République Algérienne Démocratique et Populaire
الجمهورية الجزائرية الديمقراطية الشعبية

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Chadli Bendjedid
Faculté des Sciences et de la Technologie
Département d'Informatique

وزارة التعليم العالي والبحث العلمي
جامعة الشاذلي بن جديد
كلية العلوم و التكنولوجيا
قسم الاعلام الالي

جامعة الشاذلي بن جديد
UNIVERSITE CHADLI BENDJEDID

MEMOIRE

Présenté par

Boudjema Hamza

Pour l'obtention du diplôme de

MASTER

Filière : Informatique

Spécialité : Systèmes Informatiques Intelligents

Thème

Conception et réalisation d'un Système de Recommandation pour les Services Web

Soutenue le : octobre /2020

Devant le Jury composé de :

Présidente	Mme Ferdeneche Ahlem	MCB	Chadli Bendjedid El-Tarf
Rapporteur	**Mme. Maâtallah Majda**	**MCB**	**Chadli Bendjedid El-Tarf**
Examinateur	Mr. Benmachiche Abdelmadjid	MCB	Chadli Bendjedid El-Tarf

Année Universitaire : 2019/2020

Remerciements

On remercie le bon dieu, qui nous a donné la force, la volonté et le

Courage pour terminer ce modeste travail.

J'exprime toute ma gratitude à ma directrice de mémoire madame

Dr. Maàtallah Majda, pour son encadrement,

Merci pour sa disponibilité indéfectible, son aide ainsi que ses remarques judicieuses qui m'ont permis de faire progresser ainsi que d'enrichir mon travail.

Nous remerciement s'adressent à Mme et Mrs les jurys pour l'intérêt qu'ils ont

Porté à ce travail en acceptant d'être examinateurs.

C'est un énorme remerciement qu'on s'adresse à nos parents et nos

Sœurs et frères et nos amis.

Merci à tout le corps professoral et administratif de l'université Chadli Bendjedid El-Tarf

Dédicace

Je dédie ce travail a ceux qui m'ont encouragé et soutenu

A mes chers parents ;

A mes sœurs et frères ;

à ma directrice de mémoire

A tous le corps professoral et administratif de l'université Chadli Bendjedid El-Tarf

A tous ceux qui m'aiment et à tous ceux que j'aime.

A tous ceux que je connais de près ou de loin.

Table des matières

Liste des figures

Liste des tableaux

Liste des Acronymes

SR	Système de Recommandation
FC	Filtrage Collaboratif
FBC	Filtrage basé contenu
SW	Service Web
GRA-RECWS	Graphe-based Recommender of Web Services

Introduction Générale

Les systèmes d'information modernes sur Internet sont des composants logiciels intégrés de services pour la prise en charge de l'interaction interopérable de machine à machine sur un réseau. Ces dernières années, les services Web ont été largement utilisés pour créer des applications orientées services.

Avec ce développement rapide des technologies basées sur les services Web, de grandes quantités de services Web sont disponibles sur Internet, et qui augmentent régulièrement sur Internet. Cependant, cette prolifération rend difficile pour un utilisateur de sélectionner un service Web approprié parmi un grand nombre de services candidats. Une sélection de service inappropriée peut causer de nombreux problèmes aux applications résultantes (par exemple des performances inadaptées et manque de nouveauté).

Afin d'aider les utilisateurs à découvrir des services qu'ils ne connaissent pas et qui répondent à leur besoins, les Systèmes de Recommandation ont été adoptés ces dernière années pour recommander des services Web avec une meilleure valeur de qualité de service QoS (quality of service), dont la façon de recommander ces services à reçu beaucoup d'attention.

Dans ce même contexte et afin de générer dynamiquement de nouveaux services web à l'utilisateur, nous proposons dans ce mémoire un système de recommandation de services web basé sur le contenu. La liste de recommandation générée par notre système est construite en sélectionnant les Top-N services similaires au profil de l'utilisateur, càd les services qu'il a déjà évalués (évaluation de leur qualité de service) ou bien qui répondent à ses besoins, en plus d'un petit ensemble de services dissimilaires et *nouveaux* pour lui, afin de lui donner plus de chance à découvrir ces nouveaux services de façon automatique.

Afin de représenter l'ensemble de nos services web, nous avons utilisé la notion de graphe, où les nœuds représentent les services et les arcs représentent les similarités entre eux. Cela nous permet de sélectionner les services dissimilaires au profil de l'utilisateur mais qui sont pertinents pour lui, en parcourant le graphe et en cherchant les plus longs chemins possibles à partir du profil de l'utilisateur, càd en sélectionnant les voisins les plus lointains des voisins des services constituants le profil de l'utilisateur. Pour ce faire, nous avons utilisé l'algorithme de *Dijikstra* dans sa version inverse pour la recherche du plus long chemin.

Le mémoire est organisé en trois chapitres comme suit :

Le premier chapitre est composé de trois grandes parties, dont la première partie contient une définition des services web, leur architecture et les technologies associées. Dans la deuxième partie, nous présentons la théorie des systèmes de recommandation, leurs origines, les concepts de base et notions liées, leur formalisation, les différentes approches et problèmes liés, en terminant par la présentation des métriques d'évaluation de ces systèmes. Dans la troisième partie, nous avons cité quelques travaux connexes de la littérature qui ont utilisé des systèmes de recommandation de services web afin d'en se positionner par rapport à eux.

Le deuxième chapitre est consacré à la conception de notre système de recommandation de services web basé sur la notion de graphe. Nous mentionnant tout d'abord, la problématique posée et nos objectifs visés. Ensuite, nous présentons la description de notre système avec une architecture qui explique le processus général de notre système. Ensuite, nous expliquons en détail le fonctionnement de notre système et l'approche proposée.

Le troisième chapitre présente l'élaboration de la méthode proposée au sein d'un système de recommandation de films, en l'absence de *Datasets* des services web disponibles en ligne, en utilisant la base de données *MovieLens,* en vue de prouver l'efficacité de notre approche. Les performances sont évaluées qualitativement et quantitativement avec différents tests et comparaisons.

A l'issue des études présentées, nous concluons avec un bilan de nos propositions, en ouvrant la voie vers certaines perspectives envisagées pour accomplir les lacunes de l'approche proposée et améliorer sa performance.

Chapitre 1:

Les Systèmes de Recommandation et Les Services Web

Chapitre 1 : Les Systèmes de Recommandation et les Services Web

Partie 1. Les Systèmes de Recommandation

1.1. Introduction

Avec l'avènement d'internet, nous assistons aujourd'hui à une surcharge d'information. Par exemple, une personne qui désire lire un cours se retrouve devant un volume très grand de propositions de cours, ce qui rend la tâche de choix d'un cours très difficile. Les systèmes de recommandation sont apparus pour remédier à ce problème.

Dans ce chapitre, nous commençons par définir ce qu'un système de recommandation. En suite, nous présentons les trois approches de recommandation les plus utilisées dans la littérature, et nous présentons les différentes mesures de similarité qui permettent aux systèmes de recommandation de faire l'appariement entre les concepts. Enfin, nous terminons en citant les avantages et inconvénients de ces systèmes ainsi que quelques principaux travaux dans le domaine.

1.2. Historique

Les systèmes de recommandation ont été utilisés afin de faire face au problème de surcharge et de richesse d'informations disponibles notamment à travers le Web ou les e-services. Les systèmes de recommandation visent à proposer à un utilisateur actif une ou des recommandations d'items susceptibles de l'intéresser. Ces recommandations peuvent concerner un article à lire, un livre à commander, un film à regarder, un restaurant à choisir, etc.

« Information Lens System » peut être considérée comme le premier système de recommandation. A l'époque, l'approche la plus commune pour le problème de partage d'informations dans l'environnement de messagerie électronique était la liste de recommandation basée sur les intérêts de groupes.

Quelques années plus tard, un certain nombre de systèmes de recommandation académiques ont vu le jour en 1994 et en 1995, tels que le système de recommandation d'articles d'actualités et de films développé par "GroupLens" et le système de recommandation de musique "Ringo" proposé par [1]. Ces deux systèmes sont également basés sur le filtrage collaboratif, des livres , des vidéos, des films, des pages Web, des articles de nouvelles Usenet et des liens Internet.

1.3. Définition des Systèmes de Recommandation :

Les systèmes de filtrage ou les systèmes de recommandation peuvent être définis de plusieurs façons, vue la diversité des classifications proposées pour ces systèmes, mais il existe une définition générale de Robin Burke [2] qui les définit comme suit :

"*Des systèmes capable de fournir des recommandations personnalisées permettant de guider l'utilisateur vers des ressources intéressantes et utiles au sein d'un espace de données important*".

Un système de filtrage d'information, ou un système de recommandation (Recommander System), est un filtre de flux d'information entrant de façon personnalisée pour chaque utilisateur. Autrement dit, dans le but de personnaliser la recherche d'information dans un domaine d'application particulier, un système de filtrage collecte, sélectionne, classifie et suggère à l'utilisateur les informations qui répondent vraisemblablement à ses intérêts à long termes.

Les deux entités de base qui apparaissent dans tous les systèmes de recommandations sont l'utilisateur et l'item.

- L'«utilisateur» est la personne qui utilise un système de recommandation, donne son opinion sur diverses items et reçoit les nouvelles recommandations du système.

- L'« Item » est le terme général utilisé pour designer ce que le système recommande aux usagers.

Les données d'entrée pour un système de recommandation dépendent du type de l'algorithme du filtrage employé. Généralement, elles appartiennent à l'une des catégories suivantes :

- ***Les estimations*** :(également appelées les votes), expriment l'opinion des utilisateurs sur les items (exemple : 1 mauvais à 5 excellent).
- ***Les données démographiques*** : se réfèrent à des informations telles que l'âge, le sexe, le pays et l'éducation des utilisateurs. Ce type de données est généralement difficile à obtenir et est normalement collecté explicitement.
- ***Les données de contenu*** : qui sont fondées sur une analyse textuelle des documents liés aux items évalués par l'utilisateur. Les caractéristiques extraites de cette analyse sont utilisées comme entrées dans l'algorithme du filtrage afin d'en déduire un profil de l'utilisateur. [3].

Pour réaliser le filtrage, le système de recommandation (SR) utilise les profils représentant des préférences relativement stables des utilisateurs pour calculer des recommandations. Ce calcul se

fait par la prédiction des scores qu'un utilisateur est susceptible d'attribuer aux contenus. Le SR adapte ce profil au cours du temps en exploitant au mieux le retour de pertinence que les utilisateurs fournissent sur les informations (documents) reçues.

1.4. Classification des systèmes de recommandation

Il existe plusieurs classifications des systèmes de recommandations:

- **La classification classique :** cette classification est reconnue par trois types de filtrage : le filtrage collaboratif(CF), le filtrage basé sur le contenu(CBF) et le filtrage hybride.
  - Le Filtrage Basé sur le Contenu compare les nouveaux documents au profil de l'utilisateur, et recommande ceux qui sont les plus proches.
  - Le Filtrage Collaboratif compare les utilisateurs entre eux sur la base de leurs jugements passés pour créer des communautés, et chaque utilisateur reçoit les documents jugés pertinents par sa communauté.
  - Le Filtrage Hybride combine le filtrage basé sur le contenu et le filtrage collaboratif pour exploiter au mieux les avantages de chacun.
- **La Classification de [4]** : elle est utilisée dans les systèmes de collaboration. Ils proposent une sous-classification qui comprend les techniques hybrides et les classe dans les méthodes de collaboration hybrides. [4] classent le filtrage collaboratif en trios catégories :
  - ***Approches du FC à base de mémoire*** : pour les K-plus proches voisins.
  - ***Approches du FC basé sur un modèle :*** englobent une variété de techniques telles que: le *clustering*, les réseaux bayésiens, la factorisation de matrices, les processus de décision de Markov.
  - ***Le FC hybride*** : qui combine une technique de recommandation du FC avec un ou plusieurs autres méthodes.
- **La classification de [5] :** c'est une classification en fonction de la source d'information utilisée.

Pour tous les systèmes de recommandation développés jusqu'à nos jours, la collecte de données relatives aux utilisateurs et/ou aux items, représente une phase clé dans le processus de personnalisation. La sous-section qui suit décrit les techniques utilisées par les systèmes de recommandation.

1.5. Les approches de Recommandation :

On s'intéresse dans cette section à la première classification (vue dans la section précédente), et qui propose trois grandes approches du filtrage : Le filtrage basé sur le contenu, Le filtrage collaboratif et le filtrage hybride.

1.5.1. La Recommandation basé sur le contenu (Content-based Filtering)

1.5.1.1. Définition

Le filtrage basé sur le contenu, qui est une évolution générale des études sur le filtrage d'information, s'appuie sur le contenu des documents (thèmes abordés) pour les comparer à un profil lui-même constitué de thèmes. Chaque utilisateur du système possède alors un profil qui décrit ses centres d'intérêts. Par exemple, le profil peut contenir une liste de thèmes ou préférences que l'utilisateur aime bien ou qu'il n'aime pas. Lors de l'arrivée d'un nouveau document, le système compare le descriptif du document avec le profil de l'utilisateur pour prédire l'utilité de ce document pour cet utilisateur.

L'avantage des systèmes de filtrage basés contenu est qu'ils permettent d'associer des documents à un profil utilisateur. Notamment, en utilisant des techniques d'indexation et d'intelligence artificielle. L'utilisateur est indépendant des autres ce qui lui permet d'avoir des recommandations même s'il est le seul utilisateur du système. Par exemple, Afin de recommander des films à un utilisateur, le système analyse les corrélations entre ces films et les films consultés antérieurement par cet utilisateur. Ces corrélations sont évaluées en considérant des attributs comme le titre et le genre. De ce fait, parmi ces films, ceux qui seront recommandés à l'utilisateur, sont les plus similaires (en termes d'attribut) aux films consultés par cet utilisateur.

Cependant, ce type de systèmes présente certaines limitations :

• *L'effet "entonnoir*" : les besoins de l'utilisateur sont de plus en plus spécifiques, ce qui l'empêche d'avoir une diversité de sujets. Même pire, un nouvel axe de recherche dans un domaine bien précis peut ne pas être pris en compte car il ne fait pas parti du profil explicite de l'utilisateur.

• Filtrage basé sur le critère thématique uniquement, absence d'autres facteurs comme la qualité scientifique, le public visé, l'intérêt porté par l'utilisateur, etc.

• Les difficultés à recommander des documents multimédia (images, vidéos, etc.) et ceci à cause de la difficulté à indexer ce type de documents, c'est en fait la même problématique dont souffrent les systèmes de recherche.

• *Problème de démarrage à froid* : Un nouvel utilisateur du système éprouve des difficultés à exprimer son profil en spécifiant des thèmes qui l'intéressent. Ceci malgré les techniques d'apprentissage où l'utilisateur fournit des textes exemples.

1.5.2.1. Descripteur d'article et profil utilisateur

L'algorithme de filtrage basé sur le contenu peut réaliser le *matching* entre un descripteur de contenu (comme par exemple, documents, livre, etc.) et un profil utilisateur et détermine le degré de pertinence de chaque article (ou contenu) pour les utilisateurs potentiels. Si de nombreux articles s'accumulent dans un certain laps de temps, l'algorithme de filtrage de contenu peut ordonner les articles en fonction de leur pertinence pour chacun des utilisateurs potentiels.

*Représentation des contenus - le descripteur d'articl*e : Un descripteur d'article se compose d'un ensemble de concept qui peuvent être représentés par une ontologie de domaine. Les concepts qui représentent un élément sont les plus spécialisés dans une branche de la hiérarchie. De toute évidence, un article peut être représenté avec de nombreux concepts de l'ontologie, chaque concept peut apparaître dans n'importe quelle branche de la hiérarchie de l'ontologie et à tout niveau cela dépend du contenu réel de cet article. Il est à noter que le profil peut inclure des concepts frères, c'est-à-dire les fils d'un même concept.

Représentation des utilisateurs - le profil d'utilisateur : Un profil utilisateur basé sur le contenu se compose d'une liste pondérée de concepts de l'ontologie, représentant ses préférences (ses intérêts). De toute évidence, le profil de l'utilisateur peut comporter de nombreux concepts de l'ontologie, chacun figurant dans les différentes branches et différents niveaux de la hiérarchie. Par exemple, le profil de l'utilisateur peut inclure uniquement « sport », ou « sport » et « football », ou « football » et « basketball », ou tous les trois - en plus de nombreux autres concepts. Cela signifie qu'un certain concept dans un descripteur d'article peut être comparé avec plus d'un concept équivalent dans le profil de l'utilisateur.

Similarité entre un descripteur d'article et un profil utilisateur : Un descripteur d'article et un profil utilisateur sont semblables à un certain degré si leurs profils comprennent des concepts communs ou des concepts relatifs, c'est-à-dire des concepts ayant une sorte de relation père-fils. Un descripteur d'article et un profil utilisateur peuvent avoir de nombreux concepts communs ou relatifs; de toute évidence, plus les concepts sont communs ou relatifs, plus forte est leur similitude. Par exemple, si le profil de l'utilisateur inclut « football » et « sport », ce profil est similaire (à un certain degré) à un article qui comprend ces deux concepts, mais il est moins semblable à un article incluant juste « sport », et il est plus semblable à un article, comprenant « sport » et « football ».

1.5.2. Recommandation basé sur le Filtrage Collaboratif (*Collaborative Filtering CF)*

1.5.2.1.. Définition

Le filtrage collaboratif a pour principe d'exploiter les évaluations faites par des utilisateurs sur certains documents (contenus), afin de recommander ces mêmes documents à d'autres utilisateurs, et sans qu'il soit nécessaire d'analyser le contenu des documents.

Tous les utilisateurs du système de filtrage collaboratif peuvent tirer profit des évaluations des autres en recevant des recommandations pour lesquelles les utilisateurs les plus proches ont émis un jugement de valeur favorable, et cela sans que le système dispose d'un processus d'extraction du contenu des documents. Grace à son indépendance vis-à-vis de la représentation des données, cette technique peut s'appliquer dans les contextes où le contenu est soit indisponible, soit difficile à analyser, et en particulier elle peut s'utiliser pour tout type de données : texte, image, audio et vidéo.

Un autre avantage du filtrage collaboratif est que les jugements de valeur des utilisateurs intègrent non seulement la dimension thématique mais aussi d'autres facteurs relatifs à la qualité des documents tels que la diversité, la nouveauté, etc.

Le CF souffre de plusieurs gros problèmes, où le problème principal étant le démarrage à froid : c'est le fait qu'un utilisateur doit voter sur beaucoup d'objet avant d'obtenir les recommandations.

1.5.2.2. Processus du Filtrage Collaboratif :

Le processus du filtrage collaboratif suit les étapes suivantes:

- *Evaluation des recommandations* : soit explicitement ou récupérées par le système implicitement.
- *Formation des communautés des utilisateurs*: en calculant la similarité entre eux pour en extraire les utilisateurs les plus proches.
- *Production des recommandations:* Dans ce dernier processus, une fois la communauté de l'utilisateur créée, le système prédit l'intérêt qu'un document particulier peut présenter pour l'utilisateur en s'appuyant sur les évaluations que les membres de la communauté ont faites sur ce même document. Lorsque l'intérêt prédit dépasse un certain seuil, le système recommande le document à l'utilisateur.

1.5.3. Le Filtrage hybride

Constatant les avantages et inconvénients de chacune des deux approches ci-dessus, on comprend que de nombreux systèmes reposent sur leur combinaison, ce qui en fait des systèmes de filtrage dits « hybrides ». En général, l'hybridation s'effectue en deux phases :

(i) appliquer séparément le filtrage collaboratif et autres techniques de filtrage pour générer des recommandations candidates, et

(ii) combiner ces ensembles de recommandations préliminaires selon certaines méthodes telles que la pondération, la mixtion, la cascade, la commutation, etc., afin de produire les recommandations finales pour les utilisateurs.

Selon Burke, on peut distinguer sept façons pour combiner les méthodes traditionnelles: Pondération (Weighted), Commutation (Switching), Technique mixte (Mixed), Combinaison de caractéristiques (Features combination), Cascade, Augmentation de caractéristiques (Feature augmentation), Méta niveau (Meta-level).

1.6. Les mesures de similarité dans le Systèmes de Recommandation :

Comme nous l'avons déjà mentionné précédemment, dans la plupart des systèmes de recommandation, on trouve un opérateur de *matching* qui mesure la similarité de deux profils utilisateurs, de deux descripteurs de contenu ou bien la similarité entre un profil utilisateur et un descripteur de contenu.

Nous présentons dans cette section, une classification des mesures de similarité, où on distingue quatre grandes catégories de mesures.

1.6.1. Approches basées sur l'espace vectoriel

Dans le domaine de la recherche d'information, les modèles de l'espace vectoriel sont largement adoptés, on parlera alors de similarité numérique. Ces approches utilisent un vecteur caractéristique, dans un espace dimensionnel, pour représenter chaque objet et calculent la similarité numérique en se basant sur la mesure cosinus ou la corrélation de Pearson. Parmi les approches citées dans la littérature on peut citer :

1.6.1.1. Similarité Cosinus

Cette mesure utilise la représentation vectorielle complète, c'est-à-dire la fréquence des objets (mots). Deux objets (documents) sont similaires si leurs vecteurs sont confondus. Si deux objets ne sont pas similaires, leurs vecteurs forment un angle (X, Y) dont le cosinus représente la valeur de la similarité. La formule est définie par le rapport du produit scalaire des vecteurs x et y et le produit de la norme de x et de y.

$$sim(Si, Sj) = \cos(\mathrm{Si}, \mathrm{Sj}) = \frac{Si*Sj}{\|.Si\|_2*\|.Sj\|_2} \quad (1.1)$$

La mesure de Cosinus [6] quantifie donc la similarité numérique entre les deux vecteurs, comme le cosinus de l'angle entre les deux vecteurs.

1.6.1.2. Similarité de Pearson

La mesure de similarité de Pearson est basée sur le calcul de la corrélation. Pour connaître le coefficient de corrélation liant deux séries X(x1, x2,…,xn) et Y(y1, y2, …,yn), on applique la formule suivante :

$$sim(x, y) = rp = \frac{\sum_{i=1}^{N}(xi-x).(yi-y)}{\sum_{i=1}^{N}(xi-x)2.\sum_{i=1}^{N}((yi-y)2} \quad (1.2)$$

Si r vaut 0, les deux courbes ne sont pas corrélées et donc ne sont pas similaire. Les deux courbes sont d'autant mieux corrélées que r est loin de 0 (proche de -1 ou 1). Avec:

$$X = \frac{1}{N}\sum_{i=1}^{N} xi \quad (1.3)$$

est la moyenne de X

$$Y = \frac{1}{N}\sum_{i=1}^{N} yi \quad (1.4)$$

est la moyenne de Y.

1.6.2. Approches basées sur les arcs

La mesure de similarité la plus intuitive des objets dans une ontologie est leurs distances. Cette similarité est évaluée par la distance qui sépare les objets de l'ontologie. Ces mesures servent de la structure hiérarchique de l'ontologie pour déterminer la similarité sémantique entre les concepts. Le calcul des distances dans l'ontologie est basé sur un graphe de spécialisation des objets. Parmi les travaux classifiés sous cette catégorie on peut citer :

1.6.2.1. Mesure de Wu & Palmer (1994)

La mesure de similarité de Wu et Palmer [7] est basée sur le principe suivant :

Etant donnée une ontologie formée d'un ensemble de nœud et un nœud racine R. Soit X et Y deux éléments de l'ontologie dont nous allons calculer la similarité. Le principe de calcul de similarité est basé sur les distances (N1 et N2) qui séparent les nœuds X et Y du nœud racine et la distance qui sépare le concept subsumant (CS) de X et de Y du nœud R.

La mesure de Wu et Palmer est définie par la formule suivante :

$$som(X, Y) = \frac{2*N}{N1+N2} \quad (1.5)$$

1.6.2.2. Mesure de Rada et al (1989)

Cette mesure est adoptée dans un réseau sémantique et elle est fondée sur le fait qu'on peut calculer la similarité en se basant sur les liens hiérarchiques «is-a». Pour calculer la similarité de concepts dans une ontologie, on doit calculer le nombre des arcs minimums qui les séparent. Cette mesure est basée sur le calcul de distance entre les nœuds par le chemin le plus court, présente un moyen des plus évidents pour évaluer la similarité sémantique dans une ontologie hiérarchique.

La métrique *distance(c1, c2)* indique le nombre d'arcs minimum à parcourir pour aller d'un concept c1 à un concept c2. [8]

1.6.3. Approches basées sur les nœuds :

Ces approches adoptent une nouvelle mesure en termes de la mesure entropique (Contenu informationnel) de la théorie de l'information. La probabilité P pour l'identification de l'utilisateur d'une classe ou de ses descendants dans un corpus désigne l'information de la classe.

On définit l'entropie d'une classe par la formule suivante :

$$E(c) = \log(Pc)) \tag{1.6}$$

Où P est la probabilité de trouver une instance du concept c. La probabilité d'un concept c est calculée en divisant le nombre des instances de c par nombre total des instances.

En associant les probabilités aux concepts d'une taxonomie, il est possible d'éviter le manque de fiabilité des distances des arcs. Parmi les travaux, recensés dans la littérature, sous cette catégorie on peut citer :

1.6.3.1. Resnik (1999)

Resnik [9] définit la similarité sémantique entre deux concepts par la quantité d'information qu'ils partagent. Cette information partagée est égale au contenu informationnel (CI) du plus petit généralisant (PPG) (Plus Petit Généralisant est le concept le plus spécifique qui subsume les deux concepts dans l'ontologie). Par exemple, Dans la figure 10, le PPG des concepts X et Y est le concept CS.

$$sim(c1, c2) = Max[C1\big(PPG(c1, c2)\big)] \tag{1.7}$$

$$C1 = \log(P(c)) \tag{1.8}$$

1.6.3.2. Mesure de Lin (1998)

Lin a défini la similarité de concepts comme le rapport entre la quantité d'informations nécessaires pour indiquer le point commun entre ces deux concepts et les informations nécessaires pour les décrire [10]. Cette mesure est légèrement différente de celle de Resnik :

$$sim(a,b) = \frac{2*C1(PPG(a,b))}{C1(a)+C1(b)} \quad (1.9)$$

1.6.4. Approches hybrides

Ces approches sont fondées sur un modèle qui combine entre les approches basées sur les arcs (Distances) en plus du contenu informationnel qui est considéré comme facteur de décision.

1.6.4.1. Mesure de Jiang et Conrath (1997)

Pour remédier au problème présenté au niveau de la mesure de Resnik, Jiang et Conrath ont apporté une nouvelle formule qui consiste à combiner l'entropie (contenu informationnel) du concept spécifique à ceux des concepts dont on cherche la similarité (combine entre les techniques basées sur les arcs et les techniques basées sur les nœuds qui consistent à compter les arcs afin d'améliorer les résultats par les calculs basés sur les nœuds. Notons que cette formule est définie par l'inverse de la distance sémantique. [11]

$$Sim(c1,c2) = \frac{1}{distance(c1,c2)} \quad (1.10)$$

Sachant que la distance entre c1 et c2 est calculée par la formule suivante :

$$Distance(c1,c2) = C1(c1) + C1(c2) - (2.C1)(PPg(c1,c2))) \quad (1.11)$$

1.6.4.2. Mesure de Leacock et Chodorow (1998)

Une autre méthode présentée combine la méthode de comptage des arcs et la méthode du contenu informationnel. La mesure proposée par Leacock et Chodorow est basée sur la longueur du plus court chemin entre deux synsets de Wordnet. Les auteurs ont limité leur attention à des liens hiérarchiques «is-a» ainsi que la longueur de chemin par la profondeur globale de la taxonomie. La formule est définie par :

$$sim(X,Y) = \log\frac{CD(X,Y)}{2*M} \quad (1.12)$$

Où M est la longueur du chemin le plus long qui sépare le concept racine, de l'ontologie, du concept le plus en bas (la profondeur globale). On dénote par CD(X, Y) la longueur du chemin le plus court qui sépare X de Y. [12]

1.7. Les Problèmes des Systèmes de Recommandation

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

1.7.1. 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 connaitre 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.

1.7.2. 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.

1.7.3. 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.

1.7.4. 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.

1.7.4. 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 couteux. 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.

1.8. Evaluation des Recommandations :

1.8.1. Mesures statistiques de précision :

Les mesures statistiques de précision consistent à évaluer la différence existant entre les notes prédites et les notes réellement attribuées par les utilisateurs.

La MAE (Mean Absolute Error) est la mesure de précision la plus célèbre pour l'évaluation des systèmes de recommandation.

La MAE mesure la moyenne d'erreur absolue entre les notes prédites et les notes réelles des utilisateurs, en rapport avec le nombre total d'items présents dans le corpus test. Plus la valeur de MAE est faible, plus les prédictions sont précises et le système de recommandation est performant.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|xi - x| \quad (1.13)$$

| x i - x | = les erreurs absolues.

N : le nombre d'erreurs.

1.8.2. Mesures permettant l'aide à la décision :

Les mesures permettant l'aide à la décision consistent à évaluer jusqu'à quel point le système de recommandation peut recommander des items susceptibles d'intéresser l'utilisateur. En d'autres termes, ces mesures calculent, dans une liste de recommandation, la proportion d'items qui sont effectivement utiles et pertinents pour l'utilisateur actif afin d'évaluer la pertinence des recommandations.

Pour les besoins d'évaluation en termes d'aide à la décision, les appréciations ou les notes des utilisateurs doivent être transformées dans le cadre d'une échelle binaire ("Aime"ou "Aime pas") afin de distinguer les items pertinents de ceux qui ne le sont pas, pour un utilisateur donné.

Les mesures permettant l'aide à la décision sont principalement issues du domaine de la recherche d'information. Ces mesures sont : la précision, le rappel et la mesure F1.

1.8.2.1. Précision :

La précision représente la possibilité qu'un item sélectionné soit réellement perçu comme étant pertinent par l'utilisateur actif Un item sélectionné représente un item qui est proposé par le système de recommandation à l'utilisateur actif et qui est contenu en même temps dans le corpus test. Les catégories d'items classifiés selon l'intersection entre les listes de recommandation et les évaluations réelles des utilisateurs, la précision est définie par le ratio entre le nombre d'items

pertinents sélectionnés Nps et le nombre d'items sélectionnés par un utilisateur actif Ns. Cette métrique est calculée sur la base de l'équation.

$$\text{précision} = \frac{\text{Nombre de documents pertinents séléctionnés}}{\text{Nombre total de documents séléctionnés}} \quad (1.14)$$

$$\hat{r}_{ui} = \frac{1}{|Ni(u)|}\sum_{v2Ni(u)} r_{vi} \quad (1.15)$$

1.8.2.2. Rappel (Recall)

Le rappel représente la probabilité qu'un item pertinent soit sélectionné par l'utilisateur actif. , le rappel peut etre défini comme étant le rapport entre le nombre d'items pertinents sélectionnés par l'utilisateur "Nps" et le nombre total d'items pertinents disponibles "Np"

$$rappel = \frac{\text{Nombre de documents pertinents séléctionnés}}{\text{Nombre total de documents pertinents}} \quad (1.16)$$

1.8.3. Couverture :

La capacité du système à générer des recommandations est connue sous le nom de « couverture ». En FC basé sur la mémoire, la couverture peut être évaluée par rapport à la capacité du système de recommandation à générer des prédictions pour toutes les notes manquantes au niveau de la matrice de notes "Utilisateur x Item".

Elle peut être également évaluée en prenant en considération uniquement les prédictions contenues dans le corpus test. En effet, L'incapacité d'un système de recommandation exploitant le FC à calculer des recommandations peut notamment être engendrée par le manque de données sur les items et/ou les utilisateurs.

Ainsi, un système de recommandation ne peut être performant que lorsqu'il est susceptible de calculer un nombre suffisant de prédictions concernant un maximum d'utilisateurs. Autrement dit, le système doit être capable de satisfaire les besoins des différents utilisateurs actifs présents dans le système.

1.9. La Nouveauté dans les Systèmes de Recommandation :

La nouveauté est l'une des métriques qui font l'objet d'intérêt ces dernières années pour évaluer la qualité des listes de recommandation. Différentes définitions ont été présentées dans la littérature pour définir le terme « Nouveauté » et qui dépend d'un domaine à l'autre.

Un article nouveau pour un utilisateur est un article qui lui est inconnu. Toutefois, la notion de nouveauté peut prendre des formes particulières selon l'approche. Par exemple, un utilisateur d'un site de streaming de musique à qui le système recommande un morceau qu'il n'a jamais

écouté auparavant peut interpréter cette proposition comme une recommandation nouvelle. En revanche si le morceau proposé provient d'un artiste populaire, la nouveauté est moindre que dans le cas d'un artiste ou d'un genre musical inconnu puisé dans la longue traîne. Il existe donc différentes façons d'aborder le concept de nouveauté.

Une définition simple peut consister à vérifier la présence de l'article dans l'historique de l'utilisateur. La nouveauté prendra alors une valeur binaire.

Il y a aussi la nouveauté par rapport aux articles non populaires. Où, on va chercher en considérant le nombre d'utilisateurs qui ont interagi avec l'article, à créer de la surprise, à proposer une expérience inattendue, en recommandant les articles peu connus.

Une notion proche de cette définition de la nouveauté est la sérendipité. Le système cherche à provoquer pour l'utilisateur une découverte heureuse. Une réaction émotionnelle positive de l'utilisateur est souhaitée dans ce cas. La sérendipité formellement est une recommandation d'un article nouveau, inconnu, pertinent et qui provoque une réaction émotionnelle positive de l'utilisateur.

Plusieurs métriques ont été proposées afin d'injecter et de calculer la nouveauté dans les listes de recommandation **[**13] [14][15][16][17]. Un bon état de l'art sur la nouveauté dans les SRs est présenté dans **[**18].

1.10. Conclusion

Dans cette partie, nous avons d'abord, présenté la notion des systèmes de recommandation, en détaillons les trois approches les plus utilisées, à savoir, l'approche du FBC, FC et hybride. Ensuite, nous avons défini la notion de profil utilisateur. Nous avons présenté les différentes classes de mesures de similarité utilisées par les SRs pour faire l'appariement entre deux profils utilisateur, deux contenus, ou un profil utilisateur et un descripteur de contenu. Enfin, nous avons terminé en citant quelques problèmes rencontrés par les systèmes de recommandation classiques.

Partie 2: Les Services Web

2.1. Introduction :

Les services web sont parmi plusieurs technologies qui permettent de résoudre le problème de communication à travers un réseau. En effet, les services web peuvent constituer un apport de rapidité et d'efficacité. La notion de service web désigne essentiellement une application (un programme) mise à disposition sur Internet par un fournisseur de services, et accessible par des clients à travers des protocoles Internet standards.

2.2. Historique

Selon une histoire, le terme de *service web* a été d'abord prononcé par Bill Gates lors d'une conférence logiciel en 2000, mais le concept de deux ordinateurs communiquant l'un avec l'autre pour partager des informations et parler dans un vocabulaire partagé avec un ensemble défini de protocole n'était finalement pas nouveau. Le concept avait été inventé bien avant qu'internet ait été inventé.

Une des premières mises en œuvre d'architecture de ce style est connue par l'acronyme EDI, (Echange de Données Informatisées). L'EDI a été documenté en 96 par l'Institut national des normes et des technologies comme un système d'échange informatique de messages fortement formatés qui représentent des documents autres que des instructions monétaires. En d'autres termes, l'EDI a été conçu pour dépasser les transactions financières et pour permettre le mouvement de données dans les deux sens sur internet. Il a été à l'origine conçu pour le commerce électronique à grande échelle, mais il a aussi été utilisé pour toutes les sortes de transactions commerciales.

2.3. Définition

Il existe différentes définitions pour les services Web. Certaines sont plus générales que d'autres. D'un point de vue technique, la définition communément admise c'est la définition de W3C[19].

Le W3C définit le service Web comme un système logiciel qui permet l'interaction entre machines sur le réseau à travers des interfaces, et vérifiant les propriétés suivantes :

- Identifié par une URI (Uniform Ressource Identifier).
- Ses interfaces et ses liens peuvent être décrits en XML.
- Sa définition peut être découverte par d'autres services Web.

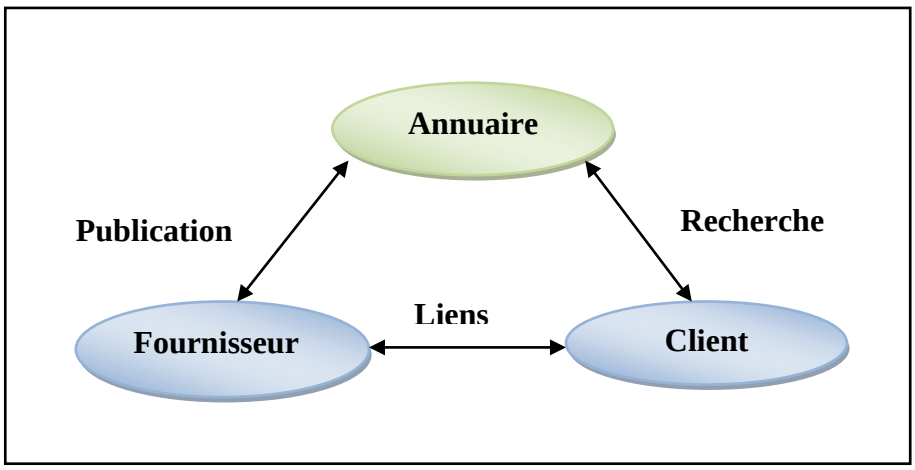
- Il peut interagir directement avec d'autres services Web à travers le langage XML et en utilisant des protocoles Internet.

Les services Web sont donc, des composants logiciels distribués qui exposent un ensemble de fonctionnalités sur un réseau (généralement Internet).

Les services Web sont décrient avec des standards fondés sur XML. Ils sont accessibles via un réseau et à travers des protocoles Internet standards (généralement HTTP2) pour les clients qui les découvrent, les sélectionnent, les invoquent et les utilisent. [19].

2.4. Architecture des Web Services

Les services Web sont définis dans le cadre des architectures orientées services SOA (Service *Oriented Architecture*). L'architecture SOA permet de distinguer le fournisseur de service, l'annuaire de services, et enfin l'utilisateur du service (le client). Ces composants fonctionnent généralement dans un système distribué et n'opèrent pas dans un même environnement de traitement et sont obligés de communiquer par échanges de messages dans le but d'accomplir le résultat souhaite. Cette architecture est illustrée dans la **Figure (1.1)**



1. 1 Architecture des services Web. [19]

L'architecture de référence des services web se base sur les trois concepts suivants :

- ***Le fournisseur de service* :** c'est le propriétaire du service.
- ***Le client (ou le consommateur de service)* :** c'est un demandeur de service. D'un point de vue technique, il est constitué par l'application qui va rechercher et invoquer un service.
- ***L'annuaire des services:*** c'est un registre de descriptions de services offrant des facilités de publication de services à l'intention des fournisseurs ainsi que des facilités de recherche de services à l'intention des clients.

Les interactions de base qui existent entre ces trois éléments sont les opérations de publication, de recherche, d'invocation et de lien. Où, le client du service interroge le registre de services pour obtenir les services disponibles qui correspondent à ses besoins. La découverte d'un service est réalisée grâce à la description du service disponible dans l'annuaire. L'utilisateur de service doit comprendre chaque service en termes de propriétés fonctionnelles et non fonctionnelles afin d'obtenir le résultat approprié.

Après avoir sélectionné le service qu'il veut utiliser, le consommateur du service peut, dans certains cas, négocier auprès du fournisseur les termes suivant lesquels il peut utiliser ce service. A la fin de la négociation, un accord de service est réalisé entre l'utilisateur et le fournisseur. La plupart du temps, cet accord de service contractualise les termes de l'utilisation du service par l'utilisateur sans garantie totale du résultat.

Grâce aux informations disponibles dans la description du service, l'utilisateur de service peut, dès lors, réaliser la liaison et appeler les fonctionnalités du service. [20]

2.5. Fonctionnement des Web services

Le fonctionnement des Web Services repose sur trois couches fondamentales présentées comme suit :

- **Invocation** : établir la communication entre le client et le fournisseur en décrivant la structure des messages échangés.
- **Découverte** : permet de localiser un Web service particulier dans un annuaire de services décrivant les fournisseurs ainsi les services fournis.
- **Description** : Le but est la description des interfaces des Web services (paramètres des fonctions, types de données).

2.6. Les technologies des services web

Les Services Web se basent principalement sur trois protocoles : SOAP, WSDL, UDDI. Dans ce qui suit, nous allons détailler le fonctionnement de chaque protocole :

2.6.1. SOAP (Simple Object Access Protocol)

Le protocole SOAP [21] est un standard proposé par le W3C pour l'échange des données. Le but de SOAP est d'assurer l'interconnexion des Web services en transportant les paquets de données encapsulés sous forme de texte structuré.

Il repose sur deux standards HTTP et XML, respectivement, pour la structure des messages et pour le transport. Le protocole SOAP ne passe pas par des ports précis mais utilise le protocole

HTTP qui lui, permet de traverser les proxys et les pare-feu (firewalls) contrairement à d'autres modes de communications telles que les communications via les sockets et RMI. Une application envoie une requête SOAP à un Web service, et le Web service retourne la réponse dans ce qu'on appelle une réponse SOAP.

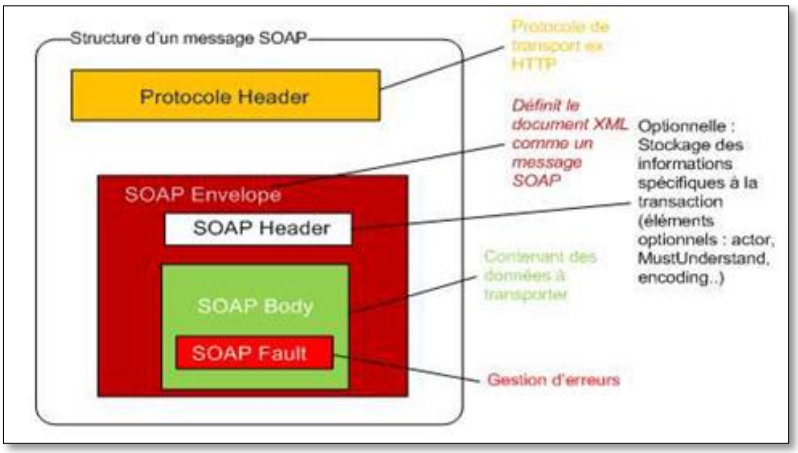
2.6.1.1. Structure d'un message SOAP

Tout d'abord un message SOAP est un document XML qui doit avoir la forme suivante :

- La structure de l'enveloppe SOAP : qui définit une structure générale décrivant le contenu, le destinataire, et la nature du message.
- Les règles d'encodage SOAP *:* qui définissent le mécanisme de sérialisation utilisé pour échanger des objets.

- SOAP RPC : qui définit, pour les utilisations synchrones, une convention de représentation des appels et des réponses des procédures distantes.

Figure (1.2) décrit la structure d'un message SOAP :



1. 2 La structure d'un message SOAP [21]

Un message SOAP est composé des parties suivantes :

a. L'entête HTTP : (HTTP Header)

Le protocole HTTP envoie une requête POST. L'entête HTTP se trouve juste avant le message SOAP, et définit le destinataire du message, les règles d'encodage HTTP, etc.

Le champ SOAP Action peut être utilisé pour indiquer l'intention de la requête SOAP.

Cette information peut être utilisée par un firewall pour filtrer les messages. Ce champ est obligatoire mais peut être vide si on n'indique pas l'intention de la requête.

b. L'enveloppe SOAP

L'enveloppe contient l'espace de nommage définissant la version de SOAP utilisée, et les règles de sérialisation, et d'encodage.

c. L'entête SOAP: (header SOAP)

Cette partie du message est optionnelle. Elle sert à transmettre des informations nécessaires pour l'exécution de la requête SOAP aux dictionnaires qui recevront le message. On y précise généralement des informations liées aux transactions, à l'authentification, etc. L'entête est composé d'un ou plusieurs champs :

- **l'attribut actor (Acteur):** désignant le destinataire du header,
- **l'attribut mustUnderstand :** qui indique si le processus est optionnel.

L'attribut actor (acteur) permet de préciser l'application à laquelle est destinée l'information contenue dans le header. L'URI http://schemas.xmlsoap.org/soap/actor/next précise en particulier que ces informations sont destinées à la première application qui reçoit le message. Dans le cas où l'acteur n'est pas précisé, le header est analysé par le destinataire final du message.

Dans l'exemple suivant, on précise des informations sur l'identification de l'utilisateur et la transaction à laquelle appartient le message :

```
<SOAP-ENV :Header
SOAP-ENV :actor="http://schemas.xmlsoap.org/soap/actor/next"
SOAP-ENV:mustUnderstand="1">
<identifiant numero="124527"/>
<transaction type="compensees"  numero="YU75X"/>
</SOAP-ENV:Header>
```

1. 3. Exemple d'un entête SOAP

d. Le Corps SOAP

Le corps d'un message SOAP contient toutes les informations que l'on veut transmettre à l'application distante. Le contenu du corps est normalisé dans SOAP RPC, pour modéliser une requête et sa réponse.

- ✓ Le corps de la requête contient l'identifiant de l'objet distant, le nom de la méthode à exécuter et les éventuels paramètres.
- ✓ Le corps de la réponse contient le résultat de l'exécution de la requête.

2.6.2. WSDL (Web Service Description Language)

Le WSDL (Web Services Description Language) [22] est un langage basé sur XML, créé dans le but de fournir une description unifiée des services web. Il se présente comme un standard actuel dans ce domaine, et il est, de plus, normalisé par le W3C. Il est issu d'une fusion des langages de descriptions NASSL (Network Accessibility Service Specification Language - IBM) et SCL (Service Conversation Langage - Microsoft).

Son objectif principal est de séparer la description abstraite du service de son implémentation, et il permet de fournir les spécifications nécessaires à l'utilisation d'un Service web en décrivant les méthodes, les paramètres et les types de retour.

Ce langage permet de décrire de façon précise les services Web, en incluant des détails tels que les protocoles, les serveurs, les ports utilisés, les opérations pouvant être effectuées, et les formats des messages d'entrée et de sortie.

2.6.2.1. Structure d'un document WSDL

Le langage de description WSDL se comporte comme un langage permettant de décrire l'interface visible (ou publiée) du service web. Il décrit, à l'aide du langage de balises XML, les différents éléments du service. Une description WSDL d'un service web est faite sur deux niveaux, un niveau abstrait et un niveau concret.

a) *Niveau abstrait :* la description du service web consiste à définir les éléments de l'interface du service web tel que les types de données (Data Types), les messages (Message), les types de ports (Port Type). Ces parties décrivent des informations abstraites indépendantes au contexte de mise en œuvre. On y trouve les types de données envoyées et reçues, les opérations utilisables et le protocole qui sera utilisé.

- **Data types :** est l'élément qui définit les types de données utilisées dans les messages échangés par le service web.
- **Message :** spécifie les types d'opérations supportées par le service web, il permet d'incorporer une séquence de messages corrélés sans avoir à spécifier les caractéristiques du flux de données.
- **Port Type :** est un groupement logique ou une collection d'opérations supportées par un ou plusieurs protocoles de transport, il est analogique à une définition d'un objet contenant un ensemble de méthodes.
- **Opération :** Décrit les opérations invoquées de manière distante sur le service.

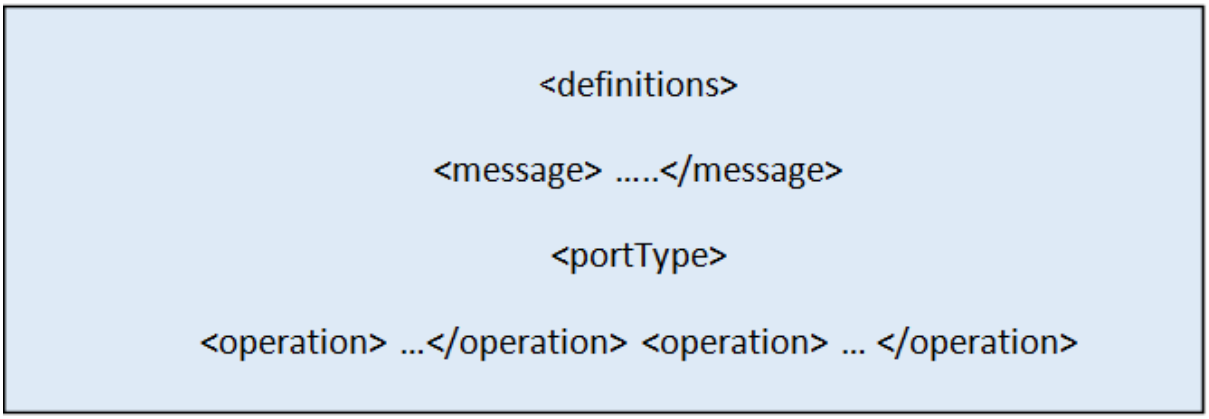
b) *Niveau concret :* le service web est défini grâce aux éléments *Bindings*, *Service* et *Port*. Ces deux derniers décrivent des informations liées à un usage contextuel du service web. On y trouve l'adresse du fournisseur implémentant le service, et le service qui est représenté par les adresses des fournisseurs :

- **Bindings :** Décrit la façon dont un type de port est mis en œuvre pour un protocole particulier (HTTP par exemple), et un mode d'invocation (SOAP par exemple). Cette description est faite par un ensemble donné d'opérations abstraites. Pour un type de port, on peut avoir plusieurs liaisons, pour différencier le mode d'invocation ou de transport de différentes opérations.
- **Port :** Spécifie une adresse URL qui correspond à l'implémentation du service web par un fournisseur et identifie une ou plusieurs liaisons aux protocoles de transport pour un Port Type donné.
- **Service :** Spécifie l'adresse complète du service web, et permet à un point d'accès d'une application distante de choisir à exposer de multiples catégories d'opérations pour divers types d'interactions.
- **<definitions>** : Cet élément contient la définition du service. C'est la racine de tout document WSDL. Cette balise peut contenir les attributs précisant le nom du service, et les espaces de nommage.

  <definitions> contient trois types d'éléments : <message>, <prototype>, <binding> et <service>. <message> et <portType>, ces éléments définissent les opérations offertes par le service, leurs paramètres d'entrée et de sortie, etc. En particulier,

  - Un <message> : correspond à un paramètre d'entrée ou de sortie d'une <operation>.
  - Un <portType> : définit un ensemble d'opérations.
  - Une <operation> : définit un couple message -entrée / message -sortie. Par exemple, dans le monde Java, une opération est une méthode et un portType une interface.
  - <binding> : Cet élément associe les <portType> à un protocole particulier. Les bindings possibles sont SOAP, CORBA ou DCOM. Actuellement seul SOAP est utilisé. Il est possible de définir un binding pour chaque protocole supporté.
  - <service> : Cet élément précise les informations complémentaires nécessaires pour invoquer le service, et en particulier l'URI du destinataire. Un <service> est modélisé comme une collection de ports,
  - Un <port> étant l'association d'un <binding> à un URI.

La **Figure (1.4)** décrit la structure d'un document WSDL :



1. 4. Structure d'un document WSDL

2.6.3. UDDI –(Universal Description Discovery and Integration)

UDDI est une spécification définissant la manière de publier et de découvrir les Services Web sur un réseau. Ainsi, lorsque l'on veut mettre à disposition un nouveau service, on crée un fichier appelé Business Registry qui décrit le service en utilisant un langage dérivé d'XML suivant les spécifications UDDI.

L'UDDI [22] est introduit en 2000 par Ariba, Microsft et IBM. Il à été crée pour faciliter la découverte et la recherche de Web services en plus de leurs publications. Les organisations publient les informations décrivant leurs Web services dans l'annuaire, et l'application client ayant besoin d'un certain service, consulte cet annuaire pour la recherche des informations concernant le Web service qui fournit le service désiré pour une éventuelle interaction.

L'UDDI peut être vu comme un annuaire contenant [22] :

- **Les pages blanches :** noms, adresses, identifiants et contacts des entreprises enregistrées. Ces informations sont décrites dans des entités de type BusinessEntity. Cette description inclus des informations de catégorisation permettant de faire des recherches spécifiques dépendantes du métier de l'entreprise.
- **Les pages jaunes :** donnent les détails sur le métier des entreprises et les services qu'elles proposent. Ces informations sont décrites dans des entités de type BusinessService.
- **Les pages vertes :** contiennent des informations techniques du service offert, la manière d'interagir avec le service, une définition du BusinessProcess et aussi un pointeur vers le fichier WSDL et une clé unique identifiant le service. La structure de données du registre UDDI contient cinq types de données :

Le modèle UDDI comporte 5 types de structures de données décrites sous forme de schéma XML.

- ***Business entity*** : elle contient des informations sur l'entreprise qui publie les services dans l'annuaire, cette structure contient les autres éléments de l'UDDI.
- ***Business service*** : ensemble d'informations sur les services publiés par l'entreprise (nom du service, les catégories…).
- ***Binding Template*** : ensemble d'informations sur le lieu d'hébergement du service (c.à.d. l'adresse du point d'accès du service).
- ***Tmodel :*** ensemble d'informations sur le mode d'accès au service (définitions wsdl), il peut s'agir aussi d'une spécification abstraite ou d'une taxonomie.
- ***Publisher Assertion*** : ensemble d'informations contractuelles entre les partenaires.

Les appels.

2.7. Avantages et inconvénients des Services Web

Les services web ont des avantages et inconvénients :

2.7.1. Avantages

- Les services Web fournissent l'interopérabilité entre divers logiciels fonctionnant sur diverses plates-formes.
- Les services Web utilisent des standards et protocoles ouverts.
- Les protocoles et les formats de données sont au format texte dans la mesure du possible, facilitant ainsi la compréhension du fonctionnement global des échanges.
- Basés sur le protocole HTTP, les services Web peuvent fonctionner au travers de nombreux pare-feux sans nécessiter des changements sur les règles de filtrage.
- Les outils de développement, s'appuyant sur ces standards, permettent la création automatique de programmes utilisant les services Web existants.

2.7.2. Inconvénients

- Les normes de services Web dans certains domaines sont actuellement récentes.
- Les services Web ont de faibles performances par rapport à d'autres approches de l'informatique répartie telles que le RMI, CORBA, ou DCOM.
- En utilisant le protocole HTTP, les services Web peuvent contourner les mesures de sécurité mises en place à travers des pare-feu.

2.8. Conclusion

Un service web n'est qu'une transaction accessible par l'échange de documents XML entre deux URL .Ce dernier offre une réelle souplesse d'utilisation et permet l'accélération du développement d'application.

Les systèmes de recommandation ont fait l'objet d'intérêt pour la suggestion des services web afin de permettre à l'utilisateur de découvrir des services qu'il n'a pas vu encore.

Un petit tour d'horizon sur les systèmes de recommandation est présenté dans la partie suivante.

Partie 3 : La Recommandation et les Services Web

Dans cette partie, nous présentons quelques travaux de la littérature qui ont employé les Systèmes de Recommandation pour les services web.

3.1. Recommandation de Services Web basée sur l'extraction d'information et la réputation symbolique [23] :

Cette méthode de recommandation de services web a identifié deux représentations différentes des services web. La première représentation proposée provient du domaine du traitement du langage naturel et est basée sur des règles pour annoter les descriptions des services et ainsi extraire les informations utiles pour l'indexation sémantique de services.

La seconde méthode proposée, appelée réputation symbolique, est calculée à partir des relations entre les services considérés et est utilisée pour la recommandation de services web.

L'impact de ces représentations pour la découverte et la recommandation est étudié et discuté à la lumière des expérimentations utilisant des services web réels

3.2. Recommandation de service Web basée sur un Filtrage Collaboratif Populaire (Popular-Dependent Collaborative Filtering PDCF) [24]

Une méthode de recommandation de services Web appelée Popular-Dependent Collaborative Filtering (PDCF) est proposée dans [24]. La méthode gère les différences de qualité de services rencontrées par les utilisateurs ainsi que la dépendance entre les utilisateurs sur un service Web spécifique en utilisant le facteur de dépendance utilisateur / service Web.

De plus, le facteur de popularité utilisateur / service Web est pris en compte dans cette méthode, ce qui améliore considérablement son efficacité.

Une autre méthode basée sur la localisation appelée LPDCF, a été aussi proposée, qui prend en considération la localisation des services Web dans le processus de recommandation du PDCF.

Un ensemble d'expériences est mené pour évaluer les performances du PDCF avec deux jeux de données du monde rée Les résultats indiquent que le PDCF surpasse les autres méthodes concurrentes dans la plupart des cas.

3.3. Recommandation de services web basée ontologie [25] :

La représentation de connaissances du domaine considéré sous forme d'ontologies, ou plus largement de bases de connaissances, peut constituer un réel atout pour améliorer les performances d'un système de recommandation automatisé. Pourtant, leur utilisation dans ce contexte reste marginale.

les services sont considérés comme des ressources identifiables par des URIs et les arcs représentent des relations sémantiques entre entités.

3.4. Recommandation de service Web basée sur la qualité de service [26] :

Les auteurs dans [26], ont proposé une méthode pour la prédiction de la valeur du qualité de service (Quality of service QoS) personnalisée pour les utilisateurs de services, en utilisant les expériences utilisateur disponibles des services Web de différents utilisateurs.

Cette approche ne nécessite aucun appel de service Web supplémentaire. Sur la base des valeurs de la qualité de service prédites des services Web, des recommandations de services personnalisées, prenant en charge la QoS, peuvent être produites pour aider les utilisateurs à sélectionner le service optimal parmi ceux fonctionnellement équivalents.

À partir d'un grand nombre de données de qualité de services réelles collectées à partir de différents emplacements, les performances de la qualité des services Web observées par l'utilisateur ont une forte corrélation avec les emplacements des utilisateurs.

Pour améliorer la précision de la prédiction, un système de recommandation de service Web sensible à l'emplacement (nommé LoRec) est proposé également dans [26]. Le système utilise à la fois les valeurs de la qualité du service Web et les emplacements des utilisateurs, pour effectuer une prédiction de la qualité de service personnalisée.

Les utilisateurs de LoRec partagent leur expérience d'utilisation des services Web et, en retour, le système leur fournit des recommandations de service personnalisées. LoRec collecte d'abord les utilisateurs qui ont observé les enregistrements QoS des différents services Web, puis regroupe les utilisateurs qui ont des observations QoS similaires pour générer des recommandations. Les informations de localisation sont également prises en compte lors du regroupement des utilisateurs et des services.

3.5. Méthode basée sur la réputation des utilisateurs [27]

Les systèmes de recommandation sensibles à la confiance sont des approches avancées qui ont été développées sur la base d'informations sociales pour fournir des suggestions pertinentes aux utilisateurs. Le manque d'informations de confiance suffisantes peut réduire l'efficacité de ces méthodes.

Une approche basée sur la réputation est proposée dans [27] pour améliorer les systèmes de recommandation sensibles à la confiance, en améliorant les profils de notation des utilisateurs, dont les notes et les informations de confiance sont insuffisantes.

Les auteurs ont proposé une mesure de fiabilité des utilisateurs pour déterminer l'efficacité des profils de notation et des réseaux de confiance des utilisateurs pour prédire les items invisibles. Puis, un nouveau modèle de réputation des utilisateurs est introduit sur la base de la combinaison des profils de notation et des réseaux de confiance. L'idée principale de la méthode proposée est d'améliorer les profils de notation des utilisateurs qui ont une faible mesure de fiabilité des utilisateurs, en ajoutant un certain nombre de notes virtuelles. À cette fin, le modèle de réputation des utilisateurs proposé est utilisé pour prédire les évaluations virtuelles.

La méthode proposée a pu améliorer certains problèmes des systèmes de recommandation, sachant le démarrage à froid, la rareté des données ainsi que les mesures de diversité, de nouveauté et de fiabilité. Les résultats expérimentaux basés sur trois ensembles de données du monde réel ont montré que la méthode proposée a atteint des performances plus élevées que les autres méthodes de recommandation.

3.6. Récapitulatif des travaux connexes :

Nous avons présenté dans la section précédente quelques travaux de la littérature sur la recommandation de services web, où quelques unes ont utilisé la recommandation de services Web basée sur l'extraction d'information à partir des descriptions textuelles disponibles pour un service web afin d'extraire les termes pertinents, d'autres ont utilisé le facteur de dépendance utilisateur / service Web pour gérer les différences de qualité de services rencontrées par les utilisateurs ainsi que la dépendance entre les utilisateurs sur un service Web spécifique. Les ontologies aussi ont été utilisées pour intégrer la sémantique dans la recommandation afin de pallier le problème de démarrage à froid. Une autre méthode est basée sur la prédiction de la qualité des services pour les services Web. Finalement, la notion de réputation des utilisateurs est aussi incorporée dans les systèmes de recommandation de services web afin d'améliorer les profils de notation des utilisateurs, en ajoutant un certain nombre de notes virtuelles qui seront utilisées pour la prédiction.

A partir des travaux qu'on a présenté, on a constaté que la recommandation est devenue un outil très important pour le domaine des services web et plusieurs applications ont prouvé son efficacité à travers différentes méthodes et stratégies. Cependant, la notion de nouveauté n'est jamais prise en considération dans les systèmes de recommandation de service web, ce qui fait l'objet d'intérêt de notre travail.

4. Conclusion

Dans ce chapitre, nous avons exposé le domaine des systèmes de recommandation, les principales techniques de recommandation et leurs problèmes et les différentes notions et mesures liées. Nous avons également présenté les services web ainsi que leur notion de base et technologies associées. A la fin du chapitre nous avons présenté quelques travaux de recherche sur la recommandation des services web, suivis par un petit récapitulatif.

Ces travaux s'attachent généralement à résoudre un ou plusieurs problèmes parmi les enjeux auxquels doit faire face ce genre de systèmes. Le chapitre suivant présente notre système de recommandation de services web à base de graphe qui vise à augmenter la notion de nouveauté dans les recommandations.

Chapitre 2:

Conception du Système

Chapitre 2 : Conception du Système

2.1. INTRODUCTION :

Dans ce chapitre, nous présentons notre approche de recommandation de services Web à base de graphes. Cette approche permet de recommander des services à un utilisateur de portail de services Web à la recherche d'un service et résoudre le problème de la nouveauté dans les systèmes de recommandation.

Nous commençons d'abord par la problématique qui donne une petite vision des défis existants pour obtenir une solution. Ensuite, nous donnons une définition abstraite du système et son architecture avant de passer aux détails de déroulement du processus de la recommandation et l'algorithme proposé.

2.2. PROBLEMATIQUE :

Les services Web sont des composants logiciels intégrés pour la prise en charge de l'interaction interopérable de machine à machine sur un réseau. Ces dernières années, les services Web ont été largement utilisés pour créer des applications orientées services dans l'industrie et le monde universitaire. Le nombre de services Web accessibles au public augmente régulièrement sur Internet. Sur cette grande augmentation de services web disponibles, la notion de la recommandation des services pertinents aux utilisateurs est survenue.

De plus, l'utilisateur est par nature un être humain qui veut toujours en savoir de plus en plus sur ce qui se passe dans ce monde et ne veut pas rester dans le même cercle. Cependant, la plupart des systèmes de recommandation existants n'offre pas la possibilité de générer des contenus différents des goûts des utilisateurs et nouveaux pour eux, ce qui les rend ennuyeux et cela finira par abandonner le système.

De ce point là, la problématique suivante se pose :

> *Comment créer un système de recommandation de services web capable de prendre en compte les besoins de l'utilisateur et de fournir des services que l'utilisateur n'a jamais vus et n'a pas apprécié encore?*

2.3. MOTIVATIONS ET OBJECTIF :

L'objectif principal de notre travail est la conception d'un système de recommandation de services web basé sur le contenu capable de fournir les besoins d'un utilisateur de façon

automatique et intelligente afin d'aider l'utilisateur à découvrir de nouveaux services qu'il n'a jamais vu mais qui sont pertinents pour lui, et d'atténuer le problème de la nouveauté pour les systèmes de recommandation.

Afin d'atteindre notre objectif, on propose de générer une liste de recommandation qui contient un mélange de services, dont quelques uns sont similaires aux préférences de l'utilisateur et les autres sont dissimilaires mais pertinents pour lui. Cette pertinence est jugée selon le degré de dissimilarité entre les services recherchés par l'utilisateur et ceux proposés par le système en utilisant la notion de graphe pour la représentation des services et la similarité entre eux.

De là, les résultats de la recommandation sont des services qui répondent aux besoins et préférences de l'utilisateur, et des services « nouveaux » pour lui qui sont différents de ses centre d'intérêts mais qui peuvent être intéressants pour lui.

2.4. UN SYSTEME DE RECOMMANDATION DE SW A BASE GRAPHE: GRA-RECWS

Dans cette section, nous allons présenter notre système de recommandation de services web à base de graphe **Gra-RecWS** (*Graphe-based Recmmender of Web Services*)**,** en donnant une petite architecture du système ainsi qu'une description formelle de l'univers de travail. Ensuite, nous expliquons comment représenter les services web avec des graphes et comment calculer la similarité entre les services et l'utilisateur actif. Finalement, nous détaillons le processus de la recommandation proposé.

2.4.1. La description du système :

En vue de répondre à la problématique posée ci-dessus, on propose dans ce travail, un système de recommandation de services web à base de graphe nommé **Gra-RecWS**. L'idée de base derrière la conception de notre système est de construire un système de recommandation capable de suggérer aux utilisateurs des services web pertinents et *nouveaux*, pour leur faciliter la phase de la découverte et leur aider à choisir et utiliser les bons services.

Afin d'assurer la nouveauté dans les listes générées à l'utilisateur, la notion de graphe est utilisée pour représenter l'ensemble des services web, où les nœuds représentent les services et les arcs représentent les similarités entre eux.

La liste de recommandation finale est construite en sélectionnant les Top-N services similaires au profil de l'utilisateur, càd les services qu'il a déjà évalués ou bien qui répondent à ses besoins, en plus d'un petit ensemble de services dissimilaires et nouveaux pour lui, afin de lui donner plus de chance à découvrir ces nouveaux services de façon automatique.

2.4.2. Architecture du système

La **Figure (2.1)** présente l'architecture générale de notre système de recommandation de services web, qui illustre les étapes du processus de la recommandation.

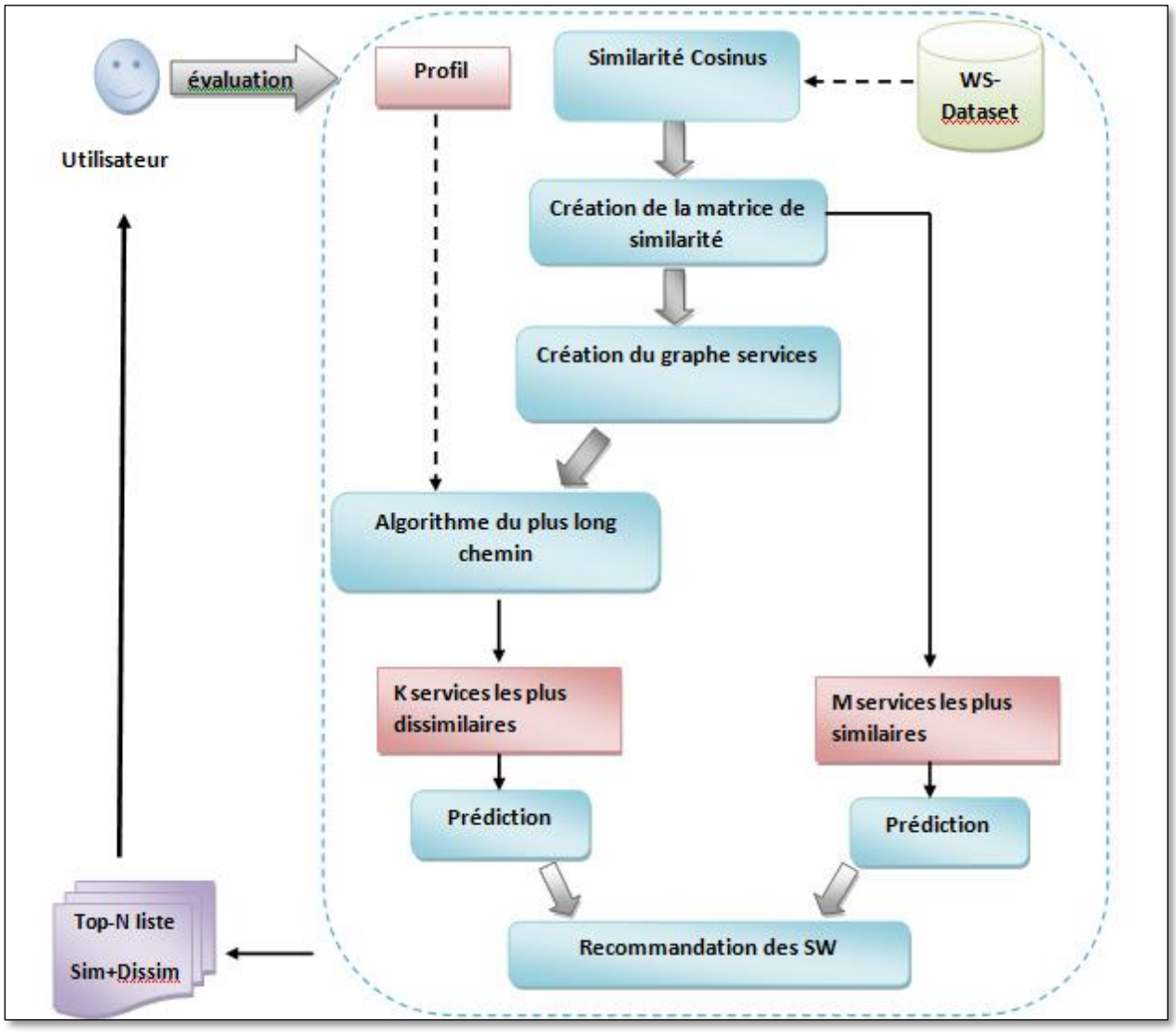


Figure2. 1 Architecture générale de notre système de recommandation

Le processus de fonctionnement commence par l'interaction de l'utilisateur avec le système. Après la consultation des services disponibles dans le système par l'utilisateur, ce dernier donne son avis sur quelques services, ce qui permet de créer son profil dans le système.

Ensuite, le système extrait les données de ses services évalués et commence la construction d'un graphe de services. Où, les nœuds du graphe sont les services web et les arcs sont des chiffres qui représentent la similarité entre eux.

Afin d'assurer la nouveauté dans les listes recommandées, un ensemble de services les plus lointains (les plus dissimilaires) est sélectionné en parcourant le graphe de service, et la prédiction de ces nouveaux services est faite.

Finalement, une liste de recommandation mixte est générée à l'utilisateur, qui contient des services similaires à son choix ainsi que d'autres nouveaux pour lui.

2.4.3. La représentation de l'univers de travail :

Pour détailler notre technique de façon claire et compréhensible, nous avons utilisé un ensemble de symboles universels afin de les conserver dans toutes les étapes d'explication.

Donc, nous envisageons un monde avec :

-Un ensemble d'utilisateurs U = {u,v}

-Un ensemble Services web S = {i,j}

- Ui : ensemble des utilisateurs qui ont apprécié un service i
- Iu : ensemble des services appréciés par un utilisateur u
- Rui : préférence de l'utilisateur u pour un service i
- r : valeur de préférence normalisée ; où r est la note donnée par l'utilisateur u au service i, et qui peut prendre deux formes : Evaluations explicites et évaluations implicites. où, les évaluations explicites sont celles exprimées directement par les utilisateurs, alors que les évaluations implicites sont celles déduites en examinant les services que l'utilisateur a suivis.
- $\hat{r}$: valeur de préférence prédite
- Un ensemble d'évaluations R= (u, i, r) ; qui représente la matrice d'évaluation.
- Chaque service web est décrit par un ensemble de descripteurs D(t) = {d1, d2, ….,dn}.

2.4.4. Processus de la recommandation à base de graphe :

Dans cette section, nous allons présenter notre approche de recommandation de service web proposée. Tout d'abord, nous détaillons comment les services web sont représentés, et comment la similarité entre les services est calculée. Ensuite, nous expliquons la méthode de sélection des services web, ayant quelques uns sont nouveaux pour l'utilisateur et les autres répondent à ses besoins et ses préférences.

2.4.4.1. Le graphe de services :

Dans cette étape, on détermine les relations entre les services. Ces relations seront utilisées pour créer un graphe de services, car dans un système de recommandation à base de contenu, la représentation des items (services) et la relation entre eux joue un rôle crucial afin de sélectionner les items les plus similaires.

Afin de représenter l'ensemble de services ainsi que les relations entre eux dans notre système, on propose d'utiliser une représentation sous forme de graphe, où les services constituent les nœuds et les arcs représentent la similarité entre eux.

Formellement, le graphe des services web noté **GSW(S, A)** est composé de :

- Un ensemble de nœuds S avec *S = (S1, S2, ..., Sn)* et
- Un ensemble d'arcs A avec A = {R(Si , Sj) ∈ S x S, les relations entre les services}.
- Deux services *Si* et *Sj* sont liés si et seulement si la similarité *SimSW(Si, Sj)* entre eux est supérieure à 0.

Donc, on obtient un graphe hétérogène qui contient deux types de nœuds : catégorie et services. Où, on peut distinguer des services similaires et d'autres non similaires (ou un peu similaires), sachant que chaque service peut appartenir à une ou plusieurs catégories comme le montre la **Figure (2.2)** suivante :

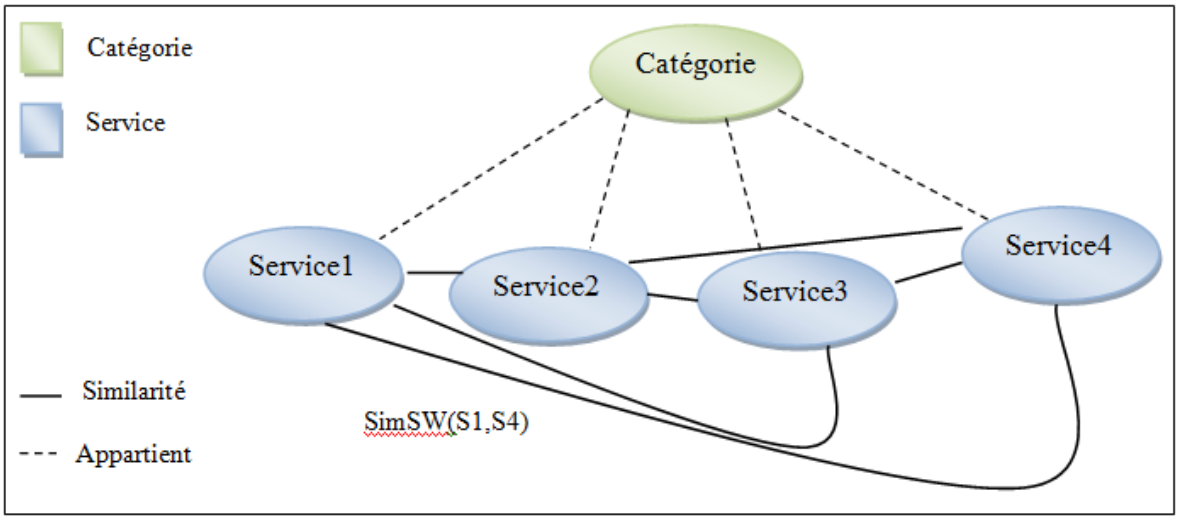


Figure2. 2. Exemple illustratif d'un graphe de services

Calcul de similarité entre les services et construction de la matrice de similarité :

Dans ce qui suit, nous expliquons comment la similarité *SimSW* est calculée et comment elle est utilisée pour la création du graphe.

- ***Similarité Cosinus :***

La similarité cosinus entre deux services web Si et Sj est définie par le rapport du produit scalaire de leurs vecteurs et le produit de la norme de Si et de Sj comme suit :

$$sim(Si, Sj) = \cos(\mathrm{Si}, \mathrm{Sj}) = \frac{Si * Sj}{\|.Si\|_2 * \|.Sj\|_2} \quad (2.1)$$

- ***Matrice de similarité*** :

Après le calcul de la similarité entre les services web, on obtient la matrice de similarité MS entre ces services de taille n×n, où n le nombre de services et les cases MS(Swi, Swj) contiennent la similarité entre ces services avec la diagonale MS(Swi, Swi) contient des 1 (similarité du service avec lui-même).

	Sw1	**Sw2**	**Sw3**	**Swn**
Sw1	**1**	….	….	…..
Sw2	….	**1**	….	….
Sw3	….	….	**1**	….
Swn	….	….	….	**1**

Figure2. 3. Exemple de la matrice de similarité

Construction du graphe de services :

La matrice de similarité sera utilisée par la suite pour la création du graphe des services, en commençant par les services évalués par l'utilisateur et qui ont eu la valeur la plus grande comme nœuds initiaux. Ensuite, on trace les relations (arrêtes) entre ces services et les autres services de la (les) même catégorie(s), ainsi que les autres services en indiquant les valeurs de similarité de chaque relation.

Enfin, on obtient le graphe des services représenté par la **Figure (2.4)** suivante :

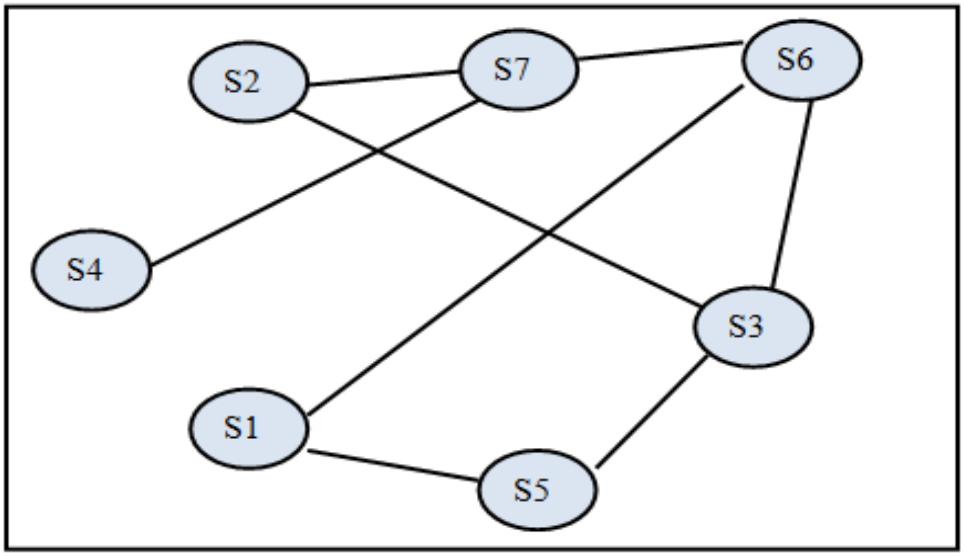


Figure2. 4. Exemple de Graphe de services

Afin de trouver les nouveaux services, que l'utilisateur n'a jamais vus, ni évalués, on propose d'utiliser l'algorithme du plus long chemin, pour le parcours de notre graphe de service, que nous expliquons comment nous avons l'adopter à notre contexte, dans la section suivante.

2.4.4.2. Parcours du graphe et Sélection des Services Dissimilaires:

Comme notre graphe de services est un graphe valué, càd chaque arc est muni d'une valeur qui est la similarité entre deux services dans notre cas, on peut le considéré comme un réseau.

Où, un réseau est par définition un graphe G=(X,U) muni d'une application : $d : U \rightarrow \Re$ qui à chaque arc fait correspondre sa longueur *d(u)*. Il est noté par: *R=(X,U,d)*, où *d(u)* peut représenter un coût, une distance, une durée, ..etc.

La recherche d'un plus long chemin sur un réseau *R=(X,U,d)* revient à rechercher un plus court chemin sur *R=(X,U,-d).* Sachant que le principe de la recherche d'un plus court chemin issu d'un sommet *x* vers un autre sommet *y* dans un graphe G, est de trouver tous les chemins élémentaires reliant *x* à *y* et d'en choisir le plus court.

Dans la littérature, ils existent plusieurs algorithmes pour résoudre les problèmes de recherche des plus courts chemins dans un réseau, où l'une des méthodes les plus performantes est celle de *Dijikstra*.

On a choisit cet algorithme, car il est applicable pour déterminer une arborescence des plus courtes distances sur un réseau où les longueurs des arcs sont positives ou nulles ($d(u) \geq 0, \forall u \in U$). Ce qui est le cas pour notre graphe de services.

a) Algorithme du plus long chemin (Dijikstra) :

- ✓ **Principe :**

L'idée de l'algorithme de *Dijkstra* est de calculer de proche en proche, l'arborescence des plus courtes distances, issue du sommet *s* à un sommet donné *p*. Une particularité de cet algorithme est que les distances s'introduisent dans l'ordre croissant.

Les étapes de l'algorithme sont présentées ci-dessous.

- ✓ **Enoncé:** *Algorithme du plus long chemin*

Données: Un réseau R=(X,U,d) avec $d(u) \geq 0 \ \forall u \in U$.

Résultat: Arborescence de plus courtes distances A.

(0) Initialisation: Soit s un sommet de X.
On pose : S={s}, $\pi(s)=0$, $\pi(x)=\infty$, $\forall x \in X / \{s\}$, $A(s)=\varnothing$ et $\alpha = s$

/* α est le dernier sommet introduit dans S

(1) Examiner tous les arcs *u* dont l'extrémité initiale est égale à $\alpha(I(u))=\alpha$ et l'extrémité terminale n'appartient pas à S(T(u)=y avec $y \notin S$).
Si $\pi(\alpha)+d(u) < \pi(y)$, on pose $\pi(y) = \pi(\alpha)+d(u)$ et $A(y)= u$ Aller en (2).

(2) Choisir un sommet $z \notin S$ tel que $\pi(z) = Min \{\pi(y) / y \notin S\}$

- Si $\pi(z) = \infty$; Terminer. Le sommet s n'est pas une racine dans R.
- Si $\pi(z) < \infty$; on pose $\alpha = z$ et $S = S \cup \{\alpha\}$.
- Si S=X; Terminer. **A** définit l'arborescence des plus courts chemins issus de *s*.
- Si $S \neq X$ Aller en (1).

Dans notre cas, nous allons calculer pour chaque service, de l'ensemble des services web que l'utilisateur actif a évalués, le plus long chemin de son réseau, càd on va considérer que chaque service évalué est un sommet initial, et on calcule le plus long chemin à partir de ce service.

A chaque fois qu'on répète l'algorithme pour un sommet, on retient le service, ou l'ensemble de services (sommets) les plus lointains, que l'utilisateur n'a pas encore évalué (s), et leur similarité avec le service actif. Où, la similarité est considérée comme la longueur du chemin entre les deux services, ce qui permet par la suite de justifier la pertinence de ces nouveaux services.

A la fin de cette étape, on obtient une liste de services dissimilaires aux services évalués par l'utilisateur, qui sont très lointains de ses centres d'intérêts et qu'il n'a pas encore vus ou évalués, mais qui peuvent lui intéresser car ils sont les voisins les plus éloignés de ses voisins.

2.4.4.3. La prédiction basée items des services inconnus :

Dans cette étape, une prédiction des services non évalués par l'utilisateur est réalisée afin de prédire leur degré de préférence par l'utilisateur, en utilisant les évaluations qu'il a donné aux autres services de son profil.

En général, la prédiction nécessite deux étapes importantes: la première étape consiste à rechercher les services les plus similaires de l'item actif, et la deuxième étape consiste à calculer la prédiction des évaluations inconnues, en utilisant les évaluations que l'utilisateur actif a données à ces services et leur similarité avec les services les plus similaires.

Dans notre cas et afin d'assurer la nouveauté, on va considérer l'ensemble des plus proches services pour le calcul de la prédiction traditionnelle (présentée ci-dessus), ainsi que les services très lointains (dissimilaires) trouvés par l'algorithme de Dijkstra.

Donc, on propose d'appliquer la formule de prédictions de deux façon différentes tout en considérant les deux ensembles de services : les *m* services les plus similaire et les *k* services les plus dissimilaires.

Dans le présent travail, nous avons exploité les similarités qu'on a calculées précédemment dans la sous section (4.4.1.a) et qui sont enregistrées dans la matrice de similarité.

Une fois que la similarité entre les services est calculée, et les T-plus proches services sont sélectionnés à partir de la matrice de similarité, la prochaine étape est de prévoir pour l'utilisateur actif, des valeurs pour les services similaires à sont profil et qu'il n'a pas encore évalués.

La valeur prévue est basée sur la somme pondérée des estimations de l'utilisateur ainsi que les déviations des estimations moyennes et peut être calculée à l'aide de la formule suivante :

$$\hat{r}_{u,s} = \bar{r}_s + \frac{\sum_{s\prime=1}^{T} \mathrm{sim}(s,s\prime)\times(r_{u,s\prime}-\bar{r}_{s\prime})}{\sum_{s\prime=1}^{T}|\mathrm{sim}(s,s\prime)|} \qquad \text{si } s\in K \qquad (2.2)$$

Où :

T : nombre de services présents dans le voisinage du service *s*, ayant déjà été votés par l'utilisateur *u*.

$\bar{r}_s$: Moyenne des évaluations pour le service.

$r_{u,s\prime}$: Évaluation de l'utilisateur u pour le service s'.

$\bar{r}_{s\prime}$: Moyenne des évaluations pour le service s'.

|*Sim(s,s')*| : Similarité entre les services.

K : ensemble de services non évalués similaires au profil de l'utilisateur.

La deuxième forme est appliquée lorsque le service à prédire appartient à l'ensemble des services dissimilaires, et afin d'éviter la valeur nulle de la similarité entre le service actif et les services évalués par l'utilisateur, on propose de remplacer la valeur de similarité par la distance (longueur du chemin) entre eux.

Donc, on obtient la nouvelle formule de prédiction suivante :

$$\hat{r}_{u,s} = \bar{r}_s + \frac{\sum_{s\prime=1}^{T} long(s,s\prime)\times(-\bar{r}_{s\prime})}{\sum_{s\prime=1}^{T}|long(s,s\prime)|} \qquad \text{Si } s \in M \qquad (2.3)$$

Où,

M : ensemble de services non évalués dissimilaires au profil de l'utilisateur.

long (s, s') : la distance entre les services s et s'.

Ces prédictions seront comparées par la suite avec les valeurs réelles omises en utilisant la mesure *MAE (Mean Absolute Error)* qui est la mesure de qualité de la prédiction très utilisée dans ce domaine.

2.4.4.4. La Recommandation Top-N des services :

Notre objectif pour cette étape, est de recommander les services les plus pertinents à l'utilisateur tout en respectant la notion de nouveauté, pour aider l'utilisateur à découvrir de nouveaux services qui peuvent lui intéressés et à personnaliser sa liste de recommandation

Pour ce faire, nous avons identifié un ensemble de services qui pourraient répondre aux besoins de l'utilisateur, et un autre contenant de nouveaux services qu'il n'a jamais vus ni évalués, en exploitant les relations de similarité issues du graphe de services.

Après avoir calculé les valeurs de prédiction pour les deux ensembles de services K et M (similaires et dissimilaires), nous classons ces ensembles de services en fonction de leurs valeurs de prédiction dans un ordre décroissant. Ensuite, on sélectionne les Top services de chaque liste pour constituer une Top-N liste à recommander à l'utilisateur.

Formellement, la valeur N des services à recommander à l'utilisateur est la somme des Top services de chaque liste :

$$N = k + m \qquad \text{avec } k \leq K \text{ et } m \leq M \qquad (2.4)$$

On propose de donner une valeur plus grande pour les services similaires *m* afin de garder la notion de pertinence pour l'utilisateur, malgré que, même les services très dissimilaires sont aussi partiellement pertinents car ils ont une relation indirecte avec les préférences de l'utilisateur (son profil) exprimée par le chemin entre eux dans le graphe, et justifiée par sa longueur non nulle. Où, les valeurs de *m* et *k* seront fixées à travers les tests.

Graphiquement, la liste des Top-10 services à recommander pour l'utilisateur actif est comme suit :

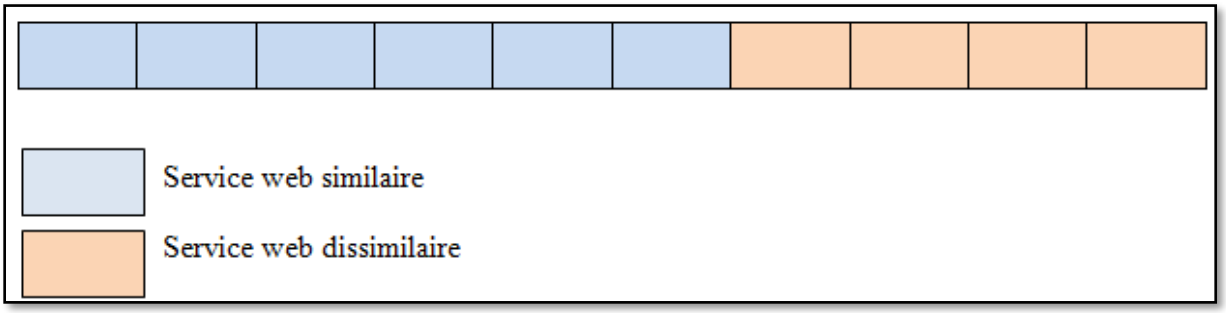


Figure2. 5. Liste Finale de Recommandation Top-N

2.5. CONCLUSION

Dans ce chapitre, nous avons décrit notre approche de recommandation à base de graphe. Cette approche consiste à exploiter les relations entre les services afin de fournir des recommandations de services à un utilisateur, qui vient de se connecter ou à la recherche d'un service, tout en respectant la notion de la nouveauté dans les listes de recommandation. Pour ce faire, nous avons construit un graphe de services, à partir du quel nous avons extrait un ensemble de services nouveaux pour l'utilisateur et très dissimilaires pour enrichir la liste de recommandation Top-N. Les expérimentations et tests de performance de notre système seront présentés dans le chapitre suivant.

Chapitre 3:

Implémentation et Evaluation du Système

Chapitre3 : Implémentation et Évaluation

3.1. Introduction

Dans ce chapitre, nous allons mener un ensemble d'expériences pour examiner et étudier l'efficacité et la faisabilité des différentes propositions que nous avons présentées dans le chapitre précédent, et ceci dans le but de trouver les meilleures valeurs pour chaque variable, et pour cela nous devons faire plusieurs expériences afin de prouver leur faisabilité.

3.2. Dataset

Nous avons utilisé dans nos expérimentations la base de données *MovieLens*[1] afin d'évaluer notre algorithme. L'ensemble de données *MovieLens* est composé de 943 utilisateurs et 1682 items avec une échelle d'évaluation comprise entre [1:5], où chaque utilisateur dispose de plus de 20 votes.

Nous avons utilisés trois ensembles de données qui contiennent 100, 200 et 300 utilisateurs avec plus de 20 évaluations et 1000 items comme ensembles d'entrainement, qui sont dénotés comme *ML_100*, *ML_200* et *ML_300*, respectivement, et un autre ensemble qui contient 200 utilisateurs comme ensemble de test.

En outre, nous avons choisit au hasard 5, 10 et 15 et 20 items évalués par l'utilisateur actif que nous avons les arrangés dans quatre différents ensembles de données qu'on a appelés *Given-5, Given-10, et Given-15, et Given-20*, respectivement, et qu'on aura les utilisés pour le test de performance de la prédiction

La base contient 3 fichiers de type CSV.

- **u.data** - L'ensemble de données u complet, 100000 évaluations par 943 utilisateurs sur 1682 éléments. Chaque utilisateur a évalué au moins 20 films. Les utilisateurs et les éléments sont numérotés consécutivement à partir de 1. Les données sont classées au hasard. Ceci est une liste d'ID utilisateur séparés par des tabulations | identifiant de l'article | note | horodatage.
- **u.user** - Le nombre d'utilisateurs, d'éléments et de notes dans l'ensemble de données u.
- **u.item** - Informations sur les éléments (films); ceci est une tabulation séparée

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

Liste des identifiants de film | titre du film | date de sortie | date de sortie de la vidéo | URL IMDb | inconnu | Action | Aventure | Animation | Enfants | Comédie | Crime | Documentaire | Drame | Fantaisie | Film-Noir | Horreur | Musical | Mystère | Romance | Science-fiction | Thriller | Guerre | Western |

Un 1 indique que le film est de ce genre, un 0 indique que ce n'est pas le cas; les films peuvent être dans plusieurs genres

3.3. Présentation des outils de développement

3.3.1. Matériel utilisée

Les expériences sont effectuées sur un PC avec un processeur 1.90 GHz Intel Core I3, 4G de mémoire, et un système d'exploitation Windows 7 professionnel type 64 bits.

3.3.2. Logiciels et Plateformes utilisés

- **Anaconda**

Anaconda est une distribution libre et open source des langages de programmation python et R appliqué au développement d'applications dédiées à la science des données et à l'apprentissage automatique (traitement de données à grande échelle, analyse prédictive, calcul scientifique), qui vise à simplifier la gestion des paquets et de déploiement. Grâce à cet outil on peut éviter les erreurs d'installation des packages de programmation pour l'apprentissage, que ce soit pour Windows ou linux et il est si simple dans l'utilisation.

- **Jupyter**

Jupyter est une application web utilisée pour programmer dans plus de 40 langages de programmation, dont Python, Julia, Ruby, R, ou encore Scala. Jupyter permet de réaliser des calepins ou notebooks, c'est-à-dire des programmes contenant à la fois du texte en markdown et du code en Julia, Python, R... Ces calepins sont utilisés en science des données pour explorer et analyser des données.

- **Pandas**

Est une bibliothèque de logiciels écrite pour le langage de programmation Python pour la manipulation et l'analyse de données. En particulier, il propose des structures de données et des opérations pour manipuler des tableaux numériques et des séries chronologiques . Il s'agit d' un

logiciel libre publié sous la licence BSD à trois clauses , Le nom est dérivé du terme « données de panel », un terme économétrique pour les ensembles de données qui comprennent des observations sur plusieurs périodes de temps pour les mêmes individus. Son nom est un jeu sur l'expression "analyse de données Python" elle-même.

- **NumPy**

Est une extension du langage de programmation Python, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux.

Plus précisément, cette bibliothèque logicielle libre et open source fournit de multiples fonctions permettant notamment de créer directement un tableau depuis un fichier ou au contraire de sauvegarder un tableau dans un fichier, et manipuler des vecteurs, matrices et polynômes. NumPy est la base de SciPy, regroupement de bibliothèques Python autour du calcul scientifique3.

3.4. Métriques d'évaluation :

3.4.1. Métrique d'évaluation de la prédiction (MAE) :

Nous avons utilisé la métrique MAE pour évaluer la performance des prédictions, définie comme suit :

$$MAE = \frac{\sum_{j,m} |r_{j,m} - \tilde{r}_{j,m}|}{|N|} \quad (3.1)$$

Où N est le nombre des évaluations utilisées dans le test. Plus MAE est inférieure, plus la performance est meilleure.

3.4.2. Métriques d'évaluation des recommandations (Top-N)

Afin d'évaluer la performance de notre système de recommandation, nous avons utilisé les métriques du Rappel, Précision et la métrique F_1. Où, F1 est calculée comme suit :

$$F_1 = \frac{2PR}{P+R} \quad (3.2)$$

Où *P* et *R* sont la *Précision* et le *Rappel* respectivement, et ils sont calculés comme suit :

$$P = \frac{N_t}{N} \quad (3.3)$$

$$R = \frac{N_t}{N_p} \tag{3.4}$$

- N_p : Nombre total des items pertinents.
- N_t : Nombre des items pertinents trouvés.
- N : Nombre total des items

3.5. Méthodologie d'implémentation

Etape 1 : Remplissage du fichier de *MovieLens* à partir de 3 autres fichiers pour obtenir la base de données nécessaire au travail.

Etape 2 : On doit trier les données selon user_id ensuite on les devise en deux : 80% pour l'apprentissage et 20% pour le test.

```
u, s, vt = svds(train_data_matrix, k = 20,which="LM")
```

Etape 3 : Traitement de données et évaluation de la performance de la prédiction fournie.

Alors, nous commencerons par la première étape, dans laquelle nous expliquerons comment charger les données et comment la création du graphe.

Tout d'abord, nous allons télécharger la base de données et commencer à travailler dessus.

```
u_cols = ['user_id', 'age', 'sex', 'occupation', 'zip_code']
users = pd.read_csv('u.user', sep='|', names=u_cols,
                    encoding='latin-1', parse_dates=True)

r_cols = ['user_id', 'movie_id', 'rating', 'unix_timestamp']
ratings = pd.read_csv('u.data', sep='\t', names=r_cols,
                      encoding='latin-1')

m_cols = ['movie_id', 'title', 'release_date', 'video_release_date', 'imdb_url']
movies = pd.read_csv('u.item', sep='|', names=m_cols, usecols=range(5),
                     encoding='latin-1')

movie_ratings = pd.merge(movies, ratings)
df = pd.merge(movie_ratings, users)
```

Figure3. 1 Télécharger la base de données

L'utilisateur actif est sélectionné par numéro d'identifiant User_id comme suit :

```
selected_user = 5
```

Figure3. 2 Sélectionner l'utilisateur actif

Dans notre exemple, nous avons choisis l'utilisateur ayant l'identifiant 5.

3.6. Tests et Résultats :

3.6.1. Création de la matrice de similarité :

Dans cette section, nous créons la matrice de similarité I*I, comme nous l'avons dit dans le chapitre 2. Ces rapports sont calculés par la similarité Cosinus, où les valeurs sont comprises entre 0 et 1, et la diagonale contient des 1 ce qui exprime la similarité du service avec lui-même.

La **Figure (3.3)** illustre ce que nous avons dit :

	0	1	2	3	4	5	6	7	8	9	...	1672	1673	1674	1675	1676
0	1.000000	0.402382	0.330245	0.454938	0.286714	0.116344	0.620979	0.481114	0.496288	0.273935	...	0.035387	0.0	0.000000	0.000000	0.035387
1	0.402382	1.000000	0.273069	0.502571	0.318836	0.083563	0.383403	0.337002	0.255252	0.171082	...	0.000000	0.0	0.000000	0.000000	0.000000
2	0.330245	0.273069	1.000000	0.324866	0.212957	0.106722	0.372921	0.200794	0.273669	0.158104	...	0.000000	0.0	0.000000	0.000000	0.032292
3	0.454938	0.502571	0.324866	1.000000	0.334239	0.090308	0.489283	0.490236	0.419044	0.252561	...	0.000000	0.0	0.094022	0.094022	0.037609
4	0.286714	0.318836	0.212957	0.334239	1.000000	0.037299	0.334769	0.259161	0.272448	0.055453	...	0.000000	0.0	0.000000	0.000000	0.000000

5 rows × 1682 columns

Figure3. 3. Création de la matrice de similarité

Après le calcul de similarité et la création de la matrice de similarité, on obtient la liste des items (services) similaires au profil de notre utilisateur actif ayant comme id 5.

La **Figure (3.4)** suivante présente les 10 items les plus similaires au profil de l'utilisateur 5.

user_id	item_id	rating	timestamp	title
5	153	5	875636375	Fish Called Wanda, A (1988)
5	89	5	875636033	Blade Runner (1982)
5	169	5	878844495	Wrong Trousers, The (1993)
5	209	5	875636571	This Is Spinal Tap (1984)
5	163	5	879197864	Return of the Pink Panther, The (1974)
5	186	5	875636375	Blues Brothers, The (1980)
5	436	5	875720717	American Werewolf in London, An (1981)
5	174	5	875636130	Raiders of the Lost Ark (1981)
5	172	5	875636130	Empire Strikes Back, The (1980)
5	181	5	875635757	Return of the Jedi (1983)

Figure3. 4. Sélection des10 items les plus similaires au profil de l'utilisateur 5.

3.6.2. Création du graphe de services

À partir de la matrice de similarité, nous avons créé le graphe de services, montré dans la **Figure(3.5)**

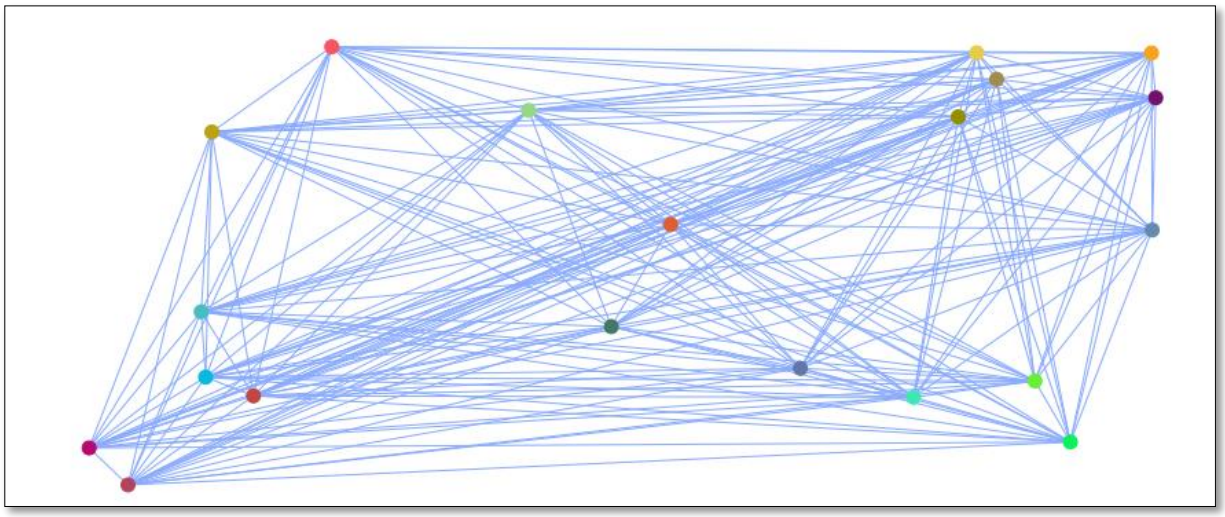


Figure3. 5. Le graphe de service

3.6.3. Parcours du graphe et sélection des plus loin voisins :

Dans cette section, nous évaluons l'algorithme de dijkstra pour le parcours des plus long chemins et la recherches des filmes dissimilaires.

Pour ce faire, on commence le parcours du graphe à partir de chaque film dans le profil de l'utilisateur 5 jusqu'à ce qu'on trouve tous les chemins les plus long et nous nous arrêtons si aucun nœud à visité, et on prend les films pour lesquels l'utilisateur n'a pas donné d'évaluation, càd rating = 0.

On obtient une liste de films dissimilaires au profil de l'utilisateur mais qui sont pertinents pour lui car ils sont les voisins les plus lointains des voisins de son profil.

La Figure (3.6) montre un ensemble de films dissimilaires au profil de l'utilisateur qui n'ont pas été déjà vus par l'utilisateur et qui sont trouvés par l'algorithmes de *Dijikstra*.

user_id	item_id	rating	timestamp	title
5	5	0	NaN	Copycat (1995)
5	15	0	NaN	Mr. Holland's Opus (1995)
5	8	0	NaN	Babe (1995)
5	7	0	NaN	Twelve Monkeys (1995)
5	2	0	NaN	GoldenEye (1995)
5	6	0	NaN	Shanghai Triad (Yao a yao yao dao waipo qiao) ...
5	9	0	NaN	Dead Man Walking (1995)
5	12	0	NaN	Usual Suspects, The (1995)
5	14	0	NaN	Postino, Il (1994)
5	3	0	NaN	Four Rooms (1995)

Figure3. 6. Sélection des 10 films dissimilaires du profil de l'utilisateur 5

3.6.4. Performance de la prédiction basée contenu :

Après l'obtention de l'ensemble de films, qui contient les films similaires et les films dissimilaires, nous appliquons notre modèle de prédiction comme suit :

```
def predict(ratings, similarity, type='user'):

    if type == 'user':
        T=50
        mean_user_rating = ratings.mean(axis=1)
        #You use np.newaxis so that mean_user_rating has same format as ratings
        ratings_diff = (ratings - mean_user_rating[:, np.newaxis])
        pred = mean_user_rating[:, np.newaxis] + similarity.dot(ratings_diff) / np.array([np.abs(similarity).sum(axis=1)]).T
    elif type == 'item':
        pred = ratings.dot(similarity) / np.array([np.abs(similarity).sum(axis=1)])
    return pred
```

Figure3. 7. Modèle de prédiction

3.6.4.1. Choix du nombre T des plus proches items à sélectionner pour la prédiction :

Cette expérience concerne la recherche de la valeur optimale des plus proches items à sélectionner **T** en utilisant l'ensemble de données *ML_200* tout en sélectionnant différentes valeurs des plus proches items T appartenant à l'intervalle [5 :5 : 50].

La Figure (3.8) montre l'évolution de l'erreur MAE en fonction du nombre des plus proches items à sélectionnés sur les deux ensembles de données *Given_20.*

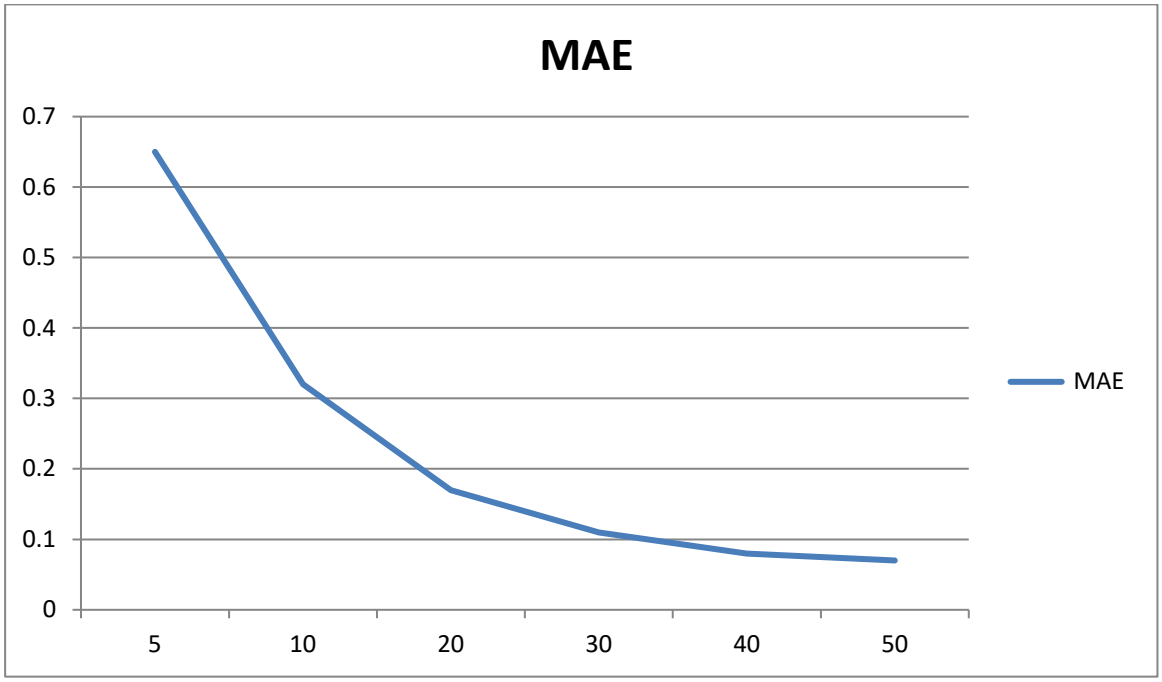


Figure3. 8. Evolution de l'erreur MAE en fonction des T-plus proches services sélectionnés

La Figure (3.8) montre que l'erreur MAE diminue progressivement avec le nombre d'items similaires à sélectionner, où elle atteint son maximum MAE= 0.65 pour une valeur de T=5. Ensuite, elle se dégrade jusqu'à ce qu'elle atteigne une valeur minimale égale à 0.07 pour T=50. Donc, nous allons fixer T à 50 items dans la formule de prédiction car c'est la valeur optimale qui nous donne la meilleure performance MAE.

3.6.4.2. Evaluation de la prédiction

Afin d'évaluer la performance de notre approche proposée, nous avons effectué des expériences sur *ML_200* avec les trois ensembles de données *Given 5*, *Given 10* et *Given 20*, en fixant T à 50, pour la prédiction des évaluations manquantes. Puis, on calcule l'erreur MAE pour tester la performance de notre système:

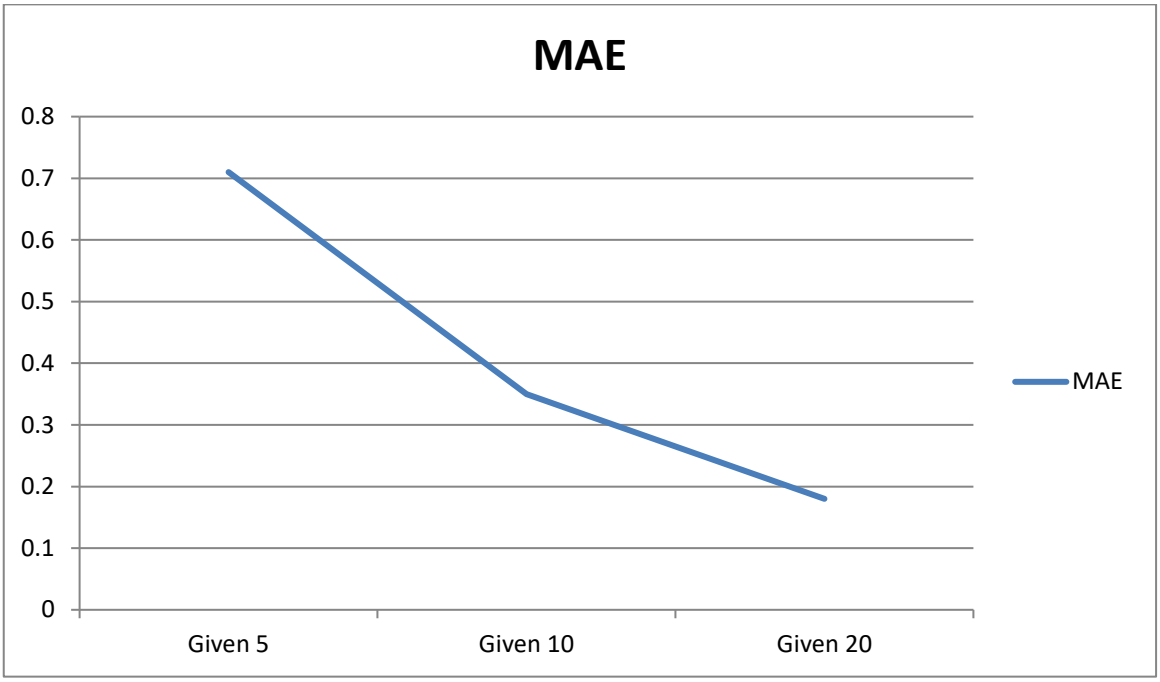


Figure3. 9. L'évolution de la courbe de l'erreur MAE

La Figure (3.9) montre l'évolution de la courbe de l'erreur MAE en fonction des ensembles de données *Given 5, Given 10 et Given 20.*

On remarque que la courbe atteint son niveau le plus élevé dans le cas de *Given5*, qui rapporte le taux d'erreur MAE 0.71. Ensuite, elle commence à diminuer progressivement en passant par *Given10*, où elle atteint un taux d'erreur égale à 0.35, et elle prend sa plus petite valeur dans le cas de l'ensemble de données *Given20* avec MAE 0.17.

3.6.5. Evaluation de la performance de la Recommandation TOP-N :

Après la prédiction des évaluations manquantes pour l'utilisateur actif, les items sont classés dans l'ordre décroissant de leurs prédictions, et les Top-N items sont générés à l'utilisateur. Sachant que la liste des Top-N films contient k films similaires et m films dissimilaires.

3.6.5.1.Choix des valeurs k et m :

Le tableau (3.1) présente les variations qu'on a pris pour les valeurs *k* et *m* afin de constituer la liste de recommandation N=10.

Liste / Items	TOP-10	TOP-10	TOP-10	TOP-10
Similaires	10	7	5	3
Dissimilaires	0	3	5	7

Tableau 3 1 Variation des valeurs *k* et *m* pour la Recommandation Top-10

Les Figures suivantes présentent les résultats affichés par le système dans les différents cas testés et mentionnés dans le tableau ci-dessus.

Cas 1 : *Recommandation TOP-10 similaires :*

user_id	item_id	rating	timestamp	title
5	153	5	875636375	Fish Called Wanda, A (1988)
5	89	5	875636033	Blade Runner (1982)
5	169	5	878844495	Wrong Trousers, The (1993)
5	209	5	875636571	This Is Spinal Tap (1984)
5	163	5	879197864	Return of the Pink Panther, The (1974)
5	186	5	875636375	Blues Brothers, The (1980)
5	436	5	875720717	American Werewolf in London, An (1981)
5	174	5	875636130	Raiders of the Lost Ark (1981)
5	172	5	875636130	Empire Strikes Back, The (1980)
5	181	5	875635757	Return of the Jedi (1983)

Figure3. 10. Recommandation TOP-10 similaires

Cas 2: *Recommandation TOP-10 (7 similaires+3 Dissimilaires) :*

distance	item_id	rating	timestamp	title	user_id
NaN	153	5	875636375.0	Fish Called Wanda, A (1988)	5
NaN	209	5	875636571.0	This Is Spinal Tap (1984)	5
NaN	42	5	875636360.0	Clerks (1994)	5
NaN	396	5	875636265.0	Serial Mom (1994)	5
NaN	430	5	875636631.0	Duck Soup (1933)	5
NaN	401	5	875636308.0	Brady Bunch Movie, The (1995)	5
NaN	163	5	879197864.0	Return of the Pink Panther, The (1974)	5
894.853978	5	0	NaN	Copycat (1995)	5
895.328040	15	0	NaN	Mr. Holland's Opus (1995)	5
111.134137	8	0	NaN	Babe (1995)	5

Figure3. 11 Recommandation TOP-10 (7 similaires + 3 Dissimilaires)

Cas 3 : *Recommandation Top-10 (5 similaires + 5 Dissimilaires) :*

distance	item_id	rating	timestamp	title	user_id
NaN	153	5	875636375.0	Fish Called Wanda, A (1988)	5
NaN	209	5	875636571.0	This Is Spinal Tap (1984)	5
NaN	42	5	875636360.0	Clerks (1994)	5
NaN	396	5	875636265.0	Serial Mom (1994)	5
NaN	430	5	875636631.0	Duck Soup (1933)	5
894.853978	5	0	NaN	Copycat (1995)	5
895.328040	15	0	NaN	Mr. Holland's Opus (1995)	5
111.134137	8	0	NaN	Babe (1995)	5
529.074469	7	0	NaN	Twelve Monkeys (1995)	5
705.824727	2	0	NaN	GoldenEye (1995)	5

Figure3. 12. Recommandation Top-10 (5 similaires + 5 Dissimilaires)

Cas 4 : *Recommandation Top-10 (7 similaires + 3 Dissimilaires)*

distance	item_id	rating	timestamp	title	user_id
NaN	153	5	875636375.0	Fish Called Wanda, A (1988)	5
NaN	209	5	875636571.0	This Is Spinal Tap (1984)	5
NaN	42	5	875636360.0	Clerks (1994)	5
894.853978	5	0	NaN	Copycat (1995)	5
895.328040	15	0	NaN	Mr. Holland's Opus (1995)	5
111.134137	8	0	NaN	Babe (1995)	5
529.074469	7	0	NaN	Twelve Monkeys (1995)	5
705.824727	2	0	NaN	GoldenEye (1995)	5
32.565214	6	0	NaN	Shanghai Triad (Yao a yao yao dao waipo qiao) ...	5
527.664996	9	0	NaN	Dead Man Walking (1995)	5

Figure3. 13. Recommandation Top-10 (7 similaires + 3 Dissimilaires)

La performance du système en termes de précision des recommandations, tout en modifiant à chaque fois le nombre d'items similaires et dissimilaires à sélectionner, est montrée dans la **Figure (3.14),** suivante :

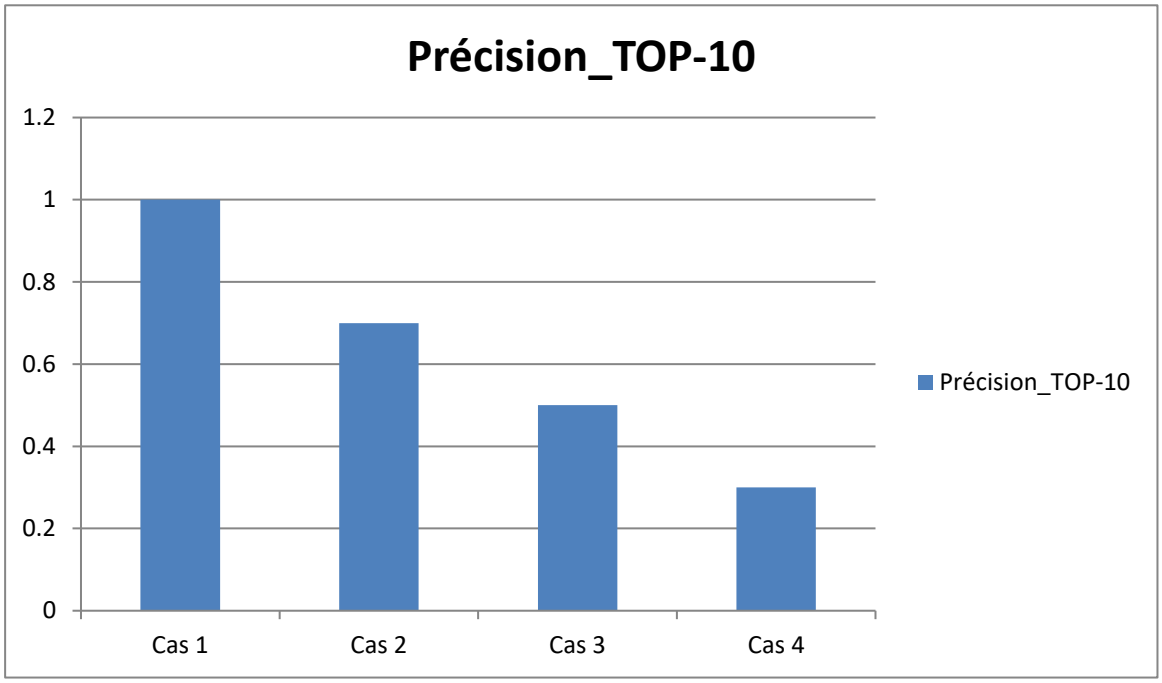


Figure3. 14 La Précision de la recommandation TOP_10 avec différentes valeur de *k* et *m*

A partir de la **Figure (3.14),** on voit clairement que la précision est égale à 1 dans le cas de la recommandation traditionnelle (càd pour une liste des Top-10 items similaires et 0 items dissimilaires). Ensuite, elle commence à se diminuer et elle prend une valeur égale à 0.7 pour le deuxième cas (Top-10 avec 7 items similaires et 3 items dissimilaires). Ainsi, pour les deux autres cas, elle atteint les valeurs 0.5 et 0.32 pour le troisième cas (Top-10 avec 5 items

similaires et 5 items dissimilaires) et le quatrième cas avec (Top-10 contenant 7 items dissimilaires et 3 items similaires), respectivement.

Notez que la précision atteint ses niveaux les plus élevés lorsque le nombre des items similaire est supérieur au nombre des items dissimilaires.

3.6.5.2. Recommandation Top-N avec nouveauté Vs. Recommandation Top-N classique:

Dans cette expérience, nous avons testé la performance générale de notre système de recommandation, en la comparant avec celle d'un système de recommandation classique (recommandation des items similaires en utilisant juste la similarité Cosinus dans la phase de la prédiction et sans l'utilisation de la représentation avec graphe).

Le Tableau suivant présente le nombre d'items à sélectionner pour constituer les différentes listes dans les deux cas de recommandation.

	Liste / **Items**	**TOP-5**	**TOP-10**	**TOP-15**	**TOP-20**
Recommandation avec nouveauté	*Similaires*	2	5	8	15
	Dissimilaires	3	5	7	5
Recommandation classique	*Similaires*	5	10	15	20
	Dissimilaires	0	0	0	0

Tableau 3 2 Variations des valeurs k et m pour différentes valeurs de N

Les résultats retournés par le système sont comme suit :

- *Recommandation Top-5 avec Nouveauté:*

distance	item_id	rating	timestamp	title	user_id
NaN	153	5	875636375.0	Fish Called Wanda, A (1988)	5
NaN	209	5	875636571.0	This Is Spinal Tap (1984)	5
NaN	42	5	875636360.0	Clerks (1994)	5
894.853978	5	0	NaN	Copycat (1995)	5
895.328040	15	0	NaN	Mr. Holland's Opus (1995)	5

Figure3. 15. Recommandation Top-5 avec nouveauté

- *Recommandation Top-5classique:*

item_id	rating	timestamp	title	user_id
153	5	875636375.0	Fish Called Wanda, A (1988)	5
209	5	875636571.0	This Is Spinal Tap (1984)	5
42	5	875636360.0	Clerks (1994)	5
396	5	875636265.0	Serial Mom (1994)	5
430	5	875636631.0	Duck Soup (1933)	5

Figure3. 16 Recommandation Top-5 classique

- *Recommandation Top-10avec nouveauté :*

distance	item_id	rating	timestamp	title	user_id
NaN	153	5	875636375.0	Fish Called Wanda, A (1988)	5
NaN	209	5	875636571.0	This Is Spinal Tap (1984)	5
NaN	42	5	875636360.0	Clerks (1994)	5
NaN	396	5	875636265.0	Serial Mom (1994)	5
NaN	430	5	875636631.0	Duck Soup (1933)	5
894.853978	5	0	NaN	Copycat (1995)	5
895.328040	15	0	NaN	Mr. Holland's Opus (1995)	5
111.134137	8	0	NaN	Babe (1995)	5
529.074469	7	0	NaN	Twelve Monkeys (1995)	5
705.824727	2	0	NaN	GoldenEye (1995)	5

Figure3. 17. Recommandation Top-5 classique

- *Recommandation Top-10 classique*

user_id	item_id	rating	timestamp	title
5	153	5	875636375	Fish Called Wanda, A (1988)
5	89	5	875636033	Blade Runner (1982)
5	169	5	878844495	Wrong Trousers, The (1993)
5	209	5	875636571	This Is Spinal Tap (1984)
5	163	5	879197864	Return of the Pink Panther, The (1974)
5	186	5	875636375	Blues Brothers, The (1980)
5	436	5	875720717	American Werewolf in London, An (1981)
5	174	5	875636130	Raiders of the Lost Ark (1981)
5	172	5	875636130	Empire Strikes Back, The (1980)
5	181	5	875635757	Return of the Jedi (1983)

Figure3. 18. Recommandation Top-5 Classique

- *Recommandation Top-15avec nouveauté:*

distance	item_id	rating	timestamp	title	user_id
NaN	153	5	875636375.0	Fish Called Wanda, A (1988)	5
NaN	209	5	875636571.0	This Is Spinal Tap (1984)	5
NaN	42	5	875636360.0	Clerks (1994)	5
NaN	396	5	875636265.0	Serial Mom (1994)	5
NaN	430	5	875636631.0	Duck Soup (1933)	5
NaN	401	5	875636308.0	Brady Bunch Movie, The (1995)	5
NaN	163	5	879197864.0	Return of the Pink Panther, The (1974)	5
894.853978	5	0	NaN	Copycat (1995)	5
895.328040	15	0	NaN	Mr. Holland's Opus (1995)	5
111.134137	8	0	NaN	Babe (1995)	5
529.074469	7	0	NaN	Twelve Monkeys (1995)	5
705.824727	2	0	NaN	GoldenEye (1995)	5
32.565214	6	0	NaN	Shanghai Triad (Yao a yao yao dao waipo qiao) ...	5
527.664996	9	0	NaN	Dead Man Walking (1995)	5
525.530131	12	0	NaN	Usual Suspects, The (1995)	5

Figure3. 19. Recommandation Top-15 avec nouveauté

- *Recommandation Top-15 classique :*

item_id	path	rating	timestamp	title	user_id
153	NaN	5	875636375.0	Fish Called Wanda, A (1988)	5
209	NaN	5	875636571.0	This Is Spinal Tap (1984)	5
42	NaN	5	875636360.0	Clerks (1994)	5
396	NaN	5	875636265.0	Serial Mom (1994)	5
430	NaN	5	875636631.0	Duck Soup (1933)	5
401	NaN	5	875636308.0	Brady Bunch Movie, The (1995)	5
163	NaN	5	879197864.0	Return of the Pink Panther, The (1974)	5
181	NaN	5	875635757.0	Return of the Jedi (1983)	5
186	NaN	5	875636375.0	Blues Brothers, The (1980)	5
228	NaN	5	875636070.0	Star Trek: The Wrath of Khan (1982)	5
174	NaN	5	875636130.0	Raiders of the Lost Ark (1981)	5
172	NaN	5	875636130.0	Empire Strikes Back, The (1980)	5
89	NaN	5	875636033.0	Blade Runner (1982)	5
100	NaN	5	875635349.0	Fargo (1996)	5
390	NaN	5	875636340.0	Fear of a Black Hat (1993)	5

Figure3. 20. Recommandation Top-15 Classique

- *Recommandation TOP-20 avec nouveauté :*

distance	item_id	rating	timestamp	title	user_id
NaN	153	5	875636375.0	Fish Called Wanda, A (1988)	5
NaN	209	5	875636571.0	This Is Spinal Tap (1984)	5
NaN	42	5	875636360.0	Clerks (1994)	5
NaN	396	5	875636265.0	Serial Mom (1994)	5
NaN	430	5	875636631.0	Duck Soup (1933)	5
NaN	401	5	875636308.0	Brady Bunch Movie, The (1995)	5
NaN	163	5	879197864.0	Return of the Pink Panther, The (1974)	5
NaN	181	5	875635757.0	Return of the Jedi (1983)	5
NaN	186	5	875636375.0	Blues Brothers, The (1980)	5
NaN	228	5	875636070.0	Star Trek: The Wrath of Khan (1982)	5
894.853978	5	0	NaN	Copycat (1995)	5
895.328040	15	0	NaN	Mr. Holland's Opus (1995)	5
111.134137	8	0	NaN	Babe (1995)	5
529.074469	7	0	NaN	Twelve Monkeys (1995)	5
705.824727	2	0	NaN	GoldenEye (1995)	5

Figure3. 21. Recommandation Top-20 avec nouveauté

- *Recommandation TOP-20 classique :*

item_id	rating	timestamp	title	user_id
153	5	875636375.0	Fish Called Wanda, A (1988)	5
209	5	875636571.0	This Is Spinal Tap (1984)	5
42	5	875636360.0	Clerks (1994)	5
396	5	875636265.0	Serial Mom (1994)	5
430	5	875636631.0	Duck Soup (1933)	5
401	5	875636308.0	Brady Bunch Movie, The (1995)	5
163	5	879197864.0	Return of the Pink Panther, The (1974)	5
181	5	875635757.0	Return of the Jedi (1983)	5
186	5	875636375.0	Blues Brothers, The (1980)	5
228	5	875636070.0	Star Trek: The Wrath of Khan (1982)	5
174	5	875636130.0	Raiders of the Lost Ark (1981)	5
172	5	875636130.0	Empire Strikes Back, The (1980)	5
89	5	875636033.0	Blade Runner (1982)	5
100	5	875635349.0	Fargo (1996)	5
390	5	875636340.0	Fear of a Black Hat (1993)	5
434	5	875637033.0	Forbidden Planet (1956)	5
408	5	878844495.0	Close Shave, A (1995)	5
169	5	878844495.0	Wrong Trousers, The (1993)	5
257	5	875635239.0	Men in Black (1997)	5
189	5	878844495.0	Grand Day Out, A (1992)	5

Figure3. 22. Recommandation Top-20 Classique

La performance du système en termes de Précision des recommandations générées sur l'ensemble de test avec différentes valeurs d'items à sélectionner N = [5, 10, 15, 20] est présentée dans la **Figure (3.2.3).**

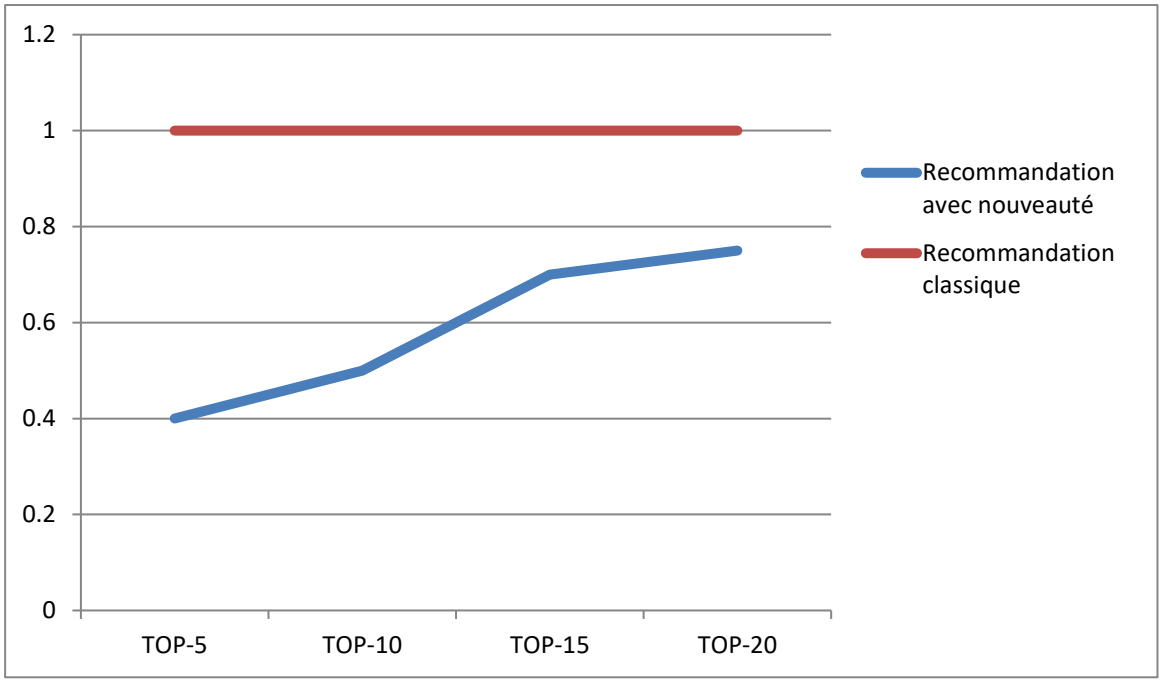


Figure3. 23. Recommandation avec Nouveauté Vs. Recommandation Classique

A partir de la **Figure (3.23),** on observe que la courbe de la précision est égale à 1 et cela veut dire que tous les items sélectionnés sont pertinents pour l'utilisateur.

Cependant, en examinant les valeurs de la précision trouvées dans le cas de la recommandation avec nouveauté, on remarque que la valeur minimale de précision est égale à 0.4 pour la recommandation TOP-5. Ensuite, elle commence à augmenter progressivement dans le cas de TOP-10 items avec une valeur de précision 0.5. Ainsi, dans la recommandation TOP- 15 la précision prend la valeur 0.7, et elle atteint sa valeur maximale égale à 0.75 pour la recommandation Top-20.

3.7. Discussion des résultats :

Les études empiriques sur la base de données *Movielens* ont démontré que la méthode proposée est moins performante que la recommandation basée contenu classique à travers l'évaluation des performances du système avec les mesures MAE et Précision, chose qui est évidente car on ne considère que les items similaires au profil de l'utilisateur donc pertinents dans la recommandation classique. Cependant, elle augmente le taux de nouveauté dans les listes de recommandation avec des items pertinents pour l'utilisateur, où cette pertinence est jugée par

l’existence des chemins les plus longs entre les items nouveaux à recommander et le profil de l’utilisateur, présentés dans la colonne Distance dans les tables des résultats présentés.

De plus, à partir des résultats trouvés et présentés ci-dessus, on peut dire que le nombre d’items voisins sélectionnés joue un rôle crucial pour l’amélioration de la performance du système, où un nombre très restreint donne de mauvais résultats alors qu’un nombre un peu important des plus proches voisins augmente la performance du système du faite qu’il aide à augmenter l’exactitude de la prédiction des items.

On peut constater aussi que la performance du système dans le cas de la recommandation avec nouveauté dépend fortement au nombre des items similaires et dissimilaires à sélectionner pour constituer la liste finale de la recommandation.

Pour la performance générale du système, la recommandation classique a dépassé la recommandation avec nouveauté dans les différents cas de listes générées N= [5 10 15 20], cela est dû à l’ensemble des items recommandés qui est toujours similaires au profil de l’utilisateur car on ne sélectionne que des items pertinents. Mais, on peut constater que la performance de notre système, en termes de précision, est toujours croissante par rapport à la taille de la liste de recommandation malgré l’ajout d’items dissimilaires. Cela confirme la faisabilité de l’approche proposée et qu’elle garantie juste une petite perte de performance par rapport à la recommandation traditionnelle.

3.8. Conclusion

Dans ce chapitre, nous avons présenté les étapes d’implémentation de notre système de recommandation de services web à base de graphe. Les résultats expérimentaux montrent que l’approche proposée donne de bonnes valeurs de précision, en présence même d’items très dissimilaires au profil de l’utilisateur, et augmente la nouveauté dans les listes générées, ce qui reflète la satisfaction des utilisateurs des recommandations fournies.

Conclusion et Perspectives

Ce mémoire s'est articulé autour du problème de la découverte des nouveaux services web par les utilisateurs de services.

La grande masse de services web accessibles sur les plateformes web a poussé à mettre en place des recommandeurs automatiques. Ces derniers recommandent à l'utilisateur des produits, qui pourraient l'intéresser, de façon automatique en se basant sur son comportement, et en s'adaptant à ses besoins et intérêts.

L'inspiration pour les systèmes de recommandation et la volonté de proposer toujours des produits pertinents aux utilisateurs, a pénalisé d'autres aspects importants à prendre en considération dans le processus de consommation des individus. Où, la découverte, et la nouveauté, sont deux aspects importants qu'il faut tenir en compte lors de la génération des listes de recommandations, car ces dernières ne peuvent pas être toujours une sélection de produits qui reflètent les comportements passés des utilisateurs pour ne pas leur faire ennuyer du système. D'un autre côté, on ne peut pas aussi donner trop de nouveauté dans les listes de recommandation, sous peine de s'éloigner des intérêts des utilisateurs.

Dans ce mémoire, nous avons proposé un système de recommandation de services web basé sur la notion de graphe, afin d'assurer de la nouveauté dans les listes de recommandation Top-N. Des services nouveaux pour l'utilisateur et dissimilaires à son profil sont sélectionnés, en parcourant le graphe et en sélectionnant les services les plus éloignés. Ensuite, ces services dissimilaires sont injectés à la liste de recommandation avec une petite quantité.

Les expériences ont prouvé que la nouvelle approche proposée augmente l'aspect de nouveauté dans les listes de recommandation, mais avec une petite réduction de la performance du système. Ainsi, la nouvelle méthode est avantageuse du faite qu'elle génère des services nouveaux mais qui sont pertinents pour l'utilisateur.

La performance des méthodes appliquées dans les systèmes de recommandation dépend d'un ensemble de données à l'autre et d'un domaine à l'autre, ce qui nous mène à envisager l'application de l'approche proposée sur une base de données de services web pour en tester sa validité réelle dans ce domaine.

Références

[1] Dustdar S, Schreiner W. 'A survey on web services composition', Int. J. Web and Grid Services, 2005, Vol. 1, No. 1, p 2.

[2] Bill N. Schilit and Marvin M. Theimer. Disseminating active map information to mobile hosts.Network, IEEE, 8(5):22-32, Sep/Oct 1994

[3] Goldberg K., Roeder T., Gupta D., Perkins C., "Eigentaste: A Constant Time Collaborative Filtering Algorithm", Information Retrieval, Vol. 4, N°. 2, pp. 133 - 151, 2001

[4] Xiaoyuan Su and Taghi M Khoshgoftaar. A survey of collaborative filtering techniques. Advances in artificial intelligence, 2009 :4, 2009.

[5] K Nageswara Rao. Application domain and functional classification of recommender systems–a survey. DESIDOC Journal of Library & Information Technology, 28(3) :17, 2008.

[6] Negre E., "Comparaison de textes : Quelques approches", Rapports techniques, Université Paris-Dauphine, 2013.

[7] D. Lin. An Information-Theoretic Definition of similarity. In Proceedings of The Fifteenth International Conference on Machine Learning (ICML'98)MorganKaufmann: Madison, WI, 1998.

[8] R. Rada, H. Mili, E. Bichnell et M. Blettner, Development and application of ametric on semantic nets. IEEE Transaction on Systems, 1989.

[9] P. Resnik. Using information content to evaluate semantic similarity intaxonomy. In Proceedings of 14th International Joint Conference on Artificial Intelligence, Montreal, 1995.

[10] D. Lin. An Information-Theoretic Definition of similarity. In Proceedings of The Fifteenth International Conference on Machine Learning (ICML'98). MorganKaufmann: Madison, WI, 1998

[11] J. Jiang et D. Conrath. Semantic similarity based on corpus statistics and lexicaltaxonomy. In Proceedings of International Conference on Research inComputational Linguistics, Taiwan, 1997

[12] C. Leacock et M. Chodorow. Combining Local Context and WordNetSimilarity for Word Sense Identification. In WordNet: An Electronic LexicalDatabase, C. Fellbaum, MIT Press, 1998

[13] Ziegler, C. N., McNee, S. M., Konstan, J. A., & Lausen, G. (2005, May). Improving recommendation lists through topic diversification. In Proceedings of the 14th international conference on World Wide Web (pp. 22-32). ACM.

[14] Shani, G., Chickering, D.M., Meek, C. (2008). Mining recommendations from the web. *In Proc. of the 2008 ACM Conf. RecSys'08*, pp. 35–42.

[15] Castells, P., Vargas, S., & Wang, J. (2011). Novelty and Diversity Metrics for Recommender Systems: Choice, Discovery and Relevance. *Inter. Workshop on DDR at the 33rd ECIR.*

[16] Vargas, S., & Castells, P. (2011). Rank and relevance in novelty and diversity metrics for recommender systems. *In the 5th ACM Conf. RecSyst'11*, pp.109-116. ACM.

[17] Bobadilla, J., Hernando, A., Ortega, F., & Bernal, J. (2011). A framework for collaborative filtering recommender systems. *ESA*, 38(12), pp. 14609-14623.

[18] Zheng, L., Li, L., Hong, W., & Li, T. (2013). PENETRATE: Personalized news recommendation using ensemble hierarchical clustering. *Expert Syst. with App.*, 40(6), pp. 2127-2136.

[19] Chantal Cherifi, « Classification et Composition des services Web : Une Perspective réseaux Complexes », Web, thèse de doctorat, Université Pascal Paoli, 2011, p8.

[20] CRISTINA MARIN, « Une approche orientée domaine pour la composition des services », thèse de doctorat, Université JOSEPH FOURIER GRENOBLE I, 2008, p19.

[21] J-M. Chauvet, Services Web aves SOAP, WSDL, UDDI, ebXML…, Edition EYROLLES;SOAP Version 1.2 Part 2: Adjuncts, W3C Recommendation, M. Gudgin, M. Hadley, N. Mendelsohn, J-J. Moreau, H. Nielsen, 24 June 2003 .

[22] BOUDJELABA Hakim. Sélection des Web Services Sémantiques Université A/Mira de Béjaia, Juin 2012.

[23] Majithia, S., A. S. Ali, O. F. Rana, et D. W. Walker. «Reputation-Based Semantic Service Discover». (2004)

[24] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In WWW10, pages 285–295, Hong Kong. ACM.

[25] Uschold, M., & Grüninger, M. (1996). Ontologies: Principles, Methods and Applications. Knowledge Engineering Review , 11 (2), 93–155

[26] E. Alrifai, T. Risse Combining Global Optimization with local selection for Efficient QoS-aware Service Composition in WWW09, April 20-24, 2009, Madrid, Spain.

[27] Burke, R. (2002). Hybrid recommender systems : Survey and experiments. User Modeling and User-Adapted Interaction, 12(4) :331–370.

Résumé

Les systèmes de recommandation classiques offrent une solution à la quantité croissante du contenu produit chaque jour. Ils fournissent aux utilisateurs des contenus qui sont les plus pertinents pour eux et qui répondent à leurs besoins et préférences. Cependant, avec le temps ça devient une sorte de routine parce que l'être humain aime sortir de l'ordinaire et veut être au courant de tout ce qui est nouveau, par exemple, découvrir de nouveaux produits, tenter de nouvelles expériences.

Dans ce mémoire, nous présentons une nouvelle approche de recommandation de services web, pour assurer de la nouveauté dans les listes recommandées et aider l'utilisateur à découvrir des services qu'il ne connait pas et qui sont nouveaux pour lui. Dans le but de représenter nos services et la relation entre eux, nous avons utilisé la notion de graphe. Ensuite, les nouveaux services sont sélectionnés en cherchant le plus long chemin entre ces services et le profil de l'utilisateur. La liste de recommandation finale générée est un mélange de services dissimilaires et services similaires, où ces derniers sont trouvés après le calcul de la similarité cosinus.

Les résultats expérimentaux sur la base de données MovieLens sont très encourageants et ont montré la faisabilité de la méthode proposée.

Mots clés :

Système de recommandation, Service Web, Filtrage basé Contenu, Nouveauté,

Graphe, Similarité, Dissimilarité, Prédiction, Recommandation Top-N.

Abstract

Traditional recommendation systems offer a solution to the increasing amount of content produced every day. They provide users with content that is most relevant to them and that meets their needs and preferences. However, over time it becomes a kind of routine because human beings like to be out of the ordinary and want to be aware of everything that is new, for example, discovering new products, trying new experiences.

In this brief, we present a new approach to recommending web services, to ensure novelty in recommended lists and to help users discover services they are unfamiliar with that are new to them. In order to represent our services and the relationship between them, we used the notion of graph. Then the new services are selected by looking for the longest path between these services and the user's profile. The final recommendation list generated is a mixture of dissimilar and similar services, where the latter are found after calculating the cosine similarity.

The experimental results on the MovieLens database are very encouraging and have shown the feasibility of the proposed method.

Keywords:

Recommendation system, Web service, Content-based filtering, Novelty, Graph, Similarity, Dissimilarity, Prediction, Top-N recommendation.

ملخّص

تقدم أنظمة التوصية التقليدية حلاً للكمية المتزايدة من المحتوى المنتج كل يوم. أنها توفر للمستخدمين المحتوى الأكثر صلة بهم والذي يلبي احتياجاتهم وتفضيلاتهم. ومع ذلك ، بمرور الوقت يصبح نوعًا من الروتين لأن البشر يحبون أن يكونوا خارج المألوف ويريدون أن يكونوا على دراية بكل ما هو جديد ، على سبيل المثال ، اكتشاف منتجات جديدة ، قيام بتجارب جديدة. في هذا الموجز، نقدم نهجًا جديدًا للتوصية بخدمات الويب ، لضمان التجديد في القوائم الموصى بها ولمساعدة المستخدمين على اكتشاف الخدمات التي لم يعتادوا عليها والتي هي جديدة بالنسبة لهم. من أجل تمثيل خدماتنا والعلاقة بينها ، استخدمنا مفهوم الرسم البياني. ثم يتم اختيار الخدمات الجديدة من خلال البحث عن أطول مسارين هذه الخدمات وملف تعريف المستخدم. قائمة التوصيات النهائية التي تم إنشاؤها هي مزيج من الخدمات المتباينة والمتشابهة ، حيث يتم العثور على الأخيرة بعد حساب تشابه جيب التمام.

النتائج التجريبية على قاعدة بيانات MovieLens مشجعة للغاية وأظهرت جدوى الطريقة المقترحة.

الكلمات المفتاحية :

نظام التوصيات ، خدمة الويب ، التصفية القائمة على المحتوى ، الجدة ، الرسم البياني ، التشابه ، الاختلاف ، التنبؤ ، توصية Top-N