

Sommaire

Chapitre 1	Introduction	17
1.1	Contexte de la thèse	17
1.2	Problématique scientifique	18
1.3	Démarche méthodologique et organisation de la thèse	21
Chapitre 2	État de l'art	23
2.1	Systèmes de recommandation.....	24
2.1.1	Historique des systèmes de recommandation	25
2.1.2	Principe des systèmes de recommandation	26
2.1.3	Filtrage collaboratif.....	26
2.1.3.1	<i>Techniques de recommandation</i>	<i>27</i>
2.1.3.2	<i>Mesure de similarité</i>	<i>30</i>
2.1.3.3	<i>Recommandation multicritères</i>	<i>31</i>
2.1.4	Filtrage basé sur le contenu	32
2.1.5	Filtrage hybride.....	32
2.1.6	Problèmes et limites des systèmes de recommandation	33
2.1.6.1	<i>Démarrage à froid</i>	<i>33</i>
2.1.6.2	<i>Sparsity.....</i>	<i>33</i>
2.1.6.3	<i>Sérendipité</i>	<i>33</i>
2.1.6.4	<i>Problème du mouton gris.....</i>	<i>34</i>
2.1.6.5	<i>Montée en charge.....</i>	<i>34</i>
2.2	Systèmes de recommandation sociales.....	34
2.2.1	Réseaux sociaux en ligne.....	34
2.2.2	Liens sociaux et leurs poids.....	35
2.2.3	Types de systèmes de recommandation sociale	35
2.2.3.1	<i>Recommandeur basé sur les amis</i>	<i>36</i>
2.2.3.2	<i>Recommandeur basé sur la popularité sociale</i>	<i>36</i>
2.2.3.3	<i>Recommandeur basé sur les distances.....</i>	<i>36</i>
2.3	Systèmes de recommandation dans les EIAH	36

2.3.1	Contexte des EIAH	36
2.3.2	Objectifs de la recommandation dans les EIAH	37
2.3.3	Framework de classification des systèmes de recommandation pour les EIAH.....	38
2.3.3.1	<i>Tâches supportées</i>	38
2.3.3.2	<i>Modèle de l'utilisateur</i>	38
2.3.3.3	<i>Modèle du domaine</i>	39
2.3.3.4	<i>Personnalisation</i>	39
2.3.3.5	<i>Architecture</i>	40
2.3.3.6	<i>Emplacement</i>	40
2.3.3.7	<i>Mode de recommandation</i>	40
2.4	Travaux connexes	41
2.4.1	Système de recommandation ISIS	41
2.4.2	Système de recommandation 3A.....	42
2.4.3	Système de recommandation Ensemble.....	42
2.4.4	Système de recommandation Topolor.....	43
2.4.5	Système de recommandation basé sur le style d'enseignement.....	44
2.4.6	Système de recommandation social basé sur les sentiments de l'apprenant	44
2.4.7	Système de recommandation social basé sur la confiance	45
2.4.8	Système de recommandation social basé sur le parcours de graphes ...	46
2.4.9	Système de recommandation sémantique pédagogique	46
2.4.10	Synthèse	47
2.5	Évaluation des systèmes de recommandation	49
2.5.1	Évaluation <i>offline</i>	50
2.5.1.1	<i>Protocoles d'évaluation</i>	51
2.5.1.2	<i>Métriques d'évaluation</i>	52
2.5.2	Étude utilisateurs.....	52
2.5.3	Évaluation <i>online</i>	53
2.5.4	Questionnaires.....	53
2.5.4.1	<i>Questionnaire Content Quality Scale</i>	54
2.5.4.2	<i>Questionnaire ResQue</i>	54
2.6	Conclusion.....	55

Chapitre 3 Contribution : approche pour la recommandation de ressources pédagogiques basée sur les liens sociaux.....	56
3.1 Aperçu de l'approche.....	57
3.2 Base formelle de l'approche.....	60
3.3 Similarité sociale	61
3.4 Types de recommandations.....	64
3.4.1 Recommandation des ressources visitées récemment	64
3.4.2 Recommandation des ressources populaires	65
3.4.3 Recommandation des ressources utiles	65
3.5 Gestion de la sparsity.....	66
3.6 Mise à jour du profil	66
3.7. Validation avec des données simulées	67
3.7.1. Objectif.....	67
3.7.2. Données de la simulation.....	68
3.7.2.1. <i>Caractéristiques des apprenants</i>	68
3.7.2.2. <i>Relations sociales entre apprenants</i>	69
3.7.2.3. <i>Visites et évaluation des ressources</i>	71
3.7.3. Similarité sociale.....	71
3.7.4. Ressources recommandées.....	71
3.7.4.1. <i>Ressources visitées récemment</i>	71
3.7.4.2. <i>Ressources populaires</i>	72
3.7.4.3. <i>Ressources utiles</i>	72
3.8 Conclusion.....	73
Chapitre 4 Implantation : L'environnement Icraa.....	74
4.1 Architecture fonctionnelle	74
4.1.1 Télé-versement de ressources	75

4.1.2	Accès aux ressources	76
4.1.3	Évaluation de ressources	78
4.1.4	Recommandation de ressources	78
4.1.5	Fonctionnalités sociales	79
4.2	Architecture technique	80
4.3	Moteur de recommandation	81
4.3.1	Modèle de données	81
4.3.2	Algorithmes	82
4.4	Conclusion.....	91
 Chapitre 5 Expérimentations		92
5.1	Expérimentation 1 : Évaluation avec des utilisateurs réels	93
5.1.1	Expérimentation	93
5.1.2	Feedback des utilisateurs.....	95
5.1.3	Résultats du Feedback des utilisateurs.....	96
5.1.4	Questionnaire	97
5.1.5	Résultats du questionnaire	99
5.1.6	Discussion	101
5.2	Expérimentation 2 : Évaluation avec un Dataset (Analyse Offline)	102
5.2.1	Objectif et hypothèses	102
5.2.2	Expérimentation	103
5.2.3	Ensemble de données (<i>Dataset</i>).....	104
5.2.4	Résultats	104
5.3	Conclusion.....	105
 Chapitre 6 Conclusion et perspectives.....		107
6.1	Principales contributions	107
6.2	Limites et perspectives	108

Bibliographie	112
Annexes	125
Annexe I : Publications et communications.....	126
A. Publications dans des revues	126
B. Chapitres de livre	126
C. Conférences internationales avec comité de lecture et actes.....	126
D. Conférences nationales avec comité de lecture et actes	126
E. Ateliers et posters.....	127
Annexe II : Questionnaire RESQUE en anglais	128
Annexe III : Comparatif entre les jeux de données EIAH (Manouselis <i>et al.</i> 2012).....	131

Liste des figures

- Figure 2.1 **Graphe du réseau social**
- Figure 3.1 **Architecture globale de l'approche proposée**
- Figure 3.2 **Principe général de l'approche proposée**
- Figure 3.3 **Processus de validation de faisabilité de l'approche**
- Figure 3.4 **Connexions entre utilisateurs**
- Figure 4.1 **Architecture fonctionnelle**
- Figure 4.2 **Télé-versement et description d'une ressource**
- Figure 4.3 **Accès à une ressource (gauche), Widget de recommandation (droite)**
- Figure 4.4 **Evaluation de la qualité et de l'utilité d'une ressource**
- Figure 4.5 **Fil d'actualité d'un groupe d'une promotion de 45 membres**
- Figure 4.6 **Architecture technique**
- Figure 4.7 **Modèle de données**
- Figure 5.1 **Processus d'évaluation avec des utilisateurs réels**
- Figure 5.2 **Taux de satisfaction pour les trois types de recommandation**
- Figure 5.3 **Moyennes des résultats du questionnaire**
- Figure 5.4 **Écart-type des résultats du questionnaire**
- Figure 5.5 **Moyennes des résultats du questionnaire groupés par hypothèses**

Figure 5.6 **Processus d'évaluation avec un dataset**

Rapport-Gratuit.com

Liste des équations

- Équation 2.1 **Filtrage collaboratif basé sur les utilisateurs**
- Équation 2.2 **Facteur de normalisation du filtrage collaboratif basé sur les utilisateurs**
- Équation 2.3 **Filtrage collaboratif basé sur les items**
- Équation 2.4 **Facteur de normalisation du filtrage collaboratif basé sur les items**
- Équation 2.5 **Similarité entre deux utilisateurs basée sur le coefficient de corrélation de Pearson**
- Équation 2.6 **Similarité entre deux items basée sur le coefficient de corrélation de Pearson**
- Équation 2.7 **Similarité entre deux utilisateurs basée sur le Cosinus**
- Équation 2.8 **Similarité entre deux items basée sur le Cosinus**
- Équation 3.1 **Similarité sociale**
- Équation 3.2 **Similarité en termes d'évaluation de qualité**
- Équation 3.3 **Similarité en termes d'utilité**
- Équation 3.4 **Similarité en termes de visites**
- Équation 3.5 **Force de lien**
- Équation 3.6 **Similarité de profil**
- Équation 3.7 **Score de recommandation des ressources visitées récemment**
- Équation 3.8 **Score de recommandation des ressources populaires**

Équation 3.9 **Facteur de normalisation de recommandation des ressources populaires**

Équation 3.10 **Score de recommandation des ressources utiles**

Équation 5.1 **Alpha de Cronbach**

Liste des tableaux

- Tableau 2.1 **exemple d'une matrice d'évaluation (échelle de 1 à 5 étoiles)**
- Tableau 2.2 **Synthèse des systèmes de recommandation pédagogique sociale**
- Tableau 3.1 **Base formelle de l'approche**
- Tableau 3.2 **Caractéristiques des utilisateurs**
- Tableau 3.3 **Visites et évaluations des ressources pédagogiques**
- Tableau 5.1 **Caractéristiques des utilisateurs**
- Tableau 5.2 **Résultats de l'analyse offline**

1 Introduction

1.1	Contexte de la thèse	17
1.2	Problématique scientifique	18
1.3	Démarche méthodologique et organisation de la thèse	21

Ce premier chapitre introduit le contexte et les objectifs de notre travail de thèse. Nous présentons tout d'abord le domaine des systèmes de recommandation en général, puis la recommandation dans le cadre des environnements informatiques pour l'apprentissage humain. Nous présentons ensuite l'objectif et les problématiques de notre recherche. Nous finirons par introduire la démarche adoptée pour réaliser nos objectifs ainsi que l'organisation de ce mémoire.

1.1 Contexte de la thèse

Les fonctionnalités d'interaction et d'entraide qu'offrent les réseaux sociaux en ligne ont permis à ces derniers d'être de plus en plus présents dans les dispositifs de formation. Que ce soit comme étant un composant intégré à un LMS (*Learning Management System*) ou en mode autonome (Popescu, 2014).

Comme cela est expliqué dans (Fazeli *et al.*, 2014), les plateformes d'apprentissage social combinent les LMS traditionnels et les réseaux sociaux pour faciliter la création de contenu, l'accès aux ressources pédagogiques, le partage de celles-ci entre les utilisateurs, etc. En plus des fonctionnalités de Forum et de Chat présentes dans les LMS, les plateformes d'apprentissage social permettent aux utilisateurs de créer plus de connexions et de développer leurs réseaux d'amis.

Les dernières années ont vu l'émergence du domaine de recherche des systèmes de recommandation pour les Environnements Informatiques pour l'Apprentissage Humaine (EIAH). Dans ce cadre, Drachsler (2012) explique que ce type de système utilise les expériences d'une communauté d'apprenants pour aider les apprenants de cette même communauté à identifier plus efficacement un contenu d'apprentissage ou des apprenants pairs parmi un ensemble potentiellement très grand de choix.

Selon Erdt *et al.* (2015), l'objectif principal des systèmes de recommandation dans les EIAH est de soutenir les apprenants pendant leur processus d'apprentissage afin d'atteindre leurs objectifs pé-

dagogiques. Les mêmes chercheurs expliquent que les systèmes de recommandation peuvent avoir des effets positifs sur l'apprentissage, tels que la performance d'apprentissage et la motivation. Manouselis *et al.* (2012) expliquent que les systèmes de recommandation dans les EIAH peuvent prendre en charge les tâches suivantes : trouver de bonnes ressources, trouver des apprenants pairs, recommander des séquences de ressources, prédire les performances d'apprentissage et suggérer des activités d'apprentissage.

Selon Ma *et al.* (2011), les techniques de recommandation ont été largement étudiées dans les domaines de la recherche d'information, de l'apprentissage automatique (*Machine Learning*) et de la fouille de données (*Data Mining*). Les deux méthodes les plus utilisées dans les systèmes de recommandation sont le filtrage collaboratif (Goldberg *et al.*, 1992) et le filtrage basé sur le contenu (Pazzani and Billsus, 2007). La première méthode recommande des ressources à partir de la similarité entre les préférences des utilisateurs. Cette méthode est utilisée par Amazon (Linden *et al.*, 2003) et LinkedIn (Wu *et al.*, 2014). Elle est également utilisée dans les EIAH par Ghauth et Abdullah (2011) et Shi *et al.* (2013). La deuxième méthode est basée sur la recommandation de ressources qui sont similaires à celles pour lesquelles l'utilisateur a montré un intérêt dans le passé. Cette méthode est utilisée par Pandora Radio (Lops *et al.*, 2011), et dans un cadre d'apprentissage humain par Karampiperis *et al.*, (2014) et Salehi (2013).

1.2 Problématique scientifique

Selon Guy et Carmel (2011), la multitude de ressources, de relations et d'interactions présentes dans les médias sociaux peut conduire les utilisateurs à subir une surcharge informationnelle qui les rend incapables d'assimiler les informations qui y sont présentées. Afin de réduire cette surcharge, il serait utile de proposer aux utilisateurs uniquement les ressources susceptibles de les intéresser. Pour ce faire, nous proposons de ne leurs recommander que des ressources pertinentes en se basant sur les **liens sociaux** entre les utilisateurs. Ainsi, l'objectif de notre travail est de proposer une approche permettant de personnaliser les ressources pédagogiques en s'appuyant sur les connexions entre individus dans les réseaux sociaux. Nous expliquerons plus bas dans cette section nos contributions qui répondent à certains verrous.

Nous nous appuyons, dans notre travail, sur deux principes des sciences sociales à savoir la régularité de la co-citation (Bhagat *et al.*, 2011) (*Co-citation regularity*) et l'influence sociale (Sun et Tang, 2011). Le premier principe stipule que les individus similaires ont tendance à se référer ou à se connecter aux mêmes ressources. Ce principe est utilisé dans les systèmes classiques de recommandation. Ces derniers se basent principalement sur les évaluations des utilisateurs similaires à un utilisateur donné pour chercher à prédire ses préférences.

Le deuxième principe indique que les personnes qui sont socialement connectées sont susceptibles de partager les mêmes intérêts ou des intérêts similaires. Ainsi, les utilisateurs d'un système peuvent être facilement influencés par leurs amis et ainsi être intéressés par leurs activités. Ce principe est utilisé par les systèmes de recommandation sociale.

Dans notre approche, nous utilisons la méthode de filtrage collaboratif car nous nous concentrons principalement sur les relations sociales entre les utilisateurs. Les systèmes de recommandation peuvent être basés sur des relations de confiance entre les utilisateurs, en partant de l'hypothèse qu'ils ont des goûts similaires avec d'autres utilisateurs auxquels ils font confiance. Les systèmes de recommandation basés sur la confiance, comme ceux dans (Bedi *et al.*, 2007) et (Ma *et al.*, 2009), utilisent les informations de confiance inférées (implicites) ou observées (explicites).

Les connexions sociales entre les utilisateurs sont de plus en plus utilisées pour améliorer la qualité et la pertinence des recommandations. L'utilisation des informations des réseaux sociaux dans les systèmes de recommandation a été étudiée dans de nombreux travaux de recherche. Vassileva (2008) présente les systèmes de recommandation sociale comme une solution qui peut aider les apprenants à trouver les ressources appropriées pendant leurs processus d'apprentissage.

L'idée de la « recommandation sociale » améliore les systèmes de recommandation en intégrant un réseau social d'amis. Selon Ma *et al.* (2011), la relation d'amitié, dans un réseau social, est liée à la coopération et aux liens mutuels qui entourent l'utilisateur tels que les camarades de classe, collègues, parents, etc.

Les utilisateurs sont intéressés d'obtenir des recommandations appropriées pour leurs besoins. Ils peuvent être intéressés par les ressources les plus populaires chez les utilisateurs avec qui ils partagent des similarités. Comme ils peuvent être intéressés par les ressources

qui ont été récemment visités par leur réseau d'amis pour collaborer avec eux sur ces ressources. Les utilisateurs peuvent également être intéressés par des ressources qui sont utiles dans leurs domaines d'intérêt. Contrairement aux systèmes de recommandation sociale existants pour les EIAH, qui ne traitent que le premier type de recommandation, l'approche présentée dans ce document considère ces trois types, à savoir la recommandation des 1) ressources populaires, 2) ressources utiles et 3) ressources récemment consultées.

Nous posons l'hypothèse que, outre les évaluations des ressources pédagogiques données par les utilisateurs, l'utilisation des informations présentes dans le réseau social, dans les profils des utilisateurs, dans les résultats des résolutions d'exercices et dans les traces laissées lors de l'utilisation du système peut contribuer à rendre les recommandations de ressources plus riches et plus adaptées aux besoins de l'apprenant. Les systèmes de recommandation sociale existants dans les EIAH utilisent principalement des évaluations des utilisateurs pour recommander des ressources appropriées aux besoins de ces utilisateurs et ne bénéficient pas de ces informations.

Tout comme les systèmes précédemment cités, notre approche est utilisée pour faire de la recommandation de ressources pédagogiques. Aucun de ces systèmes ne traite l'utilité des ressources par rapport aux domaines d'apprentissage des apprenants. Aucun d'entre eux ne traite les connaissances acquises par les apprenants pour mieux personnaliser les recommandations.

Le but de notre travail est donc de proposer une approche intégrée qui utilise toutes les informations citées ci-dessous afin d'améliorer la pertinence des recommandations proposées à l'apprenant :

- les évaluations des ressources pédagogiques données par les utilisateurs ;
- les connexions sociales entre les utilisateurs ;
- les profils des utilisateurs ;
- les résultats des résolutions d'exercices ;
- les traces laissées lors de l'utilisation du système.

En plus du problème de la précision des systèmes de recommandation pour les plateformes d'apprentissage social, nous essayons de trouver des solutions au problème de l'éparpillement (*sparsity* en anglais, terme le plus courant et que nous utiliserons par la suite).

Les préférences de l'utilisateur sont représentées dans une matrice utilisateur/ressource. Cette matrice d'évaluations est *sparse* lorsque le nombre de ressources évaluées est très faible par rapport au nombre total des ressources du système. Les méthodes de filtrage col-

laboratif utilisent la similarité entre les utilisateurs. Le problème de la *sparsity* limite le calcul de la similarité, réduisant ainsi la précision de la recommandation (Schafer *et al.*, 2007). Comme expliqué dans (Verbert *et al.*, 2011), les ensembles de données (*Dataset*) présents dans les plateformes éducatives souffrent de ce problème.

Les questions de recherche que nous proposons de traiter s'énoncent comme suit :

- QR1 : Comment générer des recommandations plus pertinentes pour les apprenants sur les plateformes d'apprentissage social, en se basant sur les relations sociales et les évaluations des utilisateurs ?
- QR2 : Comment améliorer les recommandations pour les apprenants sur les plateformes d'apprentissage social qui contiennent des évaluations éparpillées (*sparse*) ?

1.3 Démarche méthodologique et organisation de la thèse

Nous avons commencé notre thèse par une étude de l'état de l'art des systèmes de recommandation dans les réseaux sociaux et des systèmes de recommandation dans les EIAH. Nous avons, aussi, proposé un modèle formel de recommandation de ressources pédagogiques et nous avons testé sa faisabilité avec un prototype développé en Java sur un ensemble de données simulées.

Ensuite nous avons développé une plateforme, nommée *Icraa*, qui met en œuvre notre approche. Nous l'avons alors expérimentée auprès d'utilisateurs en situation réelle d'apprentissage. L'évaluation de notre approche a été réalisée en trois phases 1) utilisation d'*Icraa* et recueil des évaluations des utilisateurs sur les ressources recommandées, 2) mise en place d'un questionnaire destiné aux utilisateurs pour recueillir leurs avis sur la pertinence des ressources pédagogiques recommandées, et 3) analyse des résultats des deux phases précédentes.

Enfin, nous avons mené une analyse *offline* pour comparer la précision de notre approche à celle de quatre algorithmes de référence dans le domaine.

Ce document est organisé comme suit. Dans le chapitre 2, nous présentons un état de l'art sur la recommandation dans un contexte social et dans un contexte EIAH. Ensuite nous présentons une étude de plusieurs systèmes de recommandation dans le contexte des plateformes d'apprentissage social. Cet état de l'art se termine par une

synthèse des différentes méthodes utilisées pour évaluer les systèmes de recommandation.

Dans le chapitre 3 nous proposons une approche qui répond à la problématique de la surcharge informationnelle en utilisant les relations sociales dans le processus de recommandation. Ce chapitre décrit l'architecture générale de l'approche ainsi qu'un modèle formel qui permet de calculer la similarité entre des utilisateurs d'un EIAH afin de générer trois types de recommandation. Nous présentons également dans ce chapitre une validation de l'approche avec un ensemble de données synthétique.

Le chapitre 4 présente l'implantation des modèles formels de notre approche dans une plateforme d'apprentissage appelée Icraa. Dans ce chapitre, nous décrivons les architectures fonctionnelle et technique de cette plateforme. Nous détaillons ensuite le moteur de recommandation ainsi que les algorithmes qu'il intègre.

L'évaluation de nos propositions est présentée dans le chapitre 5. Deux expérimentations y sont décrites. La première évalue le système avec des utilisateurs réels à l'aide de leurs *feedbacks* et de leurs réponses à un questionnaire. La deuxième évalue le système en mode « hors ligne » à l'aide d'un Dataset.

Le dernier chapitre comporte une conclusion suivie des limites ainsi que des perspectives de notre travail.



2 État de l'art

2.1	Systèmes de recommandation	24
2.1.1	Historique des systèmes de recommandation.....	25
2.1.2	Principe des systèmes de recommandation	26
2.1.3	Filtrage collaboratif.....	26
2.1.3.1	<i>Techniques de recommandation</i>	27
2.1.3.2	<i>Mesure de similarité</i>	30
2.1.3.3	<i>Recommandation multicritères</i>	31
2.1.4	Filtrage basé sur le contenu	32
2.1.5	Filtrage hybride.....	32
2.1.6	Problèmes et limites des systèmes de recommandation	33
2.1.6.1	<i>Démarrage à froid</i>	33
2.1.6.2	<i>Sparsity</i>	33
2.1.6.3	<i>Sérendipité</i>	33
2.1.6.4	<i>Problème du mouton gris</i>	34
2.1.6.5	<i>Montée en charge</i>	34
2.2	Systèmes de recommandation sociales	34
2.2.1	Réseaux sociaux en ligne.....	34
2.2.2	Liens sociaux et leurs poids.....	35
2.2.3	Types de systèmes de recommandation sociale	35
2.2.3.1	<i>Recommandeur basé sur les amis</i>	36
2.2.3.2	<i>Recommandeur basé sur la popularité sociale</i>	36
2.2.3.3	<i>Recommandeur basé sur les distances</i>	36
2.3	Systèmes de recommandation dans les EIAH	36
2.3.1	Contexte des EIAH	36
2.3.2	Objectifs de la recommandation dans les EIAH	37
2.3.3	Framework de classification des systèmes de recommandation pour les EIAH.....	38
2.3.3.1	<i>Tâches supportées</i>	38
2.3.3.2	<i>Modèle de l'utilisateur</i>	38
2.3.3.3	<i>Modèle du domaine</i>	39
2.3.3.4	<i>Personnalisation</i>	39
2.3.3.5	<i>Architecture</i>	40
2.3.3.6	<i>Emplacement</i>	40
2.3.3.7	<i>Mode de recommandation</i>	40
2.4	Travaux connexes	41
2.4.1	Système de recommandation ISIS	41
2.4.2	Système de recommandation 3A.....	42
2.4.3	Système de recommandation Ensemble.....	42
2.4.4	Système de recommandation Topolor.....	43

2.4.5	Système de recommandation basé sur le style d'enseignement.....	44
2.4.6	Système de recommandation social basé sur les sentiments de l'apprenant	44
2.4.7	Système de recommandation social basé sur la confiance	45
2.4.8	Système de recommandation social basé sur le parcours de graphes ...	46
2.4.9	Système de recommandation sémantique pédagogique	46
2.4.10	Synthèse	47
2.5	Évaluation des systèmes de recommandation	49
2.5.1	Évaluation <i>offline</i>	50
2.5.1.1	Protocoles d'évaluation	51
2.5.1.2	Métriques d'évaluation.....	52
2.5.2	Étude utilisateurs.....	52
2.5.3	Évaluation <i>online</i>	53
2.5.4	Questionnaires.....	53
2.5.4.1	Questionnaire Content Quality Scale.....	54
2.5.4.2	Questionnaire ResQue	54
2.6	Conclusion.....	55

Ce deuxième chapitre présente le contexte général de notre travail. Il est scindé en 6 sections. La première présente les concepts de base des systèmes de recommandation et les techniques qu'ils utilisent ainsi que leurs limites. Dans la deuxième section, nous étudions les réseaux sociaux en ligne et les différents types de systèmes de recommandation qui peuvent s'appliquer à ces réseaux. La section qui suit présente les spécificités des EIAH dans les systèmes de recommandation. La section 4 passe en revue les différents systèmes de recommandation sociale existant dans les EIAH et se termine par une synthèse. Les trois méthodes utilisées pour évaluer les systèmes de recommandation sont détaillées dans la cinquième section.

2.1 Systèmes de recommandation

Chaque jour, nous sommes submergés par des options et des choix. Quel *news* ou article lire ? Quel livre lire ? Quelle musique écouter ou vidéo visionner ? Quel produit acheter ? Les tailles de ces do-

maines de décision sont souvent massives : Deezer¹ offre plus de 53 millions de titres de musique dans sa sélection (Deezer, 2018), et Amazon² a plus de 562 millions de produits dans son magasin (ScrapeHero, 2018).

Aider les utilisateurs à découvrir et à choisir des ressources dans un espace d'informations de cette ampleur est un défi important qui reste toujours d'actualité.

Les systèmes de recommandation constituent une solution à ce problème de surcharge d'informations. Le but principal de ces systèmes est de fournir à l'utilisateur des recommandations qui reflètent ses préférences personnelles. Ces systèmes ont été utilisés avec succès par des sites e-commerce comme Amazon, de streaming audio ou vidéo comme Netflix³ ou par des réseaux sociaux comme Facebook⁴. Selon Dalia (2014), les deux tiers des films loués sur NetFlix ont été choisis sur la base de recommandations, 38% des news Google sont générées sur la base de recommandations et 35% des achats sur Amazon.com dépendent aussi des recommandations.

2.1.1 Historique des systèmes de recommandation

Cela fait presque 25 ans que la recherche s'intéresse à la façon de recommander automatiquement des items aux utilisateurs. Une grande variété de méthodes, d'algorithmes et d'outils ont été proposés (Adomavicius et Tuzhilin, 2005 ; Su et Khoshgoftaar, 2009). Le domaine de recherche sur les systèmes de recommandation est apparu au début des années 90 et depuis, notamment avec l'intégration des réseaux sociaux, de l'apprentissage artificiel et du *big data*, ce domaine est en constante évolution.

Bridge *et al.* (2006) et Yang *et al.* (2014) expliquent que les systèmes de recommandation se basent sur plusieurs domaines de recherche tels que la recherche d'information, la modélisation de l'utilisateur, l'apprentissage artificiel, les sciences cognitives et les interactions humain-machine.

¹ <http://www.deezer.com/fr/>

² <https://www.amazon.com/>

³ <https://www.netflix.com>

⁴ <https://www.facebook.com>

Parmi les premiers systèmes de recommandation qui sont apparus dans les années 90, nous retrouvons Tapestry (Goldberg *et al.*, 1992) pour la recommandation des messages des newsgroups, GroupLens (Resnick *et al.*, 1994) pour la recommandation des articles Usenet et Ringo (Shardanand et Maes, 1995) pour la recommandation de la musique.

2.1.2 Principe des systèmes de recommandation

Les systèmes de recommandation utilisent principalement deux méthodes de filtrage pour proposer des recommandations personnalisées aux utilisateurs, à savoir le filtrage collaboratif et le filtrage basé sur le contenu. Le premier utilise les opinions des utilisateurs similaires à l'utilisateur actif. Le deuxième utilise uniquement les préférences de l'utilisateur actif. Le filtrage collaboratif est considéré comme la méthode la plus populaire et la plus répandue dans les systèmes de recommandation (Ricci *et al.*, 2011).

2.1.3 Filtrage collaboratif

Les systèmes de recommandations basés sur le filtrage collaboratif recommandent à l'utilisateur actif les items que d'autres utilisateurs ayant des préférences similaires ont aimés par le passé. Cette méthode de recommandation repose sur l'idée que, si les utilisateurs sont d'accord sur la qualité de certains items, ils seront probablement d'accord sur d'autres items.

Le filtrage collaboratif dépend des préférences exprimées par des utilisateurs sur des items. Ces préférences sont appelées évaluations et peuvent être représentées avec des échelles d'évaluation réelles ou entières, comme par exemple une évaluation de 1 à 5 étoiles. L'ensemble des préférences forme une matrice d'évaluations (de deux dimensions : Utilisateurs et Items) où chaque ligne représente un utilisateur et chaque colonne représente un item. La similarité entre deux utilisateurs dépend de la similarité de l'historique des évaluations de ces deux utilisateurs.

Le tableau 2.1 montre un exemple d'une matrice d'évaluations pour trois utilisateurs et quatre films dans un système de recommandation de film. Les « ? » indiquent que l'utilisateur n'a pas évalué ce film.

Tableau 2.1 exemple d'une matrice d'évaluation (échelle de 1 à 5 étoiles)

	Le Seigneur des anneaux	Les Avengers	La reine des neiges	Le parrain
Utilisateur 1	?	2	3	5
Utilisateur 2	4	3	?	?
Utilisateur 3	5	1	?	4

Ekstrand *et al.* (2011) explique que les systèmes de recommandation, y compris les systèmes basés sur le filtrage collaboratif, effectuent deux tâches : la prédiction et la recommandation. La première essaye de prédire les valeurs manquantes dans la matrice d'évaluations. La deuxième génère une liste d'items les mieux classées en fonctions des préférences de l'utilisateur.

2.1.3.1 Techniques de recommandation

2.1.3.1.1. Filtrage collaboratif basé sur les utilisateurs

Ekstrand *et al.* (2011) expliquent que cette technique de recommandation se base sur le principe de trouver des utilisateurs similaires à l'utilisateur courant puis d'utiliser leurs évaluations pour prédire ce que l'utilisateur courant peut aimer. Les utilisateurs similaires à l'utilisateur courant, appelés voisins de cet utilisateur, sont ceux qui ont un comportement d'évaluation similaire à celui de l'utilisateur courant.

Herlocker *et al.* (1999) présente les 3 étapes de cette technique de recommandation :

1. Calculer la similarité entre l'utilisateur courant et tous les utilisateurs du système ;
2. Sélectionner un sous ensemble d'utilisateurs à utiliser comme recommandeurs. Il s'agit des utilisateurs voisins les plus proches ;
3. Calculer les prédictions en utilisant une combinaison pondérée des évaluations appartenant aux voisins sélectionnés.

L'évaluation prédite $r(u, i)$ de l'item i par l'utilisateur u dépend de :

- la similarité entre cet utilisateur et ses voisins les plus proches notée par $\text{sim}(u, v)$,
- l'évaluation de l'utilisateur v sur l'item i notée par $r(v, i)$,
- et d'un facteur de normalisation k donné par l'équation Eq(2.2).

v appartient à l'ensemble N qui représente les voisins les plus proches de l'utilisateur u . $r(u, i)$ est décrit par la formule suivante :

$$r(u, i) = k \sum_{v \in N} r(v, i) \cdot sim(u, v) \quad \text{Eq(2.1).}$$

$$k = 1 / \sum_{v \in N} |sim(u, v)| \quad \text{Eq(2.2).}$$

2.1.3.1.2. Filtrage collaboratif basé sur les items

Le filtrage collaboratif basé sur les utilisateurs souffre de problèmes de montée en charge si la base d'utilisateurs est importante. La technique du filtrage collaboratif basé sur les items (Sarwar *et al.* 2001) a été développée pour répondre à cette problématique. Cette technique est utilisée lorsqu'il s'agit de trouver des items similaires à l'item courant.

Cette technique utilise les similarités entre les *patterns* des évaluations des items. Si deux items ont tendance à avoir les mêmes utilisateurs qui les aiment et les mêmes utilisateurs qui ne les aiment pas, alors ces items sont similaires. Les utilisateurs ont des préférences similaires pour les items similaires.

Comme défini par Gabriellsson et Gabriellsson (2006), cette technique se compose de 3 étapes :

1. Calculer la similarité entre l'item courant et tous les items du système ;
2. Sélectionner les voisins les plus proches de l'item courant. Il s'agit des items les plus proches ;
3. Calculer les prédictions en utilisant un algorithme basé sur l'évaluation par l'utilisateur courant des items appartenant au voisinage de l'item courant.

L'évaluation prédite $r(u, i)$ de l'item i par l'utilisateur u dépend de :

- la similarité entre cet item et les items évalués par l'utilisateur u notée par $sim(i, j)$,
- l'évaluation de l'utilisateur u sur l'item j notée par $r(u, j)$,
- et d'un facteur de normalisation k donné par l'équation Eq(2.4).

j appartient à l'ensemble I qui contient les items, voisins les plus proches de l'item i , évalués par l'utilisateur u . $r(u, i)$ est décrit par la formule suivante :

$$r(u, i) = k \sum_{j \in I} r(u, j) \cdot \text{sim}(i, j) \quad \text{Eq(2.3).}$$

$$k = 1 / \sum_{j \in I} | \text{sim}(i, j) | \quad \text{Eq(2.4).}$$

2.1.3.1.3. Réduction de dimension

Comme expliqué précédemment, les évaluations des utilisateurs peuvent être considérées comme des vecteurs. Ces derniers peuvent avoir une grande dimension. Un item est représenté par un vecteur à U dimensions (U étant le nombre d'utilisateurs du système) et un utilisateur est représenté par un vecteur à R dimensions (R étant le nombre des items du système). Vu que ces dimensions contiennent des redondances, cette technique de recommandation propose de réduire ces dimensions. *Singular Value Decomposition* (SVD) (Billsus et Pazzani, 1998) est l'une des techniques de réduction de dimensions.

2.1.5.1.4. Méthodes probabilistes

Il existe plusieurs méthodes de recommandation probabiliste. Ces méthodes construisent des modèles probabilistes du comportement des utilisateurs et utilise ces modèles pour prédire les futurs comportements.

Cross-sell (Kitts *et al.* 2000) est l'un des systèmes qui utilise les méthodes probabilistes, basées sur la classification bayésienne naïve.

2.1.3.1.5 Méthodes basées sur les graphes

Dalia (2014) explique que les méthodes de la théorie des graphes sont souvent utilisées pour calculer les voisins les plus proches d'un utilisateur donné dans un système de filtrage collaboratif.

Le graphe utilisateur/item est un graphe biparti dans lequel il existe deux types de sommets : les items et les utilisateurs. Si un utilisateur évalue, visite ou achète un item alors une arête est créée dans le graphe entre cet utilisateur et cet item. Dans ce graphe, une séquence du chemin utilisateur/item peut être par exemple : $U1 \rightarrow I7 \rightarrow U5 \rightarrow I4 \rightarrow U2$. Dans cet exemple les utilisateurs $U1$ et $U5$ ont visité l'item $I7$ et $U5$ et $U2$ ont visité l'item $I4$. La représentation en graphe

biparti permet d'exploiter les relations transitives entre les utilisateurs pour effectuer la recommandation.

Les systèmes de recommandation basés sur des graphes utilisent plusieurs méthodes comme le plus court chemin (Huang *et al.* 2004), la marche aléatoire (Jamali et Ester, 2009) et le PageRank (Sangkeun, 2012).

2.1.3.2 Mesure de similarité

Plusieurs mesures de similarité entre utilisateurs et entre items ont été proposées dans la littérature. Selon Beliakov *et al.*, (2011) les deux types de mesures de similarité les plus populaires sont le coefficient de corrélation de Pearson et la similarité basée sur le cosinus.

Il existe d'autres mesures de similarité telles que la différence moyenne quadratique (Shardanand et Maes, 1995) ou le coefficient de corrélation de Spearman (Herlocker *et al.* 2002) mais ils n'ont pas connu d'adoption significative par rapport aux deux mesures précédemment citées.

2.1.3.2.1. Coefficient de corrélation de Pearson

Ce coefficient calcule la corrélation statistique de Pearson entre deux vecteurs d'évaluation pour déterminer la similarité. S'il s'agit de calculer la similarité entre deux utilisateurs, la corrélation entre eux est mesurée à l'aide des deux lignes, appartenant aux deux utilisateurs, de la matrice d'évaluations. Les colonnes des items non évaluées par les deux utilisateurs sont ignorées. Seuls les items co-évalués sont utilisés dans ce calcul.

Ce coefficient se situe entre -1 et 1. Une similarité proche de -1 signifie une corrélation négative et inversement, une similarité proche de +1 signifie une corrélation positive. Il n'existe pas de corrélation entre les deux utilisateurs si la similarité est autour de 0.

La similarité $sim(u, v)$ entre les utilisateurs u et v est donnée par l'équation Eq(2.5). $\bar{r}(u, .)$ est la moyenne des évaluations de l'utilisateur u . I est l'ensemble des item co-évalués par u et v .

$$sim(u, v) = \frac{\sum_{i \in I} (r(u, i) - \bar{r}(u, .)) \cdot (r(v, i) - \bar{r}(v, .))}{\sqrt{\sum_{i \in I} (r(u, i) - \bar{r}(u, .))^2} \cdot \sqrt{\sum_{i \in I} (r(v, i) - \bar{r}(v, .))^2}} \quad Eq(2.5).$$

La similarité $sim(i, j)$ entre les items i et j est donnée par l'équation Eq(2.6). $\bar{r}(., i)$ est la moyenne des évaluations de l'item i . U est l'ensemble des utilisateurs qui ont co-évalué les items i et j .

$$\begin{aligned} & sim(i, j) \\ &= \frac{\sum_{u \in U} (r(u, i) - \bar{r}(\cdot, i)) \cdot (r(u, j) - \bar{r}(\cdot, j))}{\sqrt{\sum_{u \in U} (r(u, i) - \bar{r}(\cdot, i))^2} \cdot \sqrt{\sum_{u \in U} (r(u, j) - \bar{r}(\cdot, j))^2}} \end{aligned} \quad Eq(2.6).$$

2.1.3.2.2. Similarité basée sur le cosinus

Dans la matrice d'évaluation, les lignes associées aux utilisateurs sont considérées comme des vecteurs d'évaluation. Ce type de mesure de similarité est calculé en utilisant l'angle cosinus entre deux vecteurs d'évaluation. Cet angle est mesuré dans un espace à N dimensions où N est le nombre d'items co-évalués entre les deux utilisateurs. Cette similarité se situe entre 0 et 1 où le 0 signifie aucune similarité et 1 une forte similarité. Cette similarité entre les utilisateurs est décrite par la formule Eq(2.7) et entre les items par la formule Eq(2.8) :

$$sim(u, v) = \frac{\sum_{i \in I} (r(u, i)) \cdot (r(v, i))}{\sqrt{\sum_{i \in I} (r(u, i))^2} \cdot \sqrt{\sum_{i \in I} (r(v, i))^2}} \quad Eq(2.7).$$

$$sim(i, j) = \frac{\sum_{u \in U} (r(u, i)) \cdot (r(u, j))}{\sqrt{\sum_{u \in U} (r(u, i))^2} \cdot \sqrt{\sum_{u \in U} (r(u, j))^2}} \quad Eq(2.8).$$

2.1.3.3 Recommandation multicritères

La majorité des systèmes de recommandation basés sur le filtrage collaboratif prennent en compte les évaluations des items par des utilisateurs suivant un seul critère pour recommander une liste d'items à l'utilisateur. Cependant, d'autres systèmes intègrent plusieurs critères qui peuvent augmenter la pertinence des recommandations. Par exemple, dans un système de recommandation de films prenant en compte qu'un seul critère d'évaluation, un utilisateur peut fournir une seule évaluation pour un film donné. Cependant, dans un système de recommandation de film prenant en compte deux critères d'évaluation, les utilisateurs peuvent spécifier leurs préférences suivant deux attributs (par exemple, l'histoire et les effets spéciaux). Adomavicius *et al.* (2011) classent les techniques de recommandation utilisées dans les systèmes de recommandation multicritères en deux catégories : les techniques basées sur les heuristiques et celles basées

sur les modèles. Dans la première technique, les similitudes entre les utilisateurs sont calculées en agrégeant les similarités de chaque critère ou en utilisant des métriques de distance multidimensionnelles. La seconde technique est basée sur un modèle prédictif, utilisant des méthodes statistiques ou d'apprentissage automatique, pour prédire l'évaluation d'un item par un utilisateur. Une partie du calcul de la similarité utilisée dans notre approche est inspirée de la technique des heuristiques.

2.1.4 Filtrage basé sur le contenu

Les systèmes utilisant le filtrage basé sur le contenu recommandent des items similaires à ceux que l'utilisateur a déjà aimés. Cette méthode de recommandation analyse un ensemble de descriptions d'items préalablement évalués par un utilisateur, et construit un profil des préférences de cet utilisateur basé sur ces descriptions. Ensuite elle fait correspondre les attributs du profil utilisateur avec les attributs d'un item pour effectuer la recommandation.

À partir de cette définition, deux techniques sont nécessaires à mettre en place pour ce type de filtrage. L'une pour représenter les items et une autre pour créer le profil utilisateur. Ces techniques sont implémentées de différentes manières dans les systèmes de recommandation basés sur le contenu. Certains systèmes utilisent la classification bayésienne naïve tel que LIBRA (Mooney et Roy, 2000) et ITR (Degemmis *et al.* 2007). D'autres systèmes utilisent d'autres méthodes d'apprentissage artificiel comme le retour de pertinence tel que YourNews (Ahn *et al.* 2007) et Fab (Balabanovic et Shoham, 1997). La majorité des systèmes de recommandation basés sur le contenu utilise le *matching* de mots clés ou le modèle vectoriel (VSM : Vector Space Model) (Salton et McGill, 1983).

2.1.5 Filtrage hybride

Le filtrage hybride a pour but de combiner les points forts des techniques de recommandation précédemment expliquées afin de bénéficier de leurs avantages complémentaires. Plusieurs techniques ont été proposées (Zhao *et al.* 2015) pour combiner les techniques de base et créer un nouveau système hybride.

Par exemple, un système de recommandation utilisant la technique du filtrage collaboratif peut être complété par la technique du filtrage basé sur le contenu, afin de pallier aux problèmes de recommandation de nouveaux items. La première technique ne peut pas re-

commander des items qui n'ont pas encore eu des évaluations, alors que la deuxième peut les recommander vu que la recommandation est basée sur la description des items. Çano et Morisio (2017) fait partie des nombreux systèmes de recommandation basés sur des filtres hybrides.

2.1.6 Problèmes et limites des systèmes de recommandation

Les systèmes de recommandation se basant sur les techniques précédemment expliquées ont certaines limites (Adomavicius et Tuzhilin, 2005). Plusieurs approches ont été proposées dans la littérature pour pallier ces limites. Ci-dessous les problèmes les plus importants sont décrits.

2.1.6.1 Démarrage à froid

Les systèmes de filtrage collaboratif dépendent des évaluations des items par les utilisateurs. Ainsi, un nouvel item ne peut pas être recommandé tant qu'aucun utilisateur ne l'a évalué.

Dans les systèmes de recommandation basés sur le filtrage collaboratif et les systèmes basés sur le contenu, il est impossible de prédire les préférences des utilisateurs sans connaître leurs historiques d'évaluations d'items. Ainsi, les nouveaux utilisateurs ne recevront pas de recommandations précises avant d'avoir évalué un certain nombre d'items.

2.1.6.2 Sparsity

Un système de recommandation souffre de la *sparsity* quand le nombre d'items évalués par les utilisateurs est très faible par rapport au nombre d'items total présent dans le système. Ce fait conduit à avoir une très faible densité dans la matrice d'évaluation utilisateurs/items. Cela a des conséquences sur la capacité du système de recommandation à recommander toutes les items disponibles et sur l'exactitude des recommandations générées.

2.1.6.3 Sérendipité

Vu que les systèmes de recommandation basés sur le contenu ne recommandent que les items correspondants au profil de l'utilisateur, ce dernier ne recevra que des recommandations similaires à celles qu'il a déjà rencontrées. Il n'aura aucune chance de recevoir des recommandations inattendues. Cela peut amener l'utilisateur à se lasser des recommandations.

2.1.6.4 Problème du mouton gris

Les utilisateurs d'un système de recommandation peuvent avoir des goûts particuliers et des préférences très inhabituelles par rapport aux autres. Ces utilisateurs sont à la frontière entre deux ou plusieurs clusters d'utilisateurs. Il leur est donc difficile de trouver des utilisateurs similaires et des recommandations pertinents.

2.1.6.5 Montée en charge

Plus le nombre d'utilisateurs et d'items augmente dans le système de recommandation, plus les calculs nécessaires à la recommandation deviennent très coûteux. Souvent des algorithmes de recommandation qui ont une énorme base d'utilisateurs et d'items préfèrent avoir des recommandations moins précises avec un temps de calcul rapide.

2.2 Systèmes de recommandation sociales

Dans la vie réelle, les gens demandent souvent des conseils à leurs réseaux d'amis avant d'acheter un produit ou de consommer un service. Selon le principe d'homophilie (McPherson *et al.*, 2001), les humains ont tendance à se lier avec d'autres qui sont semblables à eux. Ces liens sociaux poussent les gens à partager leurs opinions personnelles avec leurs amis, et les aident à avoir davantage de confiance aux recommandations de leurs amis qu'à celles d'autres personnes.

Plusieurs systèmes de recommandation sociale, exploitant les liens qui existent entre les utilisateurs, ont vu le jour cette dernière décennie.

2.2.1 Réseaux sociaux en ligne

Les réseaux sociaux en ligne offrent de nouvelles opportunités pour améliorer l'exactitude des systèmes de recommandation. Par exemple, Facebook et LinkedIn offrent de nouvelles façons de communiquer et de construire des communautés virtuelles. Les réseaux sociaux en ligne permettent non seulement aux utilisateurs de partager leurs opinions, mais servent aussi de sources d'informations qui peuvent être utilisées pour améliorer la qualité des recommandations.

Ces sources d'informations peuvent être de différentes natures telles que les opinions des utilisateurs ou les évaluations explicites sur des items de ce réseau. L'importance des systèmes de recommandation dans les réseaux sociaux a augmenté dans la dernière décennie.

2.2.2 Liens sociaux et leurs poids

Selon Yang *et al.* (2014), les réseaux sociaux peuvent être généraux, tel Facebook ou spécifiques à un domaine, tel Netflix. Chaque utilisateur de ces réseaux a un ensemble de voisins directs qu'il suit ou en qui il a confiance. La relation sociale entre deux utilisateurs est une relation dirigée et pondérée (représentée par une valeur). Le poids social peut être interprété comme combien un utilisateur connaît ou fait confiance à un autre utilisateur dans un réseau social. Ce poids peut être basé sur un *feedback* explicite d'un utilisateur concernant un autre utilisateur (par exemple, l'évaluation d'un utilisateur par un autre), ou déduit d'un *feedback* implicite (par exemple, le degré d'interaction et de communication entre les deux utilisateurs). La figure 2.1 illustre un exemple de réseau social entre cinq utilisateurs où chacun d'entre eux a un groupe d'amis. Chaque lien d'amitié dirigé est pondéré par une valeur de confiance appartenant à l'intervalle $[0, 1]$.

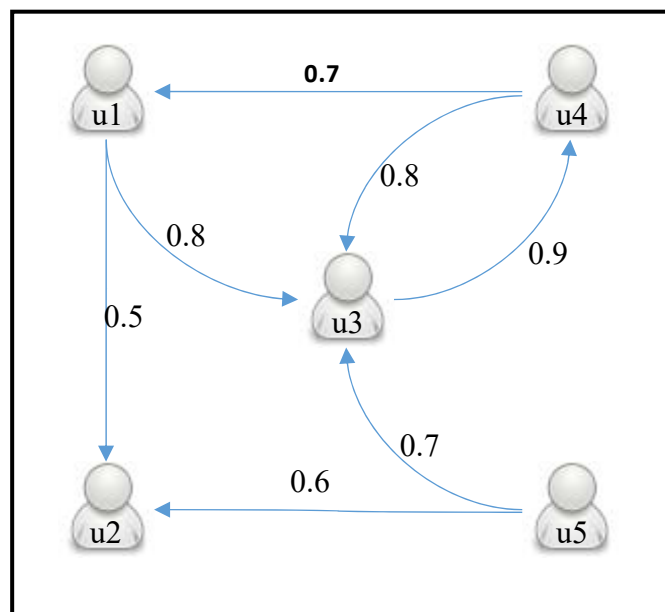


Figure 2.1 Exemple de graphe de réseau social

2.2.3 Types de systèmes de recommandation sociale

Bellogína *et al.* (2013) scindent les systèmes de recommandations sociales en quatre types que nous décrivons ci-dessous.

2.2.3.1 *Recommandeur basé sur les amis*

Cette approche a été utilisée dans (Liu et Lee, 2010). Elle incorpore l'information sociale dans le filtrage collaboratif basé sur les utilisateurs en utilisant la même formule de recommandation mais remplace l'ensemble des plus proches voisins de l'utilisateur actif par celui de ses amis.

2.2.3.2 *Recommandeur basé sur la popularité sociale*

Cette approche, très simple, est basée sur le système de recommandation proposé par Barman et Dabeer (2010). Les systèmes de ce type recommandent les items les plus populaires chez les amis de l'utilisateur courant.

2.2.3.3 *Recommandeur basé sur les distances*

L'approche de Ben Shimon *et al.* (2007) introduit explicitement les distances, entre les utilisateurs, dans le graphe social dans la formule du calcul de la recommandation. Cette approche utilise un algorithme qui calcule la distance entre deux nœuds dans un graphe, tel que l'algorithme de Dijkstra (1959).

2.3 **Systèmes de recommandation dans les EIAH**

Les Environnements Informatiques pour l'Apprentissage Humain (EIAH) sont devenus très populaires au cours des dernières décennies. Ces environnements visent à concevoir, développer et évaluer des outils informatiques pour différents types d'apprentissage (Drachler *et al.* 2015).

Certains de ces EIAH offrent un environnement social bénéfique pour les apprenants afin d'améliorer leurs apprentissages (Fazeli *et al.* 2016). Dans un EIAH, les ressources (items) pédagogiques sont régulièrement produites, organisées et publiées (Manouselis *et al.* 2012). L'une des principales préoccupations des apprenants sur ce type de plateforme est de décider quel cours suivre ou quelle ressource visualiser. Ces plateformes peuvent aider ces apprenants en leur recommandant un contenu qui peut les intéresser et qu'ils n'ont jamais visualisé.

2.3.1 Contexte des EIAH

Une bonne partie de la littérature sur les systèmes de recommandation est focalisée sur le contexte de la vente des produits en ligne, que

ce soit sur des sites de e-commerce ou sur des sites de *streaming* vidéo ou audio.

La recommandation dans le contexte des EIAH a de nombreuses particularités qui sont basées sur la richesse des théories et des modèles pédagogiques existants (Manouselis *et al.* 2012). L'apprentissage sur des plateformes EIAH est différent de l'achat des produits en ligne. L'apprentissage est en effet un effort qui prend souvent plus de temps et d'interactions par rapport à une simple transaction commerciale.

Selon Romero et Ventura (2007), le domaine des EIAH diffère du domaine du commerce électronique sur de nombreux points. Dans le commerce électronique, les données utilisées pour la recommandation sont souvent de simples journaux d'accès au serveur Web ou des évaluations d'utilisateurs sur des produits. Les systèmes de recommandation dans les EIAH utilisent plus d'informations sur les interactions des apprenants. Comme le souligne Drachsler *et al.* (2009), le modèle de l'utilisateur et les objectifs des systèmes sont également différents dans les deux domaines d'application.

Selon Drachsler *et al.* (2007), les systèmes de recommandation dans les EIAH doivent prendre en compte les caractéristiques qui sont spécifiques au contexte d'apprentissage. Ces caractéristiques sont les suivantes : but d'apprentissage, connaissances pré-requises, caractéristiques et préférences de l'apprenant, regroupement d'apprenants, ressources pédagogiques, parcours d'apprentissage et stratégies d'apprentissage.

2.3.2 Objectifs de la recommandation dans les EIAH

Les systèmes de recommandation dans les EIAH ont pour objectif de développer une stratégie de recommandation basée sur les caractéristiques propres au contexte d'apprentissage. L'objectif est de pouvoir soutenir les apprenants au cours de leur processus d'apprentissage afin d'atteindre leurs objectifs d'apprentissage (Drachsler *et al.* 2015 ; McCalla 2004).

Selon Erdt *et al.* (2015) et Buder et Schwind (2012), les systèmes de recommandation dans les EIAH recommandent une grande variété d'items tels que les ressources d'apprentissage, les logiciels, les cours, les conseils, les apprenants pairs et les séquences et activités d'apprentissage.

Plusieurs travaux (Drachsler *et al.* 2009 ; Santos *et al.* 2014 ; Erdt *et al.*, 2015) ont souligné l'existence de plusieurs effets positifs que peuvent avoir les systèmes de recommandation sur l'apprentis-

sage tels que la performance d'apprentissage, le rendement de l'apprenant et la motivation à apprendre. Plusieurs résultats d'expérimentations confirment ces effets positifs. La plupart des apprenants qui ont testé le système TAK (Chen *et al.* 2014), ont convenus qu'il peut les aider à améliorer leur efficacité. Plus de 80% des participants à une évaluation du système de recommandation proposé dans (Hsu *et al.* 2009), sont d'accord sur le fait que ce système peut améliorer leur motivation.

2.3.3 Framework de classification des systèmes de recommandation pour les EIAH

De leur côté, Manouselis *et al.* (2012) classent les systèmes de recommandation selon 7 catégories de dimensions : tâches supportées, modèle de l'utilisateur, modèle du domaine, personnalisation, architecture, emplacement et mode de recommandation.

2.3.3.1 Tâches supportées

Dans le travail cité précédemment, Manouselis *et al.* expliquent que les tâches principales supportées dans les EIAH sont :

- Trouver de nouvelles ressources : recommandation des nouvelles ressources ajoutées au système et plus particulièrement les ressources récentes ;
- Trouver des utilisateurs pairs : recommandation d'autres utilisateurs pour lesquels un utilisateur particulier peut être intéressé. Par exemple, proposer un apprenant dans la même classe ou proposer un utilisateur ayant des intérêts similaires ;
- Trouver des scénarios : recommandation d'un parcours d'apprentissage. Par exemple, proposer une liste de parcours possibles pour les mêmes ressources pour atteindre un objectif d'apprentissage.

2.3.3.2 Modèle de l'utilisateur

Le modèle de l'utilisateur, appelé aussi profil d'utilisateur, est lié à la représentation, le stockage et la mise à jour des caractéristiques de l'utilisateur dans un système de recommandation. Cette catégorie identifie les dimensions suivantes :

- **Représentation** : Le modèle de l'utilisateur peut être réalisé à l'aide de plusieurs méthodes : modèles basés sur l'historique utilisateur, modèles d'espace vectoriel, réseaux sémantiques, réseaux associatifs, modèles basés sur des classificateurs, matrices d'évaluations utilisateurs/items, caractéristiques démo-

graphiques, ainsi que les ontologies (Schafer *et al.*, 2001 ; Montaner *et al.*, 2003 ; Wei *et al.*, 2002).

- **Génération** : Il existe plusieurs méthodes de création du modèle initial de l'utilisateur et de mise à jour de ce modèle à partir de certaines données recueillies par le système (Montaner *et al.*, 2003). Le modèle initial de l'utilisateur dans un système de recommandation 1) peut être vide et se remplir progressivement lorsque l'utilisateur commence à utiliser le système, 2) peut être fourni manuellement par l'utilisateur, 3) peut être complété selon un stéréotype auquel l'utilisateur appartient, ou 4) peut être généré selon un ensemble d'exemples d'apprentissage que l'utilisateur est invité à fournir.

2.3.3.3 Modèle du domaine

Un modèle de domaine représente les propriétés des items que l'EIAH, doté d'un système de recommandation, recommande. Cette catégorie identifie les dimensions suivantes :

- **Représentation** : Les items du domaine peuvent être représentés en utilisant 1) un index simple ou un catalogue d'items présents au même niveau hiérarchique, 2) une taxonomie d'items composée d'une hiérarchie de classes d'items similaires, ou 3) une ontologie où des relations plus complexes sont définies entre des items ou des classes d'items.
- **Génération** : La génération du modèle de domaine peut utiliser plusieurs techniques telles que le *clustering*, la classification et la réduction de dimension.

2.3.3.4 Personnalisation

La personnalisation (Schafer *et al.*, 2001) se réfère à des dimensions qui décrivent la façon dont le système fournit ses recommandations, en termes de :

- **Degré** : Il s'agit du degré de personnalisation fourni par le système de recommandation. Ce dernier peut recommander les mêmes items pour tous les utilisateurs, peut fournir des recommandations éphémères basées sur les intérêts à court terme d'un utilisateur ou présenter des recommandations persistantes prenant en compte les intérêts à long terme de l'utilisateur.
- **Méthode** : Schafer *et al.*, (2001) citent les méthodes de personnalisation de recommandation suivantes 1) la récupération brute des items lorsqu'aucune méthode de personnalisation

n'est utilisée et que les items recommandés sont présentés comme des résultats de requêtes de recherche, 2) sélection manuelle des recommandations tel que la recommandation d'une liste d'items par des experts, 3) des méthodes de recommandation basées sur le contenu, 4) des méthodes de recommandation basées sur le filtrage collaboratif et 5) des approches hybrides combinant certaines des méthodes précédemment citées.

2.3.3.5 Architecture

Selon (Miller *et al.*, 2004 ; Han *et al.*, 2004), l'architecture du système de recommandation peut être :

- **Centralisée** : Lorsque le système de recommandation est installé sur un seul emplacement.
- **Distribuée** : Lorsque les composants du système de recommandation sont distribués sur plusieurs emplacements.

2.3.3.6 Emplacement

Il s'agit de l'endroit où la recommandation est générée. Elle peut être dans l'une des localisations présentées ci-dessous (Hanini *et al.*, 2001).

- **À la source d'information** : Le cas où la source d'informations ou le fournisseur fournit un système de recommandation à ses utilisateurs sans passer par une entité tierce.
- **Au serveur de recommandation** : Les recommandations sont fournies par un serveur de recommandation appartenant à une entité tierce.
- **Du côté de l'utilisateur** : Les recommandations sont produites localement du côté de l'utilisateur.

2.3.3.7 Mode de recommandation

Le mode utilisé pour fournir les recommandations des items (Herlocker *et al.*, 2004. Schafer *et al.*, 2001) peut être de trois types :

- **Mode push (actif)** : Transmettre les recommandations à l'utilisateur même lorsqu'il n'interagit pas avec le système.
- **Mode pull (actif)** : Produire des recommandations et les présenter à l'utilisateur lorsqu'il le permet ou le demande.
- **Mode passif** : Produire des recommandations dans le cadre de la procédure régulière du système.

2.4 Travaux connexes

Plusieurs systèmes de recommandation dédiés aux EIAH ont été développés ces dernières années. Parmi les premiers systèmes, nous retrouvons *Altered Vista* (Recker et Walker 2003) et *RACOFI* (Anderson *et al.*, 2003). Le premier collecte les évaluations que les utilisateurs attribuent aux ressources pédagogiques et les propage sous forme de recommandations « de bouche-à-oreille » selon les qualités des ressources. Le deuxième est similaire au premier et intègre en plus un moteur d'inférence à base de règles. *LSRS* (Yueh-Min *et al.*, 2009) est un système de recommandation qui se base sur l'analyse des groupes d'apprentissage. *ReMashed* (Drachsler *et al.*, 2009) propose aux apprenants d'évaluer des informations à partir d'un réseau d'apprentissage informel et utilise ces évaluations et les tags associés aux ressources pour effectuer la recommandation.

Drachsler *et al.* (2015) ont étudié 82 systèmes de recommandation dédiés aux EIAH conçus au cours des 15 dernières années. Cette étude présente huit systèmes qui recommandent des ressources pédagogiques dans un environnement social. Nous avons sélectionné ces huit systèmes et ajouté un neuvième qui ne figure pas dans cette étude.

Dans les sections suivantes, nous allons étudier ces neuf systèmes de recommandation, avant de conclure par une synthèse.

2.4.1 Système de recommandation ISIS

ISIS (Drachsler *et al.*, 2009) est un système qui aide l'apprenant à trouver son parcours d'apprentissage en lui recommandant la meilleure ressource suivante selon son profil et les connaissances du domaine. L'apprenant est contraint de sélectionner la seule ressource recommandée. Une recommandation d'une liste de ressources pourrait être plus intéressante pour l'apprenant qu'une ressource unique.

ISIS utilise un système de recommandation qui combine une technique de recommandation basée sur les ontologies et une technique de filtrage par stéréotypes. Le recommandeur décide laquelle des deux techniques sera utilisée pour la situation actuelle de l'apprenant. La première utilise l'information personnelle de l'apprenant et la compare aux connaissances du domaine pour recommander la ressource la plus appropriée. La deuxième technique est basée sur une sélection des ressources les plus populaires dans un groupe d'apprenants spécifique en utilisant un filtrage collaboratif. Ces groupes sont

créés en fonction des attributs des profils des apprenants. Ce système utilise la similarité entre les utilisateurs et n'utilise pas les relations sociales entre eux.

2.4.2 Système de recommandation 3A

Le système de recommandation 3A (El Helou *et al.*, 2010) classe les entités 3A (acteurs, *assets* et activités) en fonction de leur importance pour un acteur spécifique et son contexte. Dans ce système, les *acteurs* peuvent être des utilisateurs ou des agents, les *assets* représentent des ressources et les *espaces d'activité* représentent des moyens pour mener des activités individuelles ou collaboratives. Ce système cible les environnements d'apprentissage collaboratif assistés par ordinateur.

Le système 3A utilise un modèle d'interaction pour identifier et exploiter des interactions significatives avec les utilisateurs, les relations établies et les métadonnées des évaluations telles que les évaluations avec des étoiles, les *bookmarks*, les *tags* et les commentaires fournis par les utilisateurs. Ce système utilise l'algorithme de classement 3A pour classer les acteurs, les *assets* et les activités en fonction de leur popularité globale et locale (popularité dans le voisinage de l'acteur cible et son contexte).

L'approche de recommandation du système 3A se compose des quatre étapes suivantes : 1) construction du graphe, 2) définition du contexte, 3) calcul de l'importance et 4) extraction de la liste classée. Les relations dans le graphe peuvent être l'amitié entre les acteurs, la relation d'auteur entre un acteur et un *asset*, etc. Le contexte cible de l'acteur est représenté par tous les nœuds liés à l'action que cet acteur effectue. L'algorithme de classement 3A est basé sur l'idée clé de l'algorithme Google Pagerank (Page et Brin, 1999) : un nœud est récursivement important si et seulement si plusieurs autres nœuds importants pointent vers ce nœud. Dans l'étape d'extraction de la liste classée, le système extrait des listes distinctes d'acteurs, d'activités et d'*assets* en respectant leurs ordres relatifs dans la liste hétérogène d'origine générée dans la phase précédente. Ce système peut souffrir d'une *sparsity* s'il existe peu d'actions (notations, *tags*, critiques, etc.) d'utilisateurs disponibles dans le système.

2.4.3 Système de recommandation Ensemble

Brusilovsky *et al.* (2010) proposent un système de navigation sociale, appelé *Ensemble*, qui guide l'utilisateur vers les informations les plus

utiles en utilisant les traces des différents utilisateurs. Cette orientation est basée sur la génération des liens pointant vers des ressources et leur annotation adaptative. L'annotation des liens met en évidence les ressources qui sont populaires dans le groupe de l'utilisateur, tandis que la génération de liens peut recommander des ressources qui ont été visitées par le groupe de l'utilisateur dans un contexte similaire.

Ensemble utilise les deux formes existantes de navigation sociale, à savoir le filtrage collaboratif et les espaces d'information enrichis en historique (*history-enriched information spaces*) pour guider les utilisateurs.

Ensemble suit les actions de bas niveau de l'utilisateur (navigation, évaluation, commentaire et marquage avec des tags) et des actions structurelles de niveau supérieur (extraction et composition de fragments). Le premier type d'actions utilisateur permet au système d'identifier des éléments similaires et génère ensuite des liens vers des ressources similaires. Ce système recommande les liens qui ont été visités après une ressource spécifique par le groupe de l'utilisateur.

Dans *Ensemble*, les utilisateurs peuvent annoter des ressources et créer un parcours guidé de ces ressources. L'existence des ressources dans différents parcours peut aider à estimer la similarité de ressource. Cette similitude est utilisée dans le processus de recommandation de lien.

Le système présente des listes de liens vers des ressources aux utilisateurs. Une liste représente un groupe de ressources similaires. *Ensemble* utilise l'intensité de la couleur pour annoter les liens. Plus la couleur est intense, plus la ressource est populaire parmi un groupe d'utilisateurs. Le système étend cette liste à d'autres ressources qui sont similaires aux ressources de cette liste dans le contexte du groupe courant.

Cette approche propose à tous les utilisateurs d'un groupe les mêmes recommandations. Elle n'est pas personnalisée pour chaque utilisateur.

2.4.4 Système de recommandation Topolor

Topolor (Shi *et al.*, 2013) est un système social d'apprentissage adaptatif personnalisé, qui vise à améliorer les interactions sociales dans le processus d'apprentissage. Ce système offre des recommandations personnalisées et certaines fonctionnalités sociales telles que le par-

tage d'un statut d'apprentissage, l'échange simple de questions/réponses et le partage de notes.

Topolor peut recommander un contenu d'apprentissage, des parcours d'apprentissage, des experts, des pairs, des questions et des sujets d'apprentissage. Ce système utilise à la fois un filtrage collaboratif et une recommandation basée sur le contenu.

Topolor recommande des ressources pédagogiques en fonction du nombre de *tags* qui sont partagés avec la ressource que l'utilisateur voit actuellement. Malgré le fait que Topolor est une plateforme sociale, elle n'utilise pas les relations sociales pour la recommandation. Ce système utilise des méthodes classiques telles que le filtrage collaboratif et la recommandation basée sur le contenu.

2.4.5 Système de recommandation basé sur le style d'enseignement

Tout comme les apprenants, les enseignants ont également besoin de recommandations sur les ressources afin de développer leurs compétences personnelles et professionnelles. Limongelli *et al.* (2013) proposent un système de recommandation pour les enseignants. Ils proposent de regrouper les enseignants en quatre groupes sur la base de leurs styles d'enseignement. Cette classification utilise l'algorithme de *clustering* K-means en tenant en compte du style d'enseignement de chaque enseignant. Chaque groupe d'utilisateurs aura les mêmes attitudes d'enseignement.

Les auteurs considèrent les cinq styles d'enseignement de Grasha (1996) : expert, modèle personnel, autorité formelle, délégué et facilitateur. Chacun de ces styles d'enseignement est évalué dans une fourchette de 1 à 7. Grasha identifie aussi quatre groupes d'enseignants, en fonction de leurs styles d'enseignement primaires et secondaires. Chaque groupe représente un réseau d'enseignants similaires.

Le système propose à tous les enseignants d'un groupe particulier les mêmes recommandations de ressources pédagogiques. Ce système n'est pas personnalisé pour chaque utilisateur et ne traite que les enseignants et non les apprenants.

2.4.6 Système de recommandation social basé sur les sentiments de l'apprenant

Karampiperis *et al.* (2014) utilisent les techniques d'analyse des sentiments (*opinion mining*) sur les commentaires générés par l'utilisateur lorsque l'évaluation de l'utilisateur est manquante. Ce système est utilisé pour améliorer les recommandations des ressources péda-

gogiques dans les systèmes de recommandation basés sur le filtrage collaboratif.

Ce système génère une évaluation pour une ressource pédagogique à partir des activités des utilisateurs (évaluation et commentaire) sur cette ressource. Il génère également des évaluations pour les ressources qui n'ont pas d'activités. Le premier cas est considéré comme une évaluation explicite et le second comme une évaluation implicite.

L'évaluation explicite est directe lorsqu'un utilisateur évalue la ressource par le système d'étoiles et peut être indirect lorsque le système la calcule à l'aide de commentaires liés à cette ressource. L'évaluation implicite (prédite) est calculée en utilisant la similarité d'un utilisateur avec d'autres lorsque cet utilisateur n'a pas explicitement fourni une évaluation pour une ressource spécifique.

Lorsqu'un utilisateur n'a pas fourni une évaluation explicite pour une ressource donnée et n'a pas d'activité en commun avec d'autres utilisateurs, le système calcule une évaluation implicite basée sur la communauté en utilisant la moyenne des évaluations fournies par les utilisateurs appartenant au même groupe d'utilisateurs.

L'analyse du sentiment extrait les avis des utilisateurs à partir des commentaires pour refléter la satisfaction perçue de l'utilisateur sur la ressource. Le système classe alors cette satisfaction en valeurs positives, négatives ou neutres.

Ce système propose les mêmes recommandations à un groupe dans le cas d'une évaluation implicite basée sur la communauté. Il n'est pas personnalisé pour chaque utilisateur.

2.4.7 Système de recommandation social basé sur la confiance

Dans (Fazeli *et al.*, 2014 a), les auteurs proposent un système de recommandation qui aide les enseignants à trouver des ressources d'apprentissage en fonction de leurs besoins et de leurs intérêts.

Ce système utilise des méthodes de filtrage collaboratif. Il repose sur l'idée que les utilisateurs préfèrent recevoir des recommandations par des personnes en qui ils ont confiance. L'approche de ce système résout le problème de la *sparsity* en supposant que la confiance peut être transitive. Cette transitivité peut aider à établir une relation entre deux utilisateurs qui n'ont pas d'ensemble commun de ressources évaluées mais qui ont des amis en commun. Dans ce cas, une nouvelle relation de confiance entre deux utilisateurs non connectés peut être déduite.

Cette approche calcule la confiance entre les utilisateurs en fonction de l'évaluation des ressources et vise, à l'avenir, à intégrer des informations sur les interactions des utilisateurs dans ce calcul. Les auteurs n'ont pas expliqué comment ils intègrent cette dernière partie dans leur approche. Cette approche repose toujours sur les évaluations des utilisateurs et n'utilise pas les activités d'autres utilisateurs sur le système ou les informations de profil des utilisateurs.

Les auteurs ont proposé d'utiliser également un indice de confiance (indice T) pour mesurer le niveau de confiance d'un utilisateur. Cet indice a été inspiré de l'indice H (H index), qui est un indicateur des publications scientifiques d'un auteur (Hirsch, 2005). L'approche proposée par ce système est inachevée et n'a pas été évaluée.

2.4.8 Système de recommandation social basé sur le parcours de graphes

Fazeli *et al.* (2014 b) ont développé un système de recommandation sociale qui combine un système de gestion de l'apprentissage traditionnel et des réseaux sociaux commerciaux comme Facebook. Ce système utilise une méthode de parcours de graphe pour recommander des ressources pédagogiques. Il améliore la précision des recommandations même lorsque les données des évaluations sont *sparse*.

Ce système construit d'abord un graphe où les nœuds représentent les utilisateurs et les bords représentent les relations de similarité entre les utilisateurs. Deuxièmement, il collecte des recommandations pour un utilisateur donné en parcourant ses voisins.

L'algorithme utilisé pour créer le graphe prend en compte l'indice social (indice S) pour chaque utilisateur. Cet indice était également inspiré de l'indice H. L'indice social est utilisé pour aider à trouver des utilisateurs similaires

L'approche proposée par ce système infère de nouvelles relations et similarités entre les utilisateurs. Elle utilise une similarité inférée et dépend principalement des évaluations. Elle ne maximise pas l'utilisation des informations présentes dans le réseau social.

2.4.9 Système de recommandation sémantique pédagogique

Santos et Boticario (Santos et Boticario, 2015) proposent une méthode pour identifier 32 règles de recommandation dans les environnements d'apprentissage social en ligne. L'idée est de tenir compte de plusieurs types d'actions, tel que recommander à un apprenant de rejoindre un groupe, lire un contenu, donner des commentaires sur certaines contributions, discuter avec un tuteur et travailler plus dure

dans l'activité courante. Ces types d'actions sont donnés selon certaines conditions (l'apprenant n'a pas encore réalisé d'activité, l'apprenant a réussi l'évaluation, l'apprenant n'a pas encore lu le contenu prérequis, etc.). Les enseignants ont été impliqués dans le processus de définition de ces règles de recommandation.

Cette approche propose également un système de recommandation pédagogique sémantique qui fournit des recommandations appropriées aux apprenants selon ces règles.

Cette approche s'appuie sur l'utilisation d'outils sociaux, tels que le chat et les forums, et recommande des actions qui peuvent être réalisées dans ces outils.

2.4.10 Synthèse

Les liens sociaux et la similarité entre utilisateurs sont des éléments importants dans le calcul des recommandations dans les systèmes de recommandation sociale. Les systèmes précédemment cités ne tirent pas profit de toutes les informations sociales d'un environnement d'apprentissage social pour calculer la similarité. Soit ces systèmes n'utilisent pas la similarité, soit ils utilisent la même similarité pour tous les utilisateurs d'un même groupe (par exemple, dans le Système de recommandation Ensemble), soit les similarités ne dépendent que des évaluations (par exemple, dans le Système de recommandation social basé sur les sentiments de l'apprenant). À part ISIS et le « Système de recommandation sémantique pédagogique », les autres systèmes sont toujours basés sur les différentes activités des utilisateurs dans le système (évaluations, visites, tags, commentaires, etc.) et n'intègrent pas les informations du profil des utilisateurs. En outre, les systèmes présentés n'utilisent pas les informations présentes dans les profils d'utilisateurs similaires. Aucun d'entre eux, sauf le dernier système, ne traite les connaissances acquises par les apprenants pour mieux personnaliser les recommandations.

Seuls les deux systèmes « Système de recommandation social basé sur la confiance » et « Système de recommandation social basé sur le parcours de graphes » abordent le problème de la *sparsity*, en utilisant respectivement, la confiance transitive et les méthodes de parcours de graphes.

Trois des systèmes étudiés (Système de recommandation Ensemble, Système de recommandation social basé sur les sentiments de l'apprenant, et Système de recommandation basé sur le style d'enseignement) proposent à tous les utilisateurs du groupe les

mêmes recommandations. Ces dernières ne sont pas personnalisées pour chaque utilisateur.

Tous les systèmes présentés proposent un seul type de recommandation de ressources. Dans leurs recommandations, il n'est pas possible de faire une distinction entre les ressources les plus appréciées, les ressources utiles et les ressources qui sont visités récemment par leur réseau d'amis. Parfois, les utilisateurs veulent avoir des ressources qui ont été utiles pour des utilisateurs similaires à eux, avoir des ressources populaires et recevoir des recommandations des ressources que les utilisateurs similaires sont en train de consulter. Aucun des systèmes susmentionnés n'a abordé l'utilité des ressources selon les domaines d'apprentissage des étudiants.

Le tableau 2.2 présente les principales caractéristiques des dix systèmes de recommandation pédagogique sociale présentés. Pour chacun d'entre eux, nous mentionnons les types de ressources recommandées aux apprenants, la méthode de recommandation, les éléments utilisés pour calculer les recommandations et si le système traite le problème de *sparsity* des évaluations. Le *Na (not applicable)* signifie que le système n'utilise pas les évaluations. Donc cette caractéristique n'est pas applicable pour ce système.

Tableau 2.2 Synthèse des systèmes de recommandation pédagogique sociale.

Système	Types d'items recommandés	Méthode de recommandation	Éléments utilisés dans la recommandation	Traite la <i>sparsity</i> des évaluations
Système ISIS	Prochaine meilleure ressource pédagogique	Méthode basée sur les ontologies et sur les stéréotypes et filtrage collaboratif	Profil de l'apprenant et connaissances du domaine	Na
Système 3A	Utilisateurs, ressources pédagogiques et activités	Méthode basée sur les graphes	Interactions des utilisateurs, évaluations des relations, signets, tags et avis	Non
Système Ensemble	Ressources pédagogiques	Filtrage collaboratif et espaces d'information enrichis en historique	Visites, évaluations, commentaires et tags	Non
Système Topolor	Ressources pédagogiques, parcours d'appren-	Filtrage collaboratif et filtrage basé sur	Visites, tags, questions et réponses	Na

	tissage, experts, pairs, questions et sujets d'apprentissage	le contenu	des utilisateurs	
Système basé sur le style d'enseignement	Ressources pédagogiques	K-means clustering	Style d'enseignement	Na
Système social basé sur les sentiments de l'apprenant	Ressources pédagogiques	Filtrage collaboratif, analyse des sentiments et méthode de stéréotypes	Evaluations, commentaires	Oui
Système social basé sur la confiance	Ressources pédagogiques	Filtrage collaboratif et méthode basée sur les graphes	Evaluations et relations entre utilisateurs	Oui
Système social basé sur le parcours de graphes	Ressources pédagogiques	Méthode basée sur les graphes	Évaluations	Oui
Système sémantique pédagogique	Actions de l'utilisateur (Certaines actions concernent les ressources pédagogiques)	32 règles de recommandation	Actions de l'utilisateur	Na

Nous pouvons conclure que ces systèmes 1) ne bénéficient pas de toutes les informations liées à un utilisateur et à ses amis pour améliorer les recommandations, 2) ne tiennent pas compte de la nécessité d'avoir plusieurs types de recommandations (ressources populaire, utile, récemment utilisés, etc.), 3) la plupart d'entre eux ne traitent pas la *sparsity*, et 4) certains d'entre eux ne personnalisent pas leurs recommandations proposant uniquement les mêmes ressources à tous les utilisateurs d'un groupe.

Notre approche présentée dans le prochain chapitre a pour objectif de proposer une solution pour répondre à ces quatre caractéristiques cibles.

2.5 Évaluation des systèmes de recommandation

Shani et Gunawardana (2011) expliquent qu'initialement la plupart des systèmes de recommandation ont été évalués en fonction de leur

capacité à prédire avec précision les choix de l'utilisateur. Maintenant, il est largement admis que la précision des prédictions est cruciale mais insuffisante pour déployer un bon système de recommandation.

Selon Herlocker *et al.* (2004), les évaluations des systèmes de recommandations peuvent être effectuées en utilisant une analyse hors ligne (*offline analysis*) ou une expérimentation avec des utilisateurs réels (*live user experiment*).

Il existe une autre classification des méthodes d'évaluation des systèmes de recommandation. Comme expliqué dans (Erdt *et al.*, 2015), ces méthodes d'évaluation sont classées en trois types : expérimentations *offline*, études avec des utilisateurs (*user studies*) et tests réels (*real life testing*). Ce dernier type est nommé expérimentations en ligne (*Online experiments*) par Shani et Gunawardana (2011).

2.5.1 Évaluation *offline*

Une grande partie du travail d'évaluation des algorithmes des systèmes de recommandation s'est concentrée sur l'analyse hors ligne de la précision des prédictions que peuvent faire ces systèmes (Herlocker *et al.*, 2004). Les évaluations *offlines* utilisent des ensembles de données (*dataset*) constitués d'actions des utilisateurs (principalement des évaluations de ressources). Les évaluations *offlines* simulent le processus de recommandation où un sous ensemble des actions utilisateurs du *dataset* est caché et le système de recommandation prédit ces actions cachées. Le système de recommandation est évalué en fonction de sa capacité à prédire ces interactions cachées. Les résultats de ces prédictions sont analysés en utilisant une ou plusieurs métriques.

Deux types d'ensembles de données sont souvent utilisés dans ces évaluations (Erdt *et al.*, 2015) :

- Ensembles de données naturels : ils sont constitués de données issues de l'historique des interactions d'utilisateurs réels dans un système donné sur une période donnée. De nombreux *dataset* sont disponibles pour mener des évaluations sur des algorithmes de recommandation. L'annexe III présente un comparatif entre les *dataset* les plus connus dans les EIAH.
- Ensembles de données de synthèse : ils sont construits de données artificielles. Ce type de *dataset* est habituellement utilisé pour tester comment les algorithmes de recommandation fonctionnent dans certaines conditions.

Les évaluations *offlines* ont l'avantage d'être rapide et économique et peuvent être réalisées sur plusieurs ensembles de données ou plusieurs algorithmes différents à la fois. Ce type d'évaluation est une évaluation objective des résultats de la prédiction. Aucune analyse hors ligne ne peut déterminer si les utilisateurs préfèrent un système particulier, soit en raison de ses prédictions, soit en raison d'autres critères moins objectifs tels que l'esthétique ou l'ergonomie de l'interface utilisateur.

2.5.1.1 Protocoles d'évaluation

Une fois le *dataset* préparées, un protocole d'évaluation doit être sélectionné selon (Gabrielsson et Gabrielsson, 2006). Il en existe trois selon ces mêmes auteurs :

- Retrait (*Hold out*) : un pourcentage du *dataset* est utilisé comme données d'apprentissage tandis que le reste est utilisé comme données de test.
- Tous sauf 1 (*All but 1*) : Un seul élément du *dataset* est utilisé comme données de test, ce qui maximise la quantité de données utilisées comme données d'apprentissage.
- Validation croisée K-fold (*K-fold cross validation*) : Les données d'évaluation sont divisées K fois en données de formation et en données de test.

Pour chacune des k validations croisées, il est possible d'utiliser le retrait ou le tout sauf un. La validation croisée K-fold est plus intéressante que les deux premières car elle permet de maximiser la possibilité qu'un élément du *dataset* soit dans les données de tests et les données d'apprentissage.

Les mêmes auteurs expliquent également que la répartition du *dataset* en données d'apprentissage et en données de test est effectuée de l'une des trois méthodes suivantes :

- Diviser les utilisateurs : un pourcentage ou un nombre fixe des évaluations de chaque utilisateur sont utilisés comme données de test et le reste comme données d'apprentissage.
- Diviser les items : un pourcentage ou un nombre fixe d'évaluations de chaque item est utilisé comme données de test et le reste comme données d'apprentissage.
- Diviser les évaluations : un pourcentage ou un nombre fixe d'évaluations est utilisé comme données de test, et le reste comme données d'apprentissage.

Les deux premières approches garantissent respectivement qu'un utilisateur ou un item apparaît dans les données d'apprentissage et les données de test.

2.5.1.2 Métriques d'évaluation

Les métriques d'évaluation mesurent la capacité de l'algorithme de recommandation à prédire correctement les évaluations connues des utilisateurs. Les métriques d'évaluation utilisent des évaluations correctes d'un *dataset* de test et les comparent avec des prédictions proposées par l'algorithme de recommandation en se basant sur un *dataset* d'apprentissage. La majorité de ces métriques proviennent du domaine de la recherche d'information. La précision et le rappel (Cleverdon et Kean 1968) font partie de ces métriques. Ces deux métriques composent la formule de calcul d'une autre métrique qui est le F1 (Sarwar *et al.*, 2000). D'autres métriques sont utilisées pour évaluer les systèmes de recommandation telles que MAE (Mean Absolute Error) (Shardanand et Maes 1995) et RMSE (Root Mean Squared Error) (Thai-Nghe *et al.*, 2010).

Gabrielsson et Gabrielsson (2006) et Shani et Gunawardana, (2011) citent plus ce que 20 métriques qui peuvent être utilisées pour évaluer les systèmes de recommandation en *offline*. Les chercheurs qui souhaitent évaluer ces systèmes doivent choisir une ou plusieurs métriques d'évaluation. Herlocker et al. (2004) expliquent que ce choix peut dépendre des réponses à des questions telles que : Les résultats obtenus avec la métrique choisie sont-ils comparables à d'autres travaux de recherche ? Une mesure sera-t-elle suffisamment sensible pour détecter les différences réelles qui existent ?

2.5.2 Étude utilisateurs

McNee *et al.* (2006) et Knijnenburg *et al.* (2012) expliquent que l'objectif d'un système de recommandation va au-delà des métriques de précision. Les évaluations se basant sur ces métriques de précision ne répondent pas à la question de savoir si les utilisateurs sont réellement satisfaits des recommandations proposées par le système. Comme mentionné dans Fazeli *et al.* (2017), près de 50% du succès commercial d'un système de recommandation est lié aux interactions entre les utilisateurs et ce système alors que l'algorithme de recommandation ne compte que pour 5%.

Le point de vue de l'utilisateur sur le système de recommandation est encore plus important dans le domaine de l'éducation (Chatti *et al.*, 2013 ; Erdt *et al.*, 2015). En plus de la précision, les systèmes

de recommandation appartenant à ce domaine doivent prendre en compte des mesures de qualité telles que l'utilité, la nouveauté ou la diversité des recommandations.

Le point de vue de l'utilisateur peut être recueilli via une étude utilisateurs. Il s'agit d'une méthode utilisée pour découvrir comment un système de recommandation influence l'expérience, la perception et les interactions d'un utilisateur avec un système (Knijnenburg, 2012). Une étude utilisateurs est menée en recrutant un ensemble d'utilisateurs, et en leur demandant d'effectuer certaines tâches dans un environnement contrôlé pendant une courte période de temps. L'interaction entre les utilisateurs et le système de recommandation est observée et des informations sont enregistrées telles que le temps nécessaire pour terminer la tâche ou la qualité des résultats de la tâche. En plus de l'observation du comportement de l'utilisateur, il est possible de faire passer des questionnaires aux utilisateurs pour recueillir des données qui ne sont pas directement observables telles que l'appréciation de l'interface utilisateur ou de la pertinence des recommandations.

2.5.3 Évaluation *online*

L'évaluation *online* peut aussi recueillir le point de vue de l'utilisateur concernant le système de recommandation. Dans ce type d'évaluation, des utilisateurs réels utilisent le système dans des conditions réelles sur une longue période (Manouselis *et al.*, 2012). Selon Herlocker *et al.* (2004), ce type d'évaluation peut montrer les usages et les habitudes d'utilisation des utilisateurs, les problèmes et les besoins non satisfaits, et les problèmes que les chercheurs n'ont peut-être pas envisagés dans une étude utilisateurs.

Avec ces tests réels sur le terrain, la plupart des objectifs centrés sur l'utilisateur peuvent être efficacement évalués, comme l'évaluation de l'expérience utilisateur, la satisfaction des utilisateurs ou la rétention des utilisateurs (Herlocker *et al.*, 2004 ; Shani et Gunawardana, 2011).

2.5.4 Questionnaires

Les études utilisateur et les expérimentations *online* peuvent utiliser des questionnaires pour récolter les avis des utilisateurs sur leurs expériences sur le système. Ces questionnaires peuvent être posés avant, pendant et après que les utilisateurs utilisent le système. Plu-

sieurs questionnaires d'évaluation centrés sur l'utilisateur ont été proposés.

2.5.4.1 Questionnaire Content Quality Scale

Le questionnaire CQS (*Content Quality Scale*) a été développé pour évaluer l'opinion des apprenants sur le contenu recommandé par un EIAH (Shi, 2014). Ci-dessous les 9 questions qui le composent que nous avons traduit en français :

1. Le contenu recommandé par le système était ce que je voulais apprendre.
2. Le contenu recommandé par le système était ce que j'avais besoin d'apprendre.
3. Il était facile de reconnaître le contenu recommandé par le système.
4. J'étais confiant sur le fait que j'aimerais le contenu qui me sera recommandé.
5. J'ai compris pourquoi le système m'a recommandé certains contenus.
6. J'ai compris comment le contenu est organisé dans le système.
7. Le contenu fourni par le système était compréhensible.
8. Le contenu fourni par le système était suffisant.
9. Le contenu fourni par le système était utile.

2.5.4.2 Questionnaire ResQue

ResQue (*Recommender systems' Quality of user experience*) est une plateforme d'évaluation des systèmes de recommandation en général proposée par Pu *et al.* (2011). Ce questionnaire se compose de 60 questions classées en quatre dimensions principales : les qualités perçues du système, les croyances des utilisateurs, les attitudes subjectives et les intentions comportementales. ResQue propose aussi une version simplifiée avec 15 questions. La version anglaise de ce questionnaire est présentée dans l'annexe II.

ResQue sert à évaluer les qualités des systèmes de recommandation telles que leur utilisabilité, leur utilité, leurs qualités d'interface et d'interaction, la satisfaction des utilisateurs à propos de ces systèmes et l'influence de ces qualités sur les intentions de comportement de ces utilisateurs. Ces intentions peuvent être le fait d'acheter les produits qui leur sont recommandés, de revenir visiter le système à l'avenir et de parler de ce système à leurs amis.

ResQue est plus riche et plus organisé que CQS avec ses 60 questions et ses 4 dimensions. A titre d'exemple, contrairement à ResQue, CQS ne traite pas les différents aspects de la qualité des ressources recommandées (nouveau, variété des ressources, etc.). CQS ne traite aussi pas les intentions comportementales que peuvent avoir les utilisateurs sur le système de recommandation. Dans notre travail, nous avons utilisé le questionnaire ResQue par ce qu'il répond aux hypothèses que nous nous sommes fixés dans notre évaluation.

2.6 Conclusion

Nous avons présenté dans ce chapitre l'état de l'art des systèmes de recommandations dans le contexte des EIAH et dans le contexte des réseaux sociaux en ligne. Cet état de l'art détaille certaines méthodes et techniques de recommandation telles que le filtrage collaboratif et la recommandation multicritères que nous utilisons dans notre approche. Nous avons ainsi identifié 9 travaux qui traitent les systèmes de recommandations de ressources pédagogiques dans un contexte social. Nous avons noté que ces travaux présentent certains problèmes : 1) ils ne bénéficient pas de toutes les informations sur l'utilisateur, 2) ils n'offrent qu'un type de recommandation, 3) la plupart ne traitent pas la *sparsity*, et 4) certains ne personnalisent pas leurs recommandations. Ces problèmes seront traités dans notre contribution qui est présentée dans le chapitre suivant.

Nous avons présenté les différentes méthodes d'évaluation des systèmes de recommandations. Nous pouvons noter qu'il est plus facile d'effectuer des expériences *offline* en utilisant des ensembles de données existants pour estimer la précision des prédictions quand ces ensembles de données contiennent les données dont nous avons besoin. Il est plus coûteux de faire une étude utilisateur sur un petit groupe ou de faire une expérience à grande échelle sur un système déployé avec des utilisateurs réels. Cependant, ces études en situations réelles sont la seule solution pour recueillir l'avis des utilisateurs.

3 Contribution : approche pour la recommandation de ressources pédagogiques basée sur les liens sociaux

3.1	Aperçu de l'approche	57
3.2	Base formelle de l'approche	60
3.3	Similarité sociale	61
3.4	Types de recommandations	64
3.4.1	Recommandation des ressources visitées récemment	64
3.4.2	Recommandation des ressources populaires	65
3.4.3	Recommandation des ressources utiles	65
3.5	Gestion de la sparsity	66
3.6	Mise à jour du profil	66
3.7.	Validation avec des données simulées	67
3.7.1.	Objectif	67
3.7.2.	Données de la simulation	68
3.7.2.1.	<i>Caractéristiques des apprenants</i>	68
3.7.2.2.	<i>Relations sociales entre apprenants</i>	69
3.7.2.3.	<i>Visites et évaluation des ressources</i>	71
3.7.3.	Similarité sociale	71
3.7.4.	Ressources recommandées	71
3.7.4.1.	<i>Ressources visitées récemment</i>	71
3.7.4.2.	<i>Ressources populaires</i>	72
3.7.4.3.	<i>Ressources utiles</i>	72
3.8	Conclusion	73

Dans ce chapitre, nous allons présenter dans un premier temps le principe général de notre approche et la base formelle qui sera utilisée par la suite dans notre proposition. Ensuite, nous présenterons les quatre parties de notre proposition : 1) calcul de la similarité entre les utilisateurs (apprenants) dans un environnement d'apprentissage social, 2) sélection des ressources qui conviennent le mieux aux préférences des utilisateurs, 3) traitement du problème de la *sparsity* et 4) mise à jour les profils utilisateurs. À la fin de ce chapitre, avant de conclure, nous présenterons un exemple qui illustre notre approche. Cet exemple a servi de *dataset* pour faire une première évaluation de notre approche et ainsi affiner le modèle formel.

3.1 Aperçu de l'approche

L'approche proposée repose sur les principes de :

- l'influence sociale (Sun et Tang 2011) : deux amis ont tendance à avoir les mêmes intérêts ;
- la régularité co-citation (Bhagat *et al.* 2011) (*Co-citation regularity*) : les personnes qui ont des similitudes sont susceptibles d'avoir les mêmes intérêts.

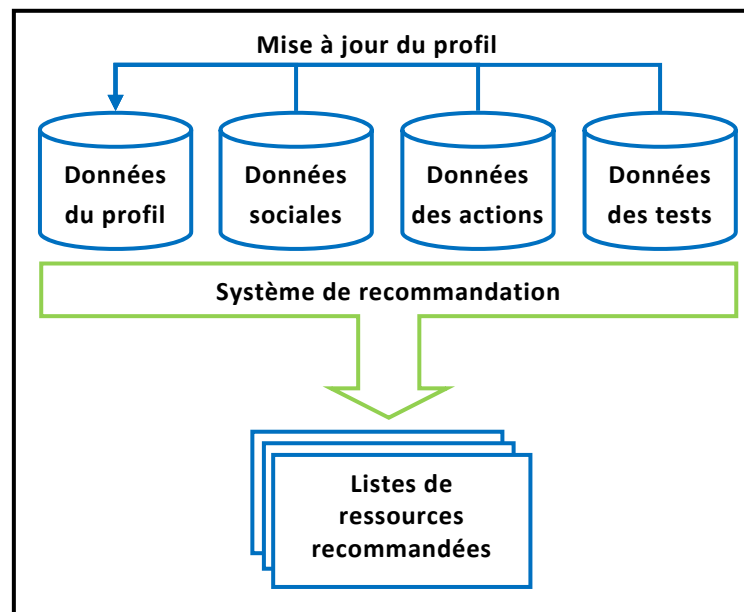


Figure 3.1 Architecture globale de l'approche proposée

Comme le montre l'architecture de la figure 3.1, notre approche repose sur quatre types d'information :

- les informations du profil utilisateur qui décrivent ses préférences, ses connaissances, ses expériences, etc.,
- les informations sociales qui décrivent les liens entre les utilisateurs et les groupes d'utilisateurs et les liens entre les utilisateurs eux-mêmes,
- les informations qui décrivent les actions effectuées par les utilisateurs sur les ressources éducatives telles que les visites ou les évaluations et
- les informations qui décrivent les résultats des exercices effectués par les apprenants, telles que les résultats d'un quiz qu'ils ont fait.

Ces données seront toutes utilisées pour recommander des ressources éducatives pertinentes. L'environnement d'apprentissage récupère les résultats des exercices effectués par les apprenants et met à jour leurs niveaux de connaissances stockés dans leurs profils. Les actions des apprenants et les données sociales sont aussi utilisées dans la mise à jour du profil.

Le principe général de notre approche (Tadlaoui 2014. Tadlaoui *et al.* 2014. Tadlaoui *et al.* 2015a. Tadlaoui *et al.* 2015b. Tadlaoui *et al.* 2018) est illustré par la figure 3.2. Chaque utilisateur de l'EIAH est décrit par des informations qui caractérisent son profil et est lié par des liens d'amitié avec d'autres utilisateurs du système. Dans cette approche, nous définissons la relation d'amitié comme un lien explicitement déclaré par un utilisateur du système. Les relations d'amitié sont déclarées par les utilisateurs via des demandes d'ajout à la liste d'amis comme sur Facebook ou LinkedIn⁵. Deux types d'amitié sont pris en charge, des amis que l'utilisateur connaît dans la vraie vie et a déclarés comme des amis sur le réseau social et ceux qu'il ne connaît que sur le réseau social. Dans les deux cas, suivant le principe de l'influence sociale, ces amis peuvent avoir les mêmes intérêts d'apprentissage, et suivant le principe de la régularité de la co-citation, si ces amis ont des points communs, ils seront donc susceptibles d'avoir les mêmes intérêts d'apprentissage. Les utilisateurs peuvent effectuer trois types d'actions sur les ressources éducatives, à savoir : visiter, évaluer la qualité des ressources et évaluer leur utilité. L'utilité est donnée suivant un domaine d'intérêt de l'utilisateur alors que la qualité est donnée par l'utilisateur indépendamment de son domaine d'intérêt (si le système est utilisé dans une université, les domaines peuvent être les modules enseignés ou les spécialités).

Le système de recommandation utilise toutes les informations ci-dessus pour générer des recommandations personnalisées pour chaque utilisateur de l'environnement d'apprentissage.

Les apprenants sont intéressés à obtenir des recommandations appropriées pour leurs besoins pédagogiques. Ils peuvent être intéressés par les ressources les plus populaires chez leurs utilisateurs similaires. Ils peuvent également être intéressés par les ressources qui ont été récemment visitées par leur réseau d'amis pour les encourager à

⁵ <https://www.linkedin.com/>

collaborer avec leurs amis sur ces ressources. Les utilisateurs peuvent également être intéressés par des ressources qui sont utiles dans leurs domaines d'intérêt. Contrairement aux systèmes de recommandation sociale existants pour les EIAH, qui ne traitent que le premier type de recommandation, notre approche traite ces trois types de ressources : populaires, récemment visités et les ressources utiles pour un domaine d'apprentissage.

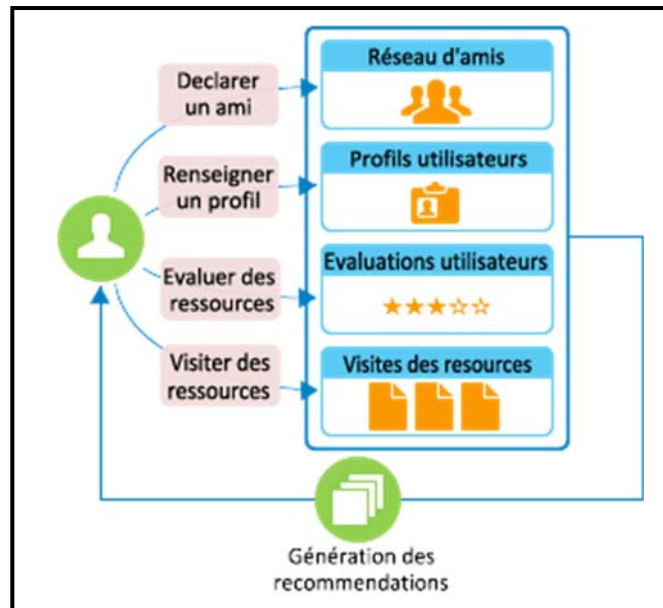


Figure 3.2 Principe général de l'approche proposée

L'algorithme de recommandation s'appuie sur des modèles formels qui calculent la similarité sociale entre les utilisateurs d'un environnement d'apprentissage pour générer trois types de recommandation, à savoir la recommandation 1) des ressources populaires, 2) des ressources récemment visitées et 3) des ressources utiles.

Le processus de recommandation comprend les étapes suivantes :

1. L'utilisateur courant sélectionne l'un des trois types de recommandation ;
2. Le système de recommandation sélectionne les amis de l'utilisateur courant ;
3. Le système calcule le niveau de similarité entre l'utilisateur courant et ses amis. Le calcul de similarité est présenté dans la section 3.3 ;

4. Le système calcule le score des ressources éducatives en se basant sur les visites et les évaluations effectuées par les amis de l'utilisateur courant sur les ressources. La formule de calcul de ce score est présentée dans la section 3.4 ;
5. Le système affiche à l'utilisateur courant une liste de ressources pédagogiques, classées par les scores en fonction du type de recommandation sélectionné.

L'approche utilise un modèle formel pour appuyer ce processus de recommandation. Il est présenté ci-dessous.

3.2 Base formelle de l'approche

Dans cette section, nous définissons les concepts de base permettant le calcul des scores pour la recommandation de ressources pédagogiques. Ces concepts, présentés dans le tableau 3.1, sont liés au profil de l'utilisateur, à ses relations sociales et aux actions des utilisateurs sur les ressources (visites et évaluation).

Tableau 3.1 Base formelle de l'approche

Élément	Description
U	L'ensemble de tous les utilisateurs de l'environnement d'apprentissage
B[u]	$B[u] = \{v \in U : v \text{ ami avec } u\}$. Il représente l'ensemble des amis de l'utilisateur u
IV[u]	L'ensemble des ressources visitées par l'utilisateur u
IE[u]	L'ensemble des ressources évaluées par rapport à leurs qualités par l'utilisateur u
IU[u]	L'ensemble des ressources évaluées en termes d'utilité par l'utilisateur u
D	L'ensemble des domaines pédagogiques (domaines d'intérêt) représentés dans l'environnement d'apprentissage. Par exemple, si l'EIAH est utilisé dans un laboratoire de recherche alors les domaines peuvent être les thèmes de recherches. S'il s'agit d'une université, les domaines peuvent être les modules enseignés ou les spécialités
E[u]	$E[u] = \{d \in D : d \text{ est un domaine de } u\}$. Il représente les domaines d'intérêt de l'utilisateur u
C	Le profil de l'utilisateur est composé de deux parties (Huynh <i>et al.</i> 2012) : une partie statique et une autre dynamique. La partie statique contient des données qui ne changent pas, telles que le nom, date de

	naissance, etc. La partie dynamique quant à elle contient des informations qui peuvent évoluer dans le temps telles que le niveau de connaissance de l'utilisateur, ses préférences, etc. Toutes ces informations, statiques et dynamiques, sont représentées par l'ensemble C
C[u]	$C[u] = \{(c, val) \mid c \in C : val \text{ est la valeur de la caractéristique } c \text{ de l'utilisateur } u\}$. Il représente l'ensemble formé de couples caractéristique/valeur qui définissent le profil de l'utilisateur u. Les valeurs de type continu peuvent être remplacées par des valeurs discrètes. Par exemple la valeur de l'âge peut être transformée en enfant, adolescent, adulte, etc.
Vis(u, i)	Utilisée pour connaître les ressources qui sont visitées par un utilisateur donné. Cette fonction est égale à 1 si l'utilisateur u a consulté la ressource i et 0 dans le cas contraire
t(u, i)	Représente le nombre de jours écoulés depuis la date de la dernière visite de la ressource i par l'utilisateur u
Eval(u, i)	Utilisée pour connaître l'évaluation d'une ressource en termes de qualité par un utilisateur. Par exemple, l'utilisateur u peut évaluer la ressource i de 1 à 5. Si la ressource n'a pas été évaluée alors cette fonction est égale à 0
Eval(u, .)	Représente l'évaluation moyenne en termes de qualité de l'utilisateur u des ressources qu'il a évalué
Util(u, i, d)	En plus de l'évaluation d'une ressource, un utilisateur peut évaluer son utilité par rapport à un domaine spécifique. Elle représente l'évaluation de l'utilisateur u par rapport à l'utilité de la ressource i dans le cadre du domaine d
Util(u, ., d)	Représente l'évaluation moyenne en termes d'utilité de l'utilisateur u des ressources qu'il a évalué dans le cadre du domaine d
Seval[u, v]	$Seval = IE[u] \cap IE[v]$: représente l'ensemble des ressources co-évaluées en termes de qualité par les utilisateurs u et v
Sutil[u, v]	$Sutil = IU[u] \cap IU[v]$: représente les ressources co-évaluées en termes d'utilité par les utilisateurs u et v

3.3 Similarité sociale

La plupart des travaux sur les systèmes de recommandation, qui se basent sur le filtrage collaboratif, utilisent le coefficient de corrélation de Pearson (Herlocker *et al.* 2004) pour calculer les similarités entre les utilisateurs d'un système. Ce coefficient mesure la corrélation linéaire entre deux variables. Dans les systèmes de recommandation, ce coefficient sert à mesurer la dépendance entre deux vecteurs d'évaluations de ressources appartenant à deux utilisateurs. Ces tra-

vous s'intéressent principalement à l'évaluation des ressources pour calculer la similarité entre les utilisateurs.

Étant donné que notre travail s'inscrit dans le contexte des réseaux sociaux à destination d'apprentissage, notre approche améliore la mesure de similarité entre les utilisateurs en intégrant l'information sociale présente sur la plateforme d'apprentissage social. La méthode proposée pour calculer la similarité entre deux utilisateurs, notée u et v , est fondée sur :

- la similarité entre u et v en termes de ressources visitées, notée $VisSim(u, v)$;
- les similarités en termes d'évaluation de ressources, notée $EvalSim(u, v)$ et $UtilSim(u, v)$;
- la similarité entre les profils de ces deux utilisateurs, notée $ProfilSim(u, v)$;
- la force de lien entre u et v , notée $LinkS(u, v)$.

Cette nouvelle formule, notée $SocialSim(u, v)$, représente la similarité sociale entre les utilisateurs u et v . Elle est calculée comme suit :

$$\begin{aligned}
 SocialSim(u, v) &= (EvalSim(u, v) + UtilSim(u, v) + VisSim(u, v) + LinkS(u, v) \\
 &+ ProfilSim(u, v)) / 5 \quad \quad \quad Eq (3.1).
 \end{aligned}$$

Pour le calcul de la similarité en termes d'évaluation $EvalSim(u, v)$, définie par l'équation Eq (3.2), nous nous sommes basés sur le coefficient de corrélation de Pearson. Nous avons adapté ce coefficient pour le calcul de la similarité en termes d'utilité $UtilSim(u, v)$, définie par l'équation Eq (3.3), en y intégrant les domaines d'intérêt des utilisateurs.

$$\begin{aligned}
 EvalSim(u, v) &= \frac{\sum_{i \in Seval[u, v]} (Eval(u, i) - \overline{Eval}(u, .)) \cdot (Eval(v, i) - \overline{Eval}(v, .))}{\sqrt{\sum_{i \in Seval[u, v]} (Eval(u, i) - \overline{Eval}(u, .))^2} \cdot \sqrt{\sum_{i \in Seval[u, v]} (Eval(v, i) - \overline{Eval}(v, .))^2}} \quad Eq (3.2).
 \end{aligned}$$

$$\begin{aligned}
 UtilSim(u, v) &= \frac{\sum_{d \in E[u] \cap E[v]} \frac{\sum_{i \in Sutil[u, v]} (Util(u, i, d) - \overline{Util}(u, ., d)) \cdot (Util(v, i, d) - \overline{Util}(v, ., d))}{\sqrt{\sum_{i \in Sutil[u, v]} (Util(u, i, d) - \overline{Util}(u, ., d))^2} \cdot \sqrt{\sum_{i \in Sutil[u, v]} (Util(v, i, d) - \overline{Util}(v, ., d))^2}}}{Card(E[u] \cap E[v])}
 \end{aligned}$$

Eq (3.3).

La similarité en termes de visites $VisSim(u,v)$, formalisée par la formule Eq (3.4), est basée sur la similarité de Jaccard (Yu *et al.* 2009). Elle est relative au nombre de ressources co-visitées par les deux utilisateurs et au nombre total de ressources visitées par les deux utilisateurs.

$$VisSim(u, v) = \frac{Card(IV[u] \cap IV[v])}{Card(IV[u] \cup IV[v])} \quad Eq(3.4).$$

Cette formule n'est pas appliquée lorsque les deux utilisateurs n'ont pas encore visité des ressources car l'union est nulle. Dans ce cas, $VisSim(u, v)$ est égal à zéro.

Dans les réseaux sociaux, le lien entre deux utilisateurs sera d'autant plus fort qu'ils ont de nombreux voisins en commun (Sun and Tang, 2011). Dans notre travail, la force du lien $LinkS(u, v)$ entre deux utilisateurs est définie à l'aide de leur nombre d'amis communs et leur nombre total d'amis. La force du lien entre les utilisateurs u et v est ainsi définie par la formule suivante :

$$LinkS(u, v) = \frac{Card(B[u] \cap B[v]) + 2}{Card(B[u] \cup B[v])} \quad Eq(3.5).$$

Le dernier élément que nous avons intégré dans la formule de la similarité sociale est la similarité liée aux caractéristiques présentes dans les profils des utilisateurs $ProfilSim(u, v)$. Elle prend en compte les similitudes entre les utilisateurs en termes de préférences, de connaissances, d'expériences, etc. Cette similarité est liée au nombre de caractéristiques partagées entre les deux utilisateurs. Comme le montre l'équation Eq (3.6), cette similarité est liée au nombre de caractéristiques communes entre les deux utilisateurs et au nombre total de caractéristiques.

$$ProfilSim(u, v) = \frac{Card(C[u] \cap C[v])}{Card(C)} \quad Eq(3.6).$$

Une partie de la formule de calcul de similarité sociale utilisé dans notre approche est inspirée de la technique des heuristiques dans les systèmes de recommandation multicritères. Cette technique utilise

plusieurs évaluations et une évaluation globale pour chaque ressource. L'approche proposée n'utilise pas l'évaluation globale mais utilise deux types d'évaluation : la qualité et l'utilité. Ce dernier est lié aux domaines de l'utilisateur et chaque utilisateur possède son ensemble de domaines associé.

La technique basée sur les heuristiques agrège la similarité entre différentes évaluations. Pour calculer la similarité dans notre approche, nous utilisons également une similarité agrégée entre les évaluations de qualité et d'utilité et nous ajoutons d'autres éléments (visites des utilisateurs et quelques informations sociales).

3.4 Types de recommandations

Notre approche propose trois types de recommandation pour répondre au mieux aux besoins pédagogiques des utilisateurs des environnements sociaux d'apprentissage.

3.4.1 Recommandation des ressources visitées récemment

Le système peut présenter à un utilisateur donné une liste de ressources qui ont été visitées récemment par des utilisateurs qui lui sont similaires. Ce type de recommandation est intéressant pour que les utilisateurs puissent consulter les ressources dans la même période que leurs amis pour pouvoir collaborer et s'entraider sur leurs différents cours.

Le score de recommandation de la ressource i est calculé suivant les ressources visitées par les amis de l'utilisateur u , le temps écoulé depuis la dernière visite des ressources et les similarités sociales avec les amis de u . Ce score est calculé comme suit :

$$S_{visu}(u, i) = \sum_{v \in B[u]} e^{-\alpha t(v,i)} \cdot Vis(v, i) \cdot SocialSim(u, v) \quad \text{Eq(3.7).}$$

α étant un facteur de décroissance. Plus la visite de la ressource, par l'utilisateur, est récente, plus le score de cette dernière augmente. Nous nous sommes inspirés des travaux de Guy *et al.* (Guy *et al.* 2009 ; Guy *et al.* 2010) pour la prise en compte du temps dans cette formule.

3.4.2 Recommandation des ressources populaires

Un utilisateur peut également se voir recommander une liste de ressources bien notées par ses amis similaires. Le score de recommandation de la ressource i à l'utilisateur u est déterminé par les valeurs des évaluations des ressources par les amis de l'utilisateur u et les similarités sociales avec ses amis. La formule de ce score est la suivante :

$$S_{eval}(u, i) = k \sum_{v \in B_i[u]} Eval(v, i) \cdot SocialSim(u, v) \quad Eq(3.8).$$

Seuls les amis de l'utilisateur u qui ont évalués la ressource i sont utilisés dans le calcul de ce score. Ce sous-ensemble de $B[u]$ est dénommé $B_i[u]$. k représente le facteur de normalisation. Il est donné par la formule :

$$k = 1 / \sum_{v \in B_i[u]} |SocialSim(u, v)| \quad Eq(3.9).$$

La formule S_{eval} utilise l'approche de la somme pondérée (Adomavicius et Tuzhilin 2011).

3.4.3 Recommandation des ressources utiles

Une liste de ressources peut être recommandée à un utilisateur selon leurs utilités par rapport aux domaines d'apprentissage de cet utilisateur. La formule de calcul du score de la recommandation des ressources est définie par l'utilité des ressources par rapport aux domaines d'apprentissage et les similarités sociales. Le score de recommandation de la ressource i à l'utilisateur u est déterminé par la formule :

$$S_{util}(u, i) = k \sum_{v \in B_i[u]} SocialSim(u, v) \cdot \frac{\sum_{d \in E_{vi}[u]} Util(v, i, d)}{Card(E_{vi}[u])} \quad Eq(3.10).$$

Seulement les domaines d'intérêts de l'utilisateur u qui ont été utilisés pour évaluer la ressource i par l'utilisateur v peuvent être in-

tégrés dans le calcul de ce score. Ce sous ensemble de $E[u]$ est dénommé $E_{vi}[u]$.

3.5 Gestion de la *sparsity*

La plupart des algorithmes de filtrage collaboratifs utilisent une mesure de similarité basée sur les évaluations des utilisateurs pour recommander des ressources.

Comme expliqué dans (Verbert *et al.* 2011), à la différence des systèmes de recommandation pour le e-commerce, ceux pour l'éducation peuvent souffrir de la *sparsity* des évaluations, puisque ce type d'application implique moins d'utilisateurs et moins d'interactions. Notre approche peut résoudre ce problème car elle utilise 5 composantes dans la mesure de similarité et seulement 2 d'entre elles sont liées à des évaluations d'utilisateurs. Nous pouvons calculer la similarité et fournir des recommandations pertinentes, même si l'environnement d'apprentissage contient des évaluations *sparse*.

3.6 Mise à jour du profil

Les données du profil d'un apprenant peuvent être partiellement renseignées, voire même périmées ou inappropriées. Comme notre approche se base, en partie, sur ces données, il est nécessaire de mettre à jour le profil de l'utilisateur pour pouvoir faire une recommandation de qualité.

Les données du profil peuvent être renseignées manuellement par l'utilisateur et/ou automatiquement par le système en fonction du comportement de cet utilisateur. En plus de ces deux méthodes, le principe de l'influence sociale (Sun et Tang 2011) nous amène à enrichir le profil de l'utilisateur à l'aide des informations qui caractérisent les profils de ses amis similaires, ceci en utilisant l'une des méthodes suivantes :

- Recommander à l'utilisateur les caractéristiques les plus communes présentes dans les profils de ses amis. Par exemple, le niveau d'étude le plus commun des amis d'un utilisateur. Le système mettra à jour le profil de l'utilisateur si ce dernier valide cette recommandation ;
- Mettre à jour directement le profil de l'utilisateur avec les valeurs des caractéristiques les plus communes de ses amis. Ces

informations peuvent être exactes ou peuvent être incertaines, donc le système leur associe des probabilités d'exactitude (degré de fiabilité) et les stocke dans le profil de l'utilisateur. Par exemple, le système stocke dans le profil le niveau d'étude le plus commun des amis d'un utilisateur avec un degré de fiabilité à 90%.

Il est possible que le système utilise plusieurs méthodes de renseignement du profil de l'utilisateur. Ces méthodes sont utilisées pour avoir un profil le plus complet possible pour pouvoir lui faire des recommandations qui répondent à ses caractéristiques, ses besoins et ses attentes.

3.7. Validation avec des données simulées

3.7.1. Objectif

Comme première étape de validation de notre approche, nous avons essayé de réaliser une analyse hors ligne à l'aide d'un jeu de données extrait d'un système de recommandation éducatif existant. Pour cela, nous avons étudié plusieurs ensembles de données, tels que Mendeley (Jack *et al.* 2010), MACE (Wolpers et Niemann 2010) et APOSDLE (Beham 2010). L'annexe III (Manouselis *et al.* 2012) présente un comparatif entre les différents *dataset* dans les EIAH. Nous avons constaté qu'il est impossible d'utiliser ces ensembles de données pour évaluer notre modèle formel parce que les données qu'ils fournissent ne contiennent pas toutes les informations nécessaires à notre approche, telles que les relations sociales entre les utilisateurs et les évaluations de ressources en termes d'utilité.

Pour répondre à ce problème, nous avons étudié la faisabilité de notre approche en utilisant une analyse hors ligne sur un ensemble de données que nous avons créé. Cette simulation nous a aidé à tester les algorithmes basés sur formules pour évaluer leur efficacité et les affiner. Nous avons développé un prototype en Java qui calcule les similarités sociales entre utilisateurs et calcule et affiche les trois types de recommandations (ressources récemment consultées, ressources populaires et ressources utiles). Ce processus de validation de faisabilité de l'approche est illustré par la Figure 3.3.

La validation proprement dite de notre approche est détaillée dans le chapitre 5.

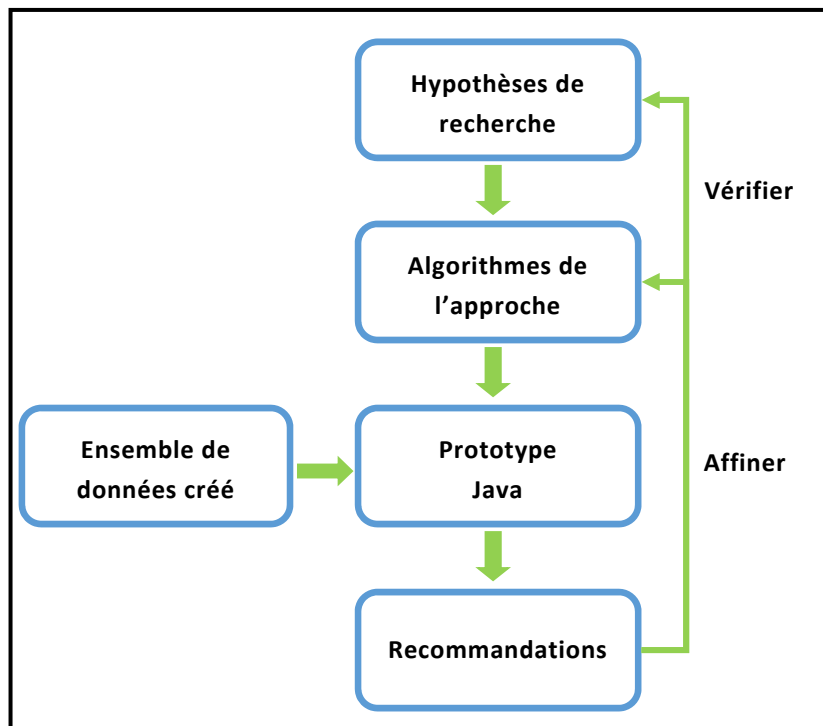


Figure 3.3 Processus de validation de la faisabilité de notre approche

3.7.2. Données de la simulation

L'ensemble de données utilisées (*dataset*) fournit des informations principalement sur 1) les caractéristiques de l'utilisateur telles que l'âge, préférences, etc. 2) les relations sociales entre les différents utilisateurs, et 3) les valeurs d'évaluation que les utilisateurs attribuent aux ressources éducatives.

3.7.2.1. Caractéristiques des apprenants

L'ensemble de données que nous avons préparé est un exemple d'un système de réseau social installé dans une école de formation continue. Le système contient 10 utilisateurs et 9 ressources. Le tableau 3.2 présente les caractéristiques des profils des utilisateurs. Les 5 premiers utilisateurs sont dans la même classe et travaillent sur les mêmes domaines : bases de données (d1), algorithmiques (d2) et génie logiciel (d3). En plus de ces trois domaines d'étude, l'utilisateur U5 a un autre domaine d'intérêt : XML (d4). Les 5 derniers utilisateurs sont dans la classe 2 et étudient les domaines suivants : algorithmes (d2), réseaux (d5) et programmation Web (d6).

Tableau 3.2 Caractéristiques des utilisateurs

Utilisateur	Classe	Préférence	Age	Expérience professionnelle (Années)	Domaines
U1	1	Graphiques	15-20	0-5	d1, d2, d3
U2	1	Vidéos	20-30	11-15	d1, d2, d3
U3	1	Vidéos	20-30	0-5	d1, d2, d3
U4	1	Vidéos	30-40	16-20	d1, d2, d3
U5	1	Graphiques	15-20	0-5	d1, d2, d3, d4
U6	2	Vidéos	20-30	0-5	d2, d5, d6
U7	2	Texte	40-50	26-30	d2, d5, d6
U8	2	Interaction	30-40	10-15	d2, d5, d6
U9	2	Graphiques	20-30	0-5	d2, d5, d6
U10	2	Interaction	20-30	6-10	d2, d5, d6

3.7.2.2. Relations sociales entre apprenants

Les relations sociales entre utilisateurs sont illustrées par la figure 3.4. Pour que ce réseau social soit le plus proche d'un réseau réel, nous avons varié le nombre d'amis par utilisateur. Certains utilisateurs ont peu d'amis tel que l'utilisateur U5 et d'autres ont plusieurs tel que l'utilisateur U6. Les utilisateurs de chaque classe ont plusieurs relations entre eux, en revanche les utilisateurs des deux classes ne sont reliés que par quelques liens (U2-U6 et U4-U6).

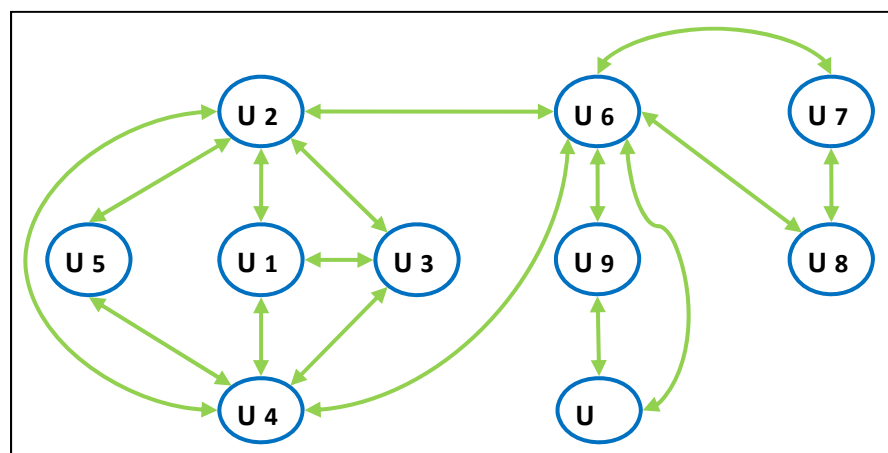


Figure 3.4 Connexions entre utilisateurs

Tableau 3.3 Visites et évaluations des ressources pédagogiques

Users	Domains	Res. 1	Res. 2	Res. 3	Res. 4	Res. 5	Res. 6	Res. 7	Res. 8	Res. 9
U1	d1, d2, d3	(1,59);4; (1,2,1)	(0,0);0; (0,0,0)	(1,1);2; (2,1,2)	(0,0);0; (0,0,0)	(1,2);1; (1,1,1)	(1,19);4; (2,1,1)	(0,0);0; (0,0,0)	(0,0);0; (0,0,0)	(1,6);4; (1,1,1)
U2	d1, d2, d3	(0,0);0; (0,0,0)	(1,8);4; (1,4,2)	(0,0);0; (0,0,0)	(1,34);5; (1,2,1)	(0,0);0; (0,0,0)	(0,0);0; (0,0,0)	(1,4);4; (4,0,0)	(1,2);2; (1,2,0)	(1,12);5; (1,1,0)
U3	d1, d2, d3	(0,0);0; (0,0,0)	(1,2);5; (1,4,2)	(0,0);0; (0,0,0)	(1,5);4; (1,3,0)	(0,0);0; (0,0,0)	(1,12);5; (3,1,1)	(1,17);5; (5,1,1)	(0,0);0; (0,0,0)	(0,0);0; (0,0,0)
U4	d1, d2, d3	(1,1);1; (1,0,0)	(1,10);2; (1,2,2)	(1,45);1; (1,0,1)	(0,0);0; (0,0,0)	(1,9);2; (1,1,1)	(1,7);4; (2,1,1)	(1,22);3; (3,1,1)	(0,0);0; (0,0,0)	(0,0);0; (0,0,0)
U5	d1, d2, d3, d4	(1,30);5; (1,2,2,4)	(1,7);5; (1,5,1,1)	(0,0);0; (0,0,0,0)	(0,0);0; (0,0,0,0)	(1,17);1; (1,1,1,0)	(1,2);5; (2,1,1,5)	(1,2);5; (5,1,1,4)	(0,0);0; (0,0,0,0)	(1,9);4; (1,1,1,0)
U6	d2, d5, d6	(1,15);4; (1,1,1)	(0,0);0; (0,0,0)	(1,34);3; (1,1,1)	(1,10);4; (2,1,4)	(1,22);2; (1,2,1)	(0,0);0; (0,0,0)	(1,2);5; (1,1,2)	(0,0);0; (0,0,0)	(1,7);5; (1,5,3)
U7	d2, d5, d6	(0,0);0; (0,0,0)	(1,8);4; (4,2,1)	(0,0);0; (0,0,0)	(1,14);5; (1,1,5)	(1,16);1; (1,1,1)	(1,12);5; (1,1,5)	(0,0);0; (0,0,0)	(1,1);2; (2,1,1)	(0,0);0; (0,0,0)
U8	d2, d5, d6	(1,23);5; (1,1,1)	(0,0);0; (0,0,0)	(1,25);2; (1,1,2)	(1,2);5; (1,1,5)	(1,35);1; (1,1,1)	(0,0);0; (0,0,0)	(1,7);4; (1,1,3)	(1,2);2; (2,1,0)	(1,10);3; (2,3,2)
U9	d2, d5, d6	(1,7);4; (1,2,1)	(1,4);0; (0,0,0)	(1,6);3; (1,1,1)	(0,0);0; (0,0,0)	(0,0);0; (0,0,0)	(1,15);5; (0,0,4)	(0,0);0; (0,0,0)	(0,0);0; (0,0,0)	(1,1);5; (2,5,3)
U10	d2, d5, d6	(1,6);5; (0,2,0)	(1,4);5; (5,1,1)	(0,0);0; (0,0,0)	(0,0);0; (0,0,0)	(0,0);0; (0,0,0)	(1,9);5; (1,1,5)	(1,12);4; (1,1,2)	(1,12);3; (3,0,1)	(0,0);0; (0,0,0)

3.7.2.3. Visites et évaluation des ressources

Le tableau 3.3 contient des informations sur la visite, l'évaluation et l'utilité des ressources pour les 10 utilisateurs du système. Pour chaque ressource et chaque utilisateur il existe un triplet sous la forme suivante : (visité ou non visité {0/1}, temps écoulé en jours à partir de la dernière visite de la ressource) ; Évaluation de la qualité de la ressource {0..5}; (Évaluation de l'utilité de la ressource par domaine {0..5}). 0 représente une absence d'évaluation et les valeurs de 1 à 5 représentent de « ne pas aimer » à « aimer beaucoup ».

3.7.3. Similarité sociale

Nous avons fait une simulation à l'aide de notre prototype écrit en Java en utilisant l'ensemble des données présenté précédemment. En termes de similarité entre utilisateurs, nous avons remarqué que :

- Les utilisateurs les plus similaires sont U1 et U3 avec une similarité sociale à 0,725. Ceci est justifié par le fait que ces 2 utilisateurs sont amis et ont les mêmes amis, ils ont évalué la qualité des ressources avec presque les mêmes valeurs et ils ont également évalué la même utilité de ressources avec presque les mêmes valeurs ;
- Les utilisateurs les moins similaires sont U6 et U10 avec une similarité sociale à 0,181. Cela est justifié par le fait que le premier utilisateur partage seulement 1 de ses 6 amis avec le deuxième utilisateur, il a également que 2 caractéristiques de son profil en commun avec le deuxième utilisateur et sur les 9 ressources disponibles, ils ont visité seulement 2 ressources communes (ressources 1 et 7).

3.7.4. Ressources recommandées

3.7.4.1. Ressources visitées récemment

Le prototype recommande des ressources à l'utilisateur 1 dans cet ordre 2, 7, 4 et 8, en termes de ressources consultées récemment. Les 2 premières ressources sont visitées par tous les amis de l'utilisateur 1, la troisième par 2 de ses amis et la dernière par un seul. La ressource 2 est recommandée comme la première parce qu'elle a été vue plus récemment par les amis de l'utilisateur 1.

L'ordre de recommandation de ressources pour l'utilisateur 5 est : 8, 3 et 4. La ressource 3 a été recommandée avant la ressource 4

parce que la première a été visitée par l'utilisateur 4 qui a un score de similarité de 0,455 avec l'utilisateur 5 et la seconde a été visitée par l'utilisateur 2 qui a un score de similarité de 0,259 avec l'utilisateur 5. Cela, malgré que la ressource 4 a été visitée par les utilisateurs 2 et 4 plus récemment que la ressource 3.

Le prototype que nous avons développé nous a permis de valider trois hypothèses sur les ressources consultées récemment : 1) Les ressources présentés à l'utilisateur sont ceux qui sont les plus visitées par ses amis. 2) Les ressources les plus récemment visitées sont fournies à l'utilisateur. 3) Les ressources visitées par des amis similaires sont mieux classées.

3.7.4.2. Ressources populaires

Les ressources 6, 2 et 8 sont recommandées dans cet ordre à l'utilisateur 6, en termes de ressources populaires. Le score de classement est calculé suivant les amis de cet utilisateur. Nous remarquons que la ressource 6 a une moyenne d'évaluation meilleure que la ressource 2 et la ressource 2 meilleure que 8.

L'ordre de recommandation de ressources pour l'utilisateur 7 est : 1, 7, 9 et 3. La ressource 1 a été recommandée avant la ressource 7 parce que la première a été évaluée par 4 étoiles par l'utilisateur 6 et par 5 étoiles par l'utilisateur 8 et la seconde a été évaluée par 5 étoiles par l'utilisateur 8 et 4 étoiles par l'utilisateur 6. L'évaluation à 5 étoiles de l'utilisateur 8 est plus importante que celle à 5 étoiles de l'utilisateur 6 parce que l'utilisateur 8 est plus similaire à l'utilisateur 7. Cela, malgré que les ressources 1 et 7 ont la même moyenne d'évaluation (4.5).

Notre prototype a validé deux hypothèses sur le type de recommandation de ressources populaires : 1) Les ressources présentées à l'utilisateur sont ceux qui sont hautement appréciées par ses amis. 2) Les ressources évaluées par des amis semblables sont mieux notées.

3.7.4.3. Ressources utiles

Le prototype recommande des ressources dans cet ordre 8, 3, 1, 9 et 5 à l'utilisateur 3, en termes de ressources utiles. Tous les amis de l'utilisateur 3 ont les mêmes domaines (d1, d2 et d3). La ressource 8 est recommandée comme étant la première parce qu'elle a une meilleure évaluation de l'utilité que les autres. Les ressources 9 et 5 ne sont pas bien classées car toutes leurs évaluations, dans les trois domaines, ne dépassent pas 1 étoile.

L'ordre de recommandation des ressources pour l'utilisateur 2 est 6, 3, 1 et 5. L'utilisateur 2 a 5 amis qui utilisent un total de 6 domaines. Il faut utiliser uniquement les trois domaines de l'utilisateur 2 à savoir d1, d2 et d3 pour calculer la recommandation. La ressource 3 a été recommandée avant la ressource 1 car un l'utilisateur 1, qui est similaire à l'utilisateur 2, a évalué l'utilité de la ressource 3 meilleure que l'utilité de la ressource 1. Cela, bien que la ressource 3 ait une meilleure utilité moyenne que la ressource 1.

Notre prototype a validé deux hypothèses sur le type de recommandation de ressources populaires : 1) Les ressources présentées à l'utilisateur sont ceux qui sont hautement appréciées par les amis de cet utilisateur sur ses domaines. 2) Les ressources évaluées par des amis similaires sont mieux notées.

3.8 Conclusion

Dans ce chapitre, nous avons présenté nos contributions pour améliorer la pertinence des recommandations de ressources pédagogiques. Nous avons développé un modèle formel pour le calcul de la similarité entre utilisateurs et la génération de trois types de recommandation de ressources pédagogique. Nous avons également présenté une illustration et une simulation que nous avons suivie pour tester, affiner et valider notre approche.

4 Implantation : L'environnement Icras

4.1 Architecture fonctionnelle	74
4.1.1 Télé-versement de ressources	75
4.1.2 Accès aux ressources	76
4.1.3 Évaluation de ressources	78
4.1.4 Recommandation de ressources	78
4.1.5 Fonctionnalités sociales	79
4.2 Architecture technique	80
4.3 Moteur de recommandation	81
4.3.1 Modèle de données	81
4.3.2 Algorithmes	82
4.4 Conclusion	91

Après avoir présenté notre approche pour la recommandation de ressources pédagogiques basée sur les liens sociaux, nous présentons dans ce chapitre, une plateforme d'apprentissage, que nous avons développée, qui implémente cette approche et qui permet de l'expérimenter et de la valider. Cette plateforme offre des ressources d'apprentissage, des fonctionnalités sociales et met en œuvre nos modèles formels pour recommander des ressources pédagogiques. Cette plateforme s'appelle Icras (*Icras is a soCial leaRning And Authoring environment*). Une partie de cette plateforme a été développée dans le cadre de projets de fin d'étude de master (Bensmaine et Bouacha 2013. Belhabib et Matahri 2014. Saidi et Bendella 2015).

Dans ce chapitre, nous décrivons en premier lieu la plateforme Icras et son architecture fonctionnelle en détaillant ses principales fonctionnalités. Ensuite, nous décrivons l'architecture technique en présentant les différents modules de cette plateforme ainsi que leurs rôles et interactions. Après cela, nous détaillons le module responsable de la recommandation de ressources pédagogiques avec son modèle de données et ses algorithmes.

4.1 Architecture fonctionnelle

Icras est composée de deux parties : le *Back Office* et le *Front Office* (Figure 4.1). La première partie permet aux enseignants de déposer des ressources sur la plateforme. Ces ressources peuvent être des supports de cours ou des QCM avec leurs corrections. Elles sont stockées et gérées dans une base de ressources. La deuxième partie

d'Icraa est à destination des apprenants plus particulièrement mais elle peut être aussi utilisée par des enseignants. Cette partie permet aux utilisateurs de visiter les supports de cours, de passer un QCM, et d'évaluer ces ressources. Le *Front Office* permet également à un utilisateur de gérer son profil et sa liste d'amis et d'interagir avec ses amis ou avec les membres d'un groupe d'utilisateurs. Ces liens d'amitiés et ces interactions sont gérés par le réseau social.

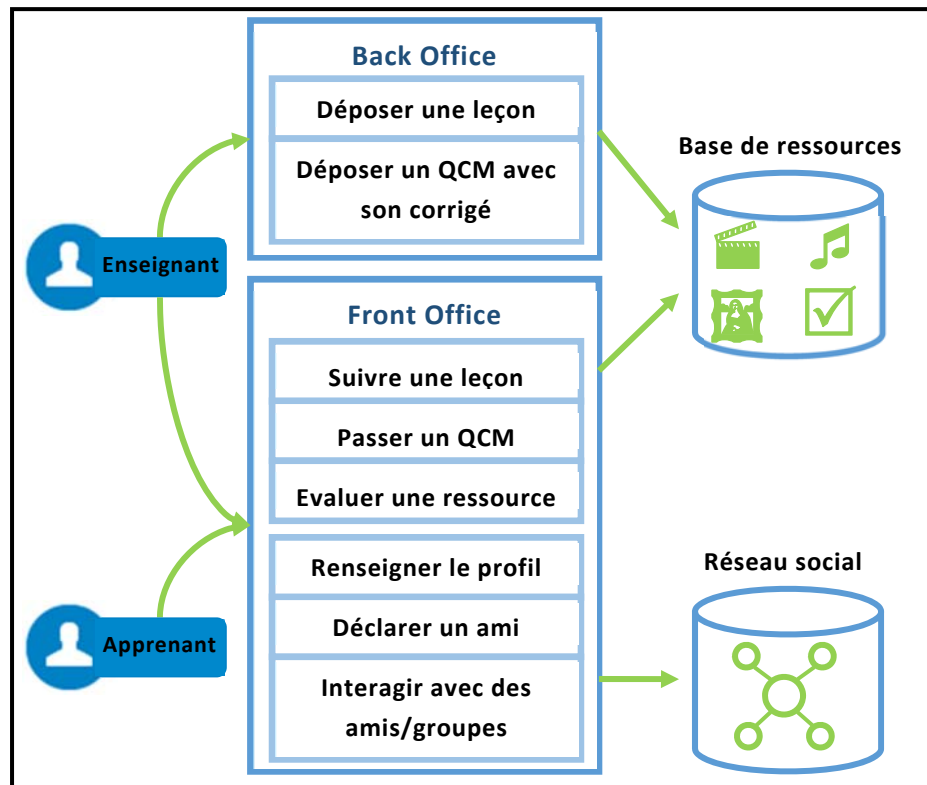


Figure 4.1 Architecture fonctionnelle

Les principales fonctionnalités de l'environnement d'Icraa sont présentées dans les sous-sections suivantes.

4.1.1 Télé-versement de ressources

Les enseignants peuvent utiliser cette fonctionnalité pour publier des ressources liées à leurs cours sur la plateforme. Ils peuvent également les décrire avec des métadonnées (Figure 4.2). Icraa permet de déposer et de publier des ressources créées sur des logiciels tiers et offre aussi la possibilité de les créer directement sur la plateforme en utilisant un outil auteur. Ce dernier permet de créer des ressources sous

forme de diapositives avec un contenu qui regroupe du texte, des images et des vidéos.

The screenshot shows a web interface for adding a resource. The main form is titled "Introduction à l'ingénierie système" and contains the following fields and sections:

- Le nom de la leçon* :** A text input field containing "Introduction à l'ingénierie système".
- Durée (Min) :** A text input field containing "30".
- Description :** A text area containing "Présentation de l'ingénierie des systèmes et de son utilité avec des cas concrets de projets."
- Associer à une image :** A "Parcourir" button and a text input field containing "http://study-press.com/icraa/wp-content/uploads/2014/1".
- Associer à un cours* :** A dropdown menu with "Ingénierie des systèmes" selected.
- Notes :** A section with a "Note..." input field and an "Ajouter" button.
- Glossaire :** A section with "Terme" and "Description" input fields and an "Ajouter" button.

On the right side, there is a panel titled "Les slides" with a "Fichier" upload area and two buttons: "Créer un nouveau slide" and "Enregistrer".

Figure 4.2 Télé-versement et description d'une ressource

4.1.2 Accès aux ressources

Les deux types d'utilisateurs (apprenants et enseignants) peuvent accéder à toutes les ressources disponibles sur la plateforme. Un menu leur permet de parcourir les promotions et les cours associés à chaque promotion. Lorsqu'un utilisateur clique sur un lien d'un cours, la liste de ressources associées s'affiche. Le côté gauche de la figure 4.3, illustre une page d'accès à une ressource. Elle contient un lien pour télécharger la ressource ainsi que certaines métadonnées décrivant la ressource, telles que son nom, son auteur, son cours, sa durée, ses notes, etc.

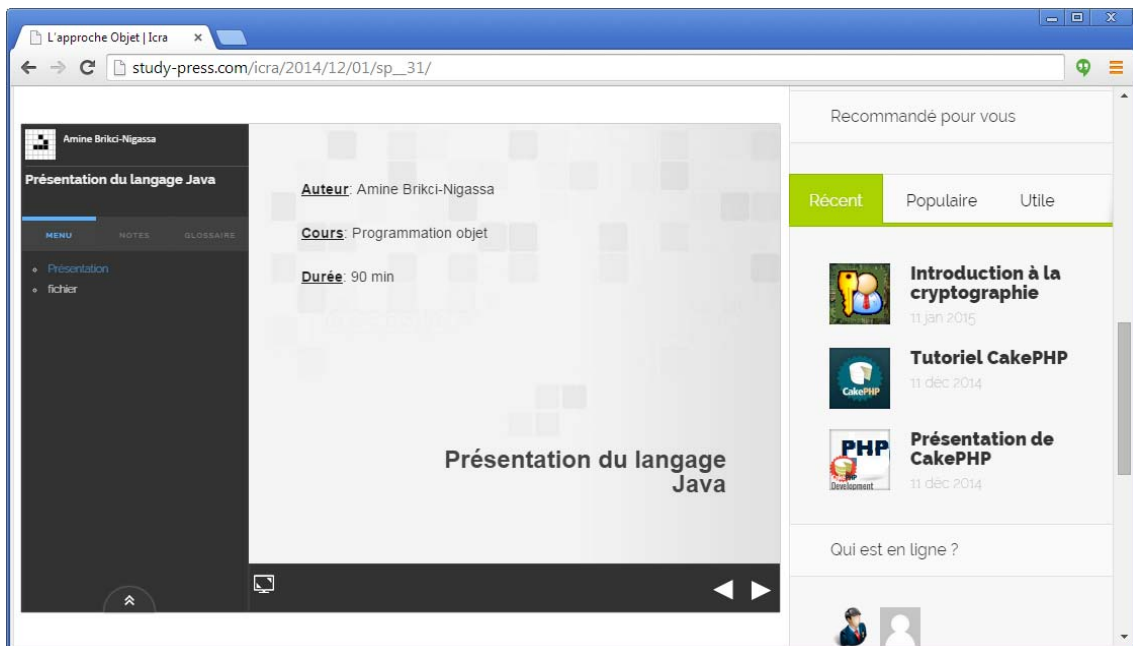


Figure 4.3 Accès à une ressource (gauche), Widget de recommandation (droite)

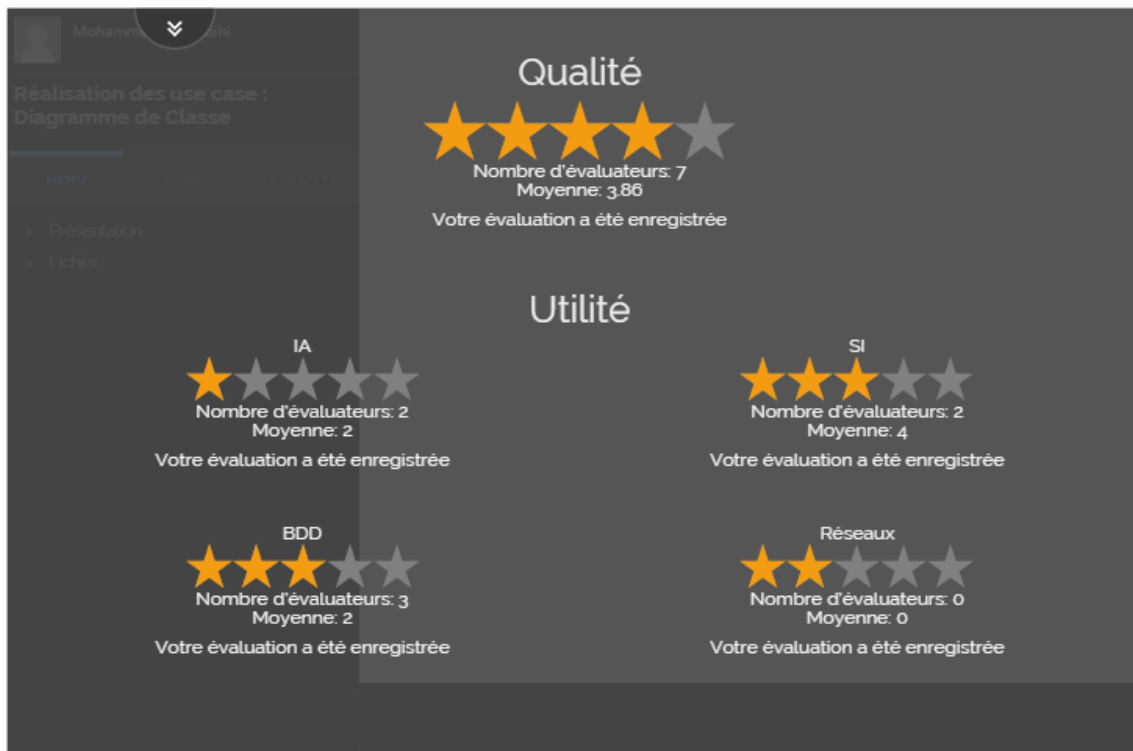


Figure 4.4 Évaluation de la qualité et de l'utilité d'une ressource

4.1.3 Évaluation de ressources

Comme illustré dans la Figure 4.4, cette fonctionnalité permet aux utilisateurs d'évaluer une ressource en termes de qualité et d'utilité selon les domaines de l'utilisateur. Les deux types de ressources (support de cours ou QCM) peuvent être évalués.

4.1.4 Recommandation de ressources

Le côté droit de la figure 4.3 montre le *widget* de recommandation. Il fournit aux utilisateurs les trois types de recommandations offerts par l'approche, à savoir les ressources récemment visitées, les ressources populaires et les ressources utiles. Ce widget affiche une liste des trois ressources les mieux recommandées pour chaque type de recommandation.



Figure 4.5 Fil d'actualité d'un groupe d'utilisateurs

4.1.5 Fonctionnalités sociales

L'environnement d'apprentissage que nous avons développé offre de multiples fonctionnalités sociales, telles que nous pouvons trouver sur les réseaux sociaux en ligne comme Facebook ou Google+⁶. Icras permet de poster des messages, de partager des documents, de rejoindre un groupe, d'ajouter un ami, etc. La figure 4.5 montre le fil d'actualité d'un groupe sociale attaché à une promotion de 45 étudiants et enseignants.

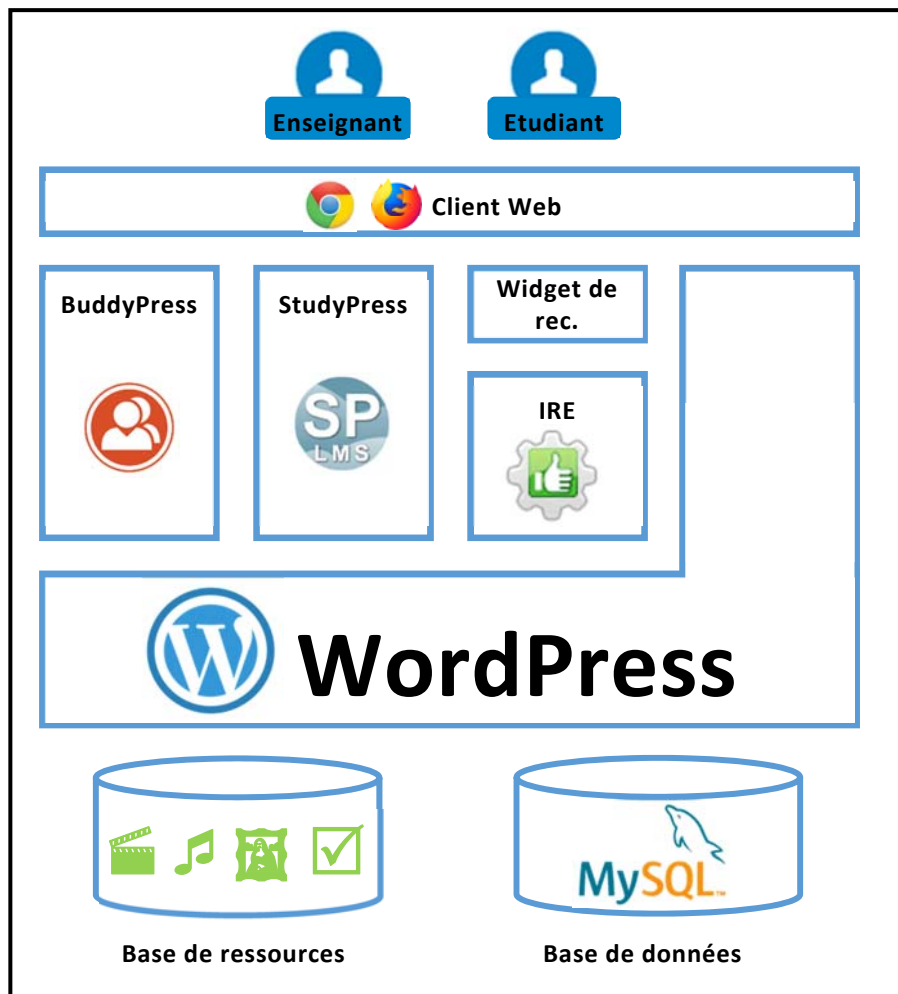


Figure 4.6 Architecture technique

⁶ <https://plus.google.com>

4.2 Architecture technique

Ictraa est un LMS social (*Learning Management System*) basé sur Wordpress⁷, BuddyPress⁸, StudyPress⁹, le *widget* de recommandation et l'IRE (*Ictraa Recommender Engine*). Wordpress est un CMS (*Content Management System*) qui fournit à Ictraa les fonctionnalités de base qui permettent la gestion du contenu Web. BuddyPress est un *plugin* Wordpress qui permet aux utilisateurs de se connecter entre eux et d'interagir. StudyPress est un *plugin* Wordpress open source que nous avons développé. Il transforme Wordpress en un LMS en intégrant des fonctionnalités d'apprentissage telles que la gestion des ressources pédagogiques et la gestion des enseignants et des apprenants. StudyPress permet aussi de sauvegarder dans la base de données les traces utilisateurs tel que le nom des ressources visitées, la date/heure de la dernière visite, etc. IRE recommande des ressources aux utilisateurs d'Ictraa sur la base de l'approche proposée. Ces recommandations sont gérées et affichées par le *widget* de recommandation.

Les ressources déposées sur la plateforme sont stockées dans la base de ressources et leurs méta-données sont stockées dans la base de données MySQL. Cette dernière stocke aussi d'autres informations telles que les profils des utilisateurs, les relations sociales, les traces utilisateurs, les résultats des QCM, etc.

Comme expliqué dans la section 4.1.4, le *widget* de recommandation affiche une liste des trois principales ressources recommandées pour chaque type de recommandation. Au chargement d'une page Web ce *widget* récupère l'identifiant de l'utilisateur connecté et appelle le moteur de recommandation en passant cet identifiant comme paramètre. Le moteur de recommandation calcule les meilleures ressources à recommander à cet utilisateur ensuite il les envoie au *widget* pour les afficher. L'IRE utilise la base MySQL et la base de ressources pour effectuer les recommandations.

⁷ <https://wordpress.org>

⁸ <https://wordpress.org/plugins/buddypress/>

⁹ <https://wordpress.org/plugins/studypress/>

4.3 Moteur de recommandation

4.3.1 Modèle de données

La figure 4.7 montre le modèle entité-relation simplifié qui représente les données manipulées par le moteur de recommandation. Ce modèle reprend les entités utilisées par le moteur de recommandation et il ne détaille pas d'autres parties tels que la gestion de contenu, l'outil auteur ou la gestion de quiz. Ce modèle de données est représenté par quatre entités et six relations. Il est matérialisé par dix tables dans la base de données MySQL.

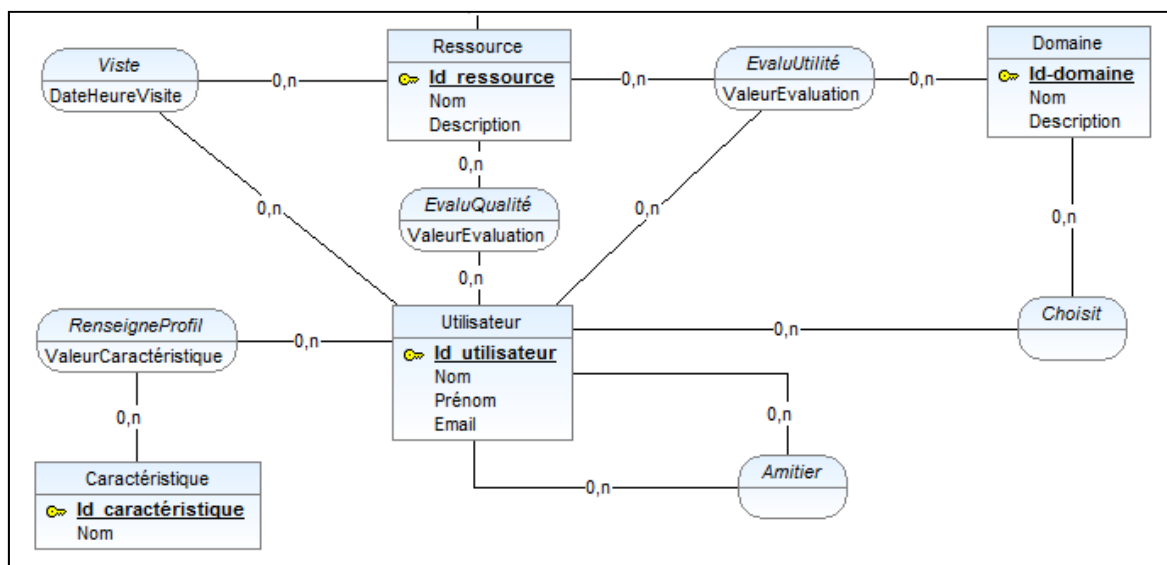


Figure 4.7 Modèle de données

Un utilisateur peut visiter plusieurs ressources. À chaque visite d'un utilisateur, Icras stocke la date et l'heure de visite. L'utilisateur peut par la suite évaluer la qualité de ces ressources et leurs utilités par rapport à des domaines d'enseignements prédéfinis par l'administrateur. Un utilisateur peut renseigner les caractéristiques de son profil. Les caractéristiques disponibles sur Icras sont : Niveau d'étude (Licence ou Master), Option d'étude (Génie logiciel, Réseaux et systèmes distribués, Système d'information et connaissances, Modèle intelligent et décision), Langage de programmation préféré, Nombre d'années d'expérience en programmation (1-2, 3-4, 5-6 et 7 ou plus), et style d'apprentissage (Texte, Graphique et Vidéo). Un

utilisateur peut se lier d'amitié avec un ou plusieurs d'autres utilisateurs.

L'interface d'administration d'Icraa donne la possibilité à l'administrateur de paramétrer les domaines et les caractéristiques des utilisateurs.

4.3.2 Algorithmes

Les algorithmes implémentés au sein du moteur de recommandation d'Icraa se basent sur les formules du calcul de la similarité sociale expliquées dans la section 3.3 et des formules de recommandation présentées dans la section 3.4.

Le moteur de recommandation (IRE) a été implanté comme un objet qui peut être invoqué par le *widget* de recommandation en lui passant l'identifiant de l'utilisateur courant et le type de recommandation choisi par cet utilisateur. Le *widget* de recommandation récupère la liste des ressources recommandées pour l'utilisateur et le type sélectionnés via la fonction *recommend* (user, recType), ensuite il affiche les 3 meilleures ressources. Ci-dessous l'algorithme de cette fonction :

Algorithme 4.1 Recommandation de ressources à un utilisateur

```

function recommend(user, recType) {
  initSets(user);
  notVisitedResSet = getNotVisitedRes(user);
  switch (recType) {
    case 1:
      for(i=0; i < count(notVisitedResSet); i++)
        recommendationList[notVisitedResSet[i]] =
          calcScoreVisited(user, notVisitedResSet[i]);
      break;
    case 2:
      for(i=0; i < count(notVisitedResSet); i++)
        recommendationList[notVisitedResSet[i]] =
          calcScoreEval(user, notVisitedResSet[i]);
      break;
    case 3:
      for(i=0; i < count(notVisitedResSet); i++)
        recommendationList[notVisitedResSet[i]] =
          calcScoreUtil(user, notVisitedResSet[i]);
      break;
  }
}

```

```

}
sortedList = descendingSort(recommendationList);
return sortedList;
}

```

Le moteur de recommandation exécute des requêtes SQL sur la base de données MySQL pour effectuer la recommandation. Pour éviter l'exécution répétée de certaines requêtes SQL, la fonction *recommend* initialise des listes avec le résultat de ces requêtes. Ceci permet à différentes fonctions du moteur de recommandation d'utiliser ces listes au lieu réinterroger la base de données plusieurs fois.

Après l'initialisation, la fonction renseigne une liste contenant les ressources non visitées par l'utilisateur courant. Pour chacune des ressources de cette liste, la fonction calcule son score de recommandation en appelant la fonction *calcScoreVisited*, *calcScoreEval* ou *calcScoreUtil* suivant le type de recommandation passé en paramètre (1 pour les recommandations des ressources visitées récemment, 2 pour les recommandations des ressources populaires et 3 pour les recommandations des ressources utiles). Les scores sont stockés dans un tableau clé/valeur où la clé représente l'identifiant de la ressource et la valeur le score de recommandation de cette ressource.

Une fois que les scores sont calculés, la fonction *recommend* effectue un tri par ordre décroissant des scores du tableau clé/valeur et retourne ce tableau. Il s'agit de la liste des ressources non visitées par l'utilisateur, ordonnées par leurs scores de recommandation.

Algorithme 4.2 Calcul du score des ressources visualisées récemment

```

function calcScoreVisited(user, ressource){
    scoreVisited = 0;
    friendsSet = getFriends(user);
    for(i=0;i<count(friendsSet);i++){
        nbrDays = calcNbrDays(friendsSet[i], ressource);
        visitedRessource = isVisitedRessource(friendsSet[i],
                                                ressource);
        socialSimilarity = calcSocialSimilarity(user, friendsSet[i]);
        scoreVisited = scoreVisited + (exp(0.025*nbrDays) *
                                        visitedRessource * socialSimilarity);
    }
    return scoreVisited;
}

```

Le calcul du score des ressources visualisées récemment, présenté dans l'algorithme 4.2, est basé sur la formule 3.7 expliquée dans la section 3.4.1. La fonction *calcScoreVisited* initialise le score à zéro. Ensuite elle renseigne une liste contenant les amis de l'utilisateur courant. Pour chaque utilisateur (ami) de cette liste, la fonction calcule un score de recommandation et le cumule avec le score de l'itération précédente.

La formule du calcul de ce score est composée du nombre de jours écoulés depuis la dernière visite de l'ami à la ressource passée en paramètre, du fait que l'utilisateur ami a visité ou pas la ressource (1 pour oui et 0 pour non), et de la similarité sociale entre l'utilisateur courant et son ami sélectionné. Après le calcul du score final, ce dernier est retourné par la fonction.

Algorithme 4.3 Calcul du score des ressources populaires

```

function calcScoreEval(user, ressource){
    scoreEval = 0;
    sumEval = 0;
    sumSocialSim = 0;
    friendsSet = getFriends(user);
    for(i=0;i<count(friendsSet);i++){
        evalRessource = getEvalRessource(friendsSet[i], ressource);
        if (evalRessource != null){
            socialSimilarity = calcSocialSimilarity(user,
                friendsSet[i]);
            sumEval = sumEval + evalRessource * socialSimilarity;
            sumSocialSim = sumSocialSim + abs(socialSimilarity);
        }
    }
    if (sumSocialSim != 0){
        scoreEval = sumEval / sumSocialSim;
    }
    return scoreEval;
}

```

Le calcul du score des ressources populaires, présenté dans l'algorithme 4.3, est basé sur la formule 3.8 présentée dans la section 3.4.2. La fonction *calcScoreEval* initialise le score, la somme des évaluations de qualité et la somme des similarités sociales à zéro. En-

suite, elle renseigne une liste contenant les amis de l'utilisateur courant. Pour chaque utilisateur ami de cette liste, la fonction récupère l'évaluation de la qualité effectuée par cet utilisateur sur la ressource passée en paramètre. Si l'évaluation existe, alors le moteur de recommandation calcule un score de recommandation et le cumule avec le score de l'itération précédente et calcule aussi la similarité sociale et la cumule avec la similarité de l'itération précédente.

Le calcul du score de recommandation de chaque itération utilise l'évaluation de la qualité de la ressource de l'ami sélectionné et de la similarité sociale entre l'utilisateur courant et son ami sélectionné. À la fin, la somme des similarités sociales est testée. Si elle est différente de zéro, le score final est calculé puis retournée par la fonction.

Algorithme 4.4 Calcul du score des ressources utiles

```

function calcScoreUtil(user, ressource){
    scoreUtil = 0;
    sumUtil = 0;
    sumSocialSim = 0;
    friendsSet = getFriends(user);
    domainsSet = getDomains(user);
    for(i=0;i<count(friendsSet);i++){
        socialSimilarity = calcSocialSimilarity(user, friendsSet[i]);
        sumUtilDomain = 0;
        cardEvi = 0;
        for(j=0;j<count(domainsSet);j++){
            utilRessource = getUtilRessource(friendsSet[i],
                ressource, domainsSet[j]);
            if (utilRessource != null){
                sumUtilDomain = sumUtilDomain + utilRessource;
                cardEvi++;
            }
        }
        if (cardEvi != 0){
            sumUtil = sumUtil + (socialSimilarity * (sumUtilDomain /
                cardEvi));
            sumSocialSim = sumSocialSim + abs(socialSimilarity);
        }
    }
    if (sumSocialSim != 0){

```

```

    scoreUtil = sumUtil / sumSocialSim;
  }
  return scoreUtil;
}

```

Le calcul du score des ressources utiles, présenté dans l'algorithme 4.4, est basé sur la formule 3.10 expliquée dans la section 3.4.3. La fonction *calcScoreUtil* initialise le score, la somme des évaluations d'utilité et la somme des similarités sociales à zéro. Ensuite elle renseigne deux listes. Une contenant les domaines d'intérêt de l'utilisateur courant et une autre contenant ses amis.

Pour chaque utilisateur ami de cette liste, la fonction récupère la similarité sociale entre l'utilisateur courant et son ami sélectionné et initialise la somme des évaluations d'utilité par domaine à zéro et le nombre d'évaluation d'utilité de l'utilisateur ami par domaine à zéro aussi. Pour chaque domaine appartenant à la liste précédemment renseignée, la fonction récupère l'évaluation de l'utilité effectuée par l'utilisateur ami sélectionné, sur la ressource passé en paramètre et sur ce domaine. Si l'évaluation existe alors le moteur de recommandation cumule cette évaluation avec celles des itérations précédentes et incrémente le nombre d'évaluation d'utilité de l'utilisateur ami par domaine. Après la dernière itération sur les domaines, la fonction teste ce nombre. Dans le cas où il est différent de zéro, la fonction calcule un score de recommandation et le cumule avec le score de l'itération précédente et calcule aussi la similarité sociale et la cumule avec la similarité de l'itération précédente.

Le calcul du score de recommandation de chaque itération utilise la similarité sociale entre l'utilisateur courant et son ami sélectionné, le cumul des évaluations d'utilité de la ressource par l'ami sélectionné et le nombre d'évaluation d'utilité de l'utilisateur ami par domaine. À la fin de la fonction, la somme des similarités sociales est testée. Si elle est différente de zéro le score final est calculé ensuite retournée par la fonction.

Algorithme 4.5 Similarité en termes de force de lien

```

function calcLinkS(userU, userV){
  uFriendsSet = getFriends(userU);
  vFriendsSet = getFriends(userV);
  commonFriendsSet = intersect(uFriendsSet, vFriendsSet);
  totalFriendsSet = union(uFriendsSet, vFriendsSet);
}

```



```

linkS = (count(commonFriendsSet) + 2) / count(totalFriendsSet);
return linkS;
}

```

Le calcul de la similarité en termes de force de lien entre deux utilisateurs, présenté dans l'algorithme 4.5, est basé sur la formule 3.5 expliquée dans la section 3.3. La fonction *calcLinkS* récupère les deux listes d'amis associées aux deux utilisateurs passés en paramètre. Ensuite elle renseigne une liste contenant les amis en commun entre les deux utilisateurs et une autre contenant tous les amis des deux utilisateurs. À la fin, la similarité est calculée en fonction du nombre d'éléments dans ces deux listes.

Algorithme 4.6 Similarité en termes de profil

```

function calcProfilSim(userU, userV){
  uProfilSet = getProfil(userU);
  vProfilSet = getProfil(userV);
  commonCharacteristicsSet = intersect(uProfilSet, vProfilSet);
  profilSim = count(commonCharacteristicsSet) /
              count(getProfil(userU));
  return profilSim;
}

```

Le calcul de la similarité en termes de profil entre deux utilisateurs, présenté dans l'algorithme 4.6, est basé sur la formule 3.6 expliquée dans la section 3.3. La fonction *calcProfilSim* récupère les deux listes contenant les informations sur les profils des deux utilisateurs passés en paramètre. Ces listes sont composées de couples « caractéristique/valeur ». Ensuite, la fonction renseigne une liste contenant les couples « caractéristique/valeur » similaires entre les deux utilisateurs. À la fin, la similarité est calculée en fonction du nombre d'éléments de cette liste et du nombre total de caractéristiques.

Algorithme 4.7 Similarité en termes de visites

```

function calcVisSim(userU, userV){
  uVisitedResSet = getVisitedRessources(userU);
  vVisitedResSet = getVisitedRessources(userV);
  commonVisitedResSet = intersect(uVisitedResSet, vVisitedResSet);
}

```

```

totalVisitedResSet = union(uVisitedResSet, vVisitedResSet);
if (count(totalVisitedResSet) != 0)
    visSim = count(commonVisitedResSet) /
              count(totalVisitedResSet);
else
    visSim = 0;
return visSim;
}

```

Le calcul de la similarité en termes de visites, présenté dans l'algorithme 4.7, est basé sur la formule 3.4 expliquée dans la section 3.3. La fonction *calcVisSim* récupère deux listes. Chacune contient les ressources visitées par son utilisateur correspondant. Ensuite, la fonction renseigne une liste contenant les ressources co-visitées par les deux utilisateurs et une autre contenant toutes les ressources visitées par ces utilisateurs. Si les deux utilisateurs n'ont visité aucune ressource, alors la similarité est nulle. Sinon elle est calculée en fonction du nombre d'éléments présents dans la liste d'intersection et la liste d'union.

Algorithme 4.8 Similarité en termes d'évaluation de qualité

```

function calcEvalSim(userU, userV){
    numerator = 0;
    sumEvalU = 0;
    sumEvalV = 0;
    evalSim = 0;
    sEval = getQualCoevaluatedRes(userU, userV);
    if(count(sEval) > 0){
        for(i=0;i<count(sEval);i++){
            evalU = getEvalRessource(userU,sEval[i]) -
                    getEvalAvgRessource(userU, sEval);
            evalV = getEvalRessource(userV,sEval[i]) -
                    getEvalAvgRessource(userV, sEval);
            numerator = numerator + (evalU * evalV);
            sumEvalU = sumEvalU + power(evalU,2);
            sumEvalV = sumEvalV + power(evalV,2);
        }
        if ((sumEvalU != 0) and (sumEvalV != 0))
            evalSim = numerator / (sqrt(sumEvalU) * sqrt(sumEvalV));
    }
}

```

```

return evalSim;
}

```

Le calcul du score des ressources populaires, présenté dans l'algorithme 4.8, est basé sur la formule 3.2 expliquée dans la section 3.3. La fonction *calcEvalSim* initialise la similarité et trois autres variables à zéro. Ensuite elle renseigne une liste contenant les ressources co-évaluées en termes de qualité par les deux utilisateurs passés en paramètre. Si cette liste n'est pas vide alors la fonction calcule les trois composantes, nécessaires au calcul de la similarité, présentes dans la formule 3.2. Il s'agit d'un numérateur, de la première partie du dénominateur liée à l'utilisateur U et la deuxième liée à l'utilisateur V. Ces trois composantes sont calculées sur plusieurs itérations en utilisant les évaluations effectuées par les utilisateurs U et V sur la qualité de la ressource sélectionnée dans l'itération et les évaluations moyennes des utilisateurs U et V de leurs évaluations de qualités respectives. À la fin, la fonction teste les deux parties du dénominateur de la formule de similarité. S'ils sont différents de zéro alors la similarité est calculée en utilisant les 3 composantes.

Algorithme 4.9 Similarité en termes d'évaluation d'utilité

```

function calcUtilSim(userU, userV){
  sUtil = getUtilCoevaluatedRes(userU, userV);
  domainsSetU = getDomains(userU);
  domainsSetV = getDomains(userV);
  commonDomainsSet = intersect(domainsSetU, domainsSetV);
  utilSim = 0;
  sumUtil = 0;
  if((count(commonDomainsSet) > 0) and (count(sUtil) > 0)) {
    countValidDom = 0;
    for(j=0;j<count(commonDomainsSet);j++){//domains
      numerator = 0;
      sumUtilU = 0;
      sumUtilV = 0;
      validDom = 0;
      for(i=0;i<count(sUtil);i++){
        utilResU = getUtilRessource(userU, sUtil(i),
                                   commonDomainsSet(j));
        utilResV = getUtilRessource(userV, sUtil(i),
                                   commonDomainsSet(j));
      }
    }
  }
}

```

```

    if ((utilResU != 0) and (utilResV != 0)){
        validDom = 1;
        utilU = utilResU - getUtilAvgRessource(userU,
                                                commonDomainsSet(j), sUtil);
        utilV = utilResV - getUtilAvgRessource(userV,
                                                commonDomainsSet(j), sUtil);
        numerator = numerator + (utilU * utilV);
        sumUtilU = sumUtilU + power(utilU, 2);
        sumUtilV = sumUtilV + power(utilV, 2);
    }
}
if (validDom == 1)
    countValidDom++;
if ((sumUtilU != 0) and (sumUtilV != 0))
    sumUtil = sumUtil + (numerator/ (sqrt(sumUtilU) *
                                     sqrt(sumUtilV)));
}
utilSim = sumUtil / countValidDom;
}
return utilSim;
}

```

Le calcul du score des ressources utiles, présenté dans l'algorithme 4.9, est basé sur la formule 3.4 expliquée dans la section 3.3. La fonction *calcUtilSim* renseigne une liste contenant les ressources co-évaluées en termes de qualité par les deux utilisateurs passés en paramètre. La fonction récupère aussi une liste des domaines en commun entre les deux utilisateurs. Si ces deux listes ne sont pas vides alors la fonction calcule les différentes composantes nécessaires au calcul de la similarité. Il s'agit d'un numérateur, de la première partie du dénominateur lié à l'utilisateur U, la deuxième lié à l'utilisateur V et du nombre de domaines co-évalués. Ces composantes sont calculées sur plusieurs itérations de deux boucles imbriquées en utilisant les évaluations effectuées par les utilisateurs U et V sur l'utilité de la ressource et au domaine sélectionnés dans l'itération et les évaluations moyennes des utilisateurs U et V de leurs évaluations d'utilité respectives.

À la fin de chaque itération sur les domaines, la fonction teste les deux parties du dénominateur de la formule de similarité. S'ils sont différents de zéro alors la similarité par rapport au domaine sélec-

tionnée est cumulée avec celles des itérations précédentes. A la fin de la fonction, la valeur finale de la similarité est calculée et retournée.

4.4 Conclusion

Dans ce chapitre, nous avons présenté la plateforme Icras, utilisée pour l'expérimentation et la validation de notre approche. Les architectures fonctionnelle et technique de cette plateforme ont été présentées. Nous avons aussi identifié les différents composants de cette plateforme et leurs interactions. Icras se base sur une architecture modulaire et elle bénéficie de modules applicatifs largement utilisés dans les plateformes d'apprentissage tel que Wordpress et BuddyPress.

Dans ce chapitre, nous nous sommes concentrés sur la description du moteur de recommandation. Ce dernier utilise des informations sur les utilisateurs et les ressources publiées sur Icras pour proposer des recommandations personnalisées à chaque apprenant. Nous avons présenté et décrit les principales fonctions parmi les 23 fonctions qui composent ce moteur. Le reste des fonctions effectuent principalement des opérations de sélection sur la base de données MySQL pour retourner des valeurs (évaluation de qualité, évaluation moyenne, etc.) ou des listes d'éléments (ressources, utilisateurs, domaines, etc.).

5 Expérimentations

5.1	Expérimentation 1 : Évaluation avec des utilisateurs réels	93
5.1.1	Expérimentation	93
5.1.2	Feedback des utilisateurs	95
5.1.3	Résultats du Feedback des utilisateurs	96
5.1.4	Questionnaire	97
5.1.5	Résultats du questionnaire	99
5.1.6	Discussion	101
5.2	Expérimentation 2 : Évaluation avec un Dataset (Analyse Offline)	102
5.2.1	Objectif et hypothèses	102
5.2.2	Expérimentation	103
5.2.3	Ensemble de données (<i>Dataset</i>)	104
5.2.4	Résultats	104
5.3	Conclusion	105

Après avoir présenté l'environnement social d'apprentissage Icras, qui intègre nos modèles de recommandation, nous allons présenter dans ce chapitre les résultats d'une expérimentation menée dans un contexte éducatif réel. L'objectif de cette expérimentation est de mesurer la pertinence, la qualité et l'utilité des ressources recommandées par le système de recommandation que nous proposons. Selon Herlocker *et al.* (2004), 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 synthétisé.

Comme expliqué dans le chapitre 3, pour une première étape d'évaluation, nous avons fait une analyse hors ligne à l'aide d'un jeu de données synthétisé que nous avons créé. En plus de ce travail, 1) nous avons développé la plateforme Icras, qui met en œuvre notre approche et nous l'avons expérimenté avec des utilisateurs dans un contexte réel d'apprentissage, et 2) nous avons comparé la précision de l'approche avec quatre algorithmes de recommandation de références en utilisant une analyse hors ligne sur les données recueillies à partir d'Icras.

5.1 Expérimentation 1 : Évaluation avec des utilisateurs réels

5.1.1 Expérimentation

Comme le montre la figure 5.1, la première expérimentation (Tadlaoui *et al.* 2017) a été menée en quatre phases : 1) utilisation d'Icraa dans un contexte réel d'apprentissage, 2) récolte des feedbacks des utilisateurs sur les ressources recommandées par le système, 3) conduite d'un questionnaire auprès des utilisateurs, en recherchant la pertinence perçue des utilisateurs sur les ressources recommandées, et 4) analyse des résultats des deux dernières phases.

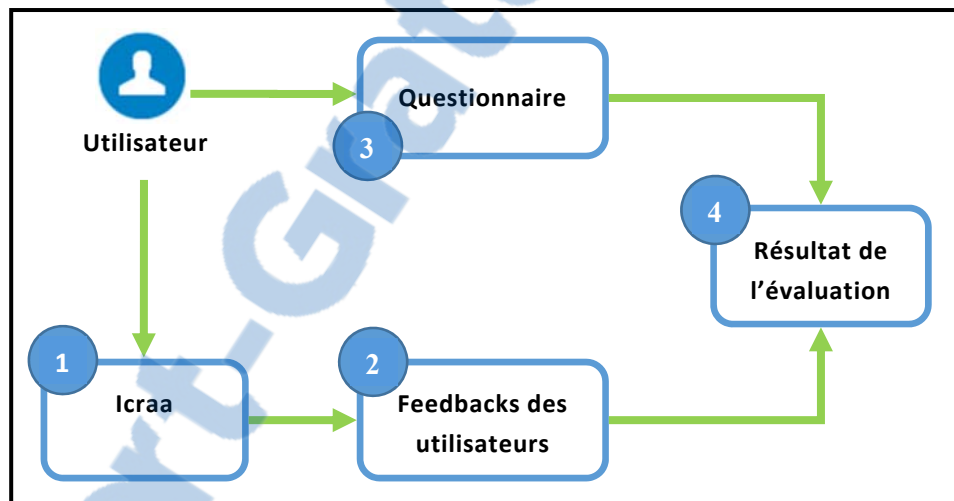


Figure 5.1 Processus d'évaluation avec des utilisateurs réels

Nous avons testé notre approche dans une situation d'apprentissage réel en utilisant Icraa. L'expérience s'est déroulée avec 238 utilisateurs (192 étudiants et 46 enseignants) du département d'informatique de l'Université de Tlemcen en Algérie. L'expérience a duré 6 mois, de novembre 2014 à mai 2015.

Dans le département d'informatique, les enseignants utilisaient le courrier électronique, moodle¹⁰, edmodo¹¹, Google Sites¹², les

¹⁰ <https://moodle.org>

blogs WordPress¹³ et Facebook pour partager leurs ressources pédagogiques. Une fois Icras installé, 14 enseignants l'ont utilisé, alors que les 32 autres enseignants ont continué à utiliser leurs outils habituels. 8 promotions d'étudiants (2 Licences et 6 Masters) ont été intégrées. 146 étudiants se sont connectés plus d'une fois à Icras pendant les 6 mois de l'expérience. L'inactivité de certains étudiants sur la plateforme est due à trois principales raisons :

- Pour une promotion, aucun de leurs enseignants n'a déposé de ressources ;
- Certains étudiants ne sont pas très assidus et préfèrent récupérer les supports de cours en passant par des camarades ;
- Certains étudiants ont fait des transferts d'options ou de filières. Les promotions vers lesquelles ils ont été transférés n'étaient pas intégrées dans Icras.

Les enseignants ont été invités à publier leurs supports de cours et à interagir avec les étudiants sur la plateforme Icras. Les 14 enseignants actifs ont publié 175 ressources d'apprentissage sur 22 modules d'enseignement avec une charge d'étude globale de 224 heures. Toutes ces ressources étaient disponibles pour tous les utilisateurs d'Icras.

Les étudiants ont été invités à consulter des ressources, à évaluer la qualité et l'utilité de ces ressources, à consulter les ressources recommandées par le système s'ils le souhaitent et à interagir avec d'autres utilisateurs sur la plateforme sociale. Nous avons clairement expliqué aux étudiants que leur participation à l'expérience et au questionnaire n'aurait aucun impact sur leurs notes. En outre, ils savaient que les évaluations de ressources qu'ils donnaient n'étaient pas visibles par leurs enseignants.

Le profil utilisateur contenait sept attributs, à savoir son nom, sa promotion, son niveau d'étude, son expérience en programmation, son langage de programmation préféré, son style d'apprentissage préféré et ses domaines préférés.

¹¹ <https://www.edmodo.com>

¹² <https://gsuite.google.com/products/sites/>

¹³ <https://fr.wordpress.com>

Nous avons préparé et envoyé une vidéo aux étudiants qui leur expliquait comment utiliser la plateforme Icras et l'intérêt de faire des recommandations, de renseigner leurs profils, d'évaluer les ressources et d'ajouter leurs connaissances comme des amis. Une autre vidéo a été faite pour les enseignants. De plus, un guide d'utilisation a été envoyé aux deux types d'utilisateurs d'Icras.

Certaines questions du questionnaire exigeaient que l'utilisateur ait vraiment utilisé l'environnement d'apprentissage et ne l'ait pas seulement découvert. Par conséquent, nous avons décidé de conserver uniquement les utilisateurs très actifs dans ce processus d'évaluation. Ainsi, les 40 étudiants les plus actifs ont été invités à répondre au questionnaire. Ces étudiants actifs ont été choisis en fonction de leur nombre de connexions à la plateforme, leur nombre d'amis, leur nombre d'évaluation de qualité et d'utilité des ressources et du nombre de clics sur les recommandations faites par le système.

5.1.2 Feedback des utilisateurs

L'utilisation d'Icras sur une période de six mois nous a permis de recueillir des traces d'utilisation et de les utiliser pour valider la précision des algorithmes proposés. Cette précision est mesurée principalement par les évaluations des utilisateurs sur les recommandations proposées par Icras.

Nous avons utilisé les traces des utilisateurs enregistrées à partir d'Icras pour analyser les 3 hypothèses suivantes :

- Le système de recommandation fournit des ressources de qualité ;
- Le système de recommandation fournit des ressources utiles en fonction des domaines des utilisateurs ;
- Les ressources récemment visitées recommandées aux utilisateurs sont pertinentes.

Les trois scénarios liés à ces trois hypothèses pour lesquelles des traces utilisateurs ont été recueillies sont énumérés ci-dessous :

- Évaluation du type de recommandation des ressources populaires :
 1. Le système recommande une ressource **populaire** à un utilisateur ;
 2. L'utilisateur clique sur la ressource et l'affiche ;
 3. L'utilisateur évalue la qualité de cette ressource ensuite il évalue son utilité en fonction de ses domaines ;

4. Si l'utilisateur évalue **la qualité** de cette ressource 1) avec quatre ou cinq étoiles, nous pouvons conclure que la recommandation est pertinente, 2) avec trois étoiles, alors la recommandation est modérément pertinente et 3) avec une ou deux étoiles, alors la recommandation n'est pas pertinente.
- Évaluation du type de recommandation des ressources utiles :
La phase 1, présentée dans le scénario précédent, est utilisée en changeant « **populaire** » en « **utile** ».
Les phases 2 et 3 sont identiques à celles du premier scénario.
Dans la phase 4, « **La qualité** » est remplacée par « **l'utilité de cette ressource pour au moins un de ses domaines** ».
- Évaluation du type de recommandation des ressources récemment visitées :
La phase 1, présentée dans le premier scénario, est utilisée en changeant « **populaire** » par « **visitées récemment** ».
Les phases 2 et 3 sont identiques à celles du premier scénario.
Dans la phase 4, « **La qualité** » est remplacée par « **la qualité de cette ressource ou évalue son utilité pour au moins un de ses domaines** ».

5.1.3 Résultats du Feedback des utilisateurs

Pendant les six mois d'utilisation de l'environnement d'apprentissage, un mécanisme de suivi a conservé les traces des actions des utilisateurs. Les utilisateurs ont déclaré 255 relations d'amitié. Les 175 ressources présentes dans Icras ont été visitées 5850 fois. Certaines ressources ont été visitées plusieurs fois par le même utilisateur. Les utilisateurs ont fait 1640 évaluations de ressources (1321 évaluations d'utilité et 319 évaluations de qualité). 17,71% des ressources ont été visités en cliquant sur les recommandations, tandis que les autres ressources ont été visitées directement via le menu de l'application.

Comme le montre la figure 5.2, nous avons constaté que 89,29% des recommandations sur les ressources populaires étaient déclarées pertinentes par les utilisateurs et 10,71% d'entre elles étaient modérément pertinentes. Nous avons également remarqué que la plupart des recommandations sur les ressources utiles (83,02%) étaient pertinentes, 9,43% étaient modérément pertinentes et 7,55% n'étaient pas pertinentes. En ce qui concerne les recommandations de

ressources récemment visitées, nous avons constaté que 81,67% d'entre elles étaient pertinentes, 15% étaient modérément pertinentes et 3,33% n'étaient pas pertinentes.

Pour mieux analyser ces résultats, nous avons également effectué une évaluation subjective basée sur un questionnaire. La section suivante présente cette évaluation.

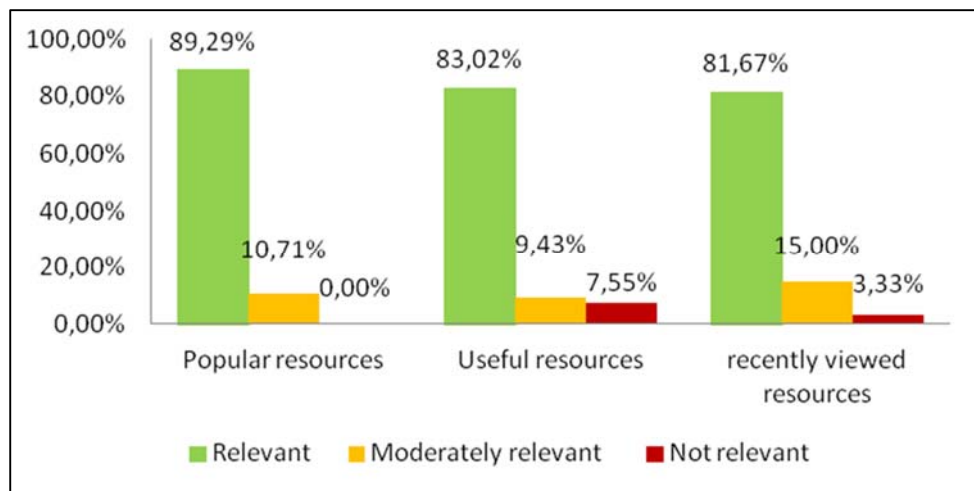


Figure 5.2 Taux de satisfaction pour les trois types de recommandation

5.1.4 Questionnaire

Pour vérifier si la recommandation proposée par notre approche est efficace et satisfaisante du point de vue de l'utilisateur, nous avons choisi d'utiliser un questionnaire basé sur ResQue (*Recommender systems' Quality of user experience*) (Pu et al., 2011). Cet outil a pour but d'évaluer les qualités perçues de l'utilisateur (utilisabilité, utilité, interface, qualité d'interaction, etc.), la satisfaction de l'utilisateur vis-à-vis du système et l'influence de ces qualités sur les intentions comportementales des utilisateurs (utiliser le système encore dans l'avenir, parler avec leurs amis du système, etc.).

ResQue comprend 60 questions. La version originale en anglais est présentée dans l'annexe II. Ses auteurs proposent une version simplifiée de ce questionnaire comprenant seulement 15 questions (marquée avec une * dans l'annexe II). Pour valider notre approche, nous avons choisi 19 questions auxquelles nous avons ajouté une question pour vérifier toutes nos hypothèses (Tableau 5.1). La question supplémentaire (question numéro 8) concerne l'incitation à la collaboration entre apprenants par le système de recommanda-

tion. Nous avons utilisé une échelle de Likert de 5 points de 1 (Pas du tout d'accord) à 5 (Tout à fait d'accord) pour déterminer les réponses des utilisateurs.

Ce questionnaire a été créé à l'aide de Google Forms¹⁴ et envoyé aux 40 étudiants actifs sélectionnés précédemment. 26 étudiants ont répondu à ce questionnaire (65% de taux de réponse).

Nous avons utilisé les réponses, de ces apprenants actifs, aux 20 questions sélectionnées pour analyser les 4 hypothèses suivantes (colonne Hypothèses du tableau 5.1) :

1. Le recommandeur fournit des ressources de qualité ;
2. Le recommandeur fournit des ressources utiles ;
3. Le recommandeur est facile à utiliser et les recommandations sont bien présentées aux étudiants ;
4. Les recommandations fournies satisferont les étudiants.

Les deux premières hypothèses sont liées à notre approche de recommandation, la troisième est liée à l'environnement d'apprentissage Icraa et la dernière est liée aux deux.

Tableau 5.1 Questionnaire ResQue

Hypothèses	Échelle ResQue	N°	Question
1	Qualité des ressources recommandées	1	Les ressources recommandées correspondaient à mes intérêts.
		2	Les recommandations que j'ai reçues correspondent mieux à mes intérêts par rapport à celles que je peux recevoir d'un ami.
		3	Les ressources qui m'ont été recommandées sont nouvelles et intéressantes.
		4	Le système de recommandation me permet de découvrir de nouvelles ressources.
		5	Les ressources qui me sont recommandées sont variées.
2	Utilité perçue	6	Les éléments recommandés m'ont aidé efficacement à trouver la ressource idéale.
		7	Je me sens soutenu pour retrouver ce que je veux avec l'aide du recommandeur.

¹⁴ <https://www.google.com/intl/fr/forms/about/>

		8	Le recommandeur m'a incité à collaborer avec mes amis sur certaines ressources.
3	Adéquation de l'interface	9	L'interface du recommandeur fournit des informations suffisantes.
		10	L'interface graphique du recommandeur est attrayante et adéquate.
	Perception de la facilité d'utilisation	11	Je me suis familiarisé avec le système de recommandation très rapidement.
		12	J'ai trouvé facile à utiliser le système de recommandation pour qu'il me recommande de différentes ressources.
		13	Il est facile pour moi d'informer le système si j'aime ou pas les éléments recommandés.
4	Attitudes	14	Globalement, je suis satisfait du recommandeur.
		15	Je suis convaincu des ressources qui me sont recommandées.
		16	Je suis confiant que je vais aimer les ressources qui me seront recommandées.
		17	Le recommandeur est digne de confiance.
	Intentions comportementales	18	J'utiliserai encore ce recommandeur.
		19	Je parlerai de ce recommandeur à mes amis.
20		Je visualiserai les éléments recommandés, quand l'opportunité se présentera.	

5.1.5 Résultats du questionnaire

Les résultats extraits du questionnaire sont présentés ci-dessous. La figure 5.3 montre la moyenne de l'échelle de Likert pour les 20 questions du questionnaire. Les moyennes des réponses sont comprises entre 2,92 et 4,38. Les valeurs d'écart type des résultats globaux, illustrées dans la figure 5.4, sont comprises entre 0,42 et 1,14. 19 valeurs moyennes sont supérieures à 3 (la réponse neutre) et une seule valeur moyenne est inférieure à 3, ce qui suggère que les attitudes des utilisateurs sont généralement positives.

Nous avons regroupé les 20 valeurs moyennes de l'échelle de Likert selon leurs catégories respectives (figure 5.5). Cette figure montre que la moyenne des 4 catégories est supérieure à 3 et donc que les 4 hypothèses posées sont vérifiées.

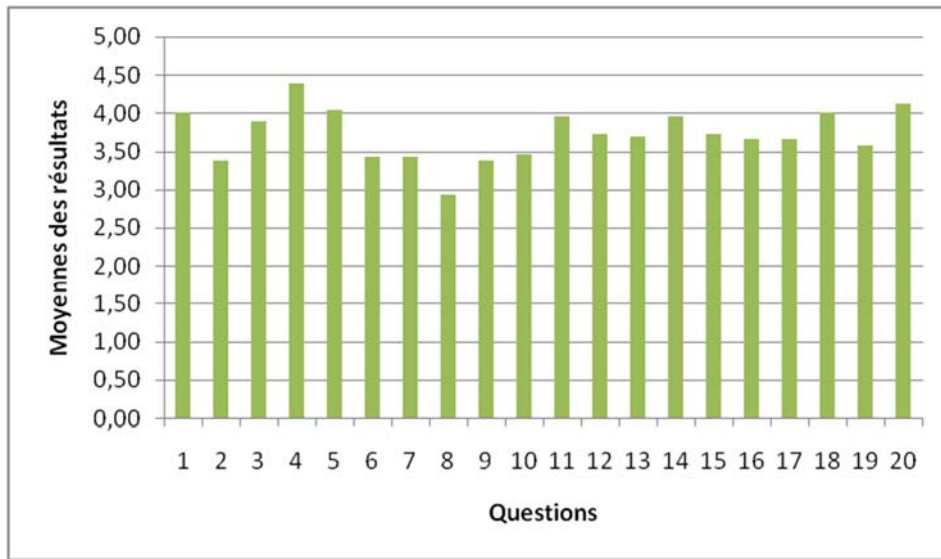


Figure 5.3 Moyennes des résultats du questionnaire

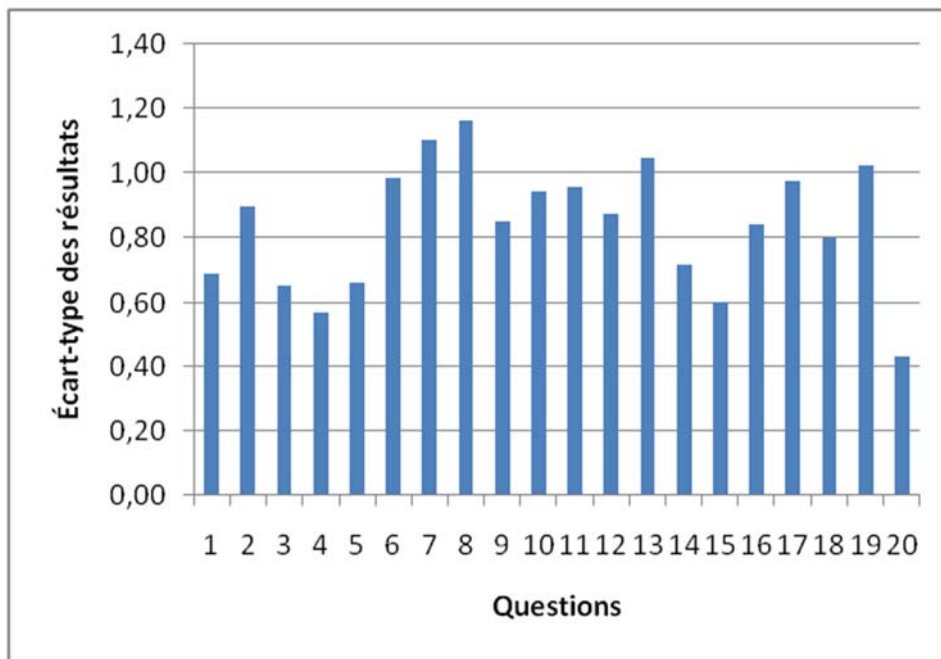


Figure 5.4 Écart-type des résultats du questionnaire

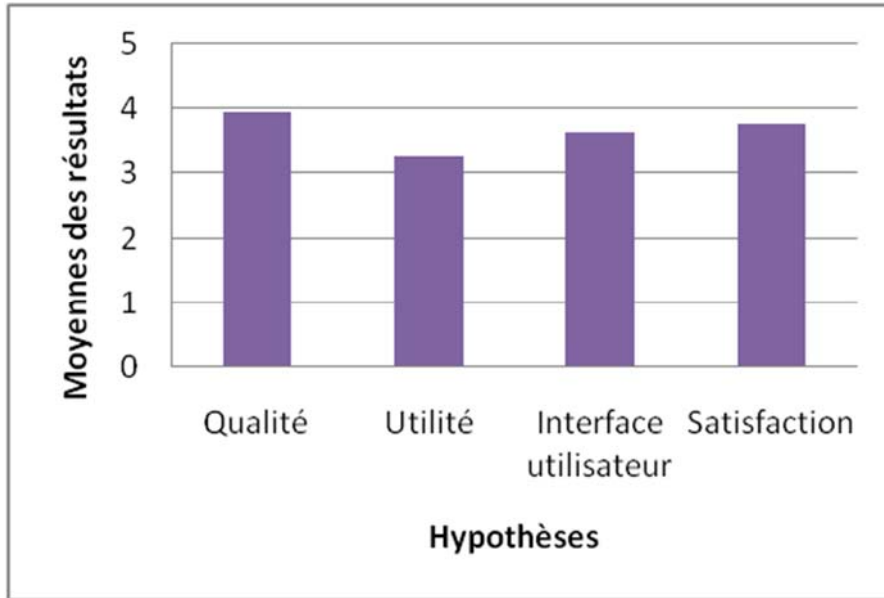


Figure 5.5 Moyennes des résultats du questionnaire groupés par hypothèses

Nous avons calculé l'alpha de Cronbach (Cronbach et Shavelson 2004) pour mesurer la fiabilité de notre questionnaire. La valeur d'un Alpha de Cronbach entre 0,9 et 1 indique un excellent niveau de fiabilité. Sa valeur pour notre questionnaire est de 0,903, ce qui suggère un haut niveau de fiabilité des résultats du questionnaire.

$$\alpha = \frac{k}{k - 1} \left(1 - \frac{\sum_{i=1}^k \sigma_{Y_i}^2}{\sigma_X^2} \right) \quad \text{Eq(5.1)}$$

La formule pour calculer la valeur α de Cronbach est représentée par l'équation Eq (5.1), où k est le nombre d'items, σ_X^2 est la variance du score total et $\sigma_{Y_i}^2$ est la variance de l'élément i .

5.1.6 Discussion

Nous avons utilisé les feedbacks des utilisateurs et les résultats d'un questionnaire pour valider notre approche. Les résultats obtenus à partir des feedbacks des utilisateurs démontrent que plus de 80% des recommandations sur les ressources fournies par les 3 types de recommandation étaient pertinentes pour les utilisateurs.

Les utilisateurs d'Icraa donnent globalement une évaluation positive. Environ 95% (19 sur 20 questions) des éléments du questionnaire ont un résultat supérieur à la moyenne (valeur moyenne ≥ 3). Dans l'ensemble, les résultats du questionnaire montrent que les

recommandations présentées aux utilisateurs étaient perçues comme pertinentes. Nous pouvons conclure que la pertinence des ressources recommandées par l'approche proposée répond à nos attentes initiales.

Seule la valeur moyenne de la question 8 est inférieure à trois. Pour cette question, 38,46% d'utilisateurs ont déclaré que le recommandeur les a aidés à collaborer avec leurs amis, 15,38% sont neutres sur cette question et 46,16% sont en désaccord avec celle-ci. Nous remarquons que cette fonctionnalité continue d'être utile pour une bonne partie des utilisateurs du système. Nous pensons que l'ajout d'une explication à la recommandation dans notre environnement d'apprentissage peut aider les utilisateurs à comprendre pourquoi le système leur a proposé ces ressources, et par la suite les incitera à collaborer avec leurs amis. Cette fonctionnalité peut leur donner une liste de leurs amis qui ont visité les ressources recommandées.

Seulement 98 sur les 238 utilisateurs de l'environnement d'apprentissage se sont connectés plus de 5 fois. Cela peut s'expliquer par les raisons suivantes :

- Les enseignants d'une promotion n'ont publié aucune ressource. Donc les étudiants de cette promotion n'avez aucune ressource à télécharger ;
- Certains étudiants se sont connectés que très peu de fois et ont téléchargé toutes leurs ressources.

Les 14 enseignants qui ont utilisé Icras ont continué à l'utiliser après la phase de l'expérimentation. Certains enseignants n'ont pas voulu essayer Icras par ce qu'ils se sont bien appropriés les systèmes cités à la section 5.1.1 et ne veulent pas en changer. Ceci peut également être lié à des défauts d'ergonomie de l'interface utilisateur de la première version d'Icras pour la publication de ressources. Depuis, nous avons amélioré cette interface des enseignants.

5.2 Expérimentation 2 : Évaluation avec un *Dataset* (Analyse Offline)

5.2.1 Objectif et hypothèses

Nous avons mené une analyse offline pour comparer la précision de notre approche avec celle de quatre algorithmes de référence (section 2.1.3.1), à savoir le filtrage collaboratif basé sur l'utilisateur, le filtrage collaboratif basé sur l'article (Item), SVD (Décomposition en valeurs singulières) et la popularité sociale. Pour rappel, le premier

l'algorithme calcule la similarité entre les utilisateurs en termes de leurs évaluations et prédit l'évaluation d'une ressource donnée. Le second suit le même processus que l'algorithme précédent, mais il est basé sur la similarité entre les ressources. SVD est une technique de factorisation de matrices utilisée dans les systèmes de recommandation. Elle élimine les utilisateurs ou les ressources insignifiants pour réduire les dimensions de la matrice Ressources/Utilisateurs. La méthode de la popularité sociale propose des recommandations basées sur la popularité générale des ressources dans le réseau des amis de l'utilisateur.

Les quatre algorithmes de référence sélectionnés recommandent des ressources en fonction des évaluations de la qualité. Ainsi, à partir des trois types de recommandations proposées dans notre approche, nous utilisons dans cette expérimentation simplement l'algorithme de la recommandation des ressources populaires, car il est également basé sur les évaluations de qualité.

5.2.2 Expérimentation

Nous avons développé une application Java qui calcule la précision des meilleures N recommandations pour chaque algorithme. Cette application utilise un ensemble de données extrait d'Icraa. Comme illustré sur la figure 5.6, l'expérimentation est composée de 3 phases. L'application Java d'abord divise les évaluations des utilisateurs de manière aléatoire dans des ensembles d'apprentissage (70%) et des ensembles de tests (30%). Dans une deuxième étape, l'application utilise l'ensemble d'apprentissage pour calculer les meilleures N ressources recommandées pour chaque utilisateur en utilisant les 5 algorithmes. Enfin, l'application utilise ces recommandations et l'ensemble de test pour calculer les scores F1.

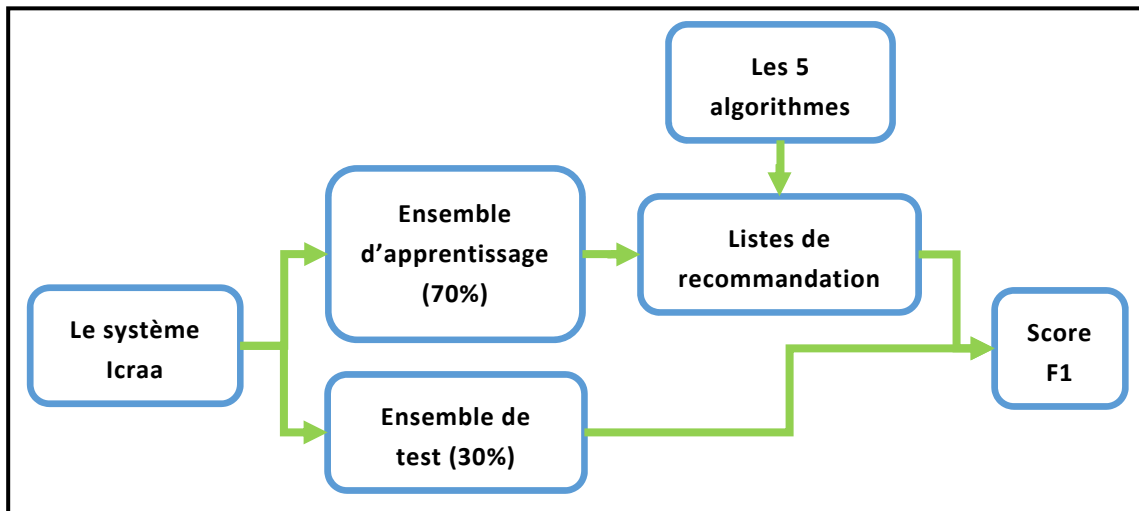


Figure 5.6 Processus d'évaluation avec un *dataset*

Le score F1 (Herlocker *et al.* 2004) est utilisé dans cette expérience pour mesurer la précision de chaque algorithme. F1 varie entre 0 et 1, et combine entre la précision et le rappel en un seul nombre. Nous calculons le F1 pour chaque utilisateur ensuite nous calculons la valeur moyenne de ces F1. Le score F1 a été calculé pour les N recommandations (N compris entre 3 et 10) de chaque utilisateur.

5.2.3 Ensemble de données (*Dataset*)

L'ensemble de données utilisé dans cette évaluation a été extrait d'Icraa après presque 2 ans d'utilisation (de novembre 2014 à août 2016). Il s'agit d'une version améliorée de l'ensemble de données utilisé dans la précédente expérimentation. L'ensemble de données contient 372 utilisateurs qui ont créé 517 relations d'amitié. Les 229 ressources d'Icraa ont été visitées 10295 fois. Les utilisateurs ont généré 344 évaluations de qualité et 1456 évaluations d'utilité. L'ensemble de données d'Icraa a une *sparsity* de 99.60% qui est semblable à la *sparsity* d'autres ensembles de données éducatifs tels que Mace (99.70%) et OpenScout (99.50%) (Fazeli *et al.* 2014).

5.2.4 Résultats

Les résultats de l'analyse offline sont présentés dans le Tableau 5.2. Ce tableau montre le score F1 moyen obtenu par les cinq algorithmes pour différentes valeurs de N. Les valeurs en surbrillance indiquent le

score F1 le plus performant. Ce tableau illustre également le pourcentage d'amélioration de notre algorithme par rapport au meilleur résultat des quatre autres algorithmes.

Pour $N = 3$, notre algorithme surpasse les autres algorithmes de référence et améliore le score F1 de 3,03% par rapport à la SVD qui occupe la deuxième place. Lorsque N varie de 4 à 9, notre algorithme dépasse aussi les autres et la précision des recommandations est améliorée de 4,92% à 10% par rapport à l'algorithme de popularité sociale qui occupe la deuxième place. Pour $N = 10$, l'algorithme proposé et la méthode de popularité sociale ont la même précision et ont le meilleur score F1 par rapport aux trois autres algorithmes.

Le score F1 améliorée de notre approche montre que celle-ci peut améliorer les recommandations dans les environnements d'apprentissage social, même si elles contiennent des évaluations *sparse*.

Tableau 5.2 Résultats de l'analyse offline

F1@N	Filtrage collaboratif basé sur l'utilisateur	Filtrage collaboratif basé sur l'item	SVD	Popularité sociale	Algorithme proposé dans Icras	Amélioration
F1@3	0.099	0.106	0.130	0.122	0.134	03.08%
F1@4	0.106	0.119	0.177	0.200	0.220	10.00%
F1@5	0.123	0.140	0.231	0.261	0.276	05.75%
F1@6	0.120	0.173	0.242	0.309	0.336	08.74%
F1@7	0.129	0.197	0.286	0.363	0.393	08.26%
F1@8	0.120	0.194	0.271	0.384	0.403	04.95%
F1@9	0.113	0.186	0.276	0.386	0.405	04.92%
F1@10	0.111	0.169	0.284	0.426	0.426	00.00%

5.3 Conclusion

Après avoir développé et mis en place l'environnement d'apprentissage Icras, qui met en œuvre notre approche. Nous avons effectué une expérience sur ce système auprès d'utilisateurs réels pour évaluer notre contribution. Globalement, l'expérience utilisateur a donné des résultats positifs. Les résultats du questionnaire que nous avons soumis aux utilisateurs d'Icras ont été concluants. Cette expérience a été suivie par une autre qui avait pour but d'analyser en hors ligne notre

approche en utilisant le *dataset* d'Icraa. Cette analyse a montré que l'approche présentée surpassait les quatre algorithmes de référence sélectionnés.

6 Conclusion et perspectives

6.1 Principales contributions	107
6.2 Limites et perspectives	108

6.1 Principales contributions

La surcharge d'informations est l'une des problématiques les plus importantes dans les plateformes d'apprentissage social. Au cours de cette thèse, nous avons présenté une approche de recommandation qui répond à cette problématique. Les approches existantes de recommandation de ressources pédagogiques fondées sur les liens sociaux ne bénéficient pas de toutes les informations liées à un utilisateur et à ses amis et ne tiennent pas compte de la nécessité d'avoir plusieurs types de recommandations. Notre approche est conçue pour répondre aux deux objectifs de recherche mentionnés dans le premier chapitre, à savoir : 1) l'amélioration de la précision des recommandations et 2) la réduction de la *sparsity* qui est un problème commun aux systèmes de recommandation pédagogique. Notre approche de recommandation se base sur le filtrage collaboratif en utilisant les relations sociales entre les utilisateurs, les profils de ces derniers et leurs évaluations des ressources pédagogiques.

Nous avons présenté une architecture générale de notre approche et développé un modèle formel pour le calcul de similarité entre les apprenants et pour la génération des trois types de recommandation que nous proposons, à savoir la recommandation des ressources : populaires, récemment visitées et utiles. Nous avons également développé un environnement d'apprentissage social, appelé *Icraa*, qui met en œuvre l'approche proposée, ce qui nous a permis de tester et affiner nos propositions.

Nous avons aussi présenté le protocole d'évaluation suivi pour valider notre approche. À l'aide de l'environnement d'apprentissage *Icraa*, nous avons mené une expérimentation en ligne avec des utilisateurs en situation réelle et une analyse hors ligne pour évaluer notre contribution par rapport à d'autres. Cette expérimentation a duré presque 2 ans. Globalement, cette expérience a donné des résultats positifs. Nous avons obtenu ainsi plus de 80% de satisfaction concernant les trois types de recommandation proposés par le système. Les résultats du questionnaire que nous avons soumis aux utilisateurs d'*Icraa* ont été concluants. Par ailleurs, l'analyse hors ligne utilisant l'ensemble de données d'*Icraa* montre que l'approche présentée sur-

passer les quatre algorithmes de référence sélectionnés, utilisés comme base comparative, ceci même si la matrice d'évaluation est très *sparse*.

6.2 Limites et perspectives

Les expérimentations nous ont permis de mettre en évidence certaines limites de nos travaux. Le résultat des réponses à la question 8 (le recommandeur m'a incité à collaborer avec mes amis sur certaines ressources) est inférieur à la moyenne (inférieur à trois sur l'échelle de Likert). Pour cette question, seulement 38,46% d'utilisateurs ont déclaré que le système de recommandation les a incités à collaborer avec leurs amis. Nous pensons que la génération et l'affichage d'une explication qui accompagne les recommandations de ressources pourrait aider les apprenants à comprendre pourquoi le système leur a proposé ces ressources, et par la suite les inciterait à collaborer avec leurs amis. Cette explication présente à l'apprenant ces amis qui ont visualisé ces ressources. Ceci incitera l'utilisateur à visualiser ces ressources et à contacter ses amis s'il rencontre des difficultés à appréhender les concepts ou à résoudre les exercices présents dans les ressources.

La formule de similarité sociale que nous proposons utilise cinq composantes avec des poids égaux, bien que les préférences implicites telles que les visites et les préférences explicites telles que les évaluations peuvent avoir des poids différents en tant que préférences de l'utilisateur. La pondération des différentes composantes de la formule de similarité sociales peut donner plus d'importance à certaines composantes et ainsi affiner la mesure de cette similarité. Le *dataset* récolté lors de l'expérimentation d'Icraa pourra aider à expérimenter cette nouvelle formule de similarité.

La première version du moteur de recommandation d'Icraa contenait de sérieux problèmes de performance. Par exemple, pour un utilisateur qui avait une trentaine d'amis, l'application prenait plus d'une minute et trente secondes pour lui afficher les recommandations. Dans un premier temps, nous avons remis en cause les ressources matérielles limitées de l'hébergement Web d'Icraa. Cependant bien que nous ayons augmenté ces ressources, le temps de réponse ne s'est amélioré que légèrement. Finalement, en optimisant le code source du moteur de recommandation nous avons pu obtenir de meilleurs résultats et le temps de chargement des recommanda-

tions est devenu presque instantané. Nous pensons que pour que notre moteur de recommandation soit utilisé dans un environnement réel, contenant plusieurs milliers d'utilisateurs et de ressources pédagogiques, son code source nécessitera davantage d'optimisation.

Les expérimentations de notre approche ont été conduites sur une communauté fermée d'utilisateurs (tous les utilisateurs faisaient partie du même département d'enseignement). Pour démontrer l'utilité et l'efficacité de notre approche il faudrait tester le moteur de recommandation dans d'autres contextes, notamment dans plusieurs domaines disciplinaires.

Actuellement, notre approche de recommandation utilise trois types d'actions utilisateurs, à savoir, la déclaration d'amitié, la visite et l'évaluation d'une ressource par un utilisateur. Nous pensons que l'intégration d'autres actions d'apprenants pourrait améliorer la précision du calcul de la similarité entre apprenants et par conséquent améliorer la précision des recommandations des ressources pédagogiques. Il s'agit d'intégrer les activités sociales (par exemple, inviter un apprenant à rejoindre le réseau d'amis, partager une ressource, donner un avis sur le contenu d'une ressource déposée par un apprenant, etc.) et les activités pédagogiques (par exemple, visualiser une ressource, résoudre un exercice, effectuer une simulation, réaliser un mini projet, etc.). Dans cette nouvelle approche, plus les apprenants effectuent les mêmes activités sur les mêmes objets (par exemple visiter ou corédiger la mêmes ressource), plus ils sont similaires. Il en est de même si les deux utilisateurs effectuent des activités équivalentes ou proches (par exemple, l'un partage une ressource avec ses amis et l'autre aime cette ressource) sur les mêmes objets.

Il s'agit ainsi de traiter la question de recherche suivante : comment générer des recommandations plus pertinentes pour les apprenants sur les plateformes d'apprentissage social, en se basant sur les activités sociales et les activités pédagogiques ?

Cette nouvelle approche est plus générique que celle qui est présentée dans cette thèse. Elle peut tirer profit de toutes les informations sociales d'un environnement d'apprentissage social. Dans la nouvelle approche, le calcul de la similarité entre deux apprenants comporte trois composantes, au lieu des cinq de l'approche actuelle :

- Similarité entre les profils des apprenants : cette similarité prend en compte les similitudes entre les apprenants en termes de buts, de préférences, de connaissances, d'expériences, etc. ;

- Force de lien entre les apprenants : cette similarité dépend des liens d'amitié, des liens de confiance et aux distances entre les apprenants ;
- Similarité entre les activités des apprenants : cette similarité prend en compte plusieurs type d'activités sociales et pédagogiques tels que la visite de ressource, partage, évaluation d'une ressource ou d'un commentaire, interactions directes (*chat*, envoi d'un message, appel audio, etc.).

Nous avons commencé à travailler sur cette nouvelle approche. Nous avons proposé deux sujets de master qui avait pour but d'enrichir les types d'activités sociales et pédagogiques présentes dans Icras. Le premier travail (Benosmane et Azzouni, 2016) avait pour objectif de créer un espace de collaboration entre enseignants/apprenants et apprenants/apprenants pour gérer la réalisation de projets d'apprentissage tels que la préparation des exposés, des travaux pratiques et des projets de fin d'études. Le deuxième travail (Mahi et Lahlouhi, 2017) améliore les activités sociales déjà présentes sur Icras, telles que l'envoi de messages et le partage d'information sur le mur, avec de nouvelles activités telles que la messagerie instantanée texte, audio et vidéo et la visio-conférence.

Dans cette nouvelle approche, une deuxième question de recherche est liée à la recommandation d'activités pédagogiques suivant le contexte de l'apprenant. Comme expliqué dans l'introduction de ce document, la multitude de ressources, de relations et d'interactions présentes dans les environnements d'apprentissage sociaux peut conduire les apprenants à subir une surcharge informationnelle. Pour mieux traiter cette problématique, le système ne doit pas juste recommander des ressources pédagogiques mais il doit aussi recommander des activités pédagogiques. Suivant le contexte dans lequel se trouve l'apprenant, le système pourra, par exemple lui recommander une liste d'activités comprenant : lire la ressource X, poser la question à Y, suivre le cours Z.

Le processus de recommandation à suivre pourra être le suivant :

- Détecter le contexte actuel de l'apprenant (détecter l'activité actuelle de l'apprenant, détecter les difficultés de l'apprenant, etc.) ;
- Calcul de la similarité avec les autres apprenants ;
- Calculer les scores pour chaque type d'activités (ressources, exercices, pairs, experts, etc.) ;

- Prioriser les scores sur les types d'activités ;
- Présenter les 3 meilleures activités.

Le *dataset* extrait d'Icraa ouvre aussi la voie à de nombreuses perspectives. Nous comptons décrire ce *dataset* et le publier pour qu'il soit disponible pour les chercheurs souhaitant faire des expérimentations sur les EIAH et/ou les systèmes de recommandation.



Bibliographie

Adomavicius, G., Manouselis, N., & Kwon, Y. (2011). Multi-criteria recommender systems. In Recommender systems handbook (pp. 769-803). Springer US.

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE transactions on knowledge and data engineering, 17(6), (pp. 734-749).

Ahn, J. W., Brusilovsky, P., Grady, J., He, D., & Syn, S. Y. (2007, May). Open user profiles for adaptive news systems: help or harm? In Proceedings of the 16th international conference on World Wide Web (pp. 11-20). ACM.

Anderson, M., Ball, M., Boley, H., Greene, S., Howse, N., Lemire, D., & McGrath, S. (2003). Racofi: A rule-applying collaborative filtering system. In Proc. IEEE/WIC COLA'03.

Balabanović, M., & Shoham, Y. (1997). Fab: content-based, collaborative recommendation. Communications of the ACM, 40(3), (pp. 66-72).

Barman, K., & Dabeer, O. (2010, June). Local popularity based collaborative filters. In Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on (pp. 1668-1672). IEEE.

Bedi, P., Kaur, H., & Marwaha, S. (2007, January). Trust Based Recommender System for Semantic Web. In IJCAI (Vol. 7, pp. 2677-2682).

Beham, G., Stern, H., & Lindstaedt, S. (2010). APOSDLE-DS—A Dataset from the APOSDLE Work-integrated Learning System. In *1st Workshop on Recommender Systems for Technology Enhanced Learning (RecSysTEL 2010)*.

Belhabib, A., & Matahri, M. A. (2014). Conception d'un système de recommandation pour un réseau sociale d'apprentissage. Mémoire de master. Université de Tlemcen.

Beliakov, G., Calvo, T., & James, S. (2011). Aggregation of preferences in recommender systems. In *Recommender systems handbook* (pp. 705-734). Springer, Boston, MA.

Bellogín, A., Cantador, I., Díez, F., Castells, P., & Chavarriga, E. (2013). An empirical comparison of social, collaborative filtering, and hybrid recommenders. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(1), 14.

Benosmane, S., & Azzouni, I. (2016). Réalisation d'une application pour l'apprentissage basé sur les projets. Mémoire de master. Université de Tlemcen.

Bensmaine, Y., & Bouacha O. (2013). Conception et réalisation d'un système d'apprentissage dans les réseaux sociaux. Mémoire de master. Université de Tlemcen.

Bhagat, S., Cormode, G., & Muthukrishnan, S. (2011). Node classification in social networks. In *Social network data analytics* (pp. 115-148). Springer, Boston, MA.

Billsus, D., & Pazzani, M. J. (1998, July). Learning Collaborative Information Filters. In *Proceedings of the 15th International Conference on Machine Learning (ICML '98)* (Vol. 98, pp. 46-54).

Brusilovsky, P., Cassel, L. N., Delcambre, L. M., Fox, E. A., Furuta, R., Garcia, D. D., ... & Yudelson, M. (2010). Social navigation for educational digital libraries. *Procedia Computer Science*, 1(2), (pp. 2889-2897).

Buder, J., & Schwind, C. (2012). Learning with personalized recommender systems: A psychological view. *Computers in Human Behavior*, 28(1), (pp. 207-216).

Çano, E., & Morisio, M. (2017). Hybrid recommender systems: A systematic literature review. *Intelligent Data Analysis*, 21(6), (pp. 1487-1524).

Chatti, M. A., Dakova, S., Thus, H., & Schroeder, U. (2013). Tag-based collaborative filtering recommendation in personal learning environments. *IEEE Transactions on learning technologies*, 6(4), (pp. 337-349).

Chen, J. M., Chen, M. C., & Sun, Y. S. (2014). A tag based learning approach to knowledge acquisition for constructing prior knowledge and enhancing student reading comprehension. *Computers & Education*, 70, (pp. 256-268).

CLEVERDON, C. & KEAN, M. (1968). Factors Determining the Performance of Indexing Systems. Aslib Cranfield Research Project, Cranfield, England.

Cronbach, L. J., & Shavelson, R. J. (2004). My current thoughts on coefficient alpha and successor procedures. *Educational and psychological measurement*, 64(3), (pp. 391-418).

D. Goldberg, D. Nichols, B. M. Oki, & D. Terry. (1992). "Using Collaborative Filtering to Weave an Information Tapestry," *Communications of the ACM*, vol. 35, no 12, (pp. 61-70).

Dalia Sulieman. (2014). Systèmes de recommandation sociaux et sémantiques. Thèse. Université de Cergy Pontoise.

Deezer, G. (2014). Laissez-vous surprendre par la musique, <https://www.deezer.com/fr/company>, visitée le 25 mars 2018.

Degemmis, M., Lops, P., & Semeraro, G. (2007). A content-collaborative recommender that exploits WordNet-based user profiles for neighborhood formation. *User Modeling and User-Adapted Interaction*, 17(3), (pp. 217-255).

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1), (pp. 269-271).

Drachsler, H. (2012). Recommender Systems for Learning, <http://fr.slideshare.net/Drachsler/recsystem-lecture-at-advanced-siks-course-nl>. Visitée le 29-04-2018.

Drachsler, H., Hummel, H. G. K., & Koper, R. (2007). Recommendations for learners are different: Applying memory-based recommender system techniques to lifelong learning. Paper presented at the SIRTEL workshop at the EC-TEL 2007 Conference. September, 17-20, 2007, Crete, Greece.

Drachsler, H., Hummel, H. G. K., & Koper, R. (2009). Identifying the goal, user model and conditions of recommender systems for formal and informal learning. *Journal of Digital Information*. 10(2), (pp. 4–24).

Drachsler, H., Hummel, H. G., Van den Berg, B., Eshuis, J., Waterink, W., Nadolski, R., Berlanga, A. J., Boers, N., & Koper, R. (2009). Effects of the ISIS Recommender System for Navigation Support in self-organised Learning Networks. *Educational Technology & Society*, 12(3), (pp. 115-126).

Drachsler, H., Pecceu, D., Arts, T., Hutten, E., Rutledge, L., Van Rosmalen, P., Hummel, H., & Koper, R. (2009, September). ReMashed—recommendations for mash-up personal learning environments. In *European Conference on Technology Enhanced Learning* (pp. 788-793). Springer, Berlin, Heidelberg.

Drachsler, H., Verbert, K., Santos, O. C., & Manouselis, N. (2015). Panorama of recommender systems to support learning. In *Recommender systems handbook* (pp. 421-451). Springer, Boston, MA.

Ekstrand, M. D., Riedl, J. T., & Konstan, J. A. (2011). Collaborative filtering recommender systems. *Foundations and Trends in Human-Computer Interaction*, 4(2), (pp. 81-173).

El Helou, S., Salzmann, C., & Gillet, D. (2010). The 3A Personalized, Contextual and Relation-based Recommender System. *J. UCS*, 16(16), (pp. 2179-2195).

Erdt, M., Fernandez, A., & Rensing, C. (2015). Evaluating recommender systems for technology enhanced learning: a quantitative survey. *IEEE Transactions on Learning Technologies*, 8(4), (pp. 326-344).

Fazeli, S., Drachsler, H., Bitter-Rijkema, M., Brouns, F., Van der Vegt, W., & Sloep, P. B. (2017). User-centric Evaluation of Recommender Systems in Social Learning Platforms: Accuracy is Just the Tip of the Iceberg. *IEEE Transactions on Learning Technologies*.

Fazeli, S., Drachsler, H., Brouns, F., & Sloep, P. (2014). Towards a social trust-aware recommender for teachers. In *Recommender systems for technology enhanced learning* (pp. 177-194). Springer, New York, NY.

Fazeli, S., Loni, B., Drachsler, H., & Sloep, P. (2014, September). Which recommender system can best fit social learning platforms?. In *European Conference on Technology Enhanced Learning* (pp. 84-97). Springer, Cham.

Fazeli, S., Rajabi, E., Lezcano, L., Drachsler, H., & Sloep, P. (2016, July). Supporting Users of Open Online Courses with Recommendations: an Algorithmic Study. In *Advanced Learning Technologies (ICALT), 2016 IEEE 16th International Conference on* (pp. 423-427). IEEE.

Gabrielsson, S. & Gabrielsson, S. 2006. The use of Self-Organizing Maps in Recommender Systems, A survey of the Recommender Systems field and a presentation of a State of the Art Highly Interactive Visual Movie Recommender System. Mémoire de master, Uppsala University.

Ghauth, K. I., & Abdullah, N. A. (2011). The effect of incorporating good learners' ratings in e-Learning content-based recommender System. *Journal of Educational Technology & Society*, 14(2), (pp. 248-257).

Grasha, A.F. (1996). *Teaching with style: A practical guide to enhancing learning by understanding teaching and learning styles*. Pittsburgh, PA: Alliance Publishers, (800), (pp. 718-4287).

Guy, I. & Carmel, D. (2011). Social Recommender Systems; Proc. of the 20th international conf. companion on World Wide Web, (pp. 283-284). ACM.

Guy, I., Zwerdling, N., Carmel, D., Ronen, I., Uziel, E., Yogev, S., & Ofek-Koifman, S. (2009, October). Personalized recommendation of social software items based on social relations. In *Proceedings of the third ACM conference on Recommender systems* (pp. 53-60). ACM.

Guy, I., Zwerdling, N., Ronen, I., Carmel, D., & Uziel, E. (2010, July). Social media recommendation based on people and tags. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval* (pp. 194-201). ACM.

Han, P., Xie, B., Yang, F., & Shen, R. (2004). A scalable P2P recommender system based on distributed collaborative filtering. *Expert systems with applications*, 27(2), (pp. 203-210).

Hanani, U., Shapira, B., & Shoval, P. (2001). Information filtering: Overview of issues, research and systems. *User modeling and user-adapted interaction*, 11(3), (pp. 203-259).

Herlocker, J. L., Konstan, J. A., & Riedl, J. (2002). An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information retrieval*, 5(4), (pp. 287-310).

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

Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1), (pp. 5-53).

Hirsch, J. E. (2005). An index to quantify an individual's scientific research output. *Proceedings of the National academy of Sciences of the United States of America*, 102(46), (pp. 16569-16572).

Hsu, C. K., Chang, C. K., & Hwang, G. J. (2009). Development of a reading material recommendation system based on a multi-expert knowledge acquisition approach. In *Advanced Learning Technologies, 2009. ICAIT 2009*, (pp. 273-277). IEEE.

Huang, Z., Chen, H., & Zeng, D. (2004). Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, 22(1), (pp. 116-142).

Huynh, K. T., Finance, B., & Bouzeghoub, M. (2012). Towards an Ambient Data Mediation System. In *Proc. of the 2nd Int. Workshop on Information Management for Mobile Applications (IMMoA 2012)*, Istanbul, Turkey (pp. 13-20).

Jack, K., Hammerton, J., Harvey, D., Hoyt, J. J., Reichelt, J., & Henning, V. (2010). Mendeleys reply to the datatel challenge. In *1st Workshop on*

Recommender Systems for Technology Enhanced Learning (RecSys- TEL 2010).

Jamali, M., & Ester, M. (2009, June). Trustwalker: a random walk model for combining trust-based and item-based recommendation. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 397-406). ACM.

Karampiperis, P., Koukourikos, A., & Stoitsis, G. (2014). Collaborative filtering recommendation of educational content in social environments utilizing sentiment analysis techniques. In Recommender Systems for Technology Enhanced Learning (pp. 3-23). Springer, New York, NY.

Kitts, B., Freed, D., & Vrieze, M. (2000, August). Cross-sell: a fast promotion-tunable customer-item recommendation method based on conditionally independent probabilities. In Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 437-446). ACM.

Knijnenburg, B. P., Willemsen, M. C., Gantner, Z., Soncu, H., & Newell, C. (2012). Explaining the user experience of recommender systems. *User Modeling and User-Adapted Interaction*, 22(4-5), (pp. 441-504).

Lee, S. (2012, April). A generic graph-based multidimensional recommendation framework and its implementations. In Proceedings of the 21st International Conference on World Wide Web (pp. 161-166). ACM.

Limongelli, C., Lombardi, M., Marani, A., & Sciarrone, F. (2013, July). A teaching-style based social network for didactic building and sharing. In *International Conference on Artificial Intelligence in Education* (pp. 774-777). Springer Berlin Heidelberg.

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

Liu, F., & Lee, H. J. (2010). Use of social network information to enhance collaborative filtering performance. *Expert systems with applications*, 37(7), (pp. 4772-4778).

Lops, P., De Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In *Recommender systems handbook* (pp. 73-105). Springer, Boston, MA.

Ma, H., King, I., & Lyu, M. R. (2009, July). Learning to recommend with social trust ensemble. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval* (pp. 203-210). ACM.

Ma, H., Zhou, D., Liu, C., Lyu, M. R., & King, I. (2011, February). Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining* (pp. 287-296). ACM.

Mahi, S., & Lahlouhi, M. (2017). Réalisation d'un système collaboratif pour StudyPress. Mémoire de master. Université de Tlemcen.

Manouselis, N., Drachsler, H., Verbert, K., and Duval, E. (2012). *Recommender Systems for Learning*. Berlin, Springer.

McCalla, G. (2004). The ecological approach to the design of e-learning environments: Purpose-based capture and use of information about learners. *Journal of Interactive Media in Education*, 2004(1).

McNee, S. M., Riedl, J., & Konstan, J. A. (2006, April). Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI'06 extended abstracts on Human factors in computing systems* (pp. 1097-1101). ACM.

McPherson, M., Smith-Lovin, L., & Cook, J. M. (2001). Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1), (pp. 415-444).

Miller, B. N., Konstan, J. A., & Riedl, J. (2004). PocketLens: Toward a personal recommender system. *ACM Transactions on Information Systems (TOIS)*, 22(3), (pp. 437-476).

Montaner, M., López, B., & De La Rosa, J. L. (2003). A taxonomy of recommender agents on the internet. *Artificial intelligence review*, 19(4), (pp. 285-330).

Mooney, R. J., & Roy, L. (2000, June). Content-based book recommending using learning for text categorization. In Proceedings of the fifth ACM conference on Digital libraries (pp. 195-204). ACM.

Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). The PageRank citation ranking: Bringing order to the web. Stanford InfoLab. Technical report, Stanford Digital Library Technologies Project.

Pazzani, M. J., & Billsus, D. (2007). Content-based recommendation systems. In *The adaptive web* (pp. 325-341). Springer, Berlin, Heidelberg.

Pintrich, P. R., & De Groot, E. V. (1990). Motivational and self-regulated learning components of classroom academic performance. *Journal of educational psychology*, 82(1), 33.

Popescu, E. (2014). Providing collaborative learning support with social media in an integrated environment. *World Wide Web*, 17(2), (pp. 199-212).

Pu, P., Chen, L., & Hu, R. (2011, October). A user-centric evaluation framework for recommender systems. In Proceedings of the fifth ACM conference on Recommender systems (pp. 157-164). ACM.

Recker, M. M., & Walker, A. (2003). Supporting "word-of-mouth" social networks through collaborative information filtering. *Journal of Interactive Learning Research*, 14(1), 79.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994, October). GroupLens: an open architecture for collaborative filtering of netnews. In Proceedings of the 1994 ACM conference on Computer supported cooperative work (pp. 175-186). ACM.

Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to recommender systems handbook. In *Recommender systems handbook* (pp. 1-35). Springer US.

Romero, C., & Ventura, S. (2007). Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications*, 33(1), (pp. 135-146). Elsevier.

Saidi, S., & Bendella, M. (2015). Réalisation d'un outil auteur interactif et social sous WordPress, Université de Tlemcen. Mémoire de master. Université de Tlemcen.

Salehi, M. (2013). Application of implicit and explicit attribute based collaborative filtering and BIDE for learning resource recommendation. *Data & Knowledge Engineering*, 87, (pp. 130-145).

Salton, G. and McGill, M. *Introduction to Modern Information Retrieval*. McGraw-Hill Education, 1983.

Santos, O. C., & Boticario, J. G. (2014, June). Exploring Arduino for building educational context-aware recommender systems that deliver affective recommendations in social ubiquitous networking environments. In *International Conference on Web-Age Information Management* (pp. 272-286). Springer, Cham.

Santos, O. C., & Boticario, J. G. (2015). User-centred design and educational data mining support during the recommendations elicitation process in social online learning environments. *Expert Systems*, 32(2), (pp. 293-311).

Santos, O. C., Boticario, J. G., & Pérez-Marin, D. (2014). Extending web-based educational systems with personalised support through User Centred Designed recommendations along the e- learning life cycle. *Science of Computer Programming*, vol. 88, no. 0, (pp. 92–109).

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000). Analysis of recommendation algorithms for e-commerce. In *Proceedings of the 2nd ACM conference on Electronic commerce* (pp. 158-167). ACM.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001, April). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web* (pp. 285-295). ACM.

Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. In *The adaptive web* (pp. 291-324). Springer, Berlin, Heidelberg.

Schafer, J. B., Konstan, J. A., & Riedl, J. (2001). E-commerce recommendation applications. *Data mining and knowledge discovery*, 5(1-2), (pp. 115-153).

ScrapeHero. How Many Products Does Amazon Sell? – January 2018, <https://www.scrapehero.com/many-products-amazon-sell-january-2018>, visitée le 25 mars 2018.

Shani, G., & Gunawardana, A. (2011). Evaluating recommendation systems. In *Recommender systems handbook* (pp. 257-297). Springer, Boston, MA.

Shardanand, U., & Maes, P. (1995, May). Social information filtering: algorithms for automating "word of mouth". In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 210-217). ACM Press/Addison-Wesley Publishing Co.

Shi, L. (2014). Defining and evaluating learner experience for social adaptive e-learning. In: *The 4th Imperial College Computing Student Workshop (ICCSW 2014)*, London, 25- 26 Sep 2014.

Shi, L., Gkotsis, G., Stepanyan, K., Al Qudah, D., & Cristea, A. I. (2013, July). Social personalized adaptive e-learning environment: Topolor-implementation and evaluation. In *International Conference on Artificial Intelligence in Education* (pp. 708-711). Springer, Berlin, Heidelberg.

Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in artificial intelligence*, vol. 2009, p. 19.

Sun, J., & Tang, J. (2011). A survey of models and algorithms for social influence analysis. In *Social network data analytics* (pp. 177-214). Springer, Boston, MA.

Tadlaoui, M. (2014). Recommandation de ressources pédagogiques basée sur les relations sociales. In *RJC EIAH 2014*, La Rochelle, France. (pp. 101-106).

Tadlaoui, M., George, S., & Sehaba, K. (2014). Recommandation de ressources pédagogiques dans les réseaux sociaux en ligne, *Journée Scientifique "Systèmes de recommandation et réseaux sociaux"*, Chambéry, France.



Tadlaoui, M., George, S., & Sehaba, K. (2015). Approche pour la recommandation de ressources pédagogiques basée sur les liens sociaux. In 7ème Conférence sur les Environnements Informatiques pour l'Apprentissage Humain (EIAH 2015), Agadir, Maroc. (pp. 192-203).

Tadlaoui, M., Sehaba, K., & George, S. (2015, May). Recommendation of Learning Resources based on Social Relations. In Proc. of the 7th International Conference on Computer Supported Education (CSEDU 2015) (pp. 425-432).

Tadlaoui, M., Sehaba, K., & George, S. (2017). Évaluation d'un système de recommandation de ressources pédagogiques fondé sur les relations sociales. In 8ème Conférence sur les Environnements Informatiques pour l'Apprentissage Humain (EIAH 2017), Strasbourg, France. (pp. 427-429).

Tadlaoui, M., Sehaba, K., George, S., Chikh, A., & Bouamrane, K. (2018). Social recommender approach for technology-enhanced learning. International Journal of Learning Technology, Inderscience 13(1), (pp. 61-89).

Thai-Nghe, N., Drumond, L., Krohn-Grimberghe, A., & Schmidt-Thieme, L. (2010). Recommender system for predicting student performance. Procedia Computer Science, 1(2), (pp. 2811-2819).

Vassileva, J. (2008). Toward social learning environments. IEEE transactions on learning technologies, 1(4), (pp. 199-214).

Verbert, K., Drachsler, H., Manouselis, N., Wolpers, M., Vuorikari, R., & Duval, E. (2011, February). Dataset-driven research for improving recommender systems for learning. In Proceedings of the 1st International Conference on Learning Analytics and Knowledge (pp. 44-53). ACM.

Wei, C. P., Shaw, M.J., & Easley, R.F. (2002). A Survey of Recommendation Systems in Electronic Commerce. In Rust, R. T. & Kannan, P. K. (Eds.), E-Serv.: New Dir. in Theor. and Pract., M. E. Sharpe Publisher.

Wolpers, M., & Niemann, K. (2010). dataTEL challenge: CAM for MACE. In 1st Workshop on Recommender Systems for Technology Enhanced Learning (RecSysTEL 2010).

Wu, L., Shah, S., Choi, S., Tiwari, M., & Posse, C. (2014, October). The Browsemaps: Collaborative Filtering at LinkedIn. In 6th Workshop on

Recommender Systems and the Social Web, collocated with ACM RecSys 2014.

Yang, X., Guo, Y., Liu, Y., & Steck, H. (2014). A survey of collaborative filtering based social recommender systems. *Computer Communications*, 41, (pp. 1-10).

Yu, C., Lakshmanan, L., & Amer-Yahia, S. (2009, March). It takes variety to make a world: diversification in recommender systems. In *Proceedings of the 12th international conference on extending database technology: Advances in database technology* (pp. 368-378). ACM.

Yueh-Min, H., Tien-Chi, H., Wang, K. T., & Hwang, W. Y. (2009). A Markov-based recommendation model for exploring the transfer of learning on the web. *Journal of Educational Technology & Society*, 12(2), 144.

Zhao, X., Niu, Z., Chen, W., Shi, C., Niu, K., & Liu, D. (2015). A hybrid approach of topic model and matrix factorization based on two-step recommendation framework. *Journal of Intelligent Information Systems*, 44(3), (pp. 335-353).

Annexes

Annexe I : Publications et communications.....	126
A. Publications dans des revues.....	126
B. Chapitres de livre.....	126
C. Conférences internationales avec comité de lecture et actes.....	126
D. Conférences nationales avec comité de lecture et actes.....	126
E. Ateliers et posters.....	127
Annexe II : Questionnaire RESQUE en anglais	128
Annexe III : Comparatif entre les jeux de données EIAH (Manouselis <i>et al.</i> 2012).....	131

Annexe I : Publications et communications

A. Publications dans des revues

Tadlaoui, M., Sehaba, K., George, S., Chikh, A., & Bouamrane, K. (2018). Social recommender approach for technology-enhanced learning. *International Journal of Learning Technology, Inderscience* 13(1), (pp. 61-89).

B. Chapitres de livre

Tadlaoui, M., Chikh, A., & Bouamrane, K. (2013). ALEM: A Reference Model for Educational Adaptive Web Applications. In *Intelligent and Adaptive Educational-Learning Systems* (pp. 25-48). Springer, Berlin, Heidelberg.

C. Conférences internationales avec comité de lecture et actes

Tadlaoui, M., Sehaba, K., & George, S. (2015). Recommendation of Learning Resources based on Social Relations. In *CSEDU 2015, Lisbonne, Portugal*. (pp. 425-432).

Tadlaoui, M., Chikh, A., & Bouamrane, K. (2010). ALEM: Un modèle de référence pour les applications web adaptatif éducatif. *Proceedings of Leafa 2010 (e-Learning For All)*. Hammamet, Tunisie.

D. Conférences nationales avec comité de lecture et actes

Tadlaoui, M., Sehaba, K., & George, S. (2017). Évaluation d'un système de recommandation de ressources pédagogiques fondé sur les relations sociales. In *8ème Conférence sur les Environnements In-*

formatiques pour l'Apprentissage Humain (EIAH 2017), Strasbourg, France. (pp. 427-429).

Tadlaoui, M., George, S., & Sehaba, K. (2015). Approche pour la recommandation de ressources pédagogiques basée sur les liens sociaux. In 7ème Conférence sur les Environnements Informatiques pour l'Apprentissage Humain (EIAH 2015), Agadir, Maroc. (pp. 192-203).

Tadlaoui, M. (2014). Recommandation de ressources pédagogiques basée sur les relations sociales. In RJC EIAH 2014, La Rochelle, France. (pp. 101-106).

E. Ateliers et posters

Tadlaoui, M., George, S., & Sehaba, K. (2014). Recommandation de ressources pédagogiques dans les réseaux sociaux en ligne, Journée Scientifique "Systèmes de recommandation et réseaux sociaux", Chambéry, France.

Annexe II : Questionnaire RESQUE en anglais

La version simplifiée de ce questionnaire comporte seulement les 15 questions marquées avec une *.

1. Quality of Recommended Items

1.1. Accuracy

- The items recommended to me matched my interests.*
- The recommender gave me good suggestions.
- I am not interested in the items recommended to me (reverse scale).

1.2. Relative Accuracy

- The recommendation I received better fits my interests than what I may receive from a friend.
- A recommendation from my friends better suits my interests than the recommendation from this system (reverse scale).

1.3. Familiarity

- Some of the recommended items are familiar to me.
- I am not familiar with the items that were recommended to me (reverse scale).

1.4. Attractiveness

- The items recommended to me are attractive.

1.5. Enjoyability

- I enjoyed the items recommended to me.

1.6. Novelty

- The items recommended to me are novel and interesting.*
- The recommender system is educational.
- The recommender system helps me discover new products.
- I could not find new items through the recommender (reverse scale).

1.7. Diversity

- The items recommended to me are diverse.*
- The items recommended to me are similar to each other (reverse scale).*

1.8. Context Compatibility

- I was only provided with general recommendations.
- The items recommended to me took my personal context requirements into consideration.
- The recommendations are timely.

2. Interaction Adequacy

- The recommender provides an adequate way for me to express my preferences.

- The recommender provides an adequate way for me to revise my preferences.
 - The recommender explains why the products are recommended to me.*
3. Interface Adequacy
- The recommender's interface provides sufficient information.
 - The information provided for the recommended items is sufficient for me.
 - The labels of the recommender interface are clear and adequate.
 - The layout of the recommender interface is attractive and adequate.*
4. Perceived Ease of Use
- 4.1. Ease of Initial Learning
- I became familiar with the recommender system very quickly.
 - I easily found the recommended items.
 - Looking for a recommended item required too much effort (reverse scale).
- 4.2. Ease of Preference Elicitation
- I found it easy to tell the system about my preferences.
 - It is easy to learn to tell the system what I like.
 - It required too much effort to tell the system what I like (reversed scale).
- 4.3. Ease of Preference Revision
- I found it easy to make the system recommend different things to me.
 - It is easy to train the system to update my preferences.
 - I found it easy to alter the outcome of the recommended items due to my preference changes.
 - It is easy for me to inform the system if I dislike/like the recommended item.
 - It is easy for me to get a new set of recommendations.
- 4.4. Ease of Decision Making
- Using the recommender to find what I like is easy.
 - I was able to take advantage of the recommender very quickly.
 - I quickly became productive with the recommender.
 - Finding an item to buy with the help of the recommender is easy.*
 - Finding an item to buy, even with the help of the recommender, consumes too much time.
5. Perceived Usefulness
- The recommended items effectively helped me find the ideal product.*
 - The recommended items influence my selection of products.
 - I feel supported to find what I like with the help of the recommender.*
 - I feel supported in selecting the items to buy with the help of the recommender.
6. Control/Transparency
- I feel in control of telling the recommender what I want.
 - I don't feel in control of telling the system what I want.

- I don't feel in control of specifying and changing my preferences (reverse scale).
- I understood why the items were recommended to me.
- The system helps me understand why the items were recommended to me.
- The system seems to control my decision process rather than me (reverse scale).

7. Attitudes

- Overall, I am satisfied with the recommender.*
- I am convinced of the products recommended to me.*
- I am confident I will like the items recommended to me. *
- The recommender made me more confident about my selection/decision.
- The recommended items made me confused about my choice (reverse scale).
- The recommender can be trusted.

8. Behavioral Intentions

8.1. Intention to Use the System

- If a recommender such as this exists, I will use it to find products to buy.

8.2. Continuance and Frequency

- I will use this recommender again.*
- I will use this type of recommender frequently.
- I prefer to use this type of recommender in the future.

8.3. Recommendation to Friends

- I will tell my friends about this recommender.*

8.4. Purchase Intention

- I would buy the items recommended, given the opportunity.*

Annexe III : Comparatif entre les jeux de données EIAH (Manouselis *et al.* 2012)

		dataTEL										PSLC dataShop						
		Mendeley	MERLOT	APOSLE	ReMashed	Organic.edunet	MACE	Travel well	ROLE	SidWeb	U3CM	CGIAR	Algebra 2008-2009	Bridge to Algebra	Geometry area	Electric Fields - Pitt	China Vocabulary	Handwriting2/Example
action type	attempt			+							+	+	+	+	+	+	+	+
	create/delete			+					+	+	+	+				+		
	write/edit			+							+	+				+		
	select/unselect	+		+			+		+	+	+	+				+		
	search			+			+		+		+	+						
	send/receive			+					+	+		+						
	register			+								+						
	open/close			+					+		+	+						
	add/remove											+						
	tag/annotate			+	+		+	+				+						
	rate/star	+	+		+	+	+	+										
	request help					+	+					+				+		
	save/download	+		+			+	+				+						
	login/logout				+	+						+						
learner /teacher	id	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
	country/school							+			+							
	gender													+				+
	language							+										
	knowledge level				+							+			+	+	+	+
	interest			+	+			+										
	goals/tasks			+														
time	timestamp	+		+	+	+	+		+	+	+	+	+	+	+	+	+	+
	time interval								+	+		+	+	+	+	+	+	+
computing	hardware																	
	software								+		+							
	network																	
resource	general	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
	technical		+				+	+		+								
	educational						+	+				+	+	+	+	+	+	
	annotation	+	+		+	+	+	+										
relation	compositional			+						+		+	+	+	+	+	+	
	social			+						+						+		
result	error message										+					+		
	result /value/grade	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+