

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de L'Enseignement Supérieur et de la Recherche Scientifique



UNIVERSITÉ FERHAT ABBAS - SETIF1

FACULTÉ DES SCIENCES

THÈSE

Présentée au Département d'informatique

Pour l'obtention du diplôme de

DOCTORAT SCIENCE

Domaine : informatique

Par

Brik Mourad

THÈME

**APPROCHE POUR L'INDEXATION ET
L'UTILISATION DES RESSOURCES
PEDAGOGIQUES DANS UN SYSTEME E-LEARNING
PERVASIF**

Soutenu le 30/10/2021 devant le Jury:

Adballah KHABABA	Professeur	Univ. Ferhat Abbas Sétif 1	Président
Sadik BESSOU	MCA	Univ. Ferhat Abbas Sétif 1	Rapporteur
Samir AKROUF	MCA	Univ. Mohamed Boudiaf M'sila	Examineur
Farid NOUIOUA	MCA	Univ. Bachir Ibrahimi BBA	Examineur
Mohamed TOUAHRIA	Professeur	Univ. Ferhat Abbas Sétif 1	Invité

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

DEDICACES

Je dédie ce modeste travail

- A la mémoire de mes parents
- A ma femme
- A la mémoire de ma sœur
- A mes enfants
abderrahim, maysoune, abdderraouf et abdelmouaine
- A mes frères et sœurs
- A tous mes amis et mes collègues
- A toutes personnes qui m'ont aidé à réaliser ce travail.

REMERCIEMENTS

Je tiens à exprimer mes remerciements à qui ne m'a jamais laissé perdu dans ce monde, qui m'a donné la volonté, la santé, la patience pour terminer ce mémoire.

J'adresse mes sincères remerciements et mes grandes gratitude tout d'abord à mon ex-encadreur professeur **Touahria mohamed**, professeur à l'université de setif1 pour son aide, ses encouragements, que je souhaite à lui un prompt rétablissement.

J'adresse aussi mes sincères remerciements à mon encadreur docteur **bessou sadik**, Maître de conférence à l'université de Sétif 1 d'avoir accepté de m'encadrer pour terminer ce travail, pour son aide et ses encouragements.

Je remercie également les membres de jury d'avoir accepté de juger cette thèse. Professeur **Khababa abdallah**, professeur à l'université de Sétif, Monsieur **Akrouf samir** Docteur à l'université de Msila et Monsieur **Nouioua Farid**, Docteur à l'université de BBA.

Je tiens à remercier madame **marie Françoise canut** docteur à l'université de paul sabatier toulouse pour ses conseils concernant ce travail.

Je tiens à remercier tous ceux qui m'ont aidé à réaliser ce travail.

Merci

Résumé

Le travail présenté dans cette thèse concerne les systèmes de recommandation sensibles au contexte (CARS), plus précisément nous concentrons sur la détection automatique des conditions contextuelles (valeurs des facteurs contextuelles) qui influencent le processus de recommandation. Dans les recherches récentes, l'intégration du contexte dans les applications de recommandation a été démontrée leur efficacité en générant des items aux utilisateurs non seulement pertinents mais également convenables à leurs contexte. Tandis que, la sélection des facteurs de contexte est une tâche exigeante en ressources qui nécessite dans la plus part des cas l'intervention d'un expert de domaine.

Le principale contribution de ce travail de thèse est basée sur l'analyse des comportements des utilisateurs au sein d'un système de recommandation à base de filtrage collaboratif afin de deviner les conditions contextuelles relatives à la consommation adéquate des items en utilisant la matrice des votes générée par les systèmes de recommandation à base de filtrage collaboratif. Un profil d'utilisateur qui englobe à la fois les données concernant les utilisateurs et les données de contexte a été proposé dans ce travail. Nous proposons aussi une méthode de modélisation de contexte à base de règles sémantiques qui se penche sur l'analyse de comportement des utilisateurs. Nous avons appliqué cette proposition dans le domaine de l'éducation dans lequel nous avons conçu un outil permettant la création des objets d'apprentissages. Ce système vise à améliorer la tâche de production des objets d'apprentissage (création, révision, édition...) en offrant un environnement collaboratif grâce aux technologies fournies par les systèmes de filtrage collaboratifs ainsi que l'analyse des comportements des utilisateurs pour améliorer le processus de recommandation dans les environnements pervasifs.

La prise en compte du contexte pour la génération de recommandations futures adopte l'approche pré-filtrage contextuel avec une méthode probabiliste (Naïve Bayes), qui permet la génération des items personnalisés avant que les recommandations soient faites.

Un test a été mené sur des utilisateurs réels montre que l'approche sémantique proposée est promoteur pour ce genre de contexte.

Mots-clés : systèmes de recommandation, sensibilité au contexte, raisonnement, profil utilisateur, règles sémantiques.

Abstract

The work presented in this thesis concerns Context Sensitive Recommendation Systems (CARS), more precisely we focus on the automatic detection of contextual conditions (values of contextual factors) that influence the recommendation process. In recent research, the integration of context in recommender applications has been shown to be effective in generating user items that are not only relevant but also appropriate to their context. Whereas, the selection of context factors is a resource intensive task which in most cases requires the intervention of a domain expert.

The main thesis contribution is based on the analysis of user behavior within a collaborative filtering based recommendation system, in order to guess the contextual conditions relating to the adequate consumption of items using the Rating matrix generated by collaborative filtering based recommendation systems. A user profile that encompasses both user data and context data has been proposed in this work. We also propose a context modeling method based on semantic rules which focuses on the analysis of user behavior. We have applied this proposal in the field of education in which we have designed a tool allowing the creation of learning objects. This system aims to improve the task of producing learning objects (creation, revision, edition, etc.) by offering a collaborative environment as well as the analysis of user behavior to improve the recommendation process in pervasive environments.

The phase of context consideration for the future recommendations generation adopts the contextual pre-filtering approach with a probabilistic method (Naive Bayes), which allows the generation of personalized items before the recommendations are made.

A test was carried out on real users showing that the proposed semantic approach is a promoter for this kind of context.

Keywords: recommender system, context awareness, reasoning, user profile, semantics rules

ملخص

يتعلق العمل المقدم في هذه الأطروحة بأنظمة التوصية الحساسة للسياق (CARS) ، وبشكل أكثر دقة نركز على الاكتشاف التلقائي للظروف السياقية (قيم العوامل السياقية) التي تؤثر على عملية التوصية. في بحث حديث، ثبت أن تكامل السياق في تطبيقات التوصية فعال في عرض عناصر ليست ذات صلة فقط ولكنها مناسبة أيضاً لسياقها. وحيث أن اختيار عوامل السياق هو مهمة تتطلب موارد كثيرة وتتطلب في معظم الحالات تدخل خبير في المجال.

تعتمد المساهمة الرئيسية في هذا العمل على تحليل سلوك المستخدم داخل نظام توصية يعتمد على التصنيفية التعاونية من أجل تخمين الظروف السياقية المتعلقة بالاستهلاك الكافي للعناصر باستخدام مصفوفة التصويت التي تم إنشاؤها بواسطة أنظمة التوصية القائمة على التصنيفية التعاونية. تم اقتراح ملف تعريف المستخدم الذي يشمل بيانات المستخدم وبيانات السياق في هذا العمل. نقترح أيضاً طريقة نمذجة السياق بناءً على القواعد الدلالية التي تركز على تحليل سلوك المستخدم. لقد طبقنا هذا الاقتراح في مجال التعليم حيث صممنا أداة تسمح بإنشاء كائنات التعلم. يهدف هذا النظام إلى تحسين مهمة إنتاج كائنات التعلم (الإنشاء والمراجعة والإصدار وما إلى ذلك) من خلال توفير بيئة تعاونية بفضل التقنيات التي توفرها أنظمة التصنيفية التعاونية بالإضافة إلى تحليل سلوك المستخدم لتحسين عملية التوصية في بيئات منتشرة.

يتبنى النظر في السياق لتوليد التوصيات المستقبلية نهج التصنيفية المسبقة السياقية مع ادراج طريقة لحساب الاحتمالات (bayes naive)، والتي تسمح بتوليد العناصر الشخصية قبل تقديم التوصيات.

تم إجراء اختبار على مستخدمين حقيقيين، من خلاله تبين أن النهج الدلالي المقترح واعد لهذا النوع من المشكلات.

كلمات مفتاحية: نظام التوصية، إدراك السياق، الاستدلال، ملف تعريف المستخدم، قواعد الدلالات

Table des matières

Chapitre 1- introduction générale

1. Contexte du travail	2
2. Problématique	4
3. Contribution	5
4. Organisation du mémoire	6
Chapitre 2- les systèmes de recommandation	
1. Introduction	8
2. Historique	9
3. Les systèmes de recommandation classiques (SR)	10
4. Techniques et méthodes utilisées pour les systèmes de recommandation	10
4.1. Systèmes de recommandation à base de contenu (CBF)	11
4.1.1. Algorithmes de recommandations	12
4.1.1.1. Recommandations à base de mots-clés pondérés	12
4.1.1.2. Recommandations sémantiques	12
4.1.1.3. Approche générale	13
4.2. Système de recommandations à base de filtrage collaboratif (FC)	15
4.2.1. Approche générale	15
4.2.1.1. Techniques basés sur la mémoire	16
4.2.1.1.1. Génération des prédictions	17
4.2.1.2. Techniques basés sur les modèles	18
4.3. Les approches hybrides	20
5. Collecte de données	21
5.1. Données explicites	21
5.2. Données implicites	22
6. Verrous scientifiques	23
6.1. Démarrage à froid (cold-start)	23
6.2. Problème de rareté des données (sparsity)	23
6.3. Évolutivité des données (scalability)	23
6.4. Robustesse	24
7. Métriques d'évaluation des systèmes de recommandations	24
7.1. Evaluation en ligne	24
7.2. Evaluation sur un échantillon	24
7.3. Evaluations hors-ligne	25
7.4. Qualité des systèmes de recommandation	26
8. Systèmes de recommandation sensibles au contexte	27
8.1. Définition du contexte	27
8.2. Type des informations contextuelles	28
8.3. Modélisation du contexte	29
8.3.1. Attributs-Valeurs	30
8.3.2. Modèle logique	30
8.3.3. Ontologie	30
8.4. Source d'informations contextuelles	31
8.5. Intégration du contexte dans les systèmes de recommandation	31
8.5.1. Pré-filtrage Contextuel (contextual pre-filtering)	31
8.5.2. Post-filtrage Contextuel	31
8.5.3. La modélisation contextuelle	32
9. Conclusion	32

Chapitre 3- Profil d'utilisateur & Web sémantique

1. Le profil utilisateur	34
1.1. Introduction	34
1.2 Les Modèles utilisateurs	35
1.3. Les caractéristiques de l'utilisateur	35
1.3.1. La connaissance	35
1.3. 2. L'expérience, les compétences	35
1.3.3. Les préférences, les objectifs	35
1.4. L'acquisition des caractéristiques	36
1.5. Représentation du profil utilisateur	36
1.5.1. Représentation vectorielle	37
1.5.2. Représentation conceptuelle	37
1.5.3. Réseau sémantique	37
1.6. Évolution du profil utilisateur	37
2. Web sémantique	38
2.1. Introduction	38
2.2 Définition	38
2.3. Définition globale du web sémantique	38
2.4. Langages du web sémantique	39
2.4.1. XML	39
2.4.2. RDF et RDFS	39
2.4.3 OWL	41
2.4.4. SPARQL (Query Language for RDF)	42
2.4.5. RDFa	42
2.5 Ressources sémantiques	43
2.5.1 Taxonomie	43
2.5.2. Thesaurus	43
2.5.3. Les ontologies	43
2.5.3.1. La notion d'ontologie	73
2.5.3.2. Composants d'une ontologie	44
2.5.3.3. Typologie des ontologies	45
2.5.3.4. Classement des ontologies	45
2.5.3.5. Construction des ontologies	46
2.5.3.5.1. Méthodes de construction	46
2.5.3.5.2. Critères de construction d'ontologies	46
3. Conclusion	47

Chapitre 4 - Approche Proposée

1. Introduction	49
2. Rappel problématique	49
3. Travaux connexes	50
4. Base de connaissance de la matrice des votes	52
5. Profil d'utilisateur	53
6. Raisonnement	55
7. Conclusion	58

Chapitre 5 - Etude de cas

Partie 1 : étude de cas détaillé	60
1.1. Introduction	60
1.2. Architecture du Système	60
1.2.1. Extracteur de métadonnées	61
1.2.2. Calculateur de similarité entre les objets pédagogiques	62
1.2.3. Filtrage des utilisateurs	62
1.2.4. Évaluation et prédiction des données manquantes	62

1.3. Ontologie des profils d'utilisateurs	63
1.4. Indexation des objets pédagogiques	64
1.4.1. Les normes de métadonnées	65
1.4.2. Construction de métadonnées	66
1.4.3. Génération des éléments de métadonnées	67
1.4.4. Evaluation des objets pédagogiques	68
1.4.4.1. Calcul de similarité	69
1.4.4.2. Filtrage des utilisateurs potentiels et prédiction	70
1.4.4.3. Intégration du contexte	73
Partie 2 : Implémentation et expérimentation	74
2.1. Présentation de l'ensemble de données (dataset)	75
2.2. Evaluation du système de recommandation proposé	77
2.3. Critique	78
2.4. Evaluation de la phase d'intégration du contexte dans le processus de recommandation	79
2.5. Implémentation	80
2.5.1 Ontologie et règles sémantiques	80
3. Conclusion	83
	Conclusion générale
	85
Bibliographie	88
Annexe	96

Table des figures

Figure 2.1. Techniques et méthodes pour les SR	11
Figure 2.2. Schéma SR à base de contenu	13
Figure 2.3. Les différents éléments du contexte	28
Figure 2.4. Partie du modèle UML de contexte	29
Figure 4.1. Base de connaissance de la matrice des votes	53
Figure 4.2. Profil d'utilisateur	55
Figure 5.1. Architecture du Système proposé	61
Figure 5.2. Ontologie du profile d'utilisateur	64
Figure 5.3. Learning Object Metadata (LOM)	66
Figure 5.4. Vecteur de données de contexte	73
Figure 5.5. Histogramme des utilisations	76
Figure 5.6. Histogramme des utilisateurs interrogés	77
Figure 5.6. Précision, rappel et f-mesure	78
Figure 5.7. Matrice de confusion	80
Figure 5.9. Interface pour l'évaluation des objets pédagogiques	81
Figure 5.9. Interface pour la création des objets pédagogiques	81
Figure 5.10. Implémentation de l'ontologie	82
Figure 5.11. Implémentation des règles sémantique	83

Liste des tableaux

Table 4.1 règles de déduction	57
Table 5.1 Elément LOM proposés	67
Table 5.2 Exemple d'utilisation	72
Table 5.3 exemple de caractéristiques matériels utilisées	75
Table 5.4 un extrait de l'ensemble de données	76
Table 5.4 utilisateurs interrogés	76
Table 5.5 données des mesures	78
Table 5.6 Mesures de performance	78
Table 5.7 données d'entraînement	79

Chapitre 1

Introduction Générale

Introduction générale

1. Contexte du travail

La croissance massive des informations et de ressources disponibles librement nécessite un système de filtrage qui permet de surmonter la surcharge de l'utilisateur devant une quantité énorme d'informations. L'apparition des systèmes de recommandation (SR) a pour but de réduire de manière considérable la tâche de l'utilisateur consacrée pour la recherche des items (articles, ressources, produits, pages web, etc.) les plus intéressants pour lui, et aussi de trouver des items qui lui sont susceptibles d'être appréciés et sans lui demander. Ces dernières années, les systèmes de recommandation ont considérablement étendus pour couvrir plusieurs domaines tels que l'e-commerce, e-Learning, ...etc. Pour parvenir à un tel but, un SR a besoin d'accumuler des données sur les utilisateurs et les items disponibles. Il exploite les traces de ses utilisateurs afin d'appliquer des méthodes pour prédire leur futur comportement à l'égard de l'item. Cependant, avec l'apparition des appareils mobiles (smartphones, tablettes, ...etc.), ces systèmes doivent s'adapter progressivement aux environnements pervasifs afin de fournir des items pertinents aux utilisateurs convenables à leurs préférences en prenant en charge leurs situations et leurs contextes. Ces systèmes sont appelés par la suite systèmes de recommandation sensibles aux contextes (CARS) qui devient rapidement un nouveau défi pour les chercheurs du domaine de systèmes de recommandation.

Les systèmes de recommandation contextuels (CARS) visent à recommander des items tout en tenant compte du contexte de l'utilisateur. Comme le mot « contexte » fait référence à des aspects de l'emplacement, des activités ou des circonstances immédiates des utilisateurs, les utilisateurs dans des environnements ubiquitaires ont besoin de services et de recommandations adaptés à leurs contextes et à leurs préférences.

Les informations contextuelles peuvent être collectées par différentes manières, à savoir : explicitement, dont en posant des questions aux utilisateurs via des formulaires Web avant de donner l'accès aux applications. Implicitement, on utilise les données fournies par l'environnement de l'utilisateur telles que l'emplacement qui peut être déduit par différents moyens tel que le GPS ou l'heure qui peut être obtenue implicitement à partir du système lui-même. Enfin, déduites dont les informations contextuelles peuvent être inférées à l'aide de techniques du data mining et de méthodes statistiques appliquées. Il est important de noter que, si le processus d'acquisition des informations contextuelles est fait explicitement ou même

implicitement, il doit être mené dans le cadre du processus global de collecte de données. Cela implique en outre que les décisions concernant les informations contextuelles qui doivent être pertinentes et collectées pour une application, doivent être prises en compte dans la phase de conception de l'application et bien avant le moment où les recommandations réelles sont générées.

La prise en charge de différents paramètres contextuels par les systèmes de recommandation est une tâche difficile, car l'ajout de divers contextes environnementaux au processus de recommandation entraîne l'expansion de sa dimensionnalité et augmente la rareté (sparsity). Par conséquent, l'incorporation d'un petit ensemble de contextes explicites prédéfinis par les experts du domaine ou déduite implicitement par le système a montré l'utilité des recommandations générées par ce type de CARS [1] [2]. Dans la littérature, Il existe trois approches principales pour intégrer le contexte dans les systèmes de recommandation [3]: le pré-filtrage, le post-filtrage et la modélisation contextuelle. Alors que les approches pré-filtrage et post-filtrage filtrent les items recommandés avant ou après le calcul de la liste de recommandations et n'intègrent pas de contexte dans le modèle de recommandation, l'approche de modélisation contextuelle est préférée, car elle applique les informations contextuelles directement sans aucune intervention de l'utilisateur. D'autre part, elle s'avère moins efficace car il n'existe aucune garantie que les données utilisées reflètent le processus de recommandation.

Dans ce mémoire, nous avons proposé une approche permettant de déduire les conditions contextuelles (valeurs des facteurs contextuelles) en se basant sur l'analyse du comportement d'utilisateur au sein d'un système de recommandation à base de filtrage collaboratif. Dans ce travail, un cas d'étude a été présenté concernant le domaine de l'éducation, dont on a proposé une solution pour déduire automatiquement les valeurs de contexte (appropriées et inappropriées) relatives à la consommation adéquate de l'objet pédagogique. Ces données de contexte ont pour but d'améliorer le processus de recommandation en tenant en compte l'aspect contexte pour générer des futures recommandations. Un système de recommandation E-learning vise à prédire le matériel d'apprentissage approprié (cours, examens, ...etc.) à partir d'une grande quantité de données d'apprentissage disponible en regard des préférences des apprenants. Ce système a obtenu une grande attention de la part des chercheurs afin de résoudre les problèmes de la surcharge d'informations dans les supports d'apprentissage et de la personnalisation [4], [5].

2- Problématique

Plusieurs systèmes de recommandation actuels rencontrent certaines limites non seulement concernant le processus de recommandation mais également le processus d'acquisition des facteurs contextuels et ces valeurs (conditions contextuelles) afin de construire le profil explicite de l'utilisateur où les utilisateurs sont invités à exprimer leurs intérêts à travers la saisie des mots-clés ou des balises.

L'obtention des conditions contextuelles qui influencent la phase de consommation d'un item est exigeante en ressources, et elle nécessite dans la plupart des cas l'intervention d'un expert de domaine.

Naturellement, tous les facteurs contextuels disponibles peuvent ne pas être pertinents ou utiles pour une application spécifique. Certains types d'informations contextuelles peuvent être plus pertinents dans une application donnée que d'autres types. Par exemple, certaines applications considèrent le temps et l'emplacement, d'autres les caractéristiques matérielles, ...etc. La décision de choix doit être effectuée dans la phase de la conception de l'application.

Le contexte explicite (contexte extrait de façon explicite) peut être mieux expliqué par les experts humains et les utilisateurs que le contexte déduit, car il décrit des situations utilisateur et/ou items connues. L'utilisation d'un contexte explicite pour un service de recommandation peut soulever des problèmes de confidentialité, puisque le contexte exact de l'utilisateur est connu du service, ce qui n'est pas le cas pour un contexte déduit qui peut être obtenu automatiquement en appliquant des techniques d'apprentissage non supervisé sur les données brutes disponibles (par exemple, des capteurs mobiles). L'obtention d'un contexte explicite est une tâche exigeante en ressources et peut interférer avec les activités de l'utilisateur.

Par cette optique, et en vue de libérer l'utilisateur de la tâche de choisir les conditions contextuelles appropriées, nous avons proposé dans ce travail une approche basée sur la modélisation des utilisateurs ainsi que leurs contextes. Cette approche nous a permis de surmonter notre problématique et d'accomplir nos objectifs de déduire les données contextuelles appropriées ainsi que les données contextuelles inappropriées. Nous adaptons le contexte implicite pour déduire les valeurs contextuelles après un choix préalable des facteurs contextuels qu'on suppose que ces facteurs ont une influence majeure sur la consommation (consultation, utilisation) usuelle de nos items. Le choix des facteurs de contexte se fait suivant la base des items de notre application, par exemple, dans notre cas d'étude nous choisissons les facteurs qui ont une relation avec ce qu'est multimédia (taille d'écran, résolution, ...etc.). Par

la suite, nous essayons de déterminer les valeurs optimales des facteurs de contextes choisis, en se basant sur des règles sémantiques et un profil d'utilisateur. Notre approche repose sur des suppositions proposées extraites à partir des travaux existants dans le domaine du CARS [6].

3- Contribution

Dans cette thèse, nous décrivons un aperçu de la plupart des techniques qui ont été proposées pour la conception des systèmes de recommandation ainsi que l'impact de l'utilisation du contexte en examinant les contributions existantes.

Par conséquent, la proposition d'un ensemble de suppositions nous permet de construire une nouvelle base de connaissance qui est primordiale pour identifier les comportements des utilisateurs dans un environnement de recommandation. Cette taxonomie est utilisée à des fins d'identifier des causes et des circonstances qui ont motivé l'évaluation des items. Les suppositions utilisées sont issues à partir de différentes utilisations des items au sein d'un système de recommandation. Dans ce travail, nous avons mis au point une analyse simple de l'activité de l'utilisateur devant une quantité de ressources au sein d'un système de recommandation à base de filtrage collaboratif en exploitant la matrice des votes (rating matrix).

Ce travail a pour objectif de répondre aux deux questions suivantes : premièrement : parmi les informations contextuelles concernant un item, quelles sont ses valeurs qui permettent de consommer l'item dans une situation confortable (valeurs appropriées) ? Et quelles sont les valeurs déconseillées pour la consommation de l'item (valeurs inappropriées) ? Deuxièmement : comment peut-on exploiter ces valeurs pour générer des recommandations pertinentes ?

Pour répondre aux questions précédentes, nous avons conçu un système de recommandation à base de filtrage collaboratif afin d'exploiter la matrice des votes. Ce système est taillé selon nos besoins et basé sur le calcul de similarité entre les utilisateurs et aussi mesuré la correspondance entre les utilisateurs et les items.

Les principales contributions de ce travail sont :

- Proposition d'une base de connaissance des évaluations des items dans un système de recommandation à base de filtrage collaboratif.
- Proposition d'un profil d'utilisateur qui englobe à la fois les données des utilisateurs et les données des items ainsi que les données de contexte

- proposer une approche pour la mise en œuvre d'un processus de prédiction du contexte basé sur des règles sémantique
- Proposition d'un cas d'étude dédié pour le domaine de l'éducation dans un environnement ubiquitaire dont on a conçu un outil permettant la création et de l'évaluation des objets pédagogiques et qui s'inscrit dans le cadre de la productivité des objets d'apprentissage. ensuite, nous avons montré comment utiliser ces données extraites dont on a adopté le Pré-filtrage Contextuel en utilisant une méthode probabiliste.

4- Organisation du mémoire

L'organisation de ce mémoire est la suivante :

Le **chapitre 2** fera l'objet d'une présentation de l'état de l'art qui aborde les notions et les concepts des systèmes de recommandations et les systèmes de recommandations sensibles au contexte à savoir le contexte, l'adaptation et la prédiction.

Le **chapitre 3** présente une vue globale sur les recherches et les approches proposées pour construire et mise à jour d'un profil d'utilisateur dans le domaine de l'informatique omniprésente ainsi, nous étudions les différentes approches de modélisation permettant par la suite de proposer un profil d'utilisateur spécifique à nos besoins.

Le **chapitre 4** présente l'approche proposée qui répond à la problématique de la déduction automatique des données contextuelles. Une nouvelle taxonomie de classification est proposée pour les phases de sélection des utilisateurs ainsi que les données de contexte. Cette taxonomie a été extraite à partir des interactions des utilisateurs avec le système de recommandation. Suivant cette taxonomie, des règles sémantiques clés ont été proposées et discutées dans ce chapitre.

Le **chapitre 5** présente notre cas d'étude qui permet de valider notre approche, nous présentons la plateforme proposée dans le domaine de l'éducation en ligne. Nous montrons en détail les étapes de conception ainsi que les obstacles reconnus lors de la réalisation de ce prototype. Une évaluation a été menée pour valider notre système en offrant une étude sur le terrain.

Chapitre 2

Les systèmes de recommandation

*“We are leaving the age of information and entering the age of
recommendation “*

Chris Anderson in the long tail

Ce chapitre présente la plateforme générale de notre travail. Il est subdivisé en 2 parties. La première partie est consacrée à l'étude des concepts de base des systèmes de recommandation classiques, les techniques qu'ils utilisent ainsi que leurs limites et les différents types de systèmes de recommandation. Dans la deuxième section, nous présentons les systèmes de recommandation sensibles au contexte (CARS) ainsi que les différents travaux incorporant le contexte comme une dimension primordiale.

1. Introduction

Actuellement, et avec la croissance massive des informations et des ressources disponibles librement, le besoin d'un système de filtrage des informations devient une revendication primordiale. La naissance des systèmes de recommandation vise à surmonter ce problème. Plusieurs domaines ont été exploités ce genre de système afin de fournir des items pertinents aux utilisateurs de l'intérêt et évitent potentiellement les utilisateurs d'être perdus devant une quantité énorme d'information.

[7] Propose une catégorisation selon le domaine d'utilisation pour la majorité des systèmes de recommandation existents aujourd'hui

Divertissement : recommandations pour les films, la musique et l'IPTV.

Contenu : journaux personnalisés, recommandation de documents, recommandations de pages Web, applications d'apprentissage en ligne et filtres de messagerie.

Commerce électronique : recommandations aux consommateurs de produits à acheter tels que des livres, appareils photo, PC, etc.

Services : recommandations de services de voyage, recommandation d'experts pour consultation, recommandation de maisons à louer ou services de jumelage

Social : recommandations de tweets ou de messages, recommandation d'amis, recommandation de groupe d'utilisateurs.

Plusieurs techniques ont été proposées pour les systèmes de recommandation afin de garantir des recommandations adéquates. Au début, ces techniques et méthodes sont basées sur deux facteurs majeurs qui sont les utilisateurs et les items (les ressources). Ces systèmes ont été démontrés rapidement leurs insuffisances à cause de d'autres facteurs qui n'étaient pas pris en compte par ces techniques tel que le contexte de l'utilisateur et le contexte de l'item.

2. Historique

Bien que les systèmes de recommandation soient devenus un axe de recherche très important ces dernières décennies, plusieurs conférences et congrès traitant ce domaine ont été organisés comme *ACM SIGKDD* et *ACM RecSys* (Recommender Systems). L'amélioration des techniques des systèmes de recommandation est un sujet de recherche relativement récent par rapport aux systèmes de recherche d'informations qui remontent au début de l'informatique

Le terme "filtrage collaboratif" a été utilisé pour la première fois avec le système *tapestry* en 1992 [8], c'est la première apparition des systèmes de recommandation. Ce système pouvait générer et envoyer des recommandations aux utilisateurs par email. Cinq ans plus tard, [9] a publié leur papier intitulé "Recommender systems" dans lequel il annonce que le filtrage collaboratif n'est pas la seule approche pour faire de la recommandation, il cite et compare plusieurs systèmes de recommandations de l'époque comme Fab [10] et PHOAKS [11]. La plupart des chercheurs précédents ont proposé des classifications des approches de la recommandation existant.

Selon *wikipedia*¹ l'historique des Systèmes de recommandations sont classifiés en trois générations. La première génération de ces systèmes a été mise au point par la création des systèmes de filtrage basés sur la logique *item to item* où les produits sont placés en relation avec d'autres produits au travers de leurs caractéristiques. Ensuite, ils se sont diversifiés, afin d'être appliqués à différents domaines. Le principe des premiers systèmes est basé sur le calcul de similarité entre les éléments (*items*) en utilisant des méthodes mathématiques.

La deuxième génération repose sur le filtrage collaboratif qui à partir des produits d'autres consommateurs déclarés, à leur insu, comme correspondants à leurs préférences. La recommandation est basée sur l'approche : "*user-to-user*" et met en relation des utilisateurs selon leurs profils et leurs préférences. Les données collectées sont utilisées pour reconstituer le profil du consommateur et vont alors mettre à jour son profil. Ce type de système de recommandation a émergé avec le web 2.0 qui a apporté une nouvelle culture sociale : le profil d'un consommateur est reconstitué au travers de profils similaires.

Troisième génération : repose sur le filtrage hybride qui vient croiser les deux générations précédentes (contenu + collaboratif). Ce troisième type de système de recommandation s'ancre dans une société où les traces numériques de l'utilisateur sont toujours plus présentes et plus nombreuses. Les innovations des NTIC au sein de la société entraînent une nouvelle conception

¹ www.wikipedia.org

et des nouveaux usages des agents de recommandation avec l'augmentation du stockage, les nouveaux dispositifs économiques et de nouveaux traitements des données. Cela s'ancre notamment dans le mouvement du Big data.

3. Les systèmes de recommandation classiques (SR)

Bien que notre travail se focalise sur le problème d'analyse du comportement des utilisateurs afin d'extraire les données contextuelles au sein d'un système de recommandation. Deux approches sont principalement connues dans cette classe : la recommandation à base de contenu [12] et le filtrage collaboratif à base d'items [13].

Les systèmes de recommandation (SR) [7] sont des outils qui tentent de prédire de façon automatique pour un utilisateur l'intérêt d'un item donné dont l'objectif est de proposer des items pertinents à l'utilisateur. Les items dans ce contexte peuvent être un produit à acheter, un film à regarder, un livre à lire, une page web à consulter, ou bien autre chose.

Un SR est basé toujours sur les préférences et les besoins de chaque utilisateur. Il tente tout d'abord d'acquérir les informations nécessaires et pertinentes pour construire des modèles d'utilisateurs. En particulier, les SR exploitent les traces et analysent les comportements laissés par les utilisateurs eux-mêmes.

Les traces (explicitement ou implicitement) laissées par les utilisateurs sont considérées comme base de traitement dont les SR opèrent ses recommandations. Les traces explicites sont fournies directement par l'utilisateur [7]. Les traces implicites sont déduites à partir des actions des utilisateurs pendant ses interactions avec le système [14], [15].

4. Techniques et méthodes utilisées pour les systèmes de recommandation

On peut classer les techniques et les méthodes utilisées pour les systèmes de recommandation selon le type de données exploitées et la manière dont est effectué le filtrage des items.

La littérature distingue généralement trois types de systèmes de recommandation, les systèmes basés sur le contenu, les systèmes à base sociale dits de filtrage collaboratif et les systèmes hybrides [16]. Nous présentons dans cette section les principales techniques de recommandation avec leurs apports et leurs limites

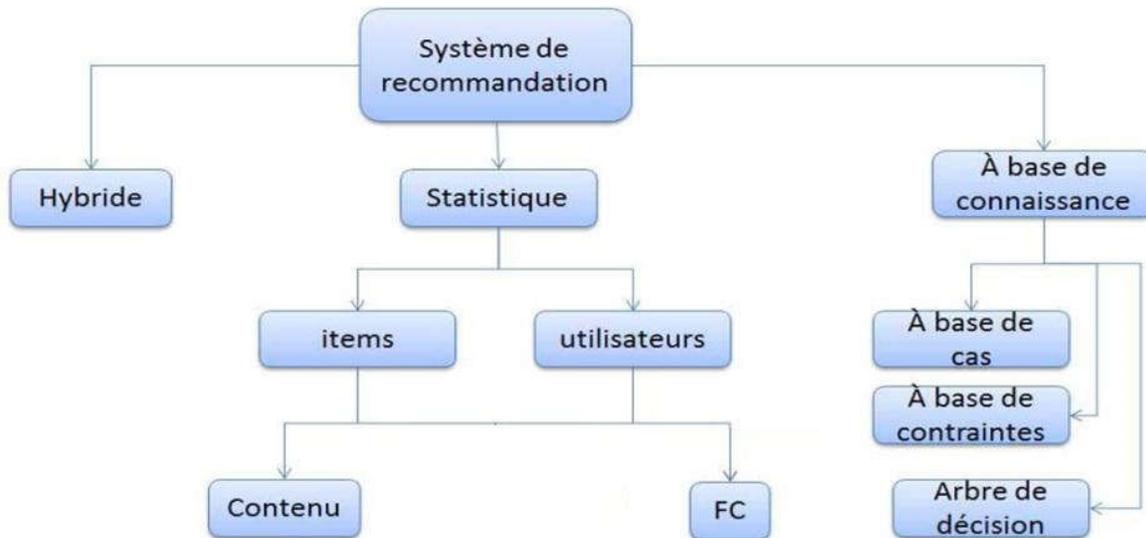


Figure 2.1. Techniques et méthodes pour les SR

4.1. Systèmes de recommandation à base de contenu (CBF)

La recommandation basée sur le contenu consiste à analyser le contenu des items candidats à la recommandation ou les descriptions de ces items. Dans cette approche, chaque item (on désigne par item tous objets peut être consommable par un utilisateur) est défini par un ensemble d'attributs et leur valeur. Les items ayant des valeurs proches pour leurs attributs sont considérés comme voisins (similaires).

Les systèmes basés sur le contenu recommandent les items similaires à ceux que l'utilisateur a aimés dans le passé en se basant sur une description de chacun d'eux. La description des items se fait en fonction du système qui est souvent influencé par le type d'items recommandés. Par exemple, IMDB² propose de décrire un film à partir de certaines informations comme l'auteur, le genre, les acteurs, etc. par exemple si un utilisateur a noté positivement un film de genre comique le système peut alors apprendre à recommander d'autres films de ce genre. Pour les items de type texte, comme des articles scientifiques, une représentation à l'aide de mots-clés contenus dans le texte est souvent utilisée. Les descriptions d'items sont comparées avec les descriptions d'utilisateur, ou profils. Cette comparaison permet d'évaluer si l'item correspond au besoin de l'utilisateur. Le profil est généralement vu comme un item idéal correspondant parfaitement aux besoins ou goûts de l'utilisateur.

Les profils utilisateurs sont, soit définis manuellement par l'utilisateur [17], soit appris en fonction de son comportement.

² <http://www.imdb.com/>

4.1.1. Algorithmes de recommandations

4.1.1.1. Recommandations à base de mots-clés pondérés

La plupart des systèmes de recommandation utilisent le modèle d'espace vectoriel de base TF-IDF pondéré (term frequency inverse document frequency). Ce modèle est une représentation de document sémantique introduite par Salton en 1975 [18]. Un document est représenté par un vecteur pondéré de termes où les poids indiquent le degré d'association entre le document et ses termes. En d'autres termes, les mots qui apparaissent fréquemment dans un document mais rarement dans le reste du corps sont plus susceptibles de représenter le sujet du document. De plus, la normalisation compense le fait que les documents longs ont plus de chances d'être trouvés. Nous détaillons cette notion dans le chapitre 5.

$$w_{ij} = \mathit{tfidf}(t_i, d_j) = \mathit{tf}(t_i, d_j) * \log \frac{|D|}{\mathit{tf}(t_i, D)}$$

D représente les documents du corpus: $D = d_1, d_2, \dots, d_n$

L'ensemble des termes du corpus: $T = t_1, t_2, \dots, t_i$

Dans les systèmes à base de contenu [19], le profil utilisateur et les items sont représentés sous forme de vecteurs pondérés. L'utilité d'un item pour un utilisateur donné peut être calculée par la similitude cosinus entre le vecteur de profil utilisateur et le vecteur d'item. Notre travail utilise cet algorithme pour calculer la correspondance entre les utilisateurs et les items afin de construire une liste des items pertinents pour chaque utilisateur du système.

4.1.1.2. Recommandations sémantiques

Les systèmes de recommandation basés sur la sémantique s'appuient sur des méthodes qui utilisent la sémantique des items traités. SiteIF [20] a été le premier système à utiliser la signification des documents pour construire le modèle d'intérêts de l'utilisateur. Le système utilise MultiWordNet³ qui est une base de données lexicale multilingue pour associer une liste de synonymes (appelés *synsets*) pour des articles de presse. Le profil utilisateur est un réseau sémantique où les nœuds représentent les *synsets* extraits des articles de presse lus par l'utilisateur. Une phase d'appariement est alors conduite dans laquelle le système reçoit en entrée les *synsets* de document et le modèle utilisateur, et produit une estimation de la pertinence du document.

³ <http://multiwordnet.fbk.eu/english/home.php>

4.1.1.3. Approche générale

La recommandation des items à base d'analyse du contenu repose sur deux ensembles élémentaires : 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. L'approche est basée sur l'analyse du contenu en comparant avec d'autres contenus précédemment notés ou consultés par un utilisateur. Le système construit donc un modèle ou un profil des intérêts de l'utilisateur sur la base des caractéristiques des items aimés ou notés par celui-ci. En fonction de ses opinions, le profil de l'utilisateur sera mis à jour périodiquement en ajoutant les caractéristiques des items qu'il a aimés et 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 l'utilisateur.

Plus le profil de l'utilisateur est enrichi, plus le système de recommandation devient efficace. [21] proposent une architecture de haut niveau (Cf. figure 2) dans laquelle le processus de recommandation est réalisé en trois étapes, chacune étant gérée par un composant spécifique :

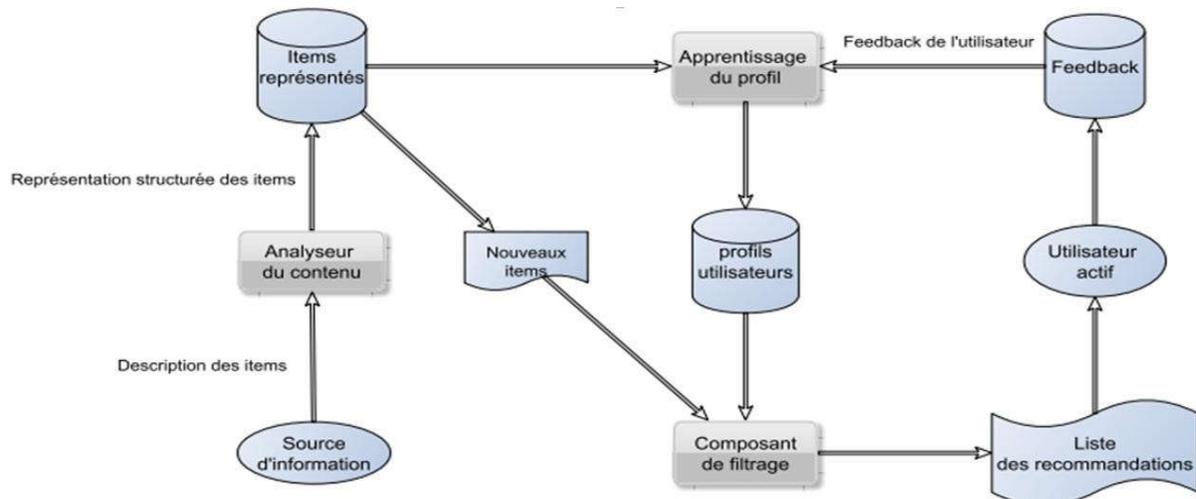


Figure 2.2. Schéma SR à base de contenu [21]

Les algorithmes de recommandation basés sur le contenu construisent un profil d'utilisateur basé sur les caractéristiques des objets évalués par l'utilisateur, qui sont supposés refléter les intérêts de l'utilisateur basés sur le contenu [21]. En général, la technique CBF est fondée sur les données sous-jacentes, généralement basée sur des techniques d'apprentissage automatique [21,22], ou utilise une fonction heuristique pour calculer les scores des items, principalement inspirée des méthodes de recherche d'information [23], [24].

Plusieurs méthodes probabilistes ont été utilisées, l'approche naïve de Bayes est l'une des méthodes la plus utilisées dans les algorithmes CBF dont elle génère un modèle probabiliste basé sur des données précédemment observées. Le modèle naïf de Bayes estime la probabilité qu'un événement se produise en fonction de conditions que nous connaissons (a priori) pour les événements en question. Le nom vient du théorème de Bayes, qui peut être écrit mathématiquement comme suit :

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

Avec X et Y sont les événements et P(X) supérieure strictement à 0.

$P(X|Y)$ est une probabilité conditionnelle. Plus précisément, c'est la probabilité que l'événement X se produise sachant que l'événement Y s'est déjà produit.

$P(Y|X)$ est une probabilité conditionnelle. Plus précisément, c'est la probabilité que l'événement Y se produise sachant que l'événement X s'est déjà produit.

P(X) et P(Y) sont les probabilités des événements X et Y indépendamment les uns des autres.

Dans le processus de recommandation, la méthode naïve de Bayes est utilisée pour estimer la probabilité qu'un document (un item) soit pertinent ou non (classe), sur la base des informations disponibles pour chaque utilisateur, c'est-à-dire que les documents déjà notés sont utilisés pour calculer les probabilités. Cette approche a été utilisée par de nombreux auteurs différents tels que [21], [22].

D'autres méthodes alternatives comme les arbres de décision et les réseaux de neurones [25] ont été utilisées pour classer les éléments d'un système comme pertinents ou non pour chaque utilisateur du système. Ces techniques, tout comme la méthode naïve de Bayes, estiment dans quelle classe chaque élément (non évalué ou non observé) est appartient et donc détermine le mieux au profil de l'utilisateur.

D'autres techniques issues des méthodes de recherche d'informations sont utilisées pour spécifier la représentation des utilisateurs et les éléments ainsi que la fonction de similitude utilisée entre eux. En général, ils utilisent un modèle d'espace vectoriel où chaque entité est pondérée d'une manière particulière. Parmi ces méthodes on trouve TF-IDF et BM25 (qui est une mise à niveau de TF-IDF) [23]. De plus, plusieurs espaces de caractéristiques différents ont été considérés dans la littérature: mots-clés [25], tags [24], et concepts sémantiques enrichis par différentes techniques [26], [27]

Dans les systèmes de recommandation, les éléments sont suggérés selon l'ordre décroissant de similitude avec l'utilisateur, dont le profil est représenté sous la même forme des items (vectorielle). Les similitudes sont calculées comme la similarité des vecteurs de caractéristiques entre chaque item (non évalué ou non observé) de la collection et le vecteur de l'utilisateur.

4.2. Système de recommandations à base de filtrage collaboratif (FC)

Le filtrage collaboratif (CF) est un algorithme de recommandation populaire basé sur les notes (appelées aussi votes) et/ou le comportement des autres utilisateurs du système afin de générer des prédictions et des recommandations. L'hypothèse fondamentale derrière cette méthode est que les opinions (en générale des notes fournies par les utilisateurs) des autres utilisateurs peuvent être sélectionnées et agrégées de manière à fournir une prédiction convenable à la préférence de l'utilisateur actif. Intuitivement, ils supposent que, si les utilisateurs sont satisfait sur la qualité ou la pertinence de certains éléments, ils seront probablement satisfait sur d'autres éléments par exemple si un groupe d'utilisateurs aime les mêmes choses que X, alors X est susceptible d'aimer les choses (qu'il n'a pas encore vu) que le groupe les aime déjà.

Aujourd'hui, La majorité des algorithmes de filtrage collaboratif, y compris tous les algorithmes détaillés dans cette section, fonctionnent en générant d'abord des prédictions concernant les préférences de l'utilisateur, ensuite produisent des recommandations en classant les éléments candidats en fonction des préférences prédites. Souvent, cette prédiction est à la même échelle que les notes fournies par les utilisateurs. Cette stratégie est similaire à la méthode courante de recherche d'informations qui consiste à produire des scores de pertinence pour chaque document (item) d'un corpus par rapport à une requête particulière et à présenter les éléments les mieux notés. En effet, la tâche de recommandation peut être considérée comme un problème de recherche d'informations dans lequel le domaine des éléments (le corpus) est interrogé avec le profil de préférence de l'utilisateur.

4.2.1. Approche générale

Contrairement à CBF, les méthodes CF visent à prédire l'utilité des éléments pour un utilisateur particulier en fonction des éléments précédemment évalués ou notés par d'autres utilisateurs partageant les mêmes idées (un groupe d'utilisateurs). Ces méthodes ont une propriété intéressante qu'aucune description d'élément n'est nécessaire pour fournir des recommandations, car ces méthodes exploitent simplement les informations sur les évaluations passées. Comparée aux approches CBF, la méthode CF a également l'avantage majeur qu'un

utilisateur peut bénéficier de l'expérience d'autres personnes, et peut être proposé des recommandations potentiellement nouvelles au-delà de sa propre expérience [16].

Pour être plus formel, une note (évaluation d'un élément) consiste en l'association de deux choses l'utilisateur et l'élément souvent au moyen d'une certaine valeur. La visualisation des notes est présentée sous forme d'une matrice (rating matrix). Nous désignons la valeur de base pour l'utilisateur U et l'élément I par $V_{u,i}$

Sans perte de généralité, une matrice de notation se compose d'un tableau dans lequel chaque ligne représente un utilisateur, chaque colonne représente un élément spécifique et le nombre à l'intersection d'une ligne et d'une colonne représente la valeur de notation de l'utilisateur. L'absence de note à cette intersection indique que l'utilisateur n'a pas encore évalué l'élément

	Élément 1	Élément 2	Élément 3	Élément 4	Élément 5	Élément 6
Utilisateur 1	$V_{1,1}$			$V_{1,4}$		
Utilisateur 2		$V_{2,2}$	$V_{2,3}$			
Utilisateur 3						$V_{3,6}$

Table 2.1 exemple de la matrice des votes (notes)

Le principe du FC se base sur une simple matrice V de notes de taille $n \times m$; où n est le nombre total d'utilisateurs U , et m est le nombre total d'items A . L'élément V_{ij} de la matrice représente la note attribuée par l'utilisateur u_i à l'item a_j . S'il se trouve que u_i n'a pas attribué une note à A_j , la valeur V_{ij} est considérée comme valeur manquante ($V_{ij} = \text{nul}$). Donc l'objectif du FC est de trouver les valeurs manquantes de la matrice.

L'utilisation de cette matrice pour les systèmes à base de filtrage collaboratif peut être divisé en deux catégories [28] filtrage collaboratif à base de mémoire et filtrage collaboratif à base du modèle

4.2.1.1. Techniques basés sur la mémoire

Pour calculer les valeurs manquantes, le FC utilise les valeurs existantes dans la matrice afin de prédire une valeur V_{ij} , il se sert des voisins de U_i qui ont déjà noté A_j . Pour les extraire, il mesure la similarité entre U_i et chacun des utilisateurs ayant noté l'item A_j

4.2.1.1.1. Génération des prédictions

Pour construire les prédictions ou les recommandations possibles pour un utilisateur U, le système combine les notes des utilisateurs voisins pour générer des prédictions concernant les préférences de l'utilisateur cible U pour un élément i. Généralement, la liste des voisins de l'utilisateur est générée en utilisant les méthodes de calcul de similarité pour générer une liste pondérée.

Le processus de prédiction utilise la différence entre la note attribuée à a_j par chaque voisin de u_i , et la moyenne de notes du voisin. Cette différence est multipliée par la similarité entre u_i et le voisin. La moyenne de valeurs obtenues est par tous les voisins est ajoutée à la note moyenne de u_i pour obtenir la note prédite $v(u_i, a_j)$. La fonction de prédiction est donnée par l'équation suivante [29]

$$v(u_i, a_j) = \underline{r} + \frac{\sum_n \text{sim}(u, u')(v(u_n, a_j) - \underline{r}_n)}{\sum_w \text{sim}(u, u')}$$

Où n est le nombre total de voisins. \underline{r} Représente la note moyenne de toutes les notes accordées par l'utilisateur cible U pour tous les autres éléments notés. $\text{sim}(u, u')$ est la similitude de l'utilisateur cible u et de l'utilisateur voisin u' .

Les mesures de similarité (appelée également métrique de similarité) sont considérées comme une clé majeur des systèmes de recommandation [30]. L'auteur dans [31] a été l'un des premiers à introduire le coefficient de corrélation de **Pearson** pour calculer la mesure de similarité entre deux entités. D'autres mesures de similitude telles que **Jaccard** et **Cosine** ont été proposées dans de nombreux travaux [31] [32].

Généralement, ces méthodes sont utilisées pour calculer les scores qui expriment à quel point les utilisateurs ou les éléments sont similaires entre eux. Ces scores peuvent être ensuite utilisés comme base de la génération de recommandations par utilisateur ou par élément. Selon le contexte d'utilisation, les métriques de similarité peuvent également être appelées métriques de corrélation ou mesures de distance. Par la suite, nous exposons les mesures de similarités qu'on a utilisées dans notre travail et aussi considérées comme les plus utilisées dans les systèmes de recommandations

Pearson corrélation

La méthode de corrélation de Pearson [78] est la méthode la plus courante à utiliser pour les variables numériques; il attribue une valeur comprise entre - 1 et 1, où 0 indique qu'il n'existe

aucune corrélation, la valeur 1 indique une corrélation positive totale et - 1 une corrélation négative totale. Ceci est interprété comme suit: par exemple, si une valeur de corrélation entre deux variables A et B est 0.5, cela explique qu'il existe une relation significative et positive entre les deux variables. Par exemple, une corrélation positive signifie que si la variable A augmente, alors B augmentera également, tandis que si la valeur de la corrélation est négative, alors si A augmente, B diminue.

Dans les systèmes de recommandations, la méthode de Pearson est utilisée pour mesurer la similarité entre deux utilisateurs x et v en calculant 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 suivante :

$$sim(x, v) = \frac{\sum_i^N (r_{x,i} - \bar{r}_x)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_i^N (r_{x,i} - \bar{r}_x)^2} \sqrt{\sum_i^N (r_{v,i} - \bar{r}_v)^2}}$$

Où N ($N = \text{item}_x \cap \text{item}_v$) représente les items notés à la fois par x et v . $r_{x,i}$ est l'ensemble des items notés par l'utilisateur x et \bar{r}_x est la note moyenne de l'utilisateur x .

Cosinus (Cosine similarity)

Dans la littérature, la similarité cosinus [34] est fréquemment utilisée afin de déterminer la similitude entre deux documents. Les documents sont représentés par des vecteurs et La similarité cosinus permet de calculer la similarité entre deux vecteurs à n dimensions en déterminant le cosinus de l'angle entre eux, la plage de la similitude résultante est comprise entre 0 et +1.

Dans le cas du filtrage collaboratif, chaque objet O (utilisateur ou item) est représenté par un vecteur. Pour pouvoir calculer la similarité entre deux objets A et B , le cosinus est calculé sur l'ensemble des items notés par les deux utilisateurs comme l'illustre la formule suivante :

$$Similarity = \frac{A \cdot B}{\|A\| \cdot \|B\|} = \frac{\sum_{i=1}^n A_i \cdot B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \sqrt{\sum_{i=1}^n (B_i)^2}}$$

Dans les systèmes de recommandations, cette mesure varie toujours 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.

4.2.1.2. Techniques basés sur les modèles

Cette technique utilise la matrice des évaluations précédentes pour construire un modèle. Le processus de création d'un modèle est effectué à l'aide de techniques d'apprentissage automatique ou d'exploration de données qui peuvent rapidement recommander un ensemble d'éléments similaires aux techniques de recommandation basées sur la mémoire. Ces techniques sont généralement incluant la technique de réduction de dimensionnalité telle que la décomposition en valeur singulière (SVD), la complétion de matrice, méthodes probabilistes, ...etc. Les techniques basées sur des modèles analysent la matrice éléments-utilisateur pour identifier les relations entre les éléments; ils utilisent ces relations pour comparer la liste des N premières recommandations. Les techniques basées sur des modèles résolvent les problèmes de rareté associés aux systèmes de recommandation.

L'utilisation d'algorithmes d'apprentissage a également changé la manière de recommandations de quoi consommer en quand consommer réellement un produit. Il est donc important de prendre un vue d'ensemble sur les algorithmes d'apprentissage utilisés dans les systèmes de recommandation basés sur des modèles:

Techniques de complétion de matrice: La plupart du temps, la matrice de notation est toujours très grande et clairsemée en raison du fait que les utilisateurs n'évaluent pas tous les éléments représentés dans la matrice. Ce problème conduit toujours à l'incapacité du système à donner des recommandations fiables et précises aux utilisateurs. Donc l'objectif de cette technique est de prédire les valeurs inconnues et manquantes dans les matrices éléments-utilisateur. Différentes variantes de modèles ont été utilisées dans la pratique pour l'achèvement de la matrice, en particulier pour l'application au filtrage collaboratif.

Modèles probabilistes: Le principe de ces techniques consiste à modéliser les évaluations des utilisateurs par un modèle probabiliste pour pouvoir prédire ses évaluations futurs. L'idée sur laquelle repose ces algorithmes consiste à calculer la probabilité $P(V_U/E)$ que l'utilisateur U attribue l'évaluation V à l'élément E connaissant ses évaluations antérieures.

Plusieurs modèles probabilistes issus des domaines de recherche sur l'intelligence artificielle et sur l'apprentissage automatique ont été appliqués au filtrage collaboratif. Certains ont défini le problème de recommandation comme étant un problème de classification [29]. [29] est l'un des travaux appliquant un modèle probabiliste dans les algorithmes de filtrage collaboratif proposant un modèle construit à partir des réseaux Bayésiens et utilisant les arbres de décision pour calculer les probabilités. PLSA Probabilistic latent semantic analysis [37], est une

technique de factorisation similaire à la décomposition en valeur singulière (SVD) mais provenant de la théorie des probabilités au lieu de la théorie de l'algèbre linéaire [35];

D'autres approches ont été appliquées des méthodes probabilistes pour réduire la dimension ou faire la classification [36]. Les techniques de réduction de la dimension utilisent des méthodes de factorisation de matrice probabiliste telle que le SVD.

4.3. Les approches hybrides

La prolifération de nouvelles stratégies de recommandation donne lieu à une variété croissante d'options disponibles pour le développement de systèmes de recommandation. La recherche en apprentissage automatique a montré depuis longtemps que la combinaison de méthodes donne généralement de meilleurs résultats que chaque méthode séparément, ce qui est également vrai dans les systèmes de recommandation.

Dans les approches hybrides, la décision la plus importante est de savoir comment combiner les approches existantes. Cependant, il faut tout d'abord décider le type d'informations qui sera utilisé dans l'ensemble. L'approche standard dans la littérature consiste à combiner les approches CBF et CF, en surmontant les problèmes de rareté et de fonctionnalités restreintes des systèmes de recommandation. Ainsi que d'autres types et sources d'informations comme les contacts sociaux ont été récemment intégrés dans la formulation des techniques de recommandation standard.

Dans [38], une taxonomie détaillée des systèmes de recommandation hybrides est présentée. Ce travail classe les approches hybrides existantes dans des types dont nous citons les suivants :

- **Cascade**: la recommandation est effectuée comme un processus séquentiel de telle manière qu'un système affine les recommandations données par l'autre.
- **Augmentation des fonctionnalités**: le résultat d'un générateur de recommandation est utilisé comme fonction d'entrée supplémentaire pour un autre système.
- **Combinaison de fonctionnalités**: les fonctionnalités utilisées par différents systèmes sont intégrées et combinées dans une seule source de données, exploitée par un seul système.
- **Mixte**: les recommandations de plusieurs systèmes sont disponibles et sont présentées se forme d'un ensemble au moyen d'une certaine stratégie de classement ou de combinaison.
- **Pondérée**: les scores fournis par les systèmes de recommandation sont agrégés à l'aide d'une combinaison linéaire ou d'un système de vote.

- **Commutation:** un cas particulier des types précédents qui considère les poids binaires, de telle sorte qu'un système de recommandation est activé et les autres sont désactivés.

L'utilisation d'un type spécifique de méthode de recommandation hybride dépend de l'application finale. En effet, [38] présente une analyse des hybrides possibles et leurs limites et incompatibilités à partir d'un sous-ensemble représentatif des techniques de recommandation disponibles. De plus, l'auteur note que certaines combinaisons se révèlent redondantes en raison de la symétrie dans le processus d'hybridation pour certaines des techniques énumérées ci-dessus.

5. Collecte de données

Les systèmes de recommandation sont des systèmes de traitement de l'information qui collectent activement divers types de données afin de formuler leurs recommandations. Les données concernent principalement les éléments à recommander et les utilisateurs qui recevront ces recommandations. Les sources des données et de connaissances disponibles pour les systèmes de recommandation peuvent être très diverses. La collecte de ces données peut se faire, soit explicitement (fournie directement par l'utilisateur), soit implicitement c'est-à-dire en analysant les traces laissées à la suite de ses interactions avec le système. Les données explicites sont des préférences que l'utilisateur a explicitement indiquées pour des éléments particuliers. Les données implicites sont déduites par le système en observant l'activité des utilisateurs comme par exemple les achats ou les clics.

En général, Les données utilisées par les systèmes de recommandation font référence à trois types d'objets: les éléments (les items), les utilisateurs et les interactions, c'est-à-dire les relations entre les utilisateurs et les éléments.

5.1. Données explicites

Ce type de données nécessite un travail supplémentaire de la part de l'utilisateur, il concerne les descriptions fournies par les utilisateurs dont il offre une description détaillée sur les préférences d'un utilisateur pour un élément. Cependant, les données explicites peuvent être difficiles à collecter (les évaluations), en particulier lors de la création d'un nouvel élément. D'autre part, les données concernant les utilisateurs d'un système de recommandation peuvent avoir des objectifs et des caractéristiques très divers. Afin de personnaliser les recommandations, les systèmes de recommandations exploitent une gamme d'informations sur l'utilisateur tel que les données démographiques. Ces informations peuvent être structurées de différentes manières et sa modélisation dépend de la technique de recommandation.

5.2. Données implicites

C'est les données qu'on les déduit automatiquement par le système en surveillant les différentes actions des utilisateurs telles que l'historique des achats, l'historique de navigation et le temps passé sur certaines pages Web, liens suivis par l'utilisateur, contenu de l'e-mail ...etc. Les données implicites facilitent la tâche des utilisateurs en déduisant les préférences de chaque utilisateur à partir de leur interaction avec le système. La méthode ne nécessite cependant aucun effort de la part de l'utilisateur, mais elle est moins précise. En outre, [39] a également noté que les données de préférence implicites (déduites) pourraient en réalité être plus objectives.

L'avantage des données implicites est qu'on les obtient en analysant le comportement (les actions) de l'utilisateur dans le système et les interactions avec les objets du système sans lui demander explicitement de noter des éléments ou de renseigner ses préférences. L'inconvénient est qu'il est difficile de déterminer le déteste de l'utilisateur ou encore de différencier les éléments qu'il consulte pour sa propre consommation des items qu'il consulte pour la consommation d'une personne tierce.

Plusieurs paramètres entre en jeu dans l'analyse du comportement de l'utilisateur tel que le temps passé à consulter l'élément et encore les endroits où se trouve l'utilisateur comme le cas de notre approche. [40] a proposé une formule permettant calculer la note implicite de l'élément i dans le contexte de recommandation des pages web:

$$note(i) = 1 + 2 * favoris(i) + 2 * recence(i) * duree(i) + liens_visite(i)$$

Tel que

$Favoris(i)$ est une valeur indiquant que l'utilisateur a ajouté l'élément i a ses favoris, sa valeur peut être de 0 ou de 1.

$recence(i)$ détermine si l'élément i a été récemment consulté

$frequence(i)$ indique la fréquence de consultation de l'élément i

$duree(i)$ représente la durée normalisée du temps de consultation de l'élément i

$Liens_Visites(i)$ correspond au nombre de liens visités divisé par le nombre de liens se trouvant sur la page de l'élément.

6. Verrous scientifiques

L'utilisation des techniques de filtrage collaboratif peut entraîner plusieurs problèmes mais l'avantage primordial de ces systèmes réside dans le fait que ces systèmes ne requièrent aucune

connaissance sur les éléments (indépendance du domaine). Ces méthodes peuvent recommander des éléments en se basant uniquement sur les notes associées aux items sans avoir des connaissances préalables sur le domaine.

Nous citons par la suite les obstacles les plus reconnus dans les systèmes de recommandation

6.1. Démarrage à froid (cold-start)

Il s'agit d'une situation dans laquelle un système de recommandation ne dispose pas assez d'informations adéquates sur un utilisateur ou sur un élément afin de faire des prédictions pertinentes. C'est l'un des problèmes majeurs qui réduisent les performances d'un système de recommandation. Ce problème concerne à la fois les nouveaux utilisateurs et les nouveaux items qui sont introduits dans le système. Un nouvel utilisateur qui n'a pas noté aucun élément ne peut pas recevoir de recommandation puisque le système ne connaît pas ses goûts. Par conséquent ce problème se pose aussi lors de l'ajout d'un nouvel élément. Celui-ci ne peut pas être recommandé avant d'avoir été noté par un certain nombre d'utilisateurs.

Une solution à ce problème consiste à recommander au départ les items les plus populaires ou de lui demander explicitement de noter un certain nombre d'éléments.

6.2. Problème de rareté des données (sparsity)

Ce problème se produit lorsque il y a un manque d'informations, c'est-à-dire lorsque seuls quelques éléments disponibles dans une base de données sont notés par les utilisateurs. Cela produit toujours une matrice d'éléments-utilisateur clairsemée, ce qui résulte à une génération de recommandations. En outre, le manque des données conduit toujours à des problèmes de couverture, c'est-à-dire le système est incapable d'identifier un nombre significatif de voisins, ce qui se répercute sur la qualité des recommandations proposées à l'utilisateur actif et sur la performance de la totalité du système.

6.3. Évolutivité des données (scalability)

Ce problème concerne les systèmes de grand échelle alors que la matrice de données d'éléments-utilisateurs de millions d'utilisateurs et des millions d'éléments est requis un temps important pour faire ses calcul (cas du l'algorithme du plus proche voisin), une évolutivité élevée du calcul entraîne un système de recommandation de ne pas réagir rapidement aux exigences en ligne et de faire immédiatement des recommandations.

6.4. Robustesse

Ce problème est relatif à la confiance des utilisateurs utilisant les systèmes de recommandation, ces derniers demeurent vulnérables aux manipulations et aux données bruitées. En effet, il n'y a pas de garantie que les données intégrées aux systèmes de recommandation reflètent les réelles appréciations des utilisateurs.

7. Métriques d'évaluation des systèmes de recommandations

L'évaluation d'un système de recommandation consiste à mesurer la capacité du système à répondre aux objectifs qui lui ont été fixés et la satisfaction des utilisateurs en termes des objets recommandés par le système. Dans le cadre de systèmes de recommandations, plusieurs métriques ont été étudiées dans de nombreux travaux dont [41] proposent une synthèse de méthodes et de stratégies d'évaluation du point de vue utilisateur, prédictions, types de données utilisées, ... etc. Cependant, pour évaluer un système, il est nécessaire de tester ce système en situation réelle c.-à-d. en présence de vrais utilisateurs. Cette forme d'évaluation est appelée évaluation en ligne ou parfois appelée en présence [42]. Ce type d'évaluation peut être très coûteuse et parfois même impossible, une alternative plus simple et plus répandue dans l'évaluation des systèmes de recommandation est l'évaluation offline, ce dernier nécessite un jeu de données tel que les votes des utilisateurs du système.

7.1. Évaluation en ligne

L'évaluation en ligne consiste à tester l'application sur de vrais utilisateurs en temps réel, cette évaluation mesure la vraie performance du système et permet de choisir l'alternative ayant mieux rempli les objectifs du système. Cependant, ce type est très coûteux et prend beaucoup de temps à réaliser. De ce fait, ce genre d'évaluation doit être utilisé après avoir effectué assez de tests hors-ligne permettant au préalable de filtrer les utilisateurs potentiels.

7.2. Evaluation sur un échantillon

Évaluer sur un échantillon consiste à choisir un groupe d'utilisateurs, auxquels on demande d'effectuer des missions bien précises tout en surveillant et enregistrant leurs comportements durant l'application. Ensuite, on peut aussi poser des questions aux participants concernant leurs déroulements sur l'expérimentation et le système de recommandation en général.

Ce type permet d'observer le comportement d'un utilisateur au cours de son interaction avec le système. Ce type d'évaluation est coûteux car il n'est pas toujours facile de recruter un nombre suffisant d'utilisateurs. Le nombre de participants est souvent limité, dont on ne peut

pas déduire les opinions de l'ensemble d'utilisateurs concernant les éléments évalués. Dans ce mémoire nous avons utilisé ce test afin d'évaluer l'utilisabilité du prototype ainsi que la satisfaction des utilisateurs en termes des données contextuelles dont on a basé sur un questionnaire élaboré à des fins précises, cela nous a permis de collecter des données qualitatives issues de différents usages. Ce questionnaire a été déployé sur un ensemble des étudiants ayant des caractéristiques contextuelles différentes.

7.3 Évaluations hors-ligne

Cette approche est préférée par rapport à l'approche précédente en raison de son faible coût ainsi que sa rapidité. Elle consiste à simuler le comportement de l'utilisateur afin de mesurer l'impact de son intervention. Cette simulation nécessite la disposition des données pouvant aider à reproduire le comportement des utilisateurs. Ces données contiennent les votes attribués par des utilisateurs pour certains éléments.

Généralement et suivant les données disponibles, l'évaluation hors ligne subdivise le corpus de données en deux parties : la partie test et la partie apprentissage. La partie test est utilisée pour valider la partie apprentissage.

Plusieurs métriques ont été utilisées pour l'évaluation hors-ligne [43]. Les mesures les plus populaires pour mesurer la précision d'un système de recommandation sont l'erreur absolue moyenne (MAE) et l'erreur quadratique moyenne (RMSE). La MAE calcule l'écart entre les notes prédites et les vraies notes. Quant à la RMSE, elle repose sur le même principe que la MAE, tout en mettant l'accent sur les écarts importants.

Soit $N \subseteq I$ un ensemble test d'items, $p(u, i)$ une prédiction de note de l'utilisateur u pour l'élément i et $v(u, i)$ la note donnée par l'utilisateur u pour l'item i .

$$MAE = \frac{1}{N} \sum |p(u, i) - v(u, i)|$$

$$RMSE = \frac{1}{N} \sqrt{\sum (p(n, i) - v(n, i))^2}$$

7.4. Qualité des systèmes de recommandation

La qualité de la recommandation est également une dimension souvent évaluée. Il s'agit de mesurer la performance de la tâche de recommandation elle-même. En pratique, Dans un tel système de recommandation il existe quatre ensembles nommés : Vrai positif (VP), Vrai négatif (VN), Faux positif (FP) et Faux négatif (FN). Par ces ensembles nous pouvons classer les résultats obtenus par un système de recommandation afin de mesurer la performance du système. Vrai positif (VP) représente les éléments pertinents et choisis proposés par le système (prédites), Vrai négatif (VN) représente les éléments pertinents et refusés, faux positif (FP) représente les éléments non pertinents et acceptés et faux négatif (FN) représente les éléments non pertinents et refusés. Donc à partir de ces quatre ensembles nous présentons les mesures les plus utilisées pour l'évaluation hors ligne des systèmes de recommandations

Précision : La précision est une mesure d'exactitude. Elle détermine la proportion des items pertinents recommandés parmi tous les items recommandés.

$$precision = \frac{VP}{VP + FP}$$

Rappel: cette métrique calcule la probabilité qu'un article pertinent soit choisi

$$Rappel = \frac{VP}{VP + FN}$$

F-Mesure: cette mesure est une façon courante de combiner le rappel et la précision dans une seule métrique afin de faciliter la comparaison.

$$F - mesure = \frac{2 * precision * rappel}{précision + rappel}$$

Chaque mesure présentée sous dessus a une signification intuitive. Le rappel représente la couverture des éléments utiles que le système de recommandation peut obtenir. Cette métrique mesure la capacité à obtenir tous les éléments utiles présents dans le corpus. Tandis que, la précision montre la capacité du système à n'afficher que les éléments pertinents, tout en minimisant le mélange d'entre eux avec les éléments inutiles.

Synthèse

Dans cette partie consacrée au système de recommandation, nous avons étudié les différentes étapes de la création d'un système de recommandation classiques, ainsi que les approches utilisées et les méthodes d'évaluation. L'inconvénient majeur retiré est que ces systèmes ne prennent pas en charge l'aspect contexte de l'utilisateur ainsi que le contexte de l'élément. Dans

Chapitre 3

Profil D'utilisateur & Web Sémantique

1. Le profil utilisateur

1.1. Introduction

Bien que les systèmes de recommandation soient souvent basés sur un modèle d'utilisateur, dans ce chapitre, nous étudions de façon générale le profil utilisateur. Nous présentons d'abord la notion de profil utilisateur ainsi que ses techniques de construction. Ensuite nous explorons les différents modèles utilisés. Noter que ces modèles peuvent être combinés afin de réaliser le modèle utilisateur qui conviendra à l'application développée [60].

Les utilisateurs différents les uns des autres par leurs objectifs, leurs acquis, leurs antécédents et leurs connaissances par rapport aux différents sujets. La collection des données de profil qui sont vaste et diversifiées se fait à travers différentes sources hétérogènes.

1.2 Les Modèles utilisateurs

Plusieurs modèles d'utilisateur ont été utilisés dans la littérature dont nous citons :

- **Individuel** : Dans ce modèle, les caractéristiques sont regroupées par catégorie. Ces caractéristiques concernent les informations d'un individu qui peuvent être soit retenues par l'utilisateur, ou bien déduites par le système lors de l'utilisation de l'application. Ce modèle contient des informations sur le domaine de connaissance de l'utilisateur, ses objectifs, ses préférences, ses intérêts ainsi que d'autres informations.
- **Stéréotype** : Il s'agit d'un modèle générique qui correspond à un ensemble de caractéristiques les plus spécifiques d'un groupe d'individus, que nous pouvons utiliser comme des valeurs par défaut (un des problèmes du démarrage à froid (voir chapitre 02)).
- **Recouvrement** : Ce modèle est représenté conjointement avec un modèle de domaine. Pour un utilisateur particulier, ce modèle est représenté par une valeur à chaque concept du modèle de domaine. Chaque valeur représente le niveau de connaissance de l'utilisateur pour un concept du domaine utilisé. Le modèle de recouvrement est facile à mettre à jour mais il est difficile à initialiser au niveau de la mesure du niveau de connaissance (problème de démarrage à froid). Les actions de l'utilisateur sont analysées pour augmenter ou réduire le niveau de connaissance des concepts du domaine [61].
- **Partagé** : Il s'agit d'un modèle réutilisable dans plusieurs services ou applications. En effet, l'idée du modèle partagé est de disposer d'une partie commune et de parties

spécifiques à chacune des applications ou tâches à réaliser. Cette approche permet d'une part de partager des informations entre applications et d'autre part redéfinir un modèle pour chacune des applications.

1.3. Les caractéristiques de l'utilisateur

Il existe deux groupes de caractéristiques relatif à un utilisateur, la première est lié à sa connaissance et son expérience (le savoir-faire) et sa compétence, la deuxième représente les préférences les objectifs [61]. nous allons regarder plus en détail chacun d'eux

1.3.1. La connaissance

La connaissance de l'utilisateur est considérée comme la caractéristique la plus importante dans un système adaptatif. La façon la plus simple de gérer les connaissances est de mémoriser ce que l'utilisateur connaît ou ne connaît pas, comme par exemple un concept provenant d'un modèle de domaine. Cette information peut être obtenue explicitement par l'utilisateur ou implicitement par le système autrement dit par une base de règles d'inférences par exemple.

1.3.2. L'expérience, les compétences

- **L'expérience** : L'expérience de l'utilisateur représente son savoir-faire qu'il possède avec le type de système qui lui est présenté. Pour un système de recommandation, l'expérience représente les capacités de l'utilisateur à comprendre les différents composants ainsi que celles de navigation dans le système.
- **Les compétences** : Les compétences possédées par l'utilisateur correspondent aux connaissances qui ne relèvent ni du domaine, ni de l'expérience mais qui sont néanmoins considérées comme pertinentes dans le fonctionnement du système. Pour les systèmes de recommandations de tourisme (par exemple POI : Ruine de Timgad). Les connaissances évoquées dans le paragraphe précédent concernent l'histoire de Timgad, alors que l'expérience porte sur les systèmes de recommandation et que les compétences peuvent regrouper ses connaissances en économie, en politique, etc.

1.3.3. Les préférences, les objectifs

- **Les préférences** : Pour différentes raisons, les préférences de l'utilisateur peuvent porter sur une généralité et/ou un point particulier. les préférences ne peuvent être déduites par le système. En effet, chaque utilisateur possède ses propres préférences, et c'est à lui de les renseigner.

- **Les objectifs** : L'objectif est un état que l'utilisateur espère atteindre, et les plans décrivent les étapes pour y arriver. Notamment, dans les systèmes information en ligne, l'utilisateur a l'habitude tendance à choisir seulement une partie du document qui dépend principalement de leurs objectifs. En effet, les plans sont souvent modélisés par le modèle de tâche et aussi l'objectif de l'utilisateur.

1.4. L'acquisition des caractéristiques

Il existe deux manières d'obtenir des informations sur les utilisateurs, *l'acquisition explicite*, selon laquelle nous utilisons une source externe pour créer et/ou compléter le modèle et *l'acquisition implicite*, selon laquelle le système infère l'information depuis les connaissances disponibles sur l'utilisateur « *Acquisition incrémentale* ». Voici quelques méthodes couramment utilisées :

- **L'observation directe** : Il s'agit de la méthode la plus précise. Elle permet d'identifier des classes d'utilisateurs ainsi que les tâches de ces derniers. De plus, on peut identifier des facteurs critiques, comme la pression sociale, qui ont des effets néfastes sur l'utilisation du système. Malheureusement, il s'agit d'une méthode très coûteuse qui nécessite des personnes qualifiées derrière chacun des individus observés.
- **Les interviews** : Cette technique permet d'obtenir un autre type d'information, l'expérience, les opinions, les motivations comportementales mais surtout les avis sur les outils existants. Ils sont plus courts et moins coûteux que la technique d'observation, néanmoins, ils nécessitent aussi du personnel qualifié.
- **Les questionnaires** : Ou « comment obtenir à moindre coût un maximum de données »? [62] Les résultats obtenus permettent des études statistiques et des généralisations plus fortes que les interviews. Les questionnaires peuvent être collectés par des personnes non expérimentées. Ils permettent d'avoir à la fois un aperçu de la situation et des points d'information plus précis

1.5. Représentation du profil utilisateur

Cette section présente les techniques de base qui permettent de représenter et de construire le profil de l'utilisateur. En particulier, nous présentons ici les types de représentation qui sont couramment utilisées : les concepts, les profils à base de vecteurs et les réseaux sémantiques.

1.5.1. Représentation vectorielle

La représentation du profil utilisateur à l'aide d'un vecteur qui contient un ensemble de mots-clés est devenue un moyen courant de personnalisation. Les mots-clés peuvent être explicitement fournis par l'utilisateur ou implicitement extraits à partir de documents.

De plus, cette technique peut traiter séparément chaque mot-clé comme un sujet d'intérêt, ou catégoriser conjointement les mots-clés en classes d'intérêts d'utilisateur. A chaque mot-clé est associé un poids qui représente son importance numérique dans le profil [80]

Il existe trois manières de représenter des profils de mots-clés : un ensemble de termes pondérés où chaque mot-clé représente un intérêt [81] ou l'ensemble du vecteur de termes [82] et finalement plusieurs vecteurs contenant des termes pondérés représentent l'intérêt de l'utilisateur [83].

1.5.2. Représentation conceptuelle

Dans la représentation conceptuelle, il existe des relations de type parent-enfant qui peut être facilement mises à jour par rapport à une représentation sémantique qui repose sur des relations multiples entre les mots [84]. Généralement, ce type de représentation nécessite l'utilisation d'une ontologie et de concepts hiérarchiques.

1.5.3. Réseau sémantique

Ce type de représentation consiste à enregistrer les intérêts de l'utilisateur dans un réseau sémantique dont les nœuds représentent les termes traduisant un intérêt de l'utilisateur et les liens entre les nœuds représentent la proximité sémantique entre les nœuds

La représentation sémantique [85] n'est pas seulement basée sur l'extraction de termes à partir de documents pertinents, mais aussi sur l'incorporation de ces termes dans un réseau de nœuds. La construction de tels profils nécessite la création de relations sémantiques entre les nœuds du réseau.

1.6. Évolution du profil utilisateur

La gestion de l'évolution du profil utilisateur est une tâche primordiale qui consiste à prendre en compte les variations ou les changements des centres d'intérêt des utilisateurs au cours du temps. L'évolution du profil utilisateur se fait souvent selon un processus basé sur l'addition de nouvelles informations dans la représentation du profil. Les approches qui gèrent l'évolution du profil reposent sur des techniques à court terme ou à long terme ou les deux à la fois. Le premier représente les centres d'intérêts liés à une activité courante de l'utilisateur, tandis que le second modélise les centres d'intérêts persistants de l'utilisateur qu'issus de son historique de recherche. Ce dernier peut être exploité dans le but d'améliorer les réponses

pour toutes requêtes soumises. Les techniques de collecte des informations utilisées dans la gestion de l'évolution du profil utilisateur sont relativement dépendantes des activités des utilisateurs. Pour le profil à court terme par exemple, la collecte de l'information est liée à des techniques d'assemblage des activités de recherche liées à une même exigence en information dans les sessions de recherche.

2. Web sémantique

2.1. Introduction

Le Web sémantique est apparu suite à des besoins concernant l'organisation des contenus Web et l'automatisation de leurs accès. Le web sémantique est considéré comme une nouvelle infrastructure permettant aux agents logiciels d'accéder efficacement aux sources d'information et aux services. Dans cette partie nous allons présenter les notions et les concepts de la modélisation sémantique.

2.2 Définition

Le terme a été inventé par **Tim Berners-Lee** pour un web de données qui peut être traité par les machines [63]. Le web sémantique est un domaine qui s'intéresse à la sémantique du contenu des pages web en particulier et le contenu des documents en générale. Le Web sémantique est un système qui permet aux machines de «comprendre» le contenu des documents et de répondre aux requêtes des humains en exploitant « le sens ». Pour se faire, les documents doivent être sémantiquement structurés. Selon le W3C¹, "Le Web sémantique fournit un cadre commun qui permet aux données d'être partagées et réutilisées par les applications, et les entreprises».

2.3. Définition globale du web sémantique

Le web sémantique est une extension du web actuel, et son contenu est compréhensible par des machines, c'est pourquoi l'utilisation du terme « sémantique », il traite le « sens » qui se trouve dans le contenu des pages [62]. Pour arriver à rendre le web compréhensible et exploitable par les agents logiciels, le W3C apprend pour élaborer des standards et des langages pour modéliser le contenu des documents, parmi ces standards nous citons les suivants : 1. L'utilisation de XML afin de permettre la structuration des documents. 2. L'utilisation de RDF pour permettre la signification de ces structures grâce aux « triplets ».

¹ World Wide Web Consortium (W3C)

2.4. Langages du web sémantique

Dans le contexte du Web Sémantique, plusieurs langages ont été élaborés, ces langages sont systématiquement exprimables et échangeables dans une syntaxe XML (eXtensible Mapping Language). Ceci permet de bénéficier de l'ensemble des technologies développées autour d'XML : XML Schemas, outils d'exploitation des ressources XML (bibliothèques JAVA, etc.), bases de données gérant des fichiers XML.

2.4.1. XML

Il s'agit seulement d'une couche syntaxique, qui permet de structurer les données et organiser selon un format de message standard. Cette opération s'appelle balisage XML. XML est un langage de balisage considéré comme une spécification pour les documents "compréhensible par les machines". Il est naturellement utilisé pour encoder les langages du Web sémantique. L'extensibilité du XML indique la différence importante avec d'autres langages précédents qui est aussi la caractéristique essentielle du XML. XML est un métalangage (une description de type de document, DTD, permet de décrire la grammaire des documents admissibles). En effet, XML fournit une structure pour représenter d'autres langages (appelé parfois *dialectes XML*) d'une manière normalisée. La structure d'un document XML est définie par une DTD (Document Type Definition). La DTD peut être écrite dans un document à part puis référencée dans le document XML ou peut être directement intégrée dans ce dernier.

XML Schéma², est un langage XML qui remplace la DTD. XML schéma est utilisé pour les types de données XML. XML Schéma est une nouvelle recommandation du W3C, il est possible de définir des types de données complexes en utilisant des éléments imbriqués. XML schéma utilise l'espace de nommage XML (*namespace*) qui permet d'identifier une définition du document spécifique avec un nom unique et de préfixer toutes les balises avec ce nom unique. XML schéma présente plusieurs avantages par rapport à la DTD car il offre une grammaire plus riche pour décrire la structure des éléments.

2.4.2. RDF et RDFS

2.4.2.1. RDF

RDF (Resource Description Framework)³ est un modèle pour le stockage d'informations sous forme de graphes⁴ permettant d'apporter de la sémantique aux ressources Web. Si XML

² <http://www.w3.org/XML/Schema>

³ <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>

est le moyen de représenter un document, alors le (RDF) est le moyen de décrire le contenu du document. RDF est considéré comme un vocabulaire au format XML conçu pour les applications de description de ressources et de métadonnées. RDF améliore le traitement automatique des ressources Web. Un des objectifs de RDF est de rendre possible la spécification de la sémantique des données d'une manière standardisée et interopérable. La spécification de la syntaxe RDF est basée sur XML, mais XML n'est pas le seul langage possible pour exprimer du RDF. En effet, le formalisme de RDF consiste en un ensemble de triplets *<ressource, propriété, valeur>* dont de nombreux outils ont été développés autour d'XML.

Un modèle RDF est un graphe étiqueté représenté par des nœuds et des arcs dont les nœuds peuvent être de deux types très généraux, ressource ou littéral. Par exemple, une personne serait une ressource, mais le nom de la personne serait un littéral. Un arc définit la relation binaire entre une ressource et une valeur (un littéral ou une autre ressource).

Les objectifs de RDF

Parmi les objectifs de RDF on cite :

- Avoir un modèle de données simple.
- Avoir une sémantique formelle et une inférence prouvable.
- Utiliser un vocabulaire extensif, basé sur les URIs (Uniform Resource Identifier).
- Supporter l'utilisation de types de données de schéma XML.

Les éléments de base de RDF

- **Ressource** : qui est représenté sous forme d'URI. Avec RDF, tout ce qui est exprimé est appelé ressource. La ressource est toute entité d'information pouvant être référencée, par un nom symbolique (littéral) ou un identificateur.
- **Propriété** : une propriété est une caractéristique ou une relation pour la description d'une ressource. Chaque propriété a une signification spécifique, des valeurs autorisées, des types de ressources qu'elle peut décrire et des relations avec d'autres propriétés.

⁴ Certains considèrent le RDF comme un langage XML de description de données.

- **Triplet** : les trois parties (ressource, propriété, valeur) constituent un *triplet* (sujet, prédicat, objet). La valeur de la propriété associée à une ressource constitue une **déclaration RDF** (statement) qui peut-être une autre déclaration RDF ou un littéral. En langage RDF, un littéral (« literal ») peut avoir un contenu XML bien formé mais qui n'est plus pris en compte par un processeur RDF.

2.4.2.2. RDFS (Resource Description Framework Schema)

Le langage RDFS est basé sur RDF [61]. RDFS a pour objectif d'étendre RDF en décrivant plus précisément les ressources utilisées pour étiqueter les graphes. Pour cela, il fournit un mécanisme permettant de spécifier les classes dont les instances seront des ressources.

RDFS s'écrit toujours à l'aide de triplets RDF en utilisant deux propriétés fondamentales *subClassOf* et *type* pour représenter respectivement les relations de subsomption entre classes et les relations d'instanciation entre instances et classes. Les classes spécifiques au domaine sont déclarées comme des instances de la ressource *Class* et les propriétés spécifiques au domaine comme des instances de la ressource *Property*. Les propriétés *subClassOf* et *subPropertyOf* permettent de définir des hiérarchies de classes et de propriétés.

RDFS ajoute également à RDF deux attributs pour définir les contraintes de domaine *rdfs:domain* et *rdfs:range*. Les instances sont décrites en utilisant le vocabulaire donné par les classes définies dans ce schéma.

Pour résumer, XML peut être vu comme la couche de transport syntaxique, RDF comme un langage relationnel de base. RDFS offre des primitives de représentation de structures ou primitives ontologiques [64].

2.4.3 OWL

OWL est créé pour d'écrire les ontologies web (voir section suivante). OWL ajoute par rapport aux RDF et RDFS qui apportent à l'utilisateur la capacité de décrire des classes et des propriétés, des outils concernant les propriétés et les classes : identité, équivalence, contraire, cardinalité, symétrie, transitivité, disjonction, etc. OWL est un langage XML, fondé sur la syntaxe de RDF/XML. . Grâce à un vocabulaire plus large et à une vraie sémantique formelle, OWL offre aux machines une grande capacité d'interprétation du contenu web que RDF et

RDFS, OWL permet de définir des terminologies pour décrire des domaines concrets. Une terminologie se constitue de concepts et de propriétés.

Le langage OWL se compose de trois sous-langages offrant une expressivité croissante : *OWL-Lite*, *OWL-DL* et *OWL-Full*.

- **OWL Lite** : Supporte des hiérarchies de classifications et des caractéristiques de contraintes simples. Par exemple, lorsqu'il supporte les contraintes de cardinalité, il permet seulement les valeurs 0 et 1.
- **OWL DL** : correspond à la logique de description (DL). Il supporte les utilisateurs qui demandent un maximum d'expressivité. OWL DL contient tous les constructeurs du langage OWL, mais sont utilisables avec des restrictions (par exemple, lorsqu'une classe peut être un sous-classe de plusieurs autres classes, une classe ne peut être une instance d'une autre classe).
- **OWL Full** : destiné aux utilisateurs qui demandent un maximum d'expressivité avec la liberté syntaxique de RDF sans aucune garantie de calculs. Par exemple, une classe peut être traitée comme une collection d'individus et en même temps peut être vue comme un seul individu. OWL Full permet aussi à une ontologie d'augmenter le sens du vocabulaire prédéfini (RDF et OWL).

Noter que, seulement les deux premiers maintiennent les tâches d'inférence principales

2.4.4. SPARQL (Query Language for RDF)

La représentation de connaissances dans la nouvelle génération du Web sémantique est importante, mais la capacité de faire des requêtes de ces connaissances joue aussi un rôle primordial dans les applications ou les bases de connaissances. Parmi certains langages de requêtes connus tels que SPARQL, TRIPLE, SQL, XPATH..., SPARQL un langage de requête proposé par W3C et dédié à RDF, est largement utilisé dans le domaine de recherche et d'extraction d'informations. RQL possède la capacité d'exprimer des variables (*?NomVariable*) de requête dans les positions du sujet, de la propriété ou de l'objet d'un triplet.

2.4.5. RDFa

RDFa [66] est une recommandation du W3C définissant une syntaxe permettant d'ajouter des données structurées dans une page HTML ou n'importe quel document XML. Le langage

RDFa fournit une syntaxe et un ensemble de balises (tags) pour décrire les données structurées en (X)HTML. Avec les informations supplémentaires insérées dans les balises, les données structurées en (X)HTML sont rajoutées les “sémantiques” qui permettront l'échange d'informations par les applications automatiques ou par les agents informatiques.

2.5 Ressources sémantiques

2.5.1 Taxonomie

Une taxonomie est une collection de termes organisés en une structure hiérarchique. Les termes dans une taxonomie sont reliés entre eux par des relations parent-enfant. Dans notre cadre nous avons proposé une taxonomie afin identifier les circonstances relatives à l'attribution des votes dans les systèmes a base de filtrage collaboratif.

2.5.2. Thesaurus

Un thésaurus est un réseau de termes d'un vocabulaire contrôlé. La différence entre taxonomie est celle que le thésaurus utilise d'autres types de relations en plus de la relation parent-enfant.

Un thésaurus a pour but de guider la phase d'indexation et la phase de recherche en sélectionnant le terme qui correspond un sujet donné. Un exemple très connu d'un thésaurus et le WORDNET.

2.5.3. Les ontologies

2.5.3.1. La notion d'ontologie

En informatique, plusieurs définitions ont été proposées à l'ontologie :

Selon **Tom Gruber** [65] « *Une ontologie est une spécification explicite d'une conceptualisation* ».

Pour [67] une ontologie est « *Un catalogue des types de choses que l'on assume qui existent dans un domaine d'intérêt D du point de vue d'une personne qui emploie le langage L dans le but de parler de D* »

La troisième définition de **Guarino** : « *Une ontologie fait référence à un artefact d'ingénierie constitué d'un vocabulaire spécifique employé pour décrire une certaine réalité, plus un ensemble d'assomptions explicites concernant la signification convenue des mots du dictionnaire.* »[68].

Une ontologie fournit Une structure conceptuelle de base à partir de laquelle il est possible de développer des systèmes à base de connaissances qui soient partageables, réutilisables et assure l'interopérabilité entre les sources d'information et de connaissances [69].

2.5.3.2. Composants d'une ontologie

Une ontologie se compose d'un certain nombre de composants différents. Les noms de ces composants diffèrent entre les ontologies en fonction du langage d'ontologie utilisé, leurs composants de base sont largement partagés entre différentes ontologies. Parmi ces composants on trouve concepts, individus et relations ...etc.

Concept / classe

Un concept représente un ensemble d'objets avec leurs propriétés communes. Les concepts peuvent être des sous concepts (sous classe). Ils sont appelés aussi termes ou classes de l'ontologie, les dimensions d'un concept peuvent être classifiés selon :

- Niveau d'abstraction (concret ou abstrait).
- Atomicité (élémentaire ou composée).
- Niveau de réalité (réel ou fictif)

Les propriétés portant sur des concepts sont :

L'abstraction : un concept est abstrait si toute instance de ce concept est aussi instance d'un de ses concepts fils. Par exemple, dans une hiérarchie comportant les concepts *Homme* et *Femme*, fils du concept *Humain*, le concept *Humain* est abstrait.

la subsumption : un concept C1 subsume un concept C2 si toute propriété sémantique de C1 est aussi une propriété sémantique de C2, c'est-à-dire C1 est plus spécifique que C2. L'extension d'un concept subsumé est forcément plus réduite que celle du concept qui le subsume. Par exemple, *Homme* subsume *Humain*.

L'équivalence : deux concepts sont équivalents s'ils ont la même extension. Par exemple, *Etoile du matin* et *Etoile du soir*.

la disjonction : deux concepts sont disjoints si leurs extensions sont disjointes. Par exemple, *Homme* et *Femme*. Si tous les concepts fils d'un concept abstrait sont disjoints deux à deux, ils forment une **partition** du concept abstrait.

Individus / Instances

L'individu est une concrétisation de concept, autrement dit, c'est l'élément instance du concept. Par exemple les individus *mohamed* et *mourad* sont des instances du concept *Personne* ».

Relation (attribut)

Désigne les liens sémantiques de la connaissance du domaine. Une relation peut être vue comme une propriété ou un attribut. Les propriétés décrivent des interactions entre concepts, elles peuvent être fortement typées, i.e., associées un domaine précis.

Axiome (règles)

Les axiomes sont distingués par des assertions générale sur les fondements de l'ontologie, les axiomes sont utilisés pour décrire les faits de l'ontologie qui seront considérés après comme vrais, cette détermination a pour but de définir les significations des composants d'ontologie, les contraintes sur les valeurs des attributs, et les arguments de relations.

2.5.3.3. Typologie des ontologies

Les auteurs des ontologies distinguent plusieurs niveaux d'ontologies selon le but pour lequel elles sont conçues

- *Les ontologies de domaine* : les plus connues, elles expriment des conceptualisations spécifiques à un domaine, elles sont réutilisables pour des applications sur ce domaine.
- *Les ontologies d'application* : elles contiennent des connaissances du domaine nécessaires à une application donnée ; elles sont spécifiques et non réutilisables. Dans notre cadre nous avons utilisé ce type d'ontologie, il s'agit de l'ontologie de profil d'utilisateur dans le contexte des systèmes de recommandations
- *Les ontologies génériques* : appelées aussi ontologies de haut niveau, elles expriment des conceptualisations très générales tels que le temps, l'espace, l'état, le processus, les composants.

2.5.3.4. Classement des ontologies

Les ontologies sont classées en plusieurs niveaux d'expression selon leur utilisation :

- *Très informelle* : l'ontologie est exprimée en langage naturel.

- *Semi-informelle* : l'ontologie est exprimée sous une forme restreinte et structurée de langage naturel pour augmenter la clarté et pour réduire l'ambiguïté.
- *Semi-formelle* : l'ontologie est exprimée en langage formel.
- *formelle* : l'ontologie est définie avec une sémantique formelle, permettant le théorème et la preuve.

2.5.3.5. Construction des ontologies

2.5.3.5.1. Méthodes de construction

Dans la littérature, il existe différentes méthodes (considérées comme des méthodes traditionnelles) de développement d'ontologies. La méthode **d'Uschold et King** [100], la méthodologie de **Grüninger et Fox** [87] et la **MÉTHODOLOGIE** [87]. D'autres méthodes plus récentes prennent en compte plusieurs aspects non techniques du processus de développement d'ontologie, tels que l'aspect de la communauté et de l'évolution. Des exemples de méthodologies plus récentes sont *Developing Ontology-grounded Methods and Applications* (DOGMA) [89], *Ontology-based Knowledge Management method* (OTKM) [90], *DILIGENT* [91],

La méthode de Uschold consiste en trois approches : *bottom-up*, *top-down* and *middle-out*

L'approche ascendante (*bottom-up*) commence par la définition des classes les plus spécifiques en fonction des instances disponibles dans le monde réel. Tandis que L'approche descendante (*top-down*) commence par identifier les concepts les plus généraux puis les organiser en une taxonomie pour générer un système d'axiomes de haut niveau. L'approche intermédiaire (*middle-out*) commence par identifier les termes de base, puis les spécifier et les généraliser selon les besoins.

La méthode de **Grüninger** est basée sur la logique du premier ordre pour le développement des systèmes à base de connaissances. Ils suggèrent d'utiliser des questions pour définir les terminologies et les relations tout en créant des ontologies.

2.5.3.5.2. Critères de construction d'ontologies

Il existe un ensemble de critères et de principes qui ont fait leurs preuves dans le développement des ontologies. Parmi celles on cite les critères proposé par **Gruber** [71]:

- **Clarté et objectivité :** L'ontologie doit fournir la signification des termes définis en fournissant des définitions objectives ainsi qu'une documentation en langue naturelle.
- **Complétude:** Une définition exprimée par des conditions nécessaires et suffisantes est préférée à une définition partielle (définie seulement par une condition nécessaire et suffisante).
- **Cohérence:** Une ontologie cohérente doit permettre des inférences conformes à ces définitions.
- **Extensibilité ontologique maximale :** De nouveaux termes généraux et spécialisés devraient être inclus dans l'ontologie d'une façon qui n'exige pas la révision des définitions existantes (des définitions sur mesure).

3. Conclusion

Dans cette partie, nous avons étudié la notion de profil utilisateur. Son utilisation dans les systèmes de recommandation est présentée dans le chapitre 2. Nous avons aussi présenté les approches de construction du profil utilisateur ainsi que les ressources sémantiques pour modéliser les différentes bases de connaissances. Le défi dans la construction du profil utilisateur est d'obtenir un profil qui satisfait les besoins de l'application. La question principale est donc : comment construire un profil qui soit pertinent par rapport aux caractéristiques et au comportement de l'utilisateur au sein d'un système de recommandation

A la différence de la technique qui prend en charge des utilisateurs ayant des relations de l'utilisateur avec d'autres. Notre profil prend en compte les profils similaires (filtrage collaboratif).

Dans ce chapitre, nous avons présenté aussi la nouvelle vision du Web sémantique. Les applications du web sémantique reposent sur deux composants essentiels et importants que sont l'ontologie et l'annotation sémantique. Nous avons examiné dans ce chapitre les ontologies qui représentent la technologie dorsale pour le Web sémantique et - plus généralement - pour la gestion des connaissances formalisées décrivant les ressources du Web. Dans le chapitre suivant nous allons présenter notre approche proposée qui repose les principaux concepts présenté dans ce chapitre tels que le profil d'utilisateur et le raisonnement à base de règles sémantiques.

Chapitre 4

Approche Proposée

1. Introduction

Dans ce chapitre, nous allons présenter l'approche générale de notre travail [102] qui nous permet d'extraire les données de contexte approprié et inapproprié pour la consommation (consultation, réutilisation, ...) d'un item au sein d'un système de recommandation à base de filtrage collaboratif. Nous rappelons, que notre plateforme de travail est un système de recommandation contextuel et qu'on suppose que les différents acteurs de ce système ayant des dispositifs informatiques (Pc, Smartphone, ...) caractérisés par différentes configuration. Nous présentons dans ce chapitre le cadre théorique de notre approche ainsi que les différents modèles utilisés. Notre approche repose sur deux suppositions majeures. La première : un score élevé donné par un utilisateur implique que le contexte de l'utilisateur est approprié, et le second : dans certains cas, l'abstention de noter un item est causée par l'incompatibilité du contexte de l'utilisateur avec le contenu de l'item. Un rappel de la problématique est donné à la section 3.1. La base de connaissance utilisée dans ce travail est présentée à la section 3.2. La section 3.3 décrit le profil d'utilisateurs en intégrant l'aspect contexte. La déduction des valeurs appropriées est inappropriée pour chaque item du système en détaillant les entrées ainsi que l'utilité de chaque règle sémantique proposée est décrite à la section 3.4. Enfin, une conclusion résume les principales notions présentées est définie dans ce chapitre

2. Rappel problématique

Le but d'un tel système de recommandation contextuel est de recommander aux utilisateurs des items pertinents parmi les items disponibles, ces recommandations doivent être conformes à leurs préférences et également à leurs situations (contexte). Dans ce travail, nous nous intéressons aux systèmes de recommandation contextuels dont le profil utilisateur est défini à partir de l'analyse du comportement de l'utilisateur dans un système de recommandation. Plusieurs méthodes de recommandation contextuelle existent dans la littérature, parmi lesquelles on trouve celles proposant des méthodes d'extraction des données contextuelles. La plupart de ces méthodes sont basées sur l'exploration et l'analyse des données en utilisant les techniques du data Mining. Le modèle utilisateur est souvent décrit par la matrice des votes fournie par le système de filtrage collaboratif, où les lignes correspondent aux utilisateurs, les colonnes aux items et les valeurs de la matrice représentent les votes (notes). Dans le paradigme de sensibilité au contexte, en ajoutant une dimension "contexte", cette dimension a pour but de décrire la situation de l'item et/ou l'utilisateur qui sera ensuite prise en charge par les différents algorithmes de recommandation contextuels.

La matrice des votes représente les opinions ou les évaluations des utilisateurs. Une valeur de cette matrice donnée explicitement ou calculée implicitement reflète le degré de satisfaction de l'utilisateur concernant un item. Pour les systèmes de recommandation traditionnels une valeur de cette matrice concerne généralement le contenu de l'item et représente le niveau d'acceptation de l'item en regard de leur contenu. Les modernes systèmes de recommandation qui intègrent le contexte comme un facteur primordial dans leurs processus de recommandation attestent que l'attribution d'une note par un utilisateur est liée par différentes causes tels que le contexte de l'utilisateur ou l'item.

La détection du contexte est un enjeu de recherche qui permet l'extraction et l'analyse des données contextuelles afin de mesurer l'effet de ces informations dans la tâche de l'attribution des notes aux items. On suppose que le contexte des systèmes de recommandation de notre cas joue un rôle important et peut affecter la note donnée par l'utilisateur. Dans les systèmes de recommandation contextuels, il existe deux types de contexte : le contexte de l'utilisateur et le contexte de l'item. Le contexte de l'utilisateur se caractérise par des informations liées à l'utilisateur elle-même, tandis que le contexte de l'item représente les différents paramètres descriptifs qui sont éventuellement mesurables.

L'objectif de ce travail est de proposer une nouvelle méthode de détection des conditions contextuelles indispensables à la consommation adéquate de l'item.

3. Travaux connexes

Les systèmes basés sur des techniques de systèmes de recommandations sont très utilisés, et cela dans de multiples domaines. Nous nous intéressons ici au domaine spécifique qui concerne la détection des informations contextuelles et plus précisément la détection des conditions contextuelles (valeurs des facteurs contextuelles) qui influencent la consommation de l'item d'un système de recommandation à base de filtrage collaboratif.

À partir de toutes les informations contextuelles possibles qui peuvent être acquises, il est nécessaire de décider ce qui est important pour un service spécifique. Des recherches antérieures sur l'identification des informations contextuelles pertinentes dans CARS ont exploré deux types de méthodes: (a) identifier a priori les facteurs qui devraient être pris en compte par le système, ou (b) sélectionner, a posteriori, après l'acquisition des évaluations et des données de contexte, les facteurs les plus agissants pour le calcul des prévisions de notation.

Les auteurs dans [92] présentent une approche basée sur une enquête visant à identifier les informations contextuelles pertinentes pour un système de recommandation dédié pour le tourisme. Ils ont d'abord estimé la dépendance des préférences de l'utilisateur à partir d'un ensemble candidat initial de facteurs contextuels. Cela a été réalisé grâce à un outil Web, dans lequel les utilisateurs ont été invités à évaluer un POI (point d'intérêt) par exemple un musée, dans des conditions contextuelles particulières (journée froide, pluvieux ...etc.). Par la suite, Ils ont pu sélectionner les facteurs contextuels les plus importants pour différents types de POI. Les notes et les informations contextuelles pour les facteurs sélectionnés ont été utilisées pour former un modèle de factorisation matricielle contextuelle afin fournir aux utilisateurs des recommandations dépendant du contexte dans une application mobile pour iPhone.

Dans [93], les auteurs ont collecté les votes et les informations contextuelles de chaque évaluateur. L'analyse de ces données est effectuée par un classificateur bayésien en examinant l'ensemble des données contextuelles qui jouent un rôle important dans la prédiction de l'évaluation. Ensuite, Ils ont appliqué la décomposition en valeurs singulières (SVD) aux variables de contexte pertinentes sélectionnées. D'autres approches récentes utilisent l'apprentissage automatique pour déterminer les conditions de contexte. D'autres méthodes basées sur des techniques statistiques pour la détection des informations pertinentes ont été utilisées dans nombreux travaux. Les auteurs de [94] ont utilisé le test χ^2 pour la détection du contexte pertinent. Cependant, ce test a posé un problème dans le cas de petites bases de données (problème du démarrage à froid).

Une autre approche similaire [95] a été appliquée dans le domaine médical. Cette approche est basée sur l'analyse des situations des patients afin de fournir le service adéquat. Une ontologie et un modèle de contexte ont été proposés. Le but de ce travail est de développer un système de raisonnement contextuel en utilisant les technologies web sémantiques.

Comme conclusion, nous pouvons identifier deux approches différentes pour déterminer la pertinence du contexte : l'évaluation à partir d'une enquête auprès des utilisateurs et la détection à partir des données d'évaluation. Notre approche diffère, elle tente de déterminer les conditions contextuelles en analysant les comportements des utilisateurs devant un système de filtrage collaboratif. Nous nous investiguons sur l'utilisation de technologie web sémantique en développant une ontologie de profil d'utilisateur qui intègre à la fois des données des utilisateurs, les données des items et les conditions contextuelles.

4. Base de connaissance de la matrice des votes

Dans les systèmes de recommandation, la matrice des votes est considérée comme un support important pour générer des recommandations pertinentes dont les valeurs de cette matrice représentent les opinions des utilisateurs concernant des items disponibles dans le système. Une valeur est considérée comme une évaluation de l'item en regard de leur contenu et de leur contexte. L'évaluation d'un item dans les systèmes de recommandation mesure généralement le degré de préférence d'un utilisateur pour un item donné. Ce degré de préférence est représenté par une certaine valeur entière (souvent de 1 à 5) qui représente l'attribution effectuée par l'utilisateur. L'utilité peut être exprimée explicitement par l'utilisateur (*étoiles, j'aime*) ou de manière implicite (par exemple le nombre de fois que l'utilisateur a écouté un genre musical, le nombre de fois qu'un utilisateur a visité la page d'un article, etc.). Dans ce travail et afin de simuler l'avis de l'utilisateur, nous adaptons une évaluation à deux niveaux "aimé (*like*)" et "détesté (*dislike*)" au lieu d'utiliser la manière populaire dans les évaluations des objets.

Lorsqu'un utilisateur aime un article, cela signifie que l'utilisateur a consommé l'item dans un contexte confortable [6]. Par cette optique, nous avons proposé une base de connaissance permettant d'extraire des données contextuelles convenables à la consommation appropriée d'un item tout en ajoutant d'autres hypothèses. La figure ci-dessous (figure 4.1) illustre notre proposition de la base de connaissance utilisée pour construire un modèle permettant l'extraction des données contextuelles convenables pour la consommation d'un item.

Cette base de connaissance est subdivisée en trois couches : couche du score de votes, couche de l'item et la couche de l'utilisateur. La couche des scores représente les deux niveaux discutés précédemment (aimé et détesté). Ces deux niveaux représentent l'opinion d'un utilisateur face à un item. La valeur "aimé" signifie que l'utilisateur a été consommé l'item dans une situation confortable, tandis que la valeur "déteste" indique que l'utilisateur n'est pas satisfait de l'item proposé. Cette dernière valeur ne donne aucune confirmation que le contexte de l'utilisateur est inapproprié et peut être causée par plusieurs facteurs comme la (figure 4.1) l'indique clairement. La valeur "aucune" indique que l'utilisateur a ignoré de donner son avis concernant l'item proposé. Cela peut être causé par plusieurs facteurs y compris le contexte inapproprié. La couche de l'item représente l'état de l'item en termes de contenu et de contenant, c'est-à-dire, les différentes appréciations possibles d'un item par l'ensemble des utilisateurs. La couche item représente l'attention de l'utilisateur vis-à-vis de l'item (aimé, intéressé, déteste, désintéressé).

Nous déterminons les propriétés "intéresse" et "désintéressé" grâce à l'utilisation d'un compteur de temps, cela nous permet de déterminer si l'utilisateur est intéressé ou pas en comptant le temps écoulé sur un item.

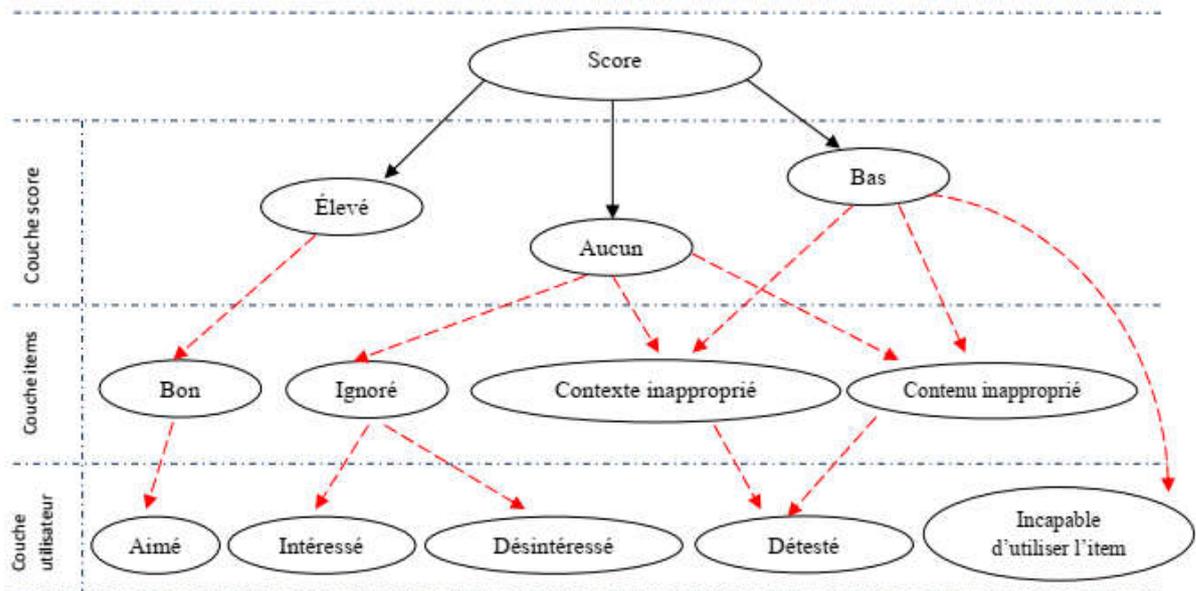


Figure 4.1. Base de connaissance de la matrice des votes

5. Profil d'utilisateur

Un profil d'utilisateur est un ensemble d'informations qui caractérisent un utilisateur spécifique dont un tel système de recommandation peut l'utiliser pour effectuer ses tâches d'adaptation et de recommandation. Généralement, un profil utilisateur est représenté comme un ensemble de mots-clés pondérés, réseaux sémantiques, concepts pondérés ou règles d'association (voir chapitre 3). Le plus courant c'est l'ensemble de mots-clés. Ces derniers peuvent être automatiquement extraits à partir de documents et / ou fournis par l'utilisateur lui-même. La construction de profil d'utilisateur est basée sur des informations qui proviennent de différentes sources d'information hétérogènes et utilise une diversité de méthodes de construction telles que les méthodes de l'extraction de l'information ou les méthodes de l'apprentissage automatique.

Dans ce travail, nous nous basons sur un modèle utilisateur simple. En effet, dans une optique d'indépendance du domaine, nous n'intégrons pas les multiples dimensions introduites dans l'état de l'art car certaines dimensions sont spécifiques à certains domaines, par exemple, le niveau de connaissances de l'utilisateur est spécifique aux systèmes adaptatifs éducatifs. Nous identifions trois catégories d'utilisateurs : les utilisateurs qui notent régulièrement les items

proposés par le système, les utilisateurs qui ignorent de noter les items à cause du contenu de ces derniers et les utilisateurs qui ignorent de noter les items proposés à cause de l'incompatibilité du contexte. Les différences de comportement des utilisateurs se basent sur les notes explicites et les notes déduites automatiquement par le système pour tous les utilisateurs, ce qui permet d'utiliser un processus de recommandation adaptable.

Le profil d'utilisateur dans ce travail se compose en trois parties essentielles : partie utilisateur, partie contexte et partie items. La partie utilisateur représente les informations personnelles de l'utilisateur et leurs compétences. Pour son première interaction, un utilisateur doit remplir un questionnaire sur les informations personnelles comme son nom, son rôle et ses compétences. Ensuite toutes activités de l'utilisateur impliquent une mise à jour des compétences de l'utilisateur à l'aide de certaines règles prédéfinies (voir section suivante). Finalement le profil de l'utilisateur sera restructuré automatiquement après tout changement dans l'historique de l'utilisateur. Les compétences de l'utilisateur sont définies sous forme clés-valeurs pondérées et qui représentent son niveau concernant un sujet. La partie contexte décrit les attributs contextuels qui influencent très souvent dans le comportement de l'utilisateur dans la phase d'attribution des notes aux items et par conséquent, influencent le processus de recommandation. Le type de contexte utilisé dans l'étude de cas de ce travail concerne le contexte matériel, c'est-à-dire le contexte qui est lié au dispositif matériel utilisé par l'utilisateur (taille de l'écran, résolution d'écran, etc.) et qui pourra servir à définir des contraintes spécifiques de présentation de documents multimédias.

Nous supposons dans ce travail que le contexte est défini sous forme d'un ensemble prédéfini d'attributs contextuels, dont la structure ne change pas avec le temps. La conséquence de cette supposition nous oblige à identifier les attributs importants qui agissent dans le processus d'attribution des notes aux items par chaque utilisateur, et ainsi, la méthode d'acquérir des informations contextuelles avant que les recommandations soient faites. Les décisions concernant les informations contextuelles qui devraient être pertinentes et collectées pour une application devraient être prises au stade de la conception de l'application et bien avant le moment où les recommandations sont fournies. Nous adaptons l'approche présentée dans [16], dont les auteurs ont proposé qu'un large éventail d'attributs contextuels soit initialement sélectionné par les experts du domaine comme candidats possibles pour les attributs contextuels pertinents de l'application.

La partie de l'item représente la description plus ou moins détaillée des items ainsi que leurs contextes d'utilisation. Le contexte d'utilisation indique les données nécessaires relatives à sa consommation adéquate par un utilisateur. Ces données sont utilisées par la suite pour générer des recommandations pertinentes pour chaque utilisateur. Dans notre étude de cas les items sont des objets pédagogiques ayant des propriétés multimédia qu'on suppose que sa consommation adéquate nécessite une certaine configuration qui devrait être présentée comme propriété du dispositif matériel demandant.

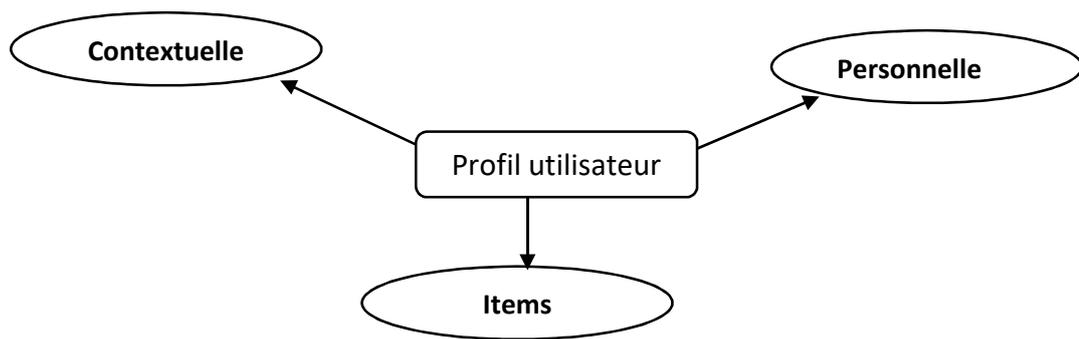


Figure 4.2. Profil d'utilisateur

6. Raisonnement

La principale contribution de ce travail est la détection des informations contextuelles appropriées et inappropriées d'un item donné en utilisant les données de notation (matrice des votes) fournies par le moteur de recommandation (explicites, prédites). La surveillance et l'analyse des comportements de l'utilisateur en tenant compte de la durée de la session utilisateur et des scores fournis offrent un moyen important de prédire les informations de contexte appropriées ainsi que les informations inappropriées. Ces données sont représentées par un ensemble d'informations concernant les ressources matérielles, ce qui nous permet de faire des recommandations pertinentes pour les utilisateurs cibles en prenant en compte toutes les informations récupérées. La stratégie que nous avons appliquée pour accomplir notre tâche est basée sur deux critères majeurs : le premier est le temps écoulé sur l'item, et le second est le taux global d'un item fourni par le moteur de recommandation.

Le raisonnement basé sur des règles sémantiques est une méthode puissante qui nous permet de dériver des informations contextuelles pertinentes et relativement facile à mettre en œuvre à l'aide de données fournies par des capteurs. Les informations qui sont obtenues à partir des capteurs de contexte ne peuvent pas être directement utilisées pour adapter un élément

arbitraire. Par conséquent, des informations contextuelles utiles peuvent être obtenues à partir de données de contexte selon un ensemble de règles définies pour chaque élément.

Il est important de déduire de nouvelles connaissances à partir des données brutes collectées (en utilisant des capteurs spécifiques) pour détecter le contexte approprié et adapter la génération des recommandations aux changements de contexte de l'utilisateur. Mais avant de pouvoir déduire de nouvelles connaissances, un certain traitement doit être pris en charge. Le traitement du contexte peut être divisé en agrégation et interprétation. Le premier fait référence à la composition des informations de contexte brutes, soit pour collecter toutes les données de contexte concernant un item spécifique, soit pour créer des informations de contexte de niveau supérieur. L'inférence peut être faite à l'aide de techniques de raisonnement sophistiquées qui reposent principalement sur la représentation du contexte. Par exemple, les techniques de raisonnement sémantique basées sur SPARQL peuvent être facilement réalisées si le modèle de représentation de contexte est basé sur OWL.

Les informations capturées sont utilisées pour formuler la base des dispositifs informatiques utilisés dans le système de recommandation ainsi que les votes fournis par les utilisateurs. Cela nous permet de définir des règles sémantiques permettant de déduire des nouveaux faits. Le tableau (table 4.1.) résume les règles proposées dans ce travail.

Règle1 détermine les utilisateurs ayant des compétences communes avec l'item en cours parmi les utilisateurs du système. En appliquant cette règle, notre système est capable de déterminer l'ensemble des utilisateurs potentiels qui peuvent participer au processus de notation. K représente les compétences de l'utilisateur sous forme d'une liste de mots-clés et k' représente la classification des items sous forme d'une liste de mots-clés, le plug-in *swrlb:ListIntersection* est utilisé afin de connaître les mots-clés communs entre les utilisateurs et l'élément, il est satisfait lorsque l'intersection entre les mots-clés de la liste (k) et la liste des mots-clés (k') n'est pas vide.

Nom	antécédent	conséquent
R1	$hasKeywords(?u, ?k), hasKeywords(?it, ?k'), swrlb: listIntersection(?k', ?k)$	$hasAbilityToRate(?u, ?it)$
R2	$hasAbilityToRate(?u, ?it), hasContext(?u, ?c), spend(?u, t), require(?it, t'), swrlb: greaterthan(t, t')$	$interest(?u, ?it)$
R3	$interest(?u, ?it), rate(?u, "high"), hasContext(?u, ?c)$	$hasSuitableContext(?u, ?c)$
R4	$hasAbilityToRate(?u, ?it), spend(?u, ?t), swrlb: lessthan(t, t'), require(?it, t'), rate(?u, "none")$	$ignoreWithoutInterest(?u, ?it)$
R5	$hasAbilityToRate(?u, ?it), spend(?u, ?t), swrlb: greaterhan(t, t'), require(?it, t'), rate(?u, "none")$	$ignoreWithInterest(?u, ?it)$
R6	$Create(?u, ?it), hasKeywords(?it, ?k)$	$hasCompetence(?u, ?k)$
R7	$rate(?u, "high"), hasKeywords(?it, ?k')$	$hasCompetence(?u, ?k')$
R8	$interest(?u, ?it), rate(?u, "none"), predictedScore(?it, "high"), globalscore(?it, "high"), hasContext(?u, ?c)$	$hasUnsuitableContext(?u, ?c)$

Table 4.1 règles de déduction

Règle2 vise à déterminer l'attention de l'utilisateur concernant un item (intéressé ou non). L'utilisateur est intéressé par un item lorsqu'il satisfait deux conditions : le premier est que l'utilisateur doit être appartient à l'ensemble des utilisateurs potentiels (les utilisateurs ayants des compétences pour évaluer l'item (regle1)) et la deuxième est que l'utilisateur doit être dépensé assez de temps sur l'item. Dans cette règle nous supposons que chaque item a besoin d'un certain temps de consultation (ou consommation) défini au début par le propriétaire de l'item. Alors, si l'utilisateur passe un temps suffisant sur l'item, le système décide que l'utilisateur a consulté complètement l'item, par conséquent, le système déduit que l'utilisateur est intéressé par cet item.

Règle3 présente notre proposition pour l'obtention des données contextuelles convenables. Par cette règle, notre système est capable de détecter les valeurs du contexte utilisateur consultant. Ces valeurs sont considérées comme contexte convenable pour la consommation de l'item de façon confortable. Cette règle est basée sur le fait que si l'utilisateur est intéressé par un item (Règle2) et ayant donné une note élevée à l'item alors le système considère que le contexte de l'utilisateur est approprié.

Règle4 et règle 5 visent également à déduire l'attention de l'utilisateur à propos d'un item. Ces deux règles sont basées sur un compteur de temps afin de connaître le temps passé sur

l'item. Les utilisateurs générés par ces deux règles sont des utilisateurs qui n'ont pas évalués les items proposés (score="aucun"). **Règle 4** a pour but de déterminer l'ensemble des utilisateurs qui ignorent les items mais restent intéressés par ces items tandis que la **règle 5** détermine l'ensemble des utilisateurs qui ignorent les items et qui font un coup d'œil rapide sur les items proposés (les utilisateurs désintéressés).

Règle6 et règle7 visent à sélectionner les acteurs qui participent généralement dans les différentes tâches fournies par le système (création d'un item, évaluation d'un item). Ces utilisateurs sont considérés comme utilisateurs potentiels s'ils appartiennent à l'un des ensembles suivants : l'ensemble qui englobe les propriétaires des items (créateurs) et l'ensemble des utilisateurs qui font évaluer un item avec un score élevé "high". Les compétences déduites sont des listes de mots-clés extraites à partir de l'item actif en utilisant les différentes méthodes d'extraction des mots clés (voir chapitre 5).

Enfin, **Règle8** a pour but de récupérer les valeurs de contexte inapproprié concernant la consommation d'un item, en se basant sur l'hypothèse discutée ci-dessus. Cette règle considère qu'un contexte inapproprié de l'utilisateur si un utilisateur n'évalue pas (ignorer) le contenu et passe un temps suffisant sur l'item et si son score prévu ou calculé est élevé "high" et que le score final de item déduit par le système de recommandation est élevé "high". Lorsque ces conditions sont vérifiées, le système enregistre les valeurs du contexte et l'identité de l'item en cours dans sa base de connaissance afin de générer des recommandations adaptables au contexte du demandeur. Par cette règle, le système suppose que les items ayant une valeur de vote prédite supérieure à un seuil donné sont identifiés comme pertinents pour l'utilisateur. Dans ce cas, le système prend en compte le contexte de ces items afin d'extraire les données contextuelles inappropriées.

7. Conclusion

Dans ce chapitre, nous avons présenté la base théorique de notre approche. Cette approche constitue un précieux outil permettant l'extraction des données contextuelles afin d'améliorer la tâche de recommandation des items. L'approche repose sur un modèle utilisateur et un ensemble de règles sémantiques en se basant sur une base de connaissance proposée dans le cadre de ce travail. Par la suite nous présentons un cas d'étude détaillé dans le domaine d'éducation permettant de valider notre approche proposée.

Chapitre 5

Etude de Cas *« E-learning System »*

Partie 1 : étude de cas détaillé

1.1. Introduction

Dans ce chapitre, nous allons présenter le cas d'étude proposé dans le cadre de ce travail. Notre cas d'étude est dédié pour l'éducation en ligne « E-Learning » dont on a proposé un environnement qui permet de créer et d'évaluer des objets pédagogiques. Notre système proposé s'inscrit dans le cadre des systèmes de production des objets pédagogiques, ce qui améliore la productivité dans le domaine de l'éducation. L'utilité de ce prototype réside dans la collaboration entre les différents utilisateurs du système afin de produire des objets pédagogiques mieux adaptés aux apprenants et convenables à leurs dispositifs informatiques. Nous avons focalisé sur les objets pédagogiques multimédia (contient des images, des portions vidéos, ...etc) afin d'étudier l'effet du contexte (contexte matérielles) sur l'utilisation de l'objet. Ces objets pédagogiques nécessitent pour ses consommations adéquates, une certaine configuration matérielle spécifique avec laquelle le dispositif consommateur doit être équipé. Les données de contexte exploitées dans ce travail sont les paramètres du dispositif ayants une influence sur la consommation régulière des objets pédagogiques notamment les objets multimédias [105]. Nous avons opté deux types de contexte physique : la taille de l'écran et la résolution d'écran, ainsi nous ajoutons un autre paramètre qu'est le débit de la connexion (la bande passante) en raison de leur rôle en transmission des données multimédias.

1.2. Architecture du Système

Le système que nous proposons est constitué d'un système de recommandation dédié pour la création des objets pédagogiques en intégrant les données de contexte et les données issues de l'analyse du comportement de l'utilisateur. L'objectif derrière cette approche est de construire une base adaptative d'objets pédagogiques accessible à la communauté d'utilisateurs (apprenants, enseignants, ...etc.). L'idée est de définir un profil utilisateur construit à partir des données de contexte sur les items et des données décrivant les items à partir de leurs appréciations pour ces mêmes items.

Notre approche repose sur deux hypothèses. La première hypothèse concerne le score fourni par un utilisateur à un item, qu'on suppose que le score élevé donné par un utilisateur implique que son contexte est approprié. La deuxième : dans certains cas, l'abstention de fournir un score à un élément est causée par l'incompatibilité des ressources de l'appareil de l'utilisateur avec l'item (exigences matériels).

Par la suite, nous décrivons notre système qui représente un outil destiné pour la communauté d'auteurs dans le domaine d'éducation, il permet aux utilisateurs de créer des objets d'apprentissage et / ou d'évaluer d'autres objets d'apprentissage créés par d'autres utilisateurs de la communauté. Nous supposons que nos objets contiennent des portions multimédias (séquence vidéo, image de type HD, ...etc.). Le système proposé présente deux avantages. Premièrement, il est destiné à aider les utilisateurs à créer de nouveaux objets d'apprentissage en fournissant un environnement collaboratif, dans lequel les utilisateurs intéressés peuvent participer à l'évaluation du contenu. Les utilisateurs qui participent à cette mission, peuvent avoir des problèmes causés par leurs contextes physiques (limitations matérielles des dispositifs) où ces utilisateurs ne fournissent aucun score à l'item proposé. Deuxièmement, nous avons essayé d'exploiter la liste de ces utilisateurs pour déterminer les données de contexte inappropriées, ce qui nous permet de générer d'éventuelles recommandations adaptables au contexte du demandeur. Notre système se compose de cinq modules : (a) Extracteur de métadonnées, (b) Calculateur de similarité, (c) Filtrage des utilisateurs, (d) Évaluation et prévision de données manquantes, dont les fonctions sont élaborées en dessous.

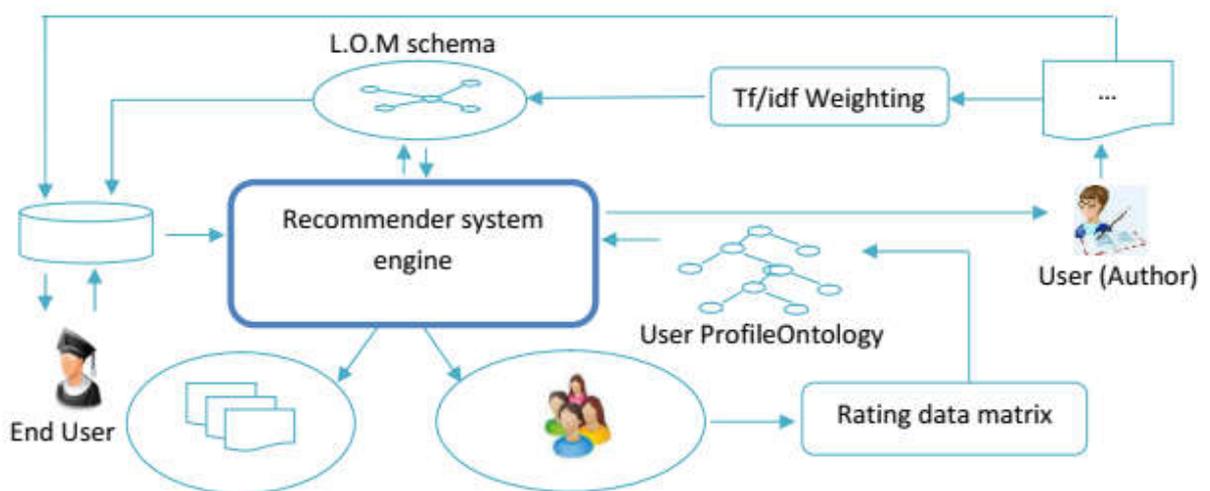


Figure 5.1. Architecture du Système proposé

1.2.1. Extracteur de métadonnées : ce module est chargé de remplir les éléments de métadonnées. Il utilise une interface pour la saisie de toutes les informations nécessaires. Ces informations peuvent être automatiques comme les nouveaux éléments proposés dans le cadre de ce travail (discutés plus tard) ainsi que toutes les informations qui peuvent être collectées de

façon automatique (date, heure, taille ...etc.) ou semi-automatique comme une liste de mots-clés générée automatiquement à l'aide de la formule *tf-idf* (détaillée ci-dessous), ou manuellement comme le nom du document, etc.

1.2.2. Calculateur de similarité entre les objets pédagogiques : ce module vise à trouver des objets d'apprentissage similaires à l'objet en cours à partir de la base de données du système en appliquant l'approche similarité cosinus. Les documents sont présentés sous forme vectorielle afin d'appliquer la formule de pondération *tf-idf*.

1.2.3. Filtrage des utilisateurs : ce module vise à récupérer l'ensemble des utilisateurs similaires en se basant sur l'algorithme *K-Nearest Neighbor*. Ce module utilise le coefficient de corrélation de Pearson et la liste de mots-clés générée par le module ci-dessus pour accomplir sa tâche. En outre, ce module a une tâche supplémentaire qui consiste à envoyer l'objet pédagogique à cet ensemble d'utilisateurs pour le processus d'évaluation.

1.2.4. Évaluation et prédiction des données manquantes : ce module est chargé de collecter les votes d'utilisateurs similaires et prédit toutes les données manquantes afin de calculer la moyenne entre eux. Ce module utilise le module de similarité des objets pédagogiques et le module de filtrage des utilisateurs pour effectuer sa tâche de prédiction.

Enfin, toutes les données produites (objet d'apprentissage et ses métadonnées) sont stockées dans une base de données pour un accès ultérieur par les étudiants, les auteurs, ...etc.

1.3. Ontologie des profils d'utilisateurs

Dans cette étude, nous adoptons le modèle représenté par les ontologies, dont on a représenté notre modèle en utilisant les langages informatiques standards tel que OWL(ontology web langage)[104]. Cette représentation nous permet de modéliser de manière structurée les éléments de notre contexte. L'ontologie est une moyenne puissante de représenter formellement les dépendances entre les différents composants du profil d'utilisateur y compris les différentes caractéristiques du contexte.

Les ontologies en tant que notion ont été déjà introduites dans le contexte du profilage des utilisateurs. Les ontologies utilisées cependant pour le profilage d'utilisateurs se limitent pour la plupart aux taxonomies des intérêts des utilisateurs. En réalité, la plupart des applications de profilage ne se limite pas aux intérêts de l'utilisateur mais englobe également d'autres caractéristiques de l'utilisateur (telles que l'éducation, l'expertise et le niveau de

connaissances), donc notre objectif est d'utiliser les dimensions nécessaires pour notre application dans la conception de notre ontologie de profil utilisateur.

Dans le présent cas d'étude, le type de contexte est représenté par deux paramètres qui sont : le débit de la connexion (bande passante) et support matériels pour les objets multimédias (qualité d'image, résolution d'écran). De plus, notre ontologie comprend les caractéristiques et les interactions des utilisateurs ainsi que les caractéristiques des items et les données du système de recommandation (les votes). Notre objectif était double. Nous avons d'abord tenté de définir le vocabulaire conceptuel mobilisé pour la représentation des savoirs dans les communautés des auteurs de ressources pédagogiques. Ensuite, nous avons tenté de réutiliser l'ontologie du domaine des ressources pédagogiques proposé dans la littérature.

Pour la création de l'ontologie, et afin de créer un modèle simple mais extensible et adaptable, les critères de conception de Gruber [71] (voir chapitre 3) (clarté, cohérence, extensibilité, codage minimal, engagement ontologique minimal) ont été pris en compte lors du processus de création. Nous avons adopté l'approche descendante (top-down); dont on a commencé par la sélection des concepts généraux importants, qui ont ensuite été enrichis et spécialisés. L'objectif de l'ontologie est le profilage de l'utilisateur dont ses caractéristiques sont plus ou moins permanentes.

Les concepts proposés dans cette ontologie sont importants pour le profilage de l'utilisateur. Nous avons opté pour notre proposition le travail de [72] qui définit un ensemble de concepts relativement importants dans le profilage statique des utilisateurs, les concepts sont : identité de l'utilisateur, caractéristiques, capacités, préférences universelles, préférences spécifiques à l'application. D'autres concepts tels que les caractéristiques du terminal actuel et les ressources pédagogiques qui sont créées ou évaluées sont inclus par cette ontologie car ils se réfèrent à un profil dynamique et représente le contexte de l'utilisateur. Les compétences (expérience de l'utilisateur) et les préférences concernant les items ont été identifiés et inclus dans l'ontologie du modèle d'utilisateur à des fins du processus de filtrage des utilisateurs ainsi que les items du système.

Les interactions des utilisateurs avec le système sont également intégrées dans notre ontologie. Elles représentent les votes ou les avis fournis par les utilisateurs aux items ainsi que leurs types qui sont organisés en hiérarchies de concepts.

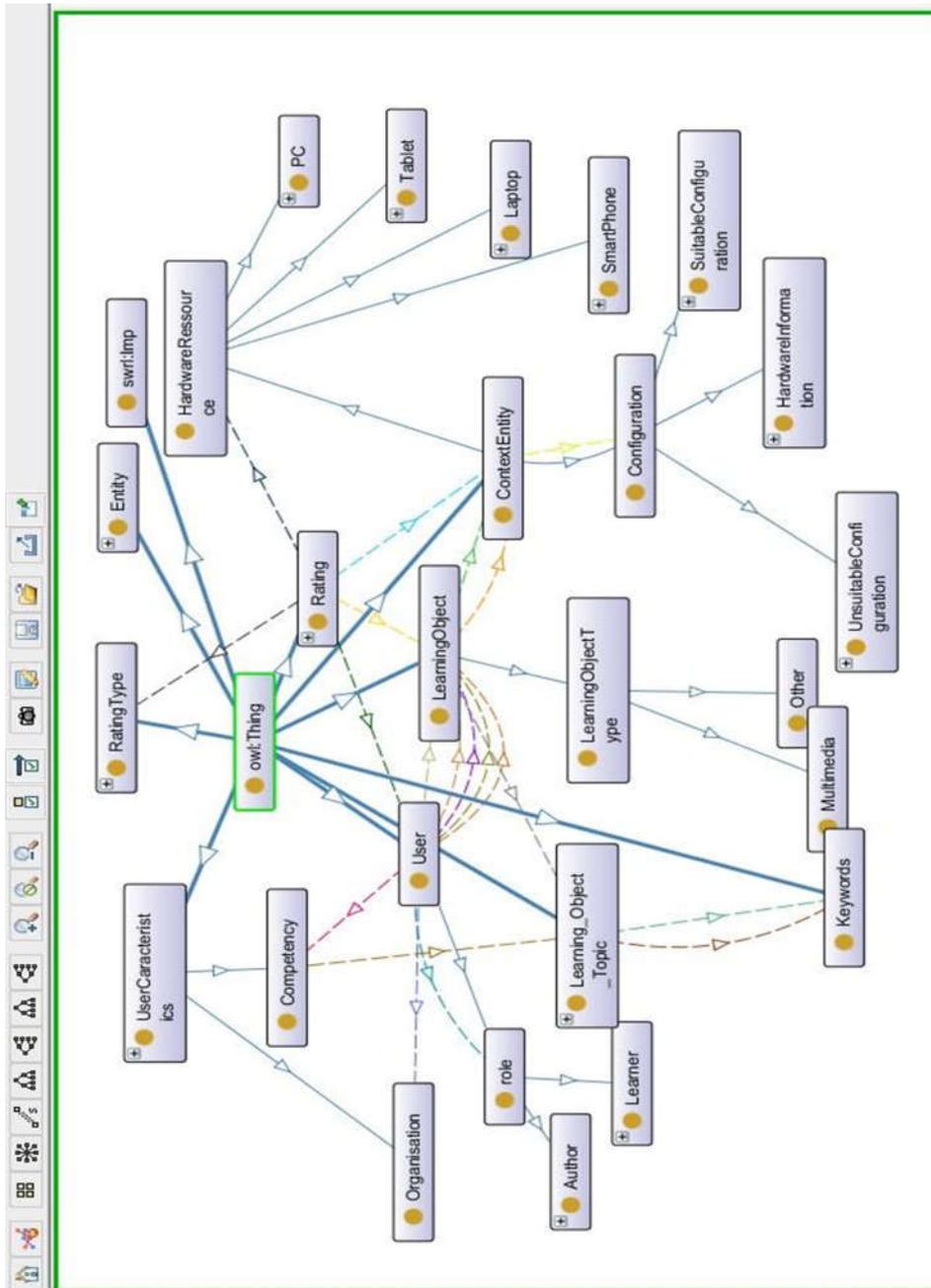


Figure 5.2. Ontologie du profil d'utilisateur

1.4. Indexation des objets pédagogiques

Dans cette section, nous présentons les étapes et les démarches suivies pour l'organisation de notre base d'objets pédagogiques. Nous commençons par une brève description des normes de métadonnées existant dans la littérature, ensuite nous présentons notre proposition concernant le profil d'application utilisé. Ce profil est proposé à des fins de préserver l'interopérabilité avec d'autres systèmes e-Learning.

1.4.1. Les normes de métadonnées

Dans de nombreux domaines de recherche, la manière la plus courante de décrire un objet est d'utiliser les métadonnées. Ces descripteurs sont importants dans le domaine de l'éducation pour l'accès, la récupération et la réutilisation de l'objet d'apprentissage. Le présent travail utilise un ensemble d'attributs de métadonnées (un schéma de métadonnées) afin de décrire le contexte de l'utilisateur et son environnement en indexant également les objets d'apprentissage.

Un objet d'apprentissage est une sorte d'élément numérique qui permet la réutilisation du contenu, l'indépendance et la flexibilité afin de donner une haute qualité de contrôle aux utilisateurs [73]. Cependant, pour obtenir une meilleure description des objets d'apprentissage, l'utilisation de métadonnées est nécessaire pour accomplir cette tâche.

La définition commune des métadonnées est celle des données sur les données, en effet, pour garantir l'interopérabilité avec d'autres systèmes, nous devons utiliser une norme. Par ce qui suit, nous détaillons les normes utilisées dans le domaine de l'éducation.

Le Dublin Core (DC), inventé par Dublin Core Metadata Initiative (DCMI), est un schéma de métadonnées simple qui est utilisé dans de nombreux travaux [74]. Ce schéma est présenté comme un ensemble de 15 éléments (titre, identifiant, langue, ...etc.), la clé principale pour utiliser ce schéma est celle compatible pour tous les domaines. En outre, de nombreux autres attributs supplémentaires sont inventés, appelés qualificatifs, qui affinent les 15 éléments de base pour augmenter l'efficacité de l'indexation des objets d'apprentissage.

Un autre schéma largement utilisé dans le contexte de l'éducation, qui est inventé par (IEEE) (Institute of Electrical and Electronics Engineers) permettant la description efficace des objets d'apprentissage. Ce schéma de métadonnées est utilisé dans de nombreux LOR (entrepôt d'objets d'apprentissage). Il est nommé *IEEE 1484.12.1-2002 Learning Object Metadata Standard* (LOM) [75]. Il fournit des catégories et chaque catégorie contient des éléments, dans son ensemble, LOM propose 76 éléments de données. La figure suivante montre le schéma LOM avec ses 76 éléments.

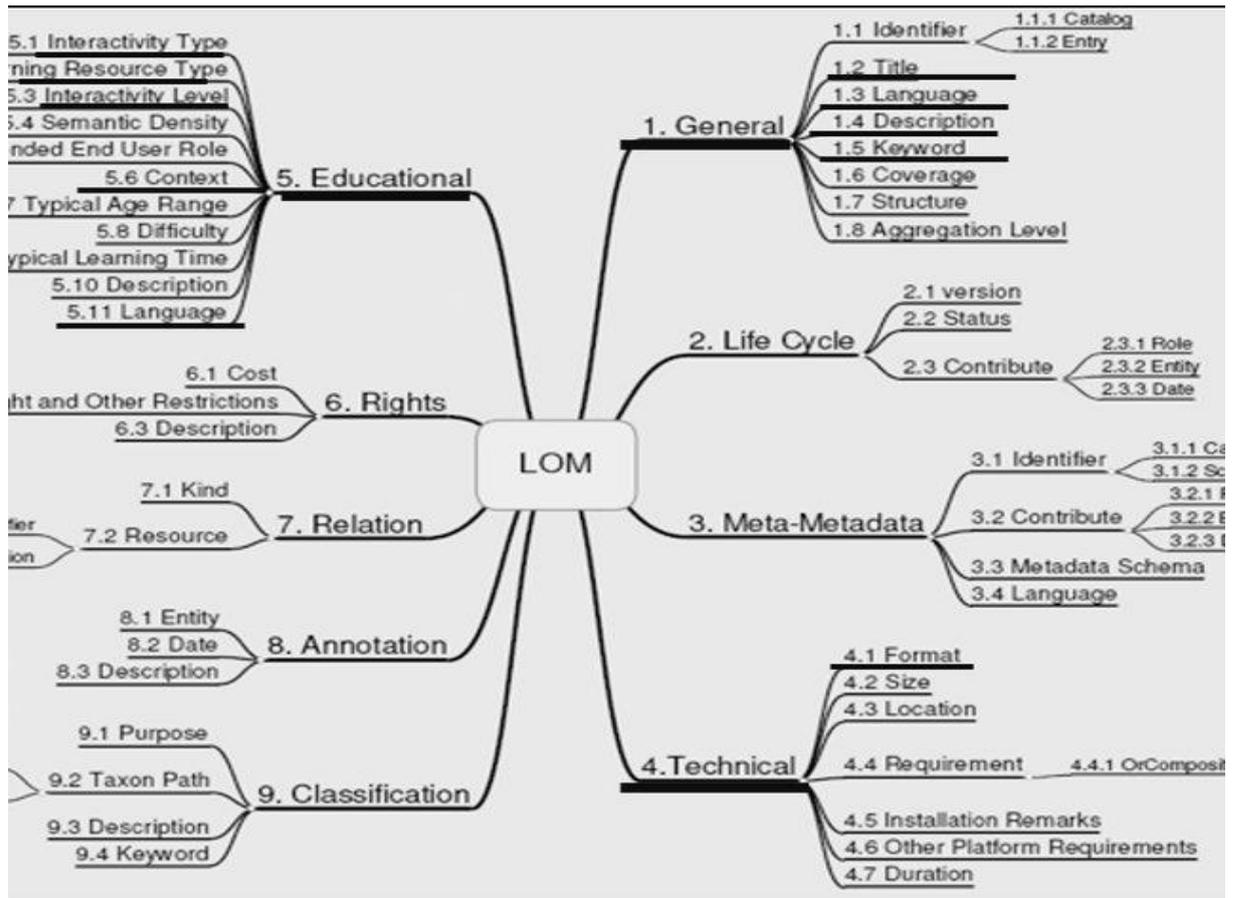


Figure 5.3. Learning Object Metadata (LOM)

1.4.2. Construction de métadonnées

Le type d'information de contexte étudié dans cette application est utile dans la phase de la distribution appropriée des objets d'apprentissage. Afin de récupérer les informations de contexte appropriées et inappropriées, nous devons collecter et stocker pour chaque participant, toutes les données de contexte utilisées dans la phase de l'évaluation (votes) (taille de l'écran, résolution de l'écran et bande passante Internet), donc pour accomplir cette tâche, nous avons proposé d'ajouter une extension au LOM [106]. Cette extension vise à préserver l'interopérabilité avec d'autres systèmes éducatifs et également à faciliter le traitement d'adaptation. Pour y parvenir, nous nous référons à [76] dont ils ont proposés une extension du LOM au MLM (Metadata Learning Mobile), les métadonnées d'apprentissage mobile, qui se composent de 3 catégories: 1) objet d'apprentissage, qui consiste en des informations décrivant la ressource d'apprentissage, 2) apprenant, qui consiste en informations décrivant l'apprenant, 3) Cadre général, qui consiste en des informations décrivant l'état du contexte de l'environnement d'apprentissage. Par conséquent, dans notre travail, nous avons proposé une extension du standard LOM afin de décrire nos objets d'apprentissages. L'extension proposée

est *Suitable_Contexte* et *Unsuitable_Contexte* à la catégorie technique du schéma LOM (branche 4.4.1.5 et 4.4.1.6). Cette extension permet de stocker les configurations appropriées (c'est-à-dire, configuration recommandée pour consommer correctement l'objet d'apprentissage) et les configurations inappropriées (configuration minimale requise pour consommer l'objet d'apprentissage). Le tableau suivant illustre les des éléments proposés

<i>Catégorie</i>	<i>Eléments</i>	<i>attributs</i>
4- Technical	4.4.1.5 Suitable_Context	4.4.1.5.1 Name
		4.4.1.5.2 Value
	4.4.1.6 Unsuitable_Context	4.4.1.5.1 Name
		4.4.1.5.2 Value

Table 5.1 Élément LOM proposés

1.4.3. Génération des éléments de métadonnées

Afin d'indexer le contenu d'apprentissage sur le sujet qu'il traite, nous avons conçu et implémenté un algorithme d'extraction de mots-clés. La formule la plus utilisée dans ce contexte est *tf-idf* (*Term frequency-inverse document frequency*). La formule *tf-idf* permet de déterminer dans quelles proportions certains mots d'un document texte, d'un corps de document ou d'un site web peuvent être évalués par rapport au reste du texte.

Pour utiliser (*tf-idf*), le document doit passer par de nombreuses phases, comme la *tokenisation* (les phrases se divisent en mots) et éliminer les *Stop-word* (c'est-à-dire les mots qui n'ont aucune signification pour le sujet) et enfin la *Racinisation* (en utilisant une analyse morphologique spécifique liée à la langue courante, chaque mot est abrégé à sa racine morphologique). La formule de *tfidf* est donné par :

$$w_{ij} = tfidf(t_i, d_j) = tf(t_i, d_j) * \log \frac{|D|}{tf(t_i, D)}$$

Où :

$tf(t_i, d_j)$ (*Term Frequency*) : Représente l'importance locale du terme dans un document. La méthode la plus simple de calcul est celle qui permet de déterminer le nombre d'occurrence du terme dans le document.

$IDF = \log \frac{|D|}{tf(t_i, D)}$ (*Inverse of Document Frequency*) : C'est la pondération globale des termes. Elle permet le calcul de l'importance d'un terme dans toute la collection. Elle permet de réduire l'importance des termes qui se trouvent souvent dans un document, mais qui ne représentent pas le sens global du document. Ces termes apparaissent fréquemment dans toute la collection.

À la suite de cette phase, nous obtenons une représentation vectorielle ordonnée du document d_j sous la forme vectorielle (terme | poids).

$$d_j = \{(w_1), (w_2), (w_3), \dots\}$$

Dont $w_1 > w_2 > w_3 > \dots$

Le résultat est trié selon w_i afin de sélectionner les N premiers mots (Top-N) qui sont considérés comme des mots-clés candidats pour le document. Notre système offre la possibilité aux auteurs de changer, d'éditer ou d'étendre la liste de mots-clés donnée par le système afin de surmonter certaines limitations reconnues par l'approche *TF-IDF* [21], [14]. L'exemple suivant montre les métadonnées encodées en XML [77].

```
<lom:general>
<lom:title>
<lom:string language="en">
    Title of the Learning Object
</lom:string>
</lom:title>
<lom:language>en</lom:language>
<lom:keyword weight="0.34">
<lom:stringlanguage="en">Keyw_1</lom:string>
</lom:keyword>
<lom:keyword weight="0.28">
<lom:stringlanguage="en">Keyw_2</lom:string>
</lom:keyword>
</lom:general>
```

1.4.4. Evaluation des objets pédagogiques

Après la construction des métadonnées, notre système accède à la base de données des utilisateurs afin de retrouver un ensemble d'utilisateurs similaires. Cette phase a pour but de collecter les différents scores de l'objet d'apprentissage en cours. L'objectif derrière cette idée est de profiter de l'expérience des auteurs afin d'obtenir une note finale du contenu

d'apprentissage. Pour y parvenir, nous nous référons à la technologie des systèmes de recommandation qui fournit les techniques pertinentes utilisées par ce travail.

Dans le domaine de l'apprentissage assisté par la technologie de l'information et de la communication (TEL), il existe de nombreux travaux axés sur le système de recommandation pour récupérer un objet d'apprentissage pertinent et adapté à l'utilisateur final (étudiants). Les auteurs dans [96] ont proposé un système pour la recommandation des ressources d'apprentissage, il intègre un module de filtrage collaboratif qui fonctionne avec les notes fournies par les utilisateurs et équipé d'un moteur de règles d'inférence. Une autre étude [6] est l'outil LORM (Learning Object Recommendation Model). Il utilise une méthode hybride qui recommande un objet d'apprentissage basé sur les préférences et la corrélation entre les objets. Cet outil a utilisé un modèle ontologique pour effectuer une découverte sémantique.

Comme une synthèse, ces travaux sont basés sur une communauté d'apprenants, ce qui présente une certaine limite car l'apprenant n'est pas considéré comme étant un expert et ses évaluations restent particulières. On trouve dans [97] un examen du système de recommandation en se concentrant sur les enseignants en tant que une communauté d'experts.

1.4.4.1. Calcule de similarité

Dans la littérature, la similarité cosinus [34] est fréquemment utilisée pour tenter de déterminer la similarité entre deux documents. Généralement, le document est représenté sous forme vectorielle et la similarité cosinus donne la similarité deux vecteurs à n dimensions en déterminant le cosinus de leur angle. Étant donné deux documents, A et B, la similarité cosinus entre A et B est calculée par la formule suivante

$$Similarity = \frac{A \cdot B}{\|A\| \cdot \|B\|} = \frac{\sum_{i=1}^n A_i \cdot B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \sqrt{\sum_{i=1}^n (B_i)^2}}$$

Dans ce travail, nous avons utilisé la similarité cosinus pour calculer la similarité entre un document donné et tous les documents du système afin de recommander un sous-ensemble de documents qui sont considérés comme pertinents.

1.4.4.2. Filtrage des utilisateurs potentiels et prédiction

Cette étape consiste à collecter les données d'évaluation (les scores) sur l'objet d'apprentissage en cours auprès de tous les participants afin de calculer le moyen des scores

fournis. Notre choix pour donner le cluster d'un enseignant est l'algorithme K-Nearest Neighbor.

K-Nearest Neighbor, connu sous le nom de filtrage collaboratif utilisateur-utilisateur, est un algorithme d'apprentissage supervisé utilisé pour la prédiction, l'estimation et la classification [78],[98]. L'objectif d'utiliser cette méthode est de donner des prédictions pour les objets d'apprentissage qui n'ont pas été évalués par des utilisateurs. Le processus se déroule comme suit :

a) Calculer les similarités entre chaque utilisateur actif (T_i) et les utilisateurs du système (T_j). La méthode la plus utilisée dans le domaine de filtrage collaboratif est la méthode PCC (Coefficient de corrélation de Pearson) [78] défini comme suit :

$$sim(x, v) = \frac{\sum_i^N (r_{x,i} - \underline{r}_x)(r_{v,i} - \underline{r}_v)}{\sqrt{\sum_i^N (r_{x,i} - \underline{r}_x)^2} \sqrt{\sum_i^N (r_{v,i} - \underline{r}_v)^2}}$$

Où N représente les objets notés à la fois par x et v , $r_{x,i}$ est l'ensemble des objets noté par l'utilisateur x et \underline{r}_x est l'évaluation moyenne de l'utilisateur x .

b). Sélectionner N utilisateurs indiqués à l'étape 1. (N représente le nombre maximum d'utilisateurs sélectionnés)

c) prédire le score de l'objet d'apprentissage :

Un des problèmes reconnus dans l'approche de filtrage collaboratif est lorsqu'un ou plusieurs utilisateurs ne souhaitent pas évaluer un l'objet. Dans cette situation, nous devons prédire les différents scores d'évaluation. Donc, après le calcul de similarité, le système construit un voisinage N pour chaque utilisateur et prédit la note de l'utilisateur U pour l'objet d'apprentissage en cours en utilisant la formule ci-dessous :

$$P = \underline{r} + \frac{\sum_w sim(u, u')(r_w - \underline{r}_w)}{\sum_w sim(u, u')}$$

Notre outil est destiné aux auteurs afin de les aider à atteindre leurs objectifs dans la tâche de création du contenu éducatif pertinent. Ce système peut être utile pour les auteurs novices qui sont fortement pris en charge par notre système. Cependant, les compétences d'un nouvel auteur sont inconnues pour notre base de données (cette situation est connue sous le nom de démarrage à froid) (voir chapitre 2).

Dans notre contexte, nous avons considéré que les nouveaux utilisateurs utilisent notre système pour créer des nouveaux objets d'apprentissage. Nous adoptons les informations de contenu du nouvel objet pour calculer la similarité avec les objets existants dans la base et extraire par la suite l'ensemble des utilisateurs potentiels. La tâche de production d'un contenu pédagogique fournit un proxy solide pour obtenir les compétences des utilisateurs [79] (règle6). Cependant, et afin d'entendre la liste des utilisateurs, nous adaptions aussi les données des évaluations fournies (règle7), car le fait qu'un score élevé pourrait impliquer que l'utilisateur a vraiment utilisé l'objet ou, au moins il était à l'aise avec l'objet [79]. Plus formellement, la liste des utilisateurs est :

$$N = \{A_c \cup A_r\}$$

Où A_c représente l'ensemble des utilisateurs qui ont créés un ou plusieurs objets d'apprentissage et A_r représente l'ensemble des utilisateurs qui ont évalués un ou plusieurs objets d'apprentissage. Cette formule vise à récupérer tous les utilisateurs qui ont participé en évaluant ou en créant un ou plusieurs objets d'apprentissage similaires à l'objet d'apprentissage en cours. Cet ensemble d'utilisateurs est donné par la règle6 et la règle7 (voir chapitre 4). Cette formule peut conduire à une grande liste d'utilisateurs (base de données augmente au fil du temps). Pour cette raison, nous utilisons la formule ci-dessous afin de pondérer la liste des utilisateurs pénitentiels (sélection des N premiers utilisateurs).

$$P_i = \sum_{k=1}^M \#(C_k) + \beta \sum_{k=1}^M \#(N_k)$$

Où C_k représente le nombre de fois où le mot-clé k apparaît dans les documents créés par l'utilisateur I et N_k représente également le nombre de fois où le mot-clé k apparaît dans les documents évalués par l'utilisateur I . Le facteur β est une constante qui peut être paramétrée en fonction de l'activité dans le système de pondération de la tâche de création vis-à-vis de la tâche de l'évaluation, sa plage est comprise entre (0,1).

Après avoir collecté tous les scores des utilisateurs (réels et prédits), le système calcule la moyenne (qui représente le score final de l'objet d'apprentissage). Ce résultat est utilisé pour mettre à jour et/ou créer le profil de l'utilisateur et/ou notifier l'utilisateur de réviser son objet d'apprentissage si le score final était inférieur à un seuil ajusté par l'utilisateur actif.

$$Avg = ScoreFinal_{Lo} = \frac{\sum_i^n r_i}{N}$$

Les nouveaux éléments de métadonnées proposés dans ce travail sont remplis automatiquement à l'aide de règles prédéfinies. Les informations de contexte appropriées et les informations de contexte inappropriées sont récupérées à l'aide de règle3 et règle8 respectivement. Ainsi, après avoir collecté les données, nous appliquons l'algorithme ci-dessous afin de récupérer des informations de contexte qui sont représentées sous forme de vecteur (la bande passante, la taille de l'écran, résolution de l'écran) relatives à l'objet d'apprentissage.

Input : dataset of suitable and unsuitable context

Output : suitable Context vector **and** unsuitable context vector

Foreach element in (suitable_Context) **do**

If suitable_Context [i] <= **one of** (Unsuitable_context[i]) **then**

Clear (Unsuitable_context [i]) ;

Suitable_Context := **min**(Suitable_Context[i]);

Foreach elements in Unsuitable_Context **do**

Unsuitable_Context := **max** (Unsuitable_Context [i])

L'exemple suivant montre les données de contexte appropriées et inappropriées obtenues. Le problème auquel nous sommes confrontés dans une telle situation est de savoir comment prendre une décision pour l'utilisateur final sur l'adéquation du contexte qui peut prendre n'importe quelle valeur

	Score	resolution (Mpixels)	Size (inch)	Bandwidth (kb/s)	suitable
U1	aimé	1,2	4	1,024	yes
U2	aimé	0,8	3,5	7,168	yes
U3	aimé	0,9	6	0,512	yes
U4	-	2,1	5	0,128	no
U5	aimé	2,2	19	0,064	yes
U6	aimé	2,1	15	7,168	yes
U7	-	1,2	3,5	2,048	no

Table 5.2 Exemple d'utilisation

1.4.4.3. Intégration du contexte dans le processus de recommandation

Après avoir exécuté l'algorithme, notre système obtiendra l'ensemble des informations appropriées et inappropriées (C_s) et (C_{us}). Cet ensemble de données est considéré comme un

ensemble de formation utilisé pour générer un modèle de décision pour toute demande de contenu d'apprentissage effectuée par les utilisateurs finaux (apprenants) en tenant compte de leur contexte (C_i). Le code ci-dessous montre la tâche de prédiction.

```

Input :  $CS, CUS, C_i$ 
Output: Suitability or Unsuitability of  $C_i$ 
If  $((C_i[K] > CS[k]) \text{ or } ((C_i[k] < CUS[k]))$  then
Begin
If  $(C_i[k] > CS[k])$  then
    The user context is suitable
If  $(C_i[k] < CUS[k])$  then
    The user context is not suitable
End
Else perform_suitability ( $C_i$ );
  
```

Où K désigne le type de contexte (résolution, taille de l'écran, bande passante) et *perform_suitability* est une fonction à un paramètre représentant les données de contexte de l'utilisateur final, et renvoie la probabilité concernant l'adéquation du contexte C_i .

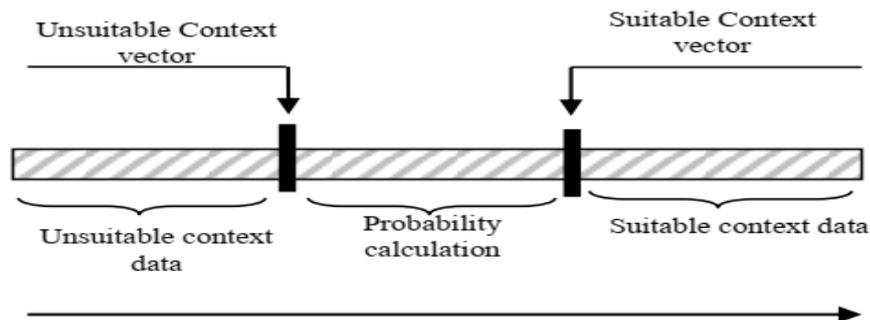


Figure 5.4. Vecteur de données de contexte

Dans ce travail, nous avons adopté la méthode bayésienne pour estimer la probabilité qu'un contexte donné appartient ou non à une classe spécifique (appropriée et inappropriée). Le naïve bayes est un algorithme puissant qui fournit un traitement rapide et haute précision avec les données de grande échelle par rapport à celui des algorithmes utilisés pour la classification tel que les réseaux neurones ou les arbres de décision [98].

Le calcul de la probabilité en utilisant le naïve bayes est donné par la formule suivante

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

Où X et Y sont deux événements différents, $P(X)$ est la probabilité que X se produise, $P(X|Y)$ est la probabilité que X se produit sachant que Y s'est déjà produit. Notant que, la probabilité a

priori $P(C_i)$ est égale à 0.5 (car nous avons deux catégories de classes (appropriées et inappropriées)).

Les attributs contextuels utilisés dans ce cas d'études sont des variables numériques ayant un nombre infini de valeurs entre deux valeurs (données contenues). Par conséquent, la distribution de probabilité de (x_i) étant donné une classe C , $p(X=x_i | C)$ peut être calculée en introduisant (x_i) dans l'équation de distribution normale (gaussienne) qui est paramétrée par la moyenne μ et écart-type σ . Plus formellement : $p(y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \exp \left(-\frac{(x_i-\mu)^2}{2\sigma^2} \right)$

$$\text{Dont } \mu = \frac{\sum_{i=1}^n x_i}{n} \text{ et } \sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2$$

Nous utilisons la formule ci-dessous pour déterminer la classe de l'item en cours en comptant la classe qui produit la probabilité la plus élevée

$$\begin{aligned} P(C_i) &= P(x_1, x_2, \dots, x_n | C_i) P(C_i) \\ &= P(x_1 | C_i) P(x_2 | C_i) \dots P(x_n | C_i) P(C_i) \\ &= P(C_i) \prod_{k=1}^n P(x_k | C_i). \end{aligned}$$

$$C = \operatorname{argmax} P(C_i) \prod_{k=1}^n P(x_k | C_i) \text{ Où } C \text{ représente la classe de l'objet en cours.}$$

Partie 2 : Implémentation et expérimentation

Après avoir présenté les démarches suivies pour la réalisation de notre prototype, qui intègre les modèles de recommandation ainsi que les modèles d'extraction du contexte, nous allons présenter dans cette partie les résultats d'une expérimentation menée dans un contexte éducatif réel. L'objectif de cette expérimentation est de mesurer la pertinence de l'approche que nous avons proposée.

Dans ce travail, nous avons développé un outil, qui implémente notre approche et nous l'avons expérimenté avec des utilisateurs dans un contexte réel d'apprentissage. Cet outil a pour but d'améliorer la tâche de la production des objets pédagogiques en offrant un environnement collaboratif. Il se compose d'un ensemble de fonctionnalités fournies afin d'aider les auteurs à savoir la fiabilité du matériel pédagogique créé. L'utilisateur de notre système doit être enregistré ou connecté via une interface fournie par le système.

2.1. Présentation de l'ensemble de données (*dataset*)

D'après [41] les systèmes de recommandations basés sur le filtrage collaboratif peuvent être évalués en utilisant 1) une expérience utilisateur avec des utilisateurs réels ou 2) une analyse hors ligne (offline) avec un *dataset* naturel ou synthétique. Dû à l'absence d'une *dataset* adéquate pour notre travail, nous avons opté pour l'expérimentation avec des utilisateurs réels.

Pour évaluer ce travail, nous avons fait une expérience avec des utilisateurs réels. L'expérience s'est déroulée avec 65 utilisateurs (enseignants et doctorants) du département de technologie de l'Université de M'sila en Algérie, dont on a mis 28 objets pédagogiques de différents formats disponibles librement pour toutes utilisations. Ces objets contiennent des portions multimédias afin de tester les supports multimédias du demandant par conséquent, les participants sont équipés par des dispositifs informatiques de différentes configurations.

Les participants ont été invités à créer des supports de cours et/ou à interagir avec les autres supports qui sont créés par d'autres participants tout en collectant les informations contextuelles de leurs appareils utilisés. Le tableau suivant présente un exemple de 11 participants avec leurs configurations matérielles ainsi que leurs évaluations données directement (7 participants) ou estimées (car elles ne sont pas fournies) par le système (4 participants) à un objet pédagogique.

	Score	Resolution (M-p)	Size (inch)	B.W (Mb/s)
Utilisateur 1	aimé	1,2	4,6	1,024
Utilisateur 2	aimé	0,8	4,5	2,048
Utilisateur 3	aimé	0,9	6	0,512
Utilisateur 4	-	0,4	3,8	0,128
Utilisateur 5	aimé	2,2	5	0,64
Utilisateur 6	aimé	2,1	5	1,048
Utilisateur 7	-	0,5	3,5	0,128
Utilisateur 8	aimé	1,3	4,4	1,024
Utilisateur 9	-	0,6	3,3	0,056
Utilisateur 10	-	0,5	3,5	0,128
Utilisateur 11	aimé	1,2	5	1,024

Table 5.3 exemple de caractéristiques matériels utilisés

Où (-) indique que le score n'est pas fourni. L'attention de l'utilisateur "aimé" impliquait que l'utilisateur a donné un score élevé à l'objet en cours.

Les participants qui n'ont pas évalué les objets qu'ils auraient été proposés ont été interrogés par la suite avec un questionnaire afin de connaître les raisons de leur abstention d'évaluer les objets proposés. 75% de ces participants ont déclaré l'incapacité de consulter ces ressources à cause de leurs configurations matérielles (notamment le débit de l'internet). Le tableau suivant illustre la participation des utilisateurs devant 04 objets pédagogiques multimédias.

	objet1	objet2	objet3	objet4
a) Nombre d'évaluations	49	32	25	30
b) Nombre des mentions « j'aime »	37	25	22	22
c) Nombre des mentions « Détesté »	12	7	3	8
d) Nombre des abstentions	6	14	25	8
e) Nombre des participants	55	46	50	38

Table 5.4 un extrait de l'ensemble de données

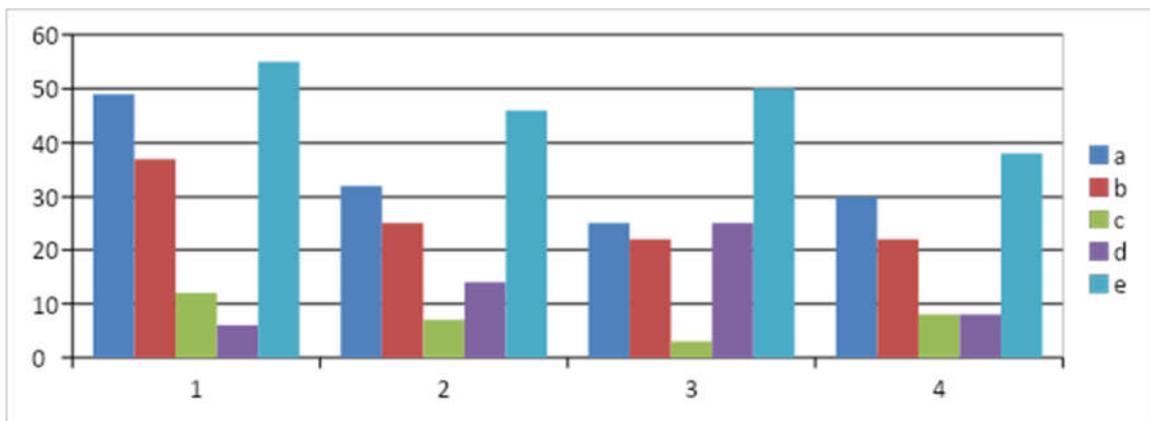


Figure 5.5. Histogramme des utilisations

Nombre des abstentions	6	14	25	8
Nombre des utilisateurs actifs	4	11	17	5
%	67%	79%	68%	88%

Table 5.4 utilisateurs interrogés

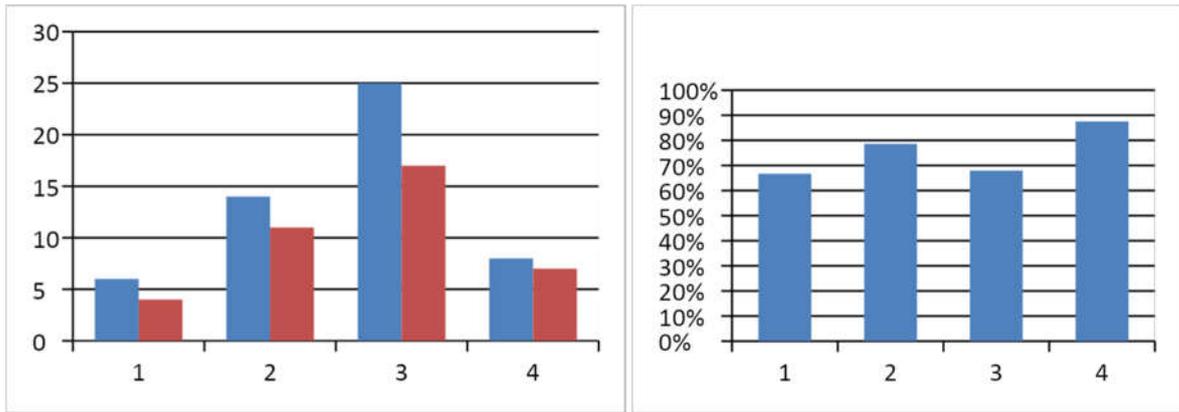


Figure 5.6. Histogramme des utilisateurs interrogés

2.2. Evaluation du système de recommandation proposé

Le travail proposé de ce manuscrit se compose de deux parties. La première est basée sur le filtrage collaboratif afin d'obtenir la liste des utilisateurs potentiels pour chaque objet pédagogique. La deuxième partie consiste à extraire les informations contextuelles et de les intégrer dans le processus de recommandation.

Pour vérifier si la recommandation proposée par notre système est efficace et satisfaisante du point de vue de l'utilisateur, nous avons utilisé le rappel, F-mesure et la précision pour évaluer les mesures d'exactitude de l'algorithme de recommandation (voir chapitre 2).

Dans la littérature, les résultats obtenus par les algorithmes de recommandation contiennent deux ensembles d'utilisateurs nommés participants positifs et participants pertinents. Dans notre cas, les participants positifs récupérés sont les utilisateurs qui ont évalués le contenu d'apprentissage et l'ensemble des participants pertinents récupérés forme l'ensemble des utilisateurs qui ont ignorés de fournir leurs évaluations. Cette dernière est composée de deux sous-ensembles : participants pertinents négatifs et de participants pertinents positifs (les utilisateurs qui ont des contextes inappropriés).

Précision et rappel sont des mesures qui donnent des résultats entre 0 et 1, l'objectif est de permettre aux systèmes de tendre vers 1. Quatre cas sont possibles :

1. Les vrais positifs VP : l'utilisateur est recommandé, et il correspond bien à l'objet.
2. Les faux positifs FP : l'utilisateur est recommandé, mais n'est pas sélectionné.
3. Les faux négatifs FN : l'utilisateur n'est pas recommandé, mais il est sélectionné.

4. Les vrais négatifs VN : l'utilisateur n'est pas recommandé, ce qui est correct.

$$precision = \frac{VP}{VP + FP}$$

$$Rappel = \frac{VP}{VP + FN}$$

$$F - mesure = \frac{2 * precision * rappel}{precision + rappel}$$

	objet1	objet2	objet3	objet4
VP	49	32	25	30
FP	4	10	5	8
VN	5	12	6	14
FN	2	3	8	3

Table 5.5 données des mesures

	objet1	objet2	objet3	objet4	%
précision	0,9245283	0,7619 048	0,833333 3	0,789473 7	83%
rappel	0,9607843	0,9142 857	0,757575 8	0,909090 9	89%
f-mesure	0,9423077	0,8311 688	0,793650 8	0,845070 4	85%

Table 5.6 Mesures de performance

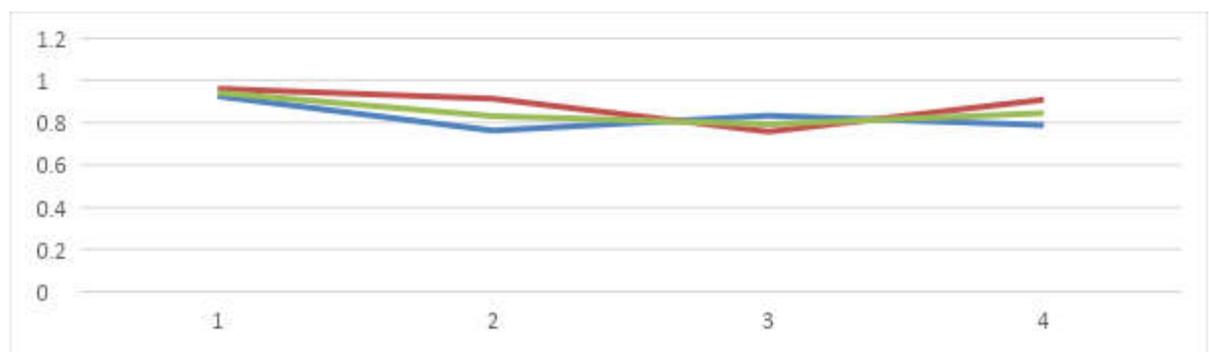


Figure 5.6. Précision, rappel et f-mesure

2.3. Critique

L'évaluation présentée au-dessus se fait sur un échantillon de 04 objets sélectionné au préalable. A travers le processus de l'évaluation, nous avons rencontré des problèmes relatifs aux utilisateurs du système ainsi que les items choisis pour la phase d'évaluation. Certain items

envoyés aux utilisateurs ont reçu que peu d'évaluations de la part des participants, tandis que d'autres objets ont reçu des évaluations de taux compris entre (30% et 40%). D'autres participants ont refusé d'utiliser le système à cause du contenu proposé. Les résultats présentés dans cette évaluation montrent l'efficacité de l'algorithme de filtrage des utilisateurs et cela revient aux objets sélectionnés qui présentent un contenu abordable pour les participants, ainsi que les portions multimédias inclus dans ces objets qui nécessitent en général une configuration moyenne pour les consulter.

2.4. Evaluation de la phase d'intégration du contexte dans le processus de recommandation

Le but de la deuxième partie de notre approche est d'avoir prendre la décision qu'une configuration spécifique d'un demandant qui est représentée sous forme d'un vecteur (i, j, \dots) est appropriée ou non pour l'utilisation d'un objet d'apprentissage. En effet, cette deuxième partie consiste à classer ce vecteur en utilisant deux classes (appropriée, inappropriée). Nous adaptons pour notre problème de classification la méthode bayésienne (discuté dans la partie précédente) dont on a construit une base de connaissance (training data) afin d'appliquer cette méthode. Le tableau suivant montre un extrait de l'ensemble de données d'entraînement.

	Score	Resolution (M-p)	Size (inch)	B.W (Mb/s)	Contexte
Utilisateur 1	aimé	1,2	4,6	1,024	adéquat
Utilisateur 2	aimé	0,8	4,5	2,048	adéquat
Utilisateur 3	aimé	0,9	6	0,512	adéquat
Utilisateur 4	aimé*	0,4	3,8	0,128	inadéquat
Utilisateur 5	aimé	2,2	5	0,64	adéquat
Utilisateur 6	aimé	2,1	5	1,048	adéquat
Utilisateur 7	aimé*	0,5	3,5	0,128	inadéquat
Utilisateur 8	aimé	1,3	4,4	1,024	adéquat
Utilisateur 9	aimé*	0,6	3,3	0,056	inadéquat
Utilisateur 10	aimé*	0,5	3,5	0,128	inadéquat
Utilisateur 11	aimé	1,2	5	1,024	adéquat

Table 5.7 données d'entraînement

Après avoir estimé les données manquantes dans l'exemple précédent, nous appliquons nos règles de déduction afin d'obtenir les deux vecteurs représentant le contexte approprié (0.8, 4.4, 0.512) et le contexte inapproprié et (0.6, 4, 0.256).

Afin de classer une donnée d'entrée par exemple (0.9, 3.2, 0.366) qui représente respectivement la résolution de l'écran, la taille de l'écran et la bande passante, nous calculons la probabilité en utilisant la méthode de classification gaussienne. Pour l'exemple ci-dessus, nous obtenons les probabilités suivantes : $P(\text{yes}) = 3.2391 \times 10^{-4}$ et $P(\text{no}) = 1.7480 \times 10^{-7}$ ce qui nous permet de conclure que cette configuration est peut être convenable pour utiliser l'objet.

Afin d'identifier les erreurs de classification courantes, nous avons calculé la matrice de confusion [99] en utilisant la plateforme Matlab¹ dont on a développé un module pour cette raison. La matrice de confusion contient des informations sur les classifications réelles et prévues effectuées par un système de classification. Une matrice de confusion illustre l'exactitude de la solution à un problème de classification.

Notre matrice de confusion montre que la précision de la classification est très encourageante avec des erreurs mineures comme la figure ci-dessous montre.

```

confusion matrix:
    10     2
     1    22

accuracy = 91.4286%
Elapsed time is 1.011179 seconds.
fx

```

Figure 5.7. Matrice de confusion

2.5. Implémentation

Dans ce travail, nous avons implémenté un serveur web constitué d'une base de données qui stocke le contenu d'apprentissage, les métadonnées, les données d'évaluation et les profils des utilisateurs. Le côté client représente une interface simple permet des fonctionnalités pour l'établissement de la tâche de création (figure1) (une partie de cette plateforme a été développée dans le cadre de projets de fin d'étude de master [103]).

¹ <https://www.mathworks.com/products/matlab.html>

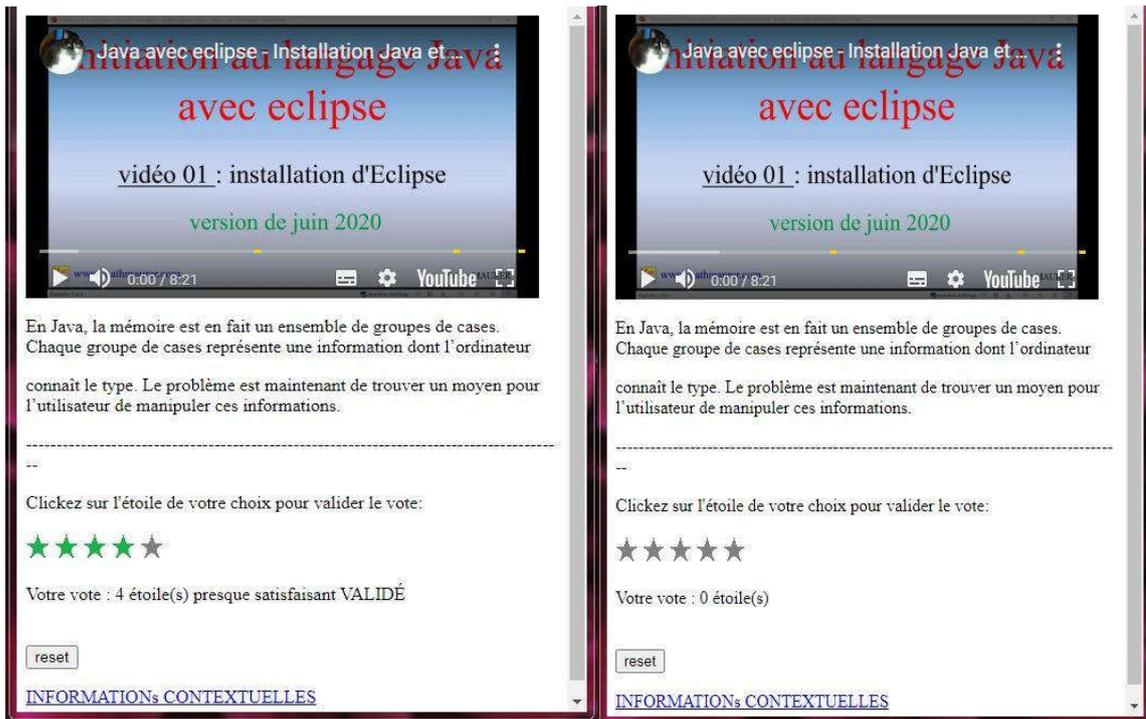


Figure 5.9. Interface pour l'évaluation des objets pédagogiques

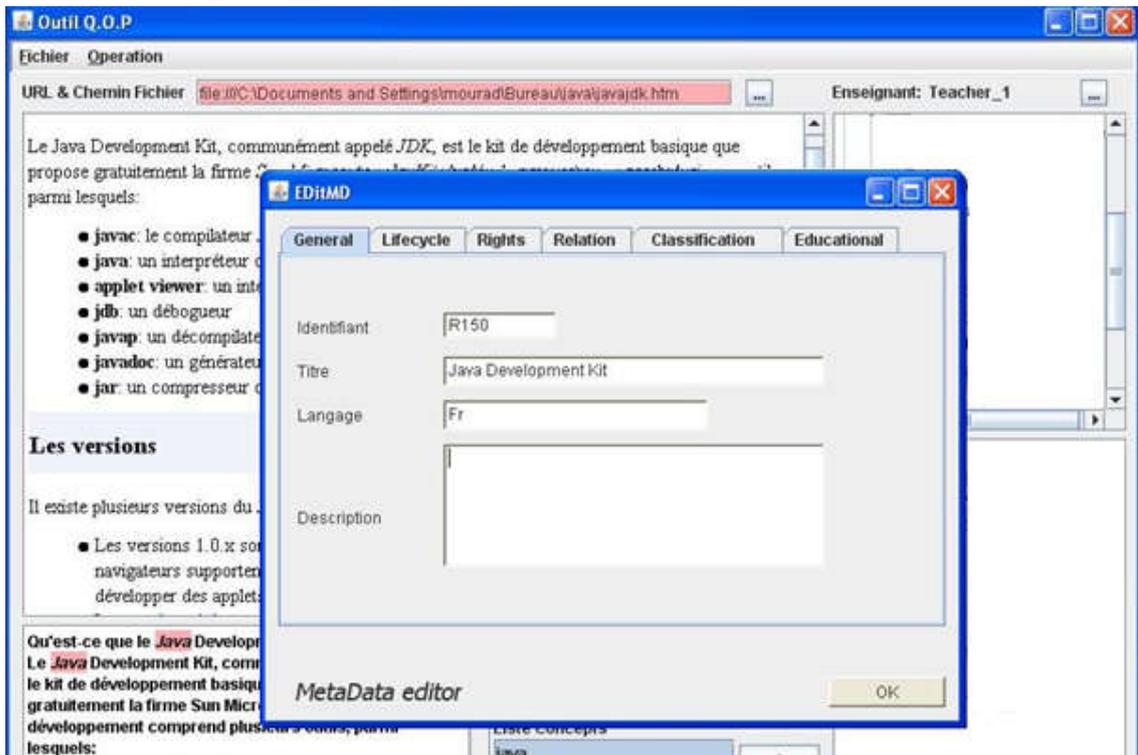


Figure 5.9. Interface pour la création des objets pédagogiques

2.5.1 Ontologie et règles sémantiques

L'ontologie a été développée avec l'outil protégé-OWL². Le choix de cet outil est fait pour sa légèreté d'exécution, son architecture ouverte (facilement extensible) et pour sa bonne ergonomie. Protégé représente une interface modulaire, permettant l'édition, la visualisation, le contrôle (vérification des contraintes) d'ontologies et l'écriture des règles sémantiques grâce au langage SWRL. Protégé-OWL est une extension de l'éditeur d'ontologie Protégé pour manipuler le format OWL. Elle permet de visualiser, d'éditer les classes et les propriétés OWL et de communiquer avec des raisonneurs logiques.

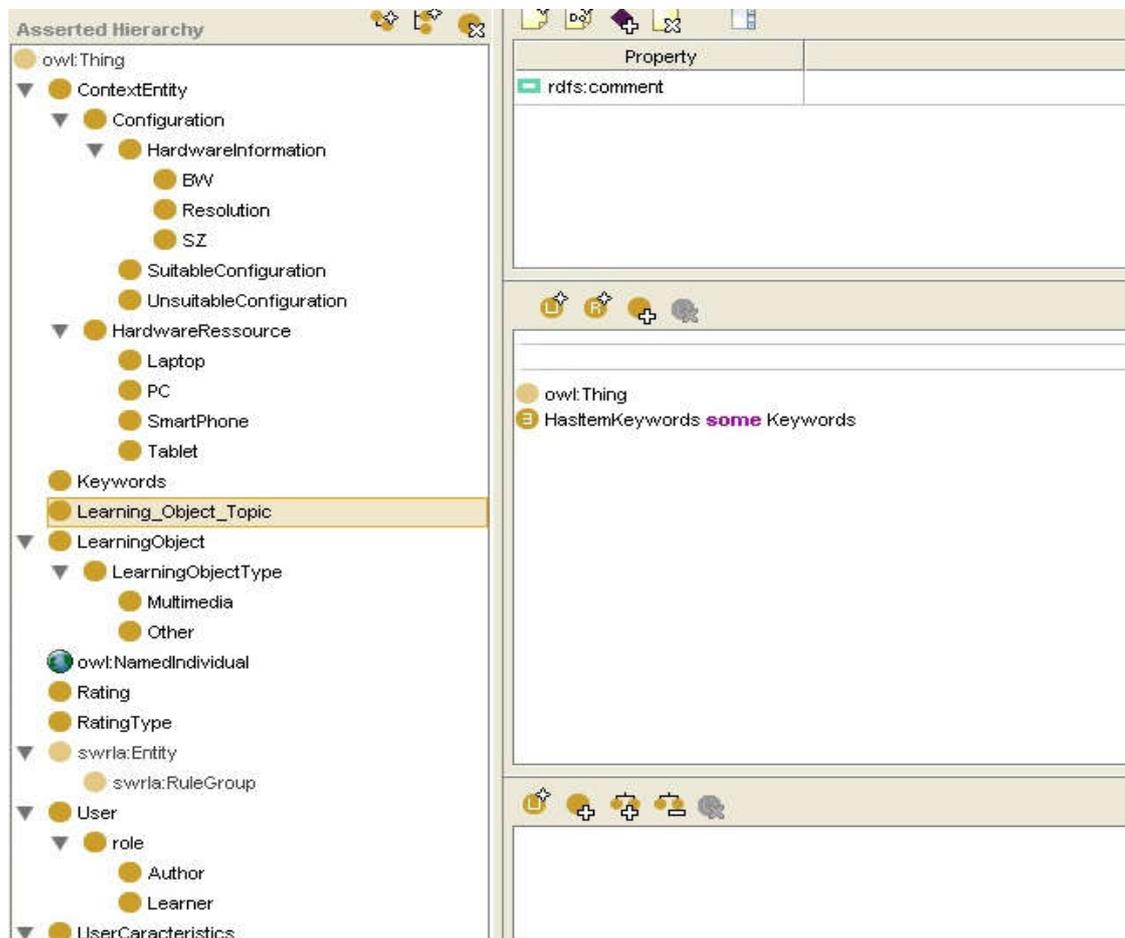


Figure 5.10. Implémentation de l'ontologie

² <https://protege.stanford.edu/>

MYOnto (http://www.semanticweb.org/mourad/ontologies/2021/5/MYOnto) : [C:\Users\Mourad\Desktop\ontology\MYOntology2.owl]

File Edit View Reasoner Tools Refactor Window Help

MYOnto (http://www.semanticweb.org/mourad/ontologies/2021/5/MYOnto)

Learning_Object_Topic

Active ontology * Entities * Classes * Individuals by class * DL Query * SWRLTab * OntoGraf * SPARQL Query *

Name	Rule
S1	HasUserCompetency(?u, ?c) ^ HasCompetencyTopic(?c, ?lot) ^ HasItemKeywords(?lot, ?k) ^ HasTopic(?it, ?lot1) ^ HasItemKeywords(?lot1, ?k1) ^ swrlb:listInterse
S2	HasAbilitytoRate(?U, ?IT) ^ Concern(?R, ?IT) ^ HasContext(?R, ?CE) ^ HasConfiguration(?CE, ?Lst) ^ Spend(?R, ?S) ^ Require(?IT, ?Rq) ^ swrlb:greaterThan(?s, ?f
S3	Interest(?U, ?IT) ^ Concern(?R, ?IT) ^ HasRatingtype(?R, ?RT) ^ HasRatingValue(?RT, "Like") ^ HasRatingKind(?RT, "Real") ^ HasContext(?R, ?CE) -> HasSuitable
S4	HasAbilitytoRate(?U, ?IT) ^ Concern(?R, ?IT) ^ Spend(?R, ?S) ^ Require(?IT, ?Rq) ^ swrlb:lessThan(?s, ?Rq) ^ HasRatingtype(?R, ?RT) ^ HasRatingValue(?RT, "Nc
S5	HasAbilitytoRate(?U, ?IT) ^ Concern(?R, ?IT) ^ Spend(?R, ?S) ^ Require(?IT, ?Rq) ^ swrlb:greaterThan(?s, ?Rq) ^ HasRatingtype(?R, ?RT) ^ HasRatingValue(?RT, "
S6	Create(?UI, ?IT) ^ HasTopic(?IT, ?Lot) ^ HasUserCompetency(?U, ?UC) -> HasCompetencyTopic(?UC, ?Lot)
S7	UserAct(?R, ?U) ^ Concern(?R, ?IT) ^ HasRatingtype(?R, ?RT) ^ HasRatingValue(?RT, "Like") ^ HasRatingKind(?RT, "Real") ^ HasTopic(?IT, ?Lot) ^ HasUserComp
S8	Interest(?u, ?IT) ^ Concern(?R, ?IT) ^ HasRatingtype(?R, ?RT) ^ HasRatingValue(?RT, "None") ^ HasRatingValue(?RT, "Like") ^ GlobalScore(?IT, "Like") ^ HasConte

Control Rules Asserted Axioms Inferred Axioms OWL 2 RL

OWL axioms successfully transferred to rule engine.
 Number of SWRL rules exported to rule engine: 8
 Number of OWL class declarations exported to rule engine: 33
 Number of OWL individual declarations exported to rule engine: 51
 Number of OWL object property declarations exported to rule engine: 20
 Number of OWL data property declarations exported to rule engine: 17
 Total number of OWL axioms exported to rule engine: 415
 The transfer took 6328 millisecond(s).

Figure 5.11. Implémentation des règles sémantique

3. Conclusion

Dans ce chapitre, une étude de cas dédiée pour le domaine de l'éducation a été présentée. Nous avons présenté aussi les démarches en détail pour la réalisation de notre système y compris l'ontologie et les règles sémantiques utilisées. L'extension proposée pour le schéma LOM a pour but de préserver l'interopérabilité entre d'autres systèmes éducatifs.

Pour ce cas d'étude nous avons choisi les facteurs contextuels relatifs à l'utilisation des objets multimédias tels que la taille d'écran, la résolution d'écran et la bande passante. Noter que le choix de d'autres facteurs contextuels est possible car l'approche repose sur l'analyse des comportements des utilisateurs, ce qui présente un avantage vital pour l'approche proposée.

L'évaluation de ce travail montre l'efficacité des approches sémantiques dans l'extraction des données contextuelles, d'autre part, elle reste peu suffisante à cause de nombre de participants réduit et ainsi l'absence d'une *dataset* convenable pour tester ce travail hors ligne.

Conclusion

Conclusion Générale

Nous nous sommes intéressés dans cette thèse aux systèmes de recommandation dans les environnements ubiquitaires, nous avons présenté dans un premier temps, leurs principes de fonctionnement, les critères permettant de leurs évaluation, et les principales techniques de recommandation: la recommandation collaborative, la recommandation basée sur le contenu et l'approche hybride. Nous avons présenté, dans un deuxième temps, les systèmes de recommandation sensibles au contexte ainsi que les différentes façons d'incorporer le contexte dans les systèmes de recommandations traditionnels. Plusieurs systèmes sensibles au contexte ont été développés dans différents domaines tels que le E-commerce [101], E-tourism [92], et le E-Learning [86], ...etc. La plupart de ces systèmes utilisent un des trois approches reconnus dans la littérature [44] qui sont : le pré-filtrage, le post filtrage et le filtrage par modélisation. Le bon choix des facteurs contextuels est une activité humaine, nécessite dans la plupart du temps un expert de domaine ou l'utilisateur elle-même. Cette tâche consiste à décider quels sont les paramètres ou les facteurs qui influencent le processus de recommandation, ce qui dérive une surcharge supplémentaire de l'utilisateur et qui peut interférer avec les activités de l'utilisateur.

Nous nous concentrons dans ce travail sur la détection automatique de conditions contextuelles appropriées et inappropriées pour la consommation adéquate des items, en utilisant un profil d'utilisateur et une base de règles sémantiques. Ces règles sont issues de l'analyse du comportement d'utilisateur au sein d'un système de recommandation. Les informations issues de l'analyse des usages sont constituées des évaluations émises par les utilisateurs sur les items qu'ils ont consultés. Les évaluations sont représentées sous forme de votes explicites ou implicites définis sur une échelle de valeurs. Pour cette raison, Une base de connaissance a été proposée pour encadrer les connaissances et identifier les causes et les circonstances permettant classifier les données issues de la matrice des votes, ce qui nous permet de formuler nos règles de base.

Le profil de l'utilisateur est généralement construit à partir de l'historique des activités de l'utilisateur, ce qui implique qu'il a besoin d'être constamment enrichi pour contenir le plus d'informations pertinentes concernant le contexte ainsi que les caractéristiques des items. Dans ce travail, le profil utilisateur est implicitement construit, il représente différentes classes ayant des relations entre eux. Il contient un modèle utilisateur relatif aux informations

démographiques sur les utilisateurs et leurs compétences ainsi que les données concernant les items évalués par l'utilisateur et les données relatives au contexte de l'utilisateur. Les compétences sont décrites sous forme de termes pondérés en utilisant la représentation attribut-valeur.

Le modèle des utilisateurs est utilisé pour calculer les similarités entre les différents couples utilisateurs afin de définir, pour chaque usager, la communauté avec laquelle il est similaire (similarité en termes de compétences). Cette communauté est en fait constituée de ses N plus proches voisins, N étant défini de façon empirique. Le contexte est défini comme étant une paire attribut-valeur et les items sont des entités ayant des caractéristiques relatives à son indexation.

Afin de valider notre approche proposée, nous avons réalisé une étude de cas dans le domaine de l'éducation et plus précisément la production d'objets pédagogiques. Cet outil est bénéfique car il décharge l'utilisateur de la mission de spécifier les données de contexte convenables, d'autre part, il représente un environnement commun entre les utilisateurs de la communauté de partager leurs opinions concernant les nouveaux objets pédagogiques créés. Le système que nous proposons est considéré comme une application de système de recommandation, il offre la possibilité d'évaluer les objets et de construire la matrice de votes en utilisant les scores fournis directement par les utilisateurs ou prédites par le système. Il s'agit d'exécuter le filtrage collaboratif sur les résultats issus d'un filtrage sur le contenu ou vice versa, soit pour résoudre le problème des données manquantes, soit pour résoudre le problème du démarrage à froid.

En outre, l'intégration du contexte dans le processus de recommandation est un aspect de notre proposition, où nous avons adopté le pré-filtrage contextuel en exploitant les données issues de règles d'inférence avec une méthode probabiliste pour décider l'adéquation du contexte du demandeur.

Comme perspectives, nous souhaitons accomplir notre travail par les points suivants :

- Choisir d'autres facteurs contextuels et par conséquent ajouter d'autres règles d'inférence.
- Enrichir le profil d'utilisateur par des données de contexte liées à son environnement tel que location, temps, et niveau de connaissance
- Entendre notre approche pour l'appliquer sur d'autres domaines tel que la santé qui semble promoteurs pour les approches sémantiques.

Bibliographie

-
- [1] Adomavicius, G., Mobasher, B., Ricci, F., Tuzhilin, A.: Context-aware recommender systems. *AI Magazine* 32(3), 67–80 (2011)
- [2] Fling, B.: *Mobile Design and Development: Practical Concepts and Techniques for Creating Mobile Sites and Web Apps*, 1st edn. O'Reilly Media, Inc. (2009)
- [3] G. Adomavicius, T. Alexander, Context-aware recommender systems, in: *Recommender Systems Handbook*, Springer, US, 2011, pp. 217–253.
- [4] C. Chen, L. Duh, and C. Liu, “A personalized courseware recommendation system based on fuzzy item response theory,” *IEEE Int. Conf. e-Technology, e-Commerce e-Service*, 2004. *EEE '04*. 2004, pp. 305–308, 2004.
- [5] M. Holenko Dlab and N. Hoic-Bozic, “Recommender system for Web 2.0 supported eLearning,” *IEEE Glob. Eng. Educ. Conf. EDUCON*, no. April, pp. 953–956, 2014.
- [6] Sergis, S., Zervas, P., & Sampson, D. G. (2014, July). Towards learning object recommendations based on teachers' ICT Competence Profiles. In *2014 IEEE 14th International Conference on Advanced Learning Technologies* (pp. 534-538). IEEE
- [7] Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to recommender systems handbook. In *Recommender systems handbook* (pp. 1-35). Springer, Boston, MA.
- [8] Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12), 61-70.
- [9] Resnick, P., & Varian, H. R. (1997). Recommender systems. *Communications of the ACM*, 40(3), 56-58.
- [10] Balabanović, M., & Shoham, Y. (1997). Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3), 66-72.
- [11] Terveen, L., Hill, W., Amento, B., McDonald, D., & Creter, J. (1997). PHOAKS: A system for sharing recommendations. *Communications of the ACM*, 40(3), 59-62.
- [12] Chumki Basu, Haym Hirsh, and William Cohen. Recommendation as classification: Using social and content-based information in recommendation. In *In Proceedings of the Fifteenth National Conference on Artificial Intelligence*, page 714–720. AAAI Press, 1998.
- [13] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, page 285–295, 2001.
- [14] Oard, D. W., & Kim, J. (1998, July). Implicit feedback for recommender systems. In *Proceedings of the AAAI workshop on recommender systems* (Vol. 83). WoUongong.
- [15] Esslimani, I., Brun, A., & Boyer, A. (2011). Densifying a behavioral recommender system by social networks link prediction methods. *Social Network Analysis and Mining*, 1(3), 159-172.
- [16] Gediminas Adomavicius and Alexander Tuzhilin, (2005). Toward the next generation of recommender systems : A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.*, 17(6), pp.734–749

- [17] Eke, C. I., Norman, A. A., Shuib, L., & Nweke, H. F. (2019). A survey of user profiling: State-of-the-art, challenges, and solutions. *IEEE Access*, 7, 144907-144924.
- [18] Salton, G. (1975). *A theory of indexing* (Vol. 18). SIAM.
- [19] Akermi, I. (2017). *A hybrid model for context-aware proactive recommendation* (Doctoral dissertation, Université de Toulouse, Université Toulouse III-Paul Sabatier).
- [20] Bernardo Magnini and Carlo Strapparava. Improving user modelling with content-based techniques. In *International Conference on User Modeling*, pages 74–83. Springer, 2001.
- [21] 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
- [22] de Gemmis, M., Lops, P., Semeraro, G., and Basile, P. (2008). Integrating tags in a semantic content-based recommender. In *Proceedings of the 2008 ACM conference on Recommender systems, RecSys '08*, pages 163–170, New York, NY, USA. ACM.
- [23] Cantador, I., Bellogín, A., and Vallet, D. (2010). Content-based recommendation in social tagging systems. In *Proceedings of the fourth ACM conference on Recommender systems, RecSys '10*, pages 237–240, New York, NY, USA. ACM.
- [24] Diederich, J. and Iofciu, T. (2006). Finding communities of practice from user profiles based on folksonomies. In Tomadaki, E., Scott, P. J., Tomadaki, E., and Scott, P. J., editors, *EC-TEL Workshops*, volume 213 of *CEUR Workshop Proceedings*. CEUR-WS.org.
- [25] Pazzani, M. and Billsus, D. (1997). Learning and Revising User Profiles: The Identification of Interesting Web Sites. *Machine Learning*, 27(3):313–331.
- [26] Magnini, B. and Strapparava, C. (2001). Improving user modelling with ContentBased techniques. In Bauer, M., Gmytrasiewicz, P. J., Vassileva, J., Bauer, M., Gmy trasiewicz, P. J., and Vassileva, J., editors, *User Modeling*, volume 2109 of *Lecture Notes in Computer Science*, pages 74–83. Springer.
- [27] Eirinaki, M., Vazirgiannis, M., and Varlamis, I. (2003). SEWeP: using site semantics and a taxonomy to enhance the web personalization process. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 99–108, New York, NY, USA. ACM Press
- [28] Bobadilla J, Ortega F, Hernando A, Gutierrez A. Recommender systems survey. *Knowl-Based Syst* 2013;46:109–32
- [29] S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithm for collaborative filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, page 43–52, 1998.
- [30] C. Ziegler, S. McNee, J. Konstan, et G. Lausen., 2005. Improving recommendation lists through topic diversification. Dans les actes de Fourteenth International World Wide Web Conference.
- [31] Resnick et Hal., 1997) P. Resnick et R. V. Hal., 1997. Recommender systems (introduction to special section.). *Communications of the ACM*. 21

- [32] F. Meyer., 2012. Recommender systems in industrial contexts. Thèse de Doctorat, University of Grenoble, France. 6, 21
- [33] B. Sarwar, G. Karypis, J. Konstan, et J. Reidl, 2001. Item-based collaborative filtering recommendation algorithms. Dans les actes de Proceedings of the 10th International Conference on World Wide Web. 21
- [34] Daniel, L. (2005). Discovering knowledge in data: An introduction to data mining. Larose, p. cm, ISBN 0-471-66657-2 (cloth), QA76, 9, D343L38.
- [35] Hofmann, 1999 Thomas Hofmann. Probabilistic latent semantic indexing. In Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '99, pages 50–57, New York, NY, USA, 1999. ACM
- [36] Hofmann, 2003 Thomas Hofmann. Collaborative filtering via gaussian probabilistic latent semantic analysis. In Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '03, pages 259–266, New York, NY, USA, 2003. ACM.
- [37] Hofmann, 2004 Thomas Hofmann, (January 2004). Latent semantic models for collaborative filtering. ACM Trans. Inf. Syst., 22(1), pp.89–115.
- [38] Burke, R. (2002), Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction, 12(4):331–370.
- [39] Buder J, Schwind C. Learning with personalized recommender systems: a psychological view. Comput Human Behav 2012;28(1): 207–16
- [40] Sylvain 2006 Sylvain Castagnos and Anne Boyer. A client/server user-based collaborative filtering algorithm : Model and implementation. In Proceedings of the 2006 conference on ECAI 2006 : 17th European Conference on Artificial Intelligence August 29 { September 1, 2006, Riva del Garda, Italy, pages 617{621, Amsterdam, The Netherlands, The Netherlands, 2006. IOS Press.
- [41] Herlocker et al., 2004 Herlocker, J. ; Konstan, J. ; Terveen, L. et Riedl, J. (2004). Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst., 22(1): 5–53.
- [42] Ekstrand et al., 2011 Michael D. Ekstrand, John T. Riedl, and Joseph A. Konstan, (February 2011). Collaborative filtering recommender systems. Found. Trends Hum.-Comput. Interact., 4(2), pp.81–173.
- [43] Shani, 2011 Shani, G., & Gunawardana, A. (2011). Evaluating recommendation systems. In Recommender systems handbook (pp. 257-297). Springer, Boston, MA.
- [44] G Adomavicius and A Tuzhilin. Context-Aware Recommender System. In Recommender Systems Handbook. 2015.
- [45] Y. Cui and S. Bull, “Context and Learner Modelling for the Mobile Foreign Language Learner,” System, vol. 33, no. 2, pp. 353-367, 2005.
- [46] R. Bader, E. Neufeld, W. Woerndl, and V. Prinz, “Context-Aware POI Recommendations in an Automotive Scenario Using MultiCriteria Decision Making

Methods,” Proc. Workshop ContextAwareness in Retrieval and Recommendation, pp. 23-30, 2011.

[47] Z. Yu, X. Zhou, D. Zhang, C.-Y. Chin, X. Wang, and J. Men, “Supporting Context-Aware Media Recommendations for Smart Phones,” *IEEE Pervasive Computing*, vol. 5, no. 3, pp. 68-75, JulySept. 2006

[48] M. Van Setten, S. Pokraev, and J. Koolwaaij, “Context-aware Recommendations in the Mobile Tourist Application COMPASS,” *Adaptive Hypermedia and Adaptive Web Based Systems*, pp. 235-244, Aug. 2004.

[49] 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 contextawareness. In *Handheld and ubiquitous computing*, pages 304–307. Springer.

[50] Zimmermann, A., Lorenz, A., & Oppermann, R. (2007, August). An operational definition of context. In *International and Interdisciplinary Conference on Modeling and Using Context* (pp. 558-571). Springer, Berlin, Heidelberg.

[51] Justin J Levandoski, Mohamed Sarwat, Ahmed Eldawy, and Mohamed F Mokbel. LARS: A Location-Aware Recommender System. In *Proc. IEEE ICDE*. IEEE, 2012

[52] Campos,2014 Pedro G Campos, Fernando D’íez, and Iv’an Cantador. Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. *UMUAI*, 24(1-2), 2014

[53] Bao, J., P. R. Smart, D. Mott and D. Braines (2010). "A formal context representation framework for network-enabled cognition."

[54] Thomas Strang and Claudia Linnhoff-Popien. A context modeling survey. In *In: Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004 – The Sixth International Conference on Ubiquitous Computing, Nottingham/England, 2004*

[55] Gerard Salton and Chu-Sing Yang. On the Specification of Term Values in Automatic Indexing. *Journal of Documentation*, 29:351–372, 1973

[56] Ahu Sieg, Bamshad Mobasher, Steve Lytinen, and Robin Burke. Using Concept Hierarchies to Enhance User Queries in Web-Base Information Retrieval. In *Artificial Intelligence and Applications, 2004*.

[57]Musumba, 2013 George Wamamu Musumba and Henry Nyongesa. Context awareness in mobile computing: A review. *International Journal of Machine Learning and Applications*, 2(1): 5–pages, 2013.

[58]Gauch, 2003 usan Gauch, Jason Chaffee, and Alaxander Pretschner. Ontology-based personalized search and browsing. *Web Intelli. and Agent Sys.*, 1(3-4):219–234, December 2003.

[59]Kim, 2003 Hyoung R. Kim and Philip K. Chan. Learning implicit user interest hierarchy for context in personalization. In *Proceedings of the 8th International Conference on Intelligent User Interfaces, IUI 03*, pages 101–108, New York, NY, USA, 2003. ACM

- [60] Serge Garlatti, thèse HDR « Les systèmes d'information sémantiques et adaptatifs fondés sur des documents virtuels et le web sémantique » L'Université de Bretagne Sud, 2006.
- [61] Brusilovsky P., Eklund J. et Schwarz E. (1998). « Web-based education for all : A tool for development adaptive courseware. » Dans Proceedings of Seventh International World Wide Web Conference, volume 30 (1-7), pages 291_300.
- [62] De Angeli A. et Convertino G. (1999). « A user-centered approach to user modeling » Dans Kay J., éditeur, UM99, Proceedings of the Seventh International Conference, pages 255_264. UM99 - User Modeling, Springer Wien NewYork
- [63] Berners-lee T., Hendler J. et Lassila O. "The Semantic Web", Scientific American , Mai 2001
- [64] Laublet, P., Reynaud, C., et Charlet, J. (2002) « Sur quelques aspects du web sémantique ». Actes des deuxièmes assises du GdR 13, pp 59-78, Cépaduès Editions, 2002.
- [65] Guarino, N., & Poli, R. (1993). Toward principles for the design of ontologies used for knowledge sharing. In In Formal Ontology in Conceptual Analysis and Knowledge Representation, Kluwer Academic Publishers, in press. Substantial revision of paper presented at the International Workshop on Formal Ontology.
- [66] Adida, B. et Birbeck, M. (). « RDFa Primer 1.0 – Embedding RDF in XHTML» . W3C Working Draft 12 March 2007
- [67] John F. Sowa, "Knowledge Representation", Brooks, 2000.
- [68] Nicola Guarino, "Understanding, Building, And Using Ontologies", LADSEBNR,Padova, Italie.
- [69] Le rôle de l'ingénierie ontologique dans le domaine des EIAH. Sticef.org (2004)
- [70] Biebow m. & szulman s., terminae : a method and a tool to build of a domain ontology, in Proceedings of the 11th European Knowledge Acquisition Workshop (EKAW'99), Springer, 1999
- [71] T. R. Gruber, (1993) Toward principles for the design of ontologies used for knowledge sharing. Originally in N. Guarino and R. Poli, (Eds.) International Workshop on Formal Ontology, Padova, Italy. Revised August 1993. Published in *International Journal of Human-Computer Studies*, Volume 43 , Issue 5-6 Nov./Dec. 1995, Pages: 907-928
- [72] Tazari, M. R. (2003, September). A Context-Oriented RDF Database. In SWDB (pp. 63-78).
- [73] Zhang, D., Gu, T., & Wang, X. (2005). Enabling context-aware smart home with semantic web technologies. *International Journal of Human-friendly Welfare Robotic Systems*, 6(4), 12-20.
- [74] Duval, E., Smith, N., & Van Coillie, M. (2006, July). Application profiles for learning. In Sixth IEEE International Conference on Advanced Learning Technologies (ICALT'06) (pp. 242-246). IEEE.

- [75] IEEE Learning Technology Standards Committee. (2002). Draft Standard for Learning Objects Metadata LOM. <http://ltsc.ieee.org/>
- [76] Chan, T., Sharples, M., Vavoula, G., & Lonsdale, P. (2004, March). Educational metadata for mobile learning. In *The 2nd IEEE International Workshop on Wireless and Mobile Technologies in Education, 2004. Proceedings.* (pp. 197-198). IEEE
- [77] Nilsson, M., Palmér, M., & Brase, J. (2003). The LOM RDF binding: principles and implementation. In *Third Annual ARIADNE conference, Leuven Belgium, 2003*
- [78]. Ekstrand, M. D., Riedl, J. T., & Konstan, J. A. (2011). Collaborative filtering recommender systems. *Foundations and Trends® in Human-Computer Interaction*, 4(2), 81-173
- [79]. Sergis, S., & Sampson, D. G. (2015). Learning object recommendations for teachers based on elicited ICT competence profiles. *IEEE Transactions on Learning Technologies*, 9(1), 67-80.
- [80]. Gauch, S., Speretta, M., Chandramouli, A., & Micarelli, A. (2007). User profiles for personalized information access. *The adaptive web*, 54-89.
- [81]. Moukas, A., & Maes, P. (1998). Amalthea: An evolving multi-agent information filtering and discovery system for the WWW. *Autonomous agents and multi-agent systems*, 1(1), 59-88.
- [82]. Lieberman, H. (1997, March). Autonomous interface agents. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems* (pp. 67-74).
- [83]. Widyantoro, D. H., Yin, J., El Nasr, M., Yang, L., Zacchi, A., & Yen, J. (1999, March). Alipes: A swift messenger in cyberspace. In *Proceedings of Spring Symposium Workshop on Intelligent Agents in Cyberspace* (pp. 62-67).
- [84]. Man, N., Xunxun, C., & Bo, W. (2016, June). Hierarchical user interest model based on large log data of mobile internet. In *2016 13th International Conference on Service Systems and Service Management (ICSSSM)* (pp. 1-5). IEEE.
- [85]. Begg, I. M., Gnocato, J., & Moore, W. E. (1993, February). A prototype intelligent user interface for real-time supervisory control systems. In *Proceedings of the 1st international conference on Intelligent user interfaces* (pp. 211-214).
- [86] Raza, S., Bashir, S. R., Hameed, M. T., & Zaheer, M. J. (2015). Design And Development Of Context-Aware Recommendation Strategy For E-Learning. *VFAST Transactions on Software Engineering*, 3(1), 32-42.
- [87]. Grüninger, M., & Fox, M. S. (1995). Methodology for the design and evaluation of ontologies.
- [88]. Fernández, M., Gómez-Pérez, A., Pazos, J., & Pazos, A. (1999). Building a chemical ontology using METHONTOLOGY and the ontology design environment. *IEEE intelligent Systems*, 14(1), 37-46.
- [89] Spyns, P., Tang, Y., & Meersman, R. (2008). An ontology engineering methodology for DOGMA. *Applied Ontology*, 3(1-2), 13-39.

- [90] Sure, Y. & Staab, S.S.,R. (2003) "On-To-Knowledge Methodology" in Handbook on Ontologies, pp. 117-132.
- [91] Tempich, C., Pinto, H.S. & Staab, S. (2006) "Ontology engineering revisited: An iterative case study", The Semantic Web Research and Applications, Springer, pp. 110-124.
- [92] Braunhofer, M., & Ricci, F. (2017). Selective contextual information acquisition in travel recommender systems. *Information Technology & Tourism*, 17(1), 5-29.
- [93] L. Liu, F. Lecue, N. Mehandjiev, and L. Xu, "Using Context Similarity for Service Recommendation," 2010 IEEE Fourth International Conference on Semantic Computing, pp. 277–284, Sept. 2010.
- [94] Gupta, R., Jain, A., Rana, S., & Singh, S. (2013, August). Contextual information based recommender system using singular value decomposition. In 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 2084-2089). IEEE.
- [95] Ko, E. J., Lee, H. J., & Lee, J. W. (2007). Ontology-based context modeling and reasoning for u-healthcare. *IEICE TRANSACTIONS on Information and Systems*, 90(8), 1262-1270.
- [96] Tsai, K. H., Chiu, T. K., Lee, M. C., & Wang, T. I. (2006, July). A learning objects recommendation model based on the preference and ontological approaches. In Sixth IEEE International Conference on Advanced Learning Technologies (ICALT'06) (pp. 36-40). IEEE.
- [97] Koren, Y., & Bell, R. (2011). *Recommender Systems Handbook*, chapter Advances in collaborative filtering Number 145-186.
- [98] Park, S. T., & Chu, W. (2009, October). Pairwise preference regression for cold-start recommendation. In Proceedings of the third ACM conference on Recommender systems (pp. 21-28).
- [98] Kim, J., Lee, D., & Chung, K. Y. (2014). Item recommendation based on context-aware model for personalized u-healthcare service. *Multimedia Tools and Applications*, 71(2), 855-872.
- [99] Margaret H. Dunham. (2003). *Data mining: introductory and advanced topics*. Prentice Hall.
- [100] Uschold, M., & Gruninger, M. (1996). *Ontologies: Principles, methods and applications*. TECHNICAL REPORT-UNIVERSITY OF EDINBURGH ARTIFICIAL INTELLIGENCE APPLICATIONS INSTITUTE AIAI TR.
- [101] Sharma, S., & Kaur, D. (2015, May). Location based context aware recommender system through user defined rules. In International Conference on Computing, Communication & Automation (pp. 257-261). IEEE.
- [102] Brik, M., & Touahria, M. (2020). Contextual Information Retrieval within Recommender System: Case Study" E-learning System". *TEM Journal*, 9(3), 1150.

[103] Rabhi, w. (2014). systeme auteur d'aide à la creation des objets pedagogiques textuels (doctoral dissertation, faculte des mathematiques et de l'informatique-universite de m'sila).

[104] Belgueliel, Y., Bourahla, M., & Brik, M. (2014). Towards an ontology for UML state machines. *Lecture Notes on Software Engineering*, 2(1), 116.

[105] Brik, M., & Touahria, M. (2012, March). EduBank: A bank of Educational Resources based on Ontologies. In *2012 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)* (pp. 92-96). IEEE.

[106] Mourad, B., & Mohamed, T. (2011). Semantic Metadata for Creating and Describing Educational Resources. *International Journal of Computer Applications*, 975, 8887.

Annexe

Règles Sémantiques

Nom	Antécédent	Conséquent
R1	<p>HasUserCompetency(?u, ?c) ^ HasCompetencyTopic(?c, ?lot) ^ HasItemKeywords(?lot, ?k) ^ HasTopic(?it, ?lot1) ^ HasItemKeywords(?lot1, ?k1) ^ swrlb:listIntersection(?k, ?k1)</p>	HasAbilitytoRate(?u, ?it)
R2	<p>HasAbilitytoRate(?U, ?IT) ^ Concern(?R, ?IT) ^ HasContext(?R, ?CE) ^ HasConfiguration(?CE, ?Lst) ^ Spend(?R, ?S) ^ Require(?IT, ?Rq) ^ swrlb:greaterThan(?s, ?Rq)</p>	Interest(?U, ?IT)
R3	<p>Interest(?U, ?IT) ^ Concern(?R, ?IT) ^ HasRatingtype(?R, ?RT) ^ HasRatingValue(?RT, "Like") ^ HasRatingKind(?RT, "Real") ^ HasContext(?R, ?CE)</p>	HasSuitableContext(?IT, ?CE)
R4	<p>HasAbilitytoRate(?U, ?IT) ^ Concern(?R, ?IT) ^ Spend(?R, ?S) ^ Require(?IT, ?Rq) ^ swrlb:lessThan(?s, ?Rq) ^ HasRatingtype(?R, ?RT) ^ HasRatingValue(?RT, "None")</p>	IgnoreWithoutInterest(?U, ?IT)
R5	<p>HasAbilitytoRate(?U, ?IT) ^ Concern(?R, ?IT) ^ Spend(?R, ?S) ^ Require(?IT, ?Rq) ^ swrlb:greaterThan(?s, ?Rq) ^ HasRatingtype(?R, ?RT) ^ HasRatingValue(?RT, "None")</p>	IgnoreWithInterest(?U, ?IT)
R6	<p>Create(?UI, ?IT) ^ HasTopic(?IT, ?Lot) ^ HasUserCompetency(?U, ?UC)</p>	HasCompetencyTopic(?UC, ?Lot)
R7	<p>UserAct(?R, ?U) ^ Concern(?R, ?IT) ^ HasRatingtype(?R, ?RT) ^ HasRatingValue(?RT, "Like") ^ HasRatingKind(?RT, "Real") ^ HasTopic(?IT, ?Lot) ^ HasUserCompetency(?U, ?UC)</p>	HasCompetencyTopic(?UC, ?Lot)
R8	<p>Interest(?u, ?IT) ^ Concern(?R, ?IT) ^ GlobalScore(?IT, "Like") ^ HasContext(?R, ?CE) ^ HasRatingtype(?R, ?RT) ^ HasRatingValue(?RT, "Like") ^ HasRatingValue(?RT, "None")</p>	HasUnsuitableContext(?IT, ?CE)

Partie de l'ontologie « Profil Utilisateur »

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
  <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
  <!ENTITY swrl "http://www.w3.org/2003/11/swrl#" >
  <!ENTITY swrlb "http://www.w3.org/2003/11/swrlb#" >
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
  <!ENTITY protege "http://protege.stanford.edu/plugins/owl/protege#" >
  <!ENTITY xsp "http://www.owl-ontologies.com/2005/08/07/xsp.owl#" >
  <!ENTITY swrla "http://swrl.stanford.edu/ontologies/3.3/swrla.owl#" >
  <!ENTITY sqwrl "http://sqwrl.stanford.edu/ontologies/built-ins/3.4/sqwrl.owl#" >
]>

<rdf:RDF xmlns="http://www.semanticweb.org/mourad/ontologies/2021/5/MYOnto#"
  xml:base="http://www.semanticweb.org/mourad/ontologies/2021/5/MYOnto"
  xmlns:swrla="http://swrl.stanford.edu/ontologies/3.3/swrla.owl#"
  xmlns:sqwrl="http://sqwrl.stanford.edu/ontologies/built-ins/3.4/sqwrl.owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
  xmlns:swrl="http://www.w3.org/2003/11/swrl#"
  xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
  xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#">
  <owl:Ontology rdf:about="">
    <owl:imports rdf:resource="http://swrl.stanford.edu/ontologies/3.3/swrla.owl"/>
    <owl:imports rdf:resource="http://sqwrl.stanford.edu/ontologies/built-ins/3.4/sqwrl.owl"/>
  </owl:Ontology>
  <owl:Class rdf:ID="Author">
    <rdfs:subClassOf rdf:resource="#role"/>
  </owl:Class>

  <owl:Class rdf:ID="Competency">
    <rdfs:subClassOf rdf:resource="#UserCharacteristics"/>
  </owl:Class>
  <owl:ObjectProperty rdf:ID="Concern">
    <rdfs:domain rdf:resource="#Rating"/>
    <rdfs:range rdf:resource="#LearningObject"/>
  </owl:ObjectProperty>
  <owl:Class rdf:ID="Configuration">
    <rdfs:subClassOf rdf:resource="#ContextEntity"/>
  </owl:Class>
  <owl:ObjectProperty rdf:ID="Create">
    <rdfs:domain rdf:resource="#User"/>
    <rdfs:range rdf:resource="#LearningObject"/>
  </owl:ObjectProperty>
  <owl:Class rdf:ID="Demographic">
    <rdfs:subClassOf rdf:resource="#UserCharacteristics"/>
  </owl:Class>
  <owl:DatatypeProperty rdf:ID="GlobalScore">
    <rdfs:domain rdf:resource="#LearningObject"/>
    <rdfs:range>
      <rdfs:Datatype/>
    </rdfs:range>
  </owl:DatatypeProperty>
  <owl:Class rdf:ID="HardwareInformation">
    <rdfs:subClassOf rdf:resource="#Configuration"/>
  </owl:Class>
  <HardwareInformation rdf:ID="HardwareInformation_2">
    <rdf:type rdf:resource="&owl;NamedIndividual"/>
  </HardwareInformation>
  <owl:Class rdf:ID="HardwareRessource">
    <rdfs:subClassOf rdf:resource="#ContextEntity"/>
  </owl:Class>
  <owl:ObjectProperty rdf:ID="HasAbilitytoRate">
```

```

    <rdfs:domain rdf:resource="#User"/>
    <rdfs:range rdf:resource="#LearningObject"/>
  </owl:ObjectProperty>
  <owl:DatatypeProperty rdf:ID="HasBw">
    <rdfs:domain rdf:resource="#BW"/>
  </owl:DatatypeProperty>
  <owl:ObjectProperty rdf:ID="HasCompetencyTopic">
    <rdfs:domain rdf:resource="#Competency"/>
    <rdfs:range rdf:resource="#Learning_Object_Topic"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="HasConfiguration">
    <rdfs:domain rdf:resource="#ContextEntity"/>
    <rdfs:range rdf:resource="#Configuration"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="HasContext">
    <rdfs:domain rdf:resource="#Rating"/>
    <rdfs:range rdf:resource="#ContextEntity"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="HasDevice">
    <rdfs:domain rdf:resource="#Rating"/>
    <rdfs:range rdf:resource="#HardwareRessource"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="HasItemKeywords">
    <rdfs:domain rdf:resource="#Learning_Object_Topic"/>
    <rdfs:range rdf:resource="#Keywords"/>
  </owl:ObjectProperty>
  <owl:DatatypeProperty rdf:ID="HasName">
    <rdfs:domain rdf:resource="#Keywords"/>
  </owl:DatatypeProperty>
  <owl:ObjectProperty rdf:ID="HasOrganisation">
    <rdfs:domain rdf:resource="#User"/>
    <rdfs:range rdf:resource="#Organisation"/>
  </owl:ObjectProperty>
  <owl:DatatypeProperty rdf:ID="HasRatingKind">
    <rdfs:domain rdf:resource="#RatingType"/>
    <rdfs:range>
      <owl:DataRange>
        <owl:oneOf>
          <rdf:List>
            <rdf:first rdf:datatype="&xsd:string">Real</rdf:first>
            <rdf:rest>
              <rdf:List>
                <rdf:first rdf:datatype="&xsd:string">None</rdf:first>
                <rdf:rest rdf:resource="&rdf:nil"/>
              </rdf:List>
            </rdf:rest>
          </rdf:List>
        </owl:oneOf>
      </owl:DataRange>
    </rdfs:range>
  </owl:DatatypeProperty>
  <owl:ObjectProperty rdf:ID="HasRatingtype">
    <rdfs:domain rdf:resource="#Rating"/>
    <rdfs:range rdf:resource="#RatingType"/>
  </owl:ObjectProperty>
  <owl:DatatypeProperty rdf:ID="HasRatingValue">
    <rdfs:domain rdf:resource="#RatingType"/>
    <rdfs:range>
      <owl:DataRange>
        <owl:oneOf>
          <rdf:List>
            <rdf:first rdf:datatype="&xsd:string">Like</rdf:first>
            <rdf:rest>
              <rdf:List>
                <rdf:first rdf:datatype="&xsd:string">Dislike</rdf:first>
                <rdf:rest rdf:resource="&rdf:nil"/>
              </rdf:List>
            </rdf:rest>
          </rdf:List>
        </owl:oneOf>
      </owl:DataRange>
    </rdfs:range>
  </owl:DatatypeProperty>

```