1998). *La représentation des PROCédures chez l'OPErateur : Description et mise en oeuvre des savoir-faire.* PhD thesis.

[Rada et al., 1989] Rada, R., Mili, H., Bicknell, E., and Blettner, M. (1989). Development and application of a metric on semantic nets. *Systems, Man and Cybernetics, IEEE Transactions on*, 19(1) :17–30.

[Resnick et al., 1994] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994). Grouplens : an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186. ACM.

[Resnik, 1995] Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007.*

[Resnik, 1993] Resnik, P. S. (1993). Selection and information : a class-based approach to lexical relationships. *IRCS Technical Reports Series*, page 200.

[Rey-López et al., 2011] Rey-López, M., Barragáns-Martínez, A. B., Peleteiro, A., Mikic-Fonte, F., and Burguillo, J. C. (2011). moretourism : Mobile recommendations for tourism. In *IEEE International Conference on Consumer Electronics (ICCE 2011). Las Vegas (USA).*

[Ricci, 2002] Ricci, F. (2002). Travel recommender systems. *IEEE Intelligent Systems*, 17(6) :55–57.

[Ricci et al., 2009] Ricci, F., Nguyen, Q. N., and Averjanova, O. (2009). Exploiting a map-based interface in conversational recommender systems for mobile travelers. *Tourism Informatics : Visual Travel Recommender Systems, Social Communities, and User Interface Design : IGI Global, Information Science Reference*, pages 73–93.

[Ricci et al., 2011] Ricci, F., Rokach, L., and Shapira, B. (2011). *Introduction to recommender systems handbook*. Springer.

[Rich, 1979] Rich, E. (1979). User modeling via stereotypes*. *Cognitive science*, 3(4) :329–354.

[Richardson et al., 1994] Richardson, R., Smeaton, A., and Murphy, J. (1994). Using wordnet as a knowledge base for measuring semantic similarity between words.

[Riordan and Sorensen, 1995] Riordan, A. and Sorensen, H. (1995). An intelligent agent for high-precision information filtering. In *Proceedings of the CIKM-95 Conference*.

[Rocchi et al., 2004] Rocchi, C., Stock, O., Zancanaro, M., Kruppa, M., and Krüger, A. (2004). The museum visit : generating seamless personalized presentations on multiple devices. In *Proceedings of the 9th international conference on Intelligent user interfaces*, pages 316–318. ACM.

[Rocchio, 1971] Rocchio, J. J. (1971). Relevance feedback in information retrieval.

[Rodríguez and Egenhofer, 2003] Rodríguez, M. A. and Egenhofer, M. J. (2003). Determining semantic similarity among entity classes from different ontologies. *Knowledge and Data Engineering, IEEE Transactions on*, 15(2) :442–456.

[Rojas and Uribe, 2013] Rojas, G. and Uribe, C. (2013). A conceptual framework to develop mobile recommender systems of points of interest. In *International workshop on advanced software engineering. Proceedings jornadas chilenas de computación, Chile, November 2013*.

[Ruiz-Montiel and Aldana-Montes, 2009] Ruiz-Montiel, M. and Aldana-Montes, J. F. (2009). Semantically enhanced recommender systems. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 604–609. Springer.

[Ruotsalo et al., 2013] Ruotsalo, T., Haav, K., Stoyanov, A., Roche, S., Fani, E., Deliai, R., Mäkelä, E., Kauppinen, T., and Hyvönen, E. (2013). Smartmuseum : A mobile recommender system for the web of data. *Web semantics : Science, services and agents on the world wide web*, 20 :50–67.

[Safoury and Salah, 2013] Safoury, L. and Salah, A. (2013). Exploiting user demographic attributes for solving cold-start problem in recommender system. *Lecture Notes on Software Engineering*, 1(3) :303.

[Salton, 1989] Salton, G. (1989). Automatic text processing : The transformation, analysis, and retrieval of. *Reading : Addison-Wesley*.

[Santiago et al., 2012] Santiago, F. M., López, F. A., Montejo-Ráez, A., and López, A. U. (2012). Geoasis : A knowledge-based geo-referenced tourist assistant. *Expert Systems with Applications*, 39(14) :11737–11745.

[Sarwar et al., 2001] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM.

[Savir et al., 2013] Savir, A., Brafman, R., and Shani, G. (2013). Recommending improved configurations for complex objects with an application in travel planning. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 391–394. ACM.

[Schafer et al., 2007] Schafer, J. B., Frankowski, D., Herlocker, J., and Sen, S. (2007). Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer.

[Sebastia et al., 2009] Sebastia, L., Garcia, I., Onaindia, E., and Guzman, C. (2009). e-tourism : a tourist recommendation and planning application. *International Journal on Artificial Intelligence Tools*, 18(05) :717–738.

[Sebastia et al., 2010] Sebastia, L., Giret, A., and Garcia, I. (2010). A multi agent architecture for tourism recommendation. In *Trends in Practical Applications of Agents and Multiagent Systems*, pages 547–554. Springer.

[Seidel et al., 2009] Seidel, I., Gärtner, M., Pöttler, M., Berger, H., Dittenbach, M., and Merkl, W. (2009). Itchy feet : a 3d e-tourism environment. *Tourism Informatics : Visual Travel Recommender Systems, Social Communities, and User Interface Design. Hershey, PA, USA : IGI Global*, pages 209–242.

[Shahabi et al., 2001] Shahabi, C., Banaei-Kashani, F., Chen, Y.-S., and McLeod, D. (2001). Yoda : An accurate and scalable web-based recommendation system. In *Cooperative Information Systems*, pages 418–432. Springer.

[Shani et al., 2002] Shani, G., Brafman, R. I., and Heckerman, D. (2002). An mdp-based recommender system. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 453–460. Morgan Kaufmann Publishers Inc.

[Shardanand and Maes, 1995a] Shardanand, U. and Maes, P. (1995a). Social information filtering : algorithms for automating ?word of mouth ? In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 210–217. ACM Press/Addison-Wesley Publishing Co.

[Shardanand and Maes, 1995b] Shardanand, U. and Maes, P. (1995b). Social information filtering : algorithms for automating ?word of mouth ? In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 210–217. ACM Press/Addison-Wesley Publishing Co.

[Sieg et al., 2007] Sieg, A., Mobasher, B., and Burke, R. D. (2007). Learning ontology-based user profiles : A semantic approach to personalized web search. *IEEE Intelligent Informatics Bulletin*, 8(1) :7–18.

[Sorensen and McElligott, 1995] Sorensen, H. and McElligott, M. (1995). Psun : a profiling system for usenet news. Citeseer.

[Sparck Jones, 1972] Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1) :11–21.

[Steck, 2011] Steck, H. (2011). Item popularity and recommendation accuracy. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 125–132. ACM.

[Stolze and Rjaibi, 2001] Stolze, M. and Rjaibi, W. (2001). Towards scalable scoring for preference-based item recommendation. *IEEE Data Eng. Bull.*, 24(3) :42–49.

[Takács et al., 2007] Takács, G., Pilászy, I., Németh, B., and Tikk, D. (2007). Major components of the gravity recommendation system. *ACM SIGKDD Explorations Newsletter*, 9(2) :80–83.

[Takács et al., 2008] Takács, G., Pilászy, I., Nemeth, B., and Tikk, D. (2008). Investigation of various matrix factorization methods for large recommender systems. In *Data Mining Workshops, 2008. ICDMW'08. IEEE International Conference on*, pages 553–562. IEEE.

[Thom-Santelli et al., 2005] Thom-Santelli, J., Toma, C., Boehner, K., and Gay, G. (2005). Beyond just the facts : Museum detective guides. In *Proceedings of the Intl Workshop Re-thinking technology in museums : Towards a new understanding of people's experience in museums*, page 2006.

[Töscher et al., 2009] Töscher, A., Jahrer, M., and Bell, R. M. (2009). The bigchaos solution to the netflix grand prize. *Netflix prize documentation*, pages 1–52.

[Towle and Quinn, 2000] Towle, B. and Quinn, C. (2000). Knowledge based recommender systems using explicit user models. In *Proceedings of the AAAI Workshop on Knowledge-Based Electronic Markets*, pages 74–77.

[Tversky, 1977] Tversky, A. (1977). Features of similarity. *Psychological review*, 84(4) :327.

[Umanets et al., 2014] Umanets, A., Ferreira, A., and Leite, N. (2014). Guideme– a tourist guide with a recommender system and social interaction. *Procedia Technology*, 17 :407–414.

[Ungar and Foster, 1998] Ungar, L. H. and Foster, D. P. (1998). Clustering methods for collaborative filtering. In *AAAI workshop on recommendation systems*, volume 1, pages 114–129.

[Van Hage et al., 2010] Van Hage, W. R., Stash, N., Wang, Y., and Aroyo, L. (2010). Finding your way through the rijksmuseum with an adaptive mobile museum guide. In *Extended Semantic Web Conference*, pages 46–59. Springer.

[Vansteenwegen and Souffriau, 2010] Vansteenwegen, P. and Souffriau, W. (2010). Trip planning functionalities : state of the art and future. *Information Technology & Tourism*, 12(4) :305–315.

[Vansteenwegen et al., 2011a] Vansteenwegen, P., Souffriau, W., Berghe, G. V., and Van Oudheusden, D. (2011a). The city trip planner : an expert system for tourists. *Expert Systems with Applications*, 38(6) :6540–6546.

[Vansteenwegen et al., 2011b] Vansteenwegen, P., Souffriau, W., and Van Oudheusden, D. (2011b). The orienteering problem : A survey. *European Journal of Operational Research*, 209(1) :1–10.

[Vargas and Castells, 2013] Vargas, S. and Castells, P. (2013). Exploiting the diversity of user preferences for recommendation. In *Proceedings of the 10th Conference on Open Research Areas in Information Retrieval*, pages 129–136. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE.

[Wang et al., 2011] Wang, W., Zeng, G., and Tang, D. (2011). Bayesian intelligent semantic mashup for tourism. *Concurrency and Computation : Practice and Experience*, 23(8) :850–862.

[Wang et al., 2009] Wang, Y., Stash, N., Aroyo, L., Hollink, L., and Schreiber, G. (2009). Using semantic relations for content-based recommender systems in cultural heritage. In *Proceedings of the 2009 International Conference on Ontology Patterns-Volume 516*, pages 16–28. CEUR-WS. org.

[Warin et al., 2005] Warin, M., Oxhammar, H., and Volk, M. (2005). M. : Enriching an ontology with wordnet based on similarity measures. In *In : MEANING-2005 Workshop*. Citeseer.

[Whan Kim and Kim, 1990] Whan Kim, Y. and Kim, J. H. (1990). A model of knowledge based information retrieval with hierarchical concept graph. *Journal of Documentation*, 46(2) :113–136.

[Wu and Palmer, 1994] Wu, Z. and Palmer, M. (1994). Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics.

[Xie et al., 2010] Xie, M., Lakshmanan, L. V., and Wood, P. T. (2010). Breaking out of the box of recommendations : from items to packages. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 151–158. ACM.

[Xie et al., 2011] Xie, M., Lakshmanan, L. V., and Wood, P. T. (2011). Comprectrip : A composite recommendation system for travel planning. In *2011 IEEE 27th International Conference on Data Engineering*, pages 1352–1355. IEEE.

[Xuan et al., 2006] Xuan, D. N., Bellatreche, L., and Pierra, G. (2006). A versioning management model for ontology-based data warehouses. In *International Conference on Data Warehousing and Knowledge Discovery*, pages 195–206. Springer.

[Yang and Hwang, 2013] Yang, W.-S. and Hwang, S.-Y. (2013). itravel : A recommender system in mobile peer-to-peer environment. *Journal of Systems and Software*, 86(1) :12–20.

[Yu et al., 2009] Yu, C., Lakshmanan, L., and Amer-Yahia, S. (2009). It takes variety to make a world : diversification in recommender systems. In *Proceedings of the 12th international conference on extending database technology : Advances in database technology*, pages 368–378. ACM.

[Yu and Chang, 2009] Yu, C.-C. and Chang, H.-P. (2009). Personalized location-based recommendation services for tour planning in mobile tourism applications. In *International Conference on Electronic Commerce and Web Technologies*, pages 38–49. Springer.

[Zancanaro et al., 2003] Zancanaro, M., Stock, O., and Alfaro, I. (2003). Using cinematic techniques in a multimedia museum guide.

[Zhang et al., 2002] Zhang, Y., Callan, J., and Minka, T. (2002). Novelty and redundancy detection in adaptive filtering. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 81–88. ACM.

[Ziegler et al., 2005] Ziegler, C.-N., McNee, S. M., Konstan, J. A., and Lausen, G. (2005). Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*, pages 22–32. ACM.

[Zigoris and Zhang, 2006] Zigoris, P. and Zhang, Y. (2006). Bayesian adaptive user profiling with explicit & implicit feedback. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 397–404. ACM.

[Zimmermann et al., 2003] Zimmermann, A., Lorenz, A., and Birlinghoven, S. (2003). Listen : Contextualized presentation for audio-augmented environments. In *Proceedings of the 11th Workshop on Adaptivity and User modeling in Interactive Systems*, pages 351–357.

[Zimmermann et al., 2007] Zimmermann, A., Lorenz, A., and Oppermann, R. (2007). An operational definition of context. In *International and Interdisciplinary Conference on Modeling and Using Context*, pages 558–571. Springer.