République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique

UNIVERSITE Dr. TAHAR MOULAY SAIDA

FACULTE : TECHNOLOGIE

DEPARTEMENT :

INFORMATIQUE

MEMOIRE

Présenté par :

Fares mohamed

Ameur mohamed

Pour l'obtention du diplôme de
MASTER en Informatique
Filière : Informatique Option :
Réseaux Informatique et Systèmes Réparties

Conception et implémentation d'un système distribué pour La recherche et recommandation sur le web à base Des graphes De connaissances

Soutenue publiquement, le 27/09/ 2021

Devant le jury composé de :

Dr Aissa FELLAH MCA, Université Dr. Tahar Moulay Saida
Dr Kheireddine MEKKAOUI MCA, Université Dr. Tahar Moulay Saida
Dr Ahmed ZAHAF Encadreur MCA Université Dr. Tahar Moulay Saida

Année Universitaire 2020-2021

Remerciements

On remercie le bon dieu, qui nous a donné la force, la volonté et le courage pour terminer ce modeste travail.

On tiens à exprimer d'abord toute notre gratitude à notre encadreur Dr. ***Ahmed ZAHAF*** *maitre de conférence à l'université Dr .MOULAY TAHAR de la wilaya de SAIDA, pour son encadrement, ses conseils, ses directives et encouragements.*

Nos remerciement s'adressent à membres des juré pour l'intérêt qu'ils ont porté à ce travail en acceptant d'être examinateurs.

C'est unénorme remerciement qu'on s'adresse à nosparents et nos sœurs et frères et nos amis.

Merci à tout le corps professoral et administratif de l'université Dr. Moulay Tahar de la wilaya de Saida.

Dédicaces

Jedédiece travail aceux qui m'ont encouragé etsoutenu ; A mes chers parents ;

A mes sœurs et frères ;

A ma femme et mes enfants ;(amira et soundose) ;

A tous mes proches et à ma belle-famille ;

A notre encadreur « Dr. ***Ahmed ZAHAF****» ;*

A mon binôme « Ameur mohamed » ;

Atouslecorpsprofessoraletadministratifdel'universitéDr.Moulay Tahar de la wilaya de Saida ;

A tous nos amis et collègues qu'on a oublié de cité.

Fares

Dédicaces

Je dédiece travail aceux qui m'ont encouragé et soutenu ; A mes chers parents ;

A mes sœurs et frères ;

A tous mes proches et à ma belle-famille ;

A notre encadreur « Dr. ***Ahmed ZAHAF****» ;*

A mon binôme « fares mohamed » ;

A tous le corps professoral et administratif del'université Dr. Moulay Tahar de la wilaya de Saida ;

A tous nos amis et collègues qu'on a oublié de cité.

Ameur

Liste des figures

Liste des tableaux

Table des matières

Chapitre 2

Les graphes de connaissances

Chapitre 3

Conception et Réalisation

Résumé

Les systèmes de recommandation apportent une solution au problème de surcharge d'information et sont capables d'estimer l'intérêt d'un utilisateur pour une ressource donnée à partir de certaines informations relatives à d'autres utilisateurs similaires et aux propriétés des ressources. Dans ce mémoire nous avons présenté un système de recommandation de cours à base d'ontologie. Dans notre approche, le graphe de connaissance de cours est intégré dans le processus de calcul de similarité sémantique entre le profil utilisateur et les descripteurs de cours, et qui est combinée avec similarité numérique, ce qui permet d'améliorer la précision de la recommandation et ainsi mieux répondre aux exigences des utilisateurs. Avec les premiers tests notre système donne des résultats encourageants.

الملخص

توفر أنظمة توصية حل لمشكلة المعلومات الزائد وقادرة على تقدير مصلحة المستخدم لمورد معين من بعض المعلومات عن المستخدمين أخرى مماثلة وخصائص الموارد. في هذا المخطط، قدمنا نظام توصية دورات ذهنية. في نهجنا، تم دمج الرسم البياني بالطبع المعرفة في عملية حساب التشابه الدلالي بين ملف تعريف المستخدم واصفات بالطبع، والتي جنبا إلى جنب مع التشابه الرقمي، مما يجعل من الممكن لتحسين دقة هذه التوصية، وبالتالي تلبية أفضل لمتطلبات المستخدمين. مع التجارب الأولى نظامنا يعطي نتائج مشجعة.

Introduction générale

Un composant du système de recommandation est généralement l'une des principales fonctionnalités de recherche des portails en ligne. Il aide les utilisateurs à découvrir des éléments qui reflètent leurs intérêts. Les recommandations personnalisées sont basées sur un profil utilisateur qui contient des informations de préférence implicites (par exemple, des statistiques d'accès ou un comportement de clic) ou explicites (par exemple, des évaluations pour des éléments).Les techniques SR courantes sont le filtrage collaboratif (FC) ou les algorithmes basés sur le contenu (BC).Les approches de FC dérivent des suggestions d'utilisateurs ayant des goûts similaires, tandis que les méthodes de BC sont basées sur des éléments similaires selon les descriptions de métadonnées .Les descriptions dans les moteurs BC sont souvent structurées dans le format de tableau des paires attribut-valeur. La structure de données plate réduit considérablement la multi-dimensionnalisé des préférences ainsi que les caractéristiques des articles et peut donc produire des recommandations faibles.

C'est pourquoi la structure de données complexe des graphiques RDF dans le cloud LOD (Linked Open Data) peut aider à améliorer la représentation des goûts des utilisateurs dans les moteurs BC.

Prenons l'exemple suivant à titre d'illustration : Supposons qu'un utilisateur ait déclaré qu'il aime un réalisateur en particulier et qu'il aimerait recevoir des suggestions pour d'autres cinéastes intéressants. Un SR conventionnel déterminerait des administrateurs similaires à partir de ces métadonnées. Cependant, les descriptions des réalisateurs peuvent être principalement composées d'informations non pertinentes, par exemple la nationalité ou les prix remportés par les cinéastes. De telles métadonnées ne donneraient pas de résultats utiles dans un système purement basé sur le contenu et n'obtiendraient que quelques résultats aléatoires.

D'autre part, une requête émise contre la collection LOD DBpedia pourrait explorer

le réseau sémantique qui entoure le réalisateur. Par ce moyen, tous les films qui ont été tournés par le réalisateur préféré sont identifiés. D'autres films intéressants, qui n'ont pas été réalisés par le cinéaste, mais qui partagent certaines caractéristiques avec ses films (par exemple, les mêmes genres ou acteurs principaux) pourraient améliorer davantage les descriptions des métadonnées et les informations de profil utilisateur. Le Web de données peut répondre à ces demandes car la pile technologique LOD fournit le protocole SPARQL et le langage de requête RDF (SPARQL) et des moteurs de requête appropriés qui permettent une récupération rapide.

Cependant, les requêtes basées sur des graphes ne suffisent pas encore à elles seules à générer des suggestions personnalisées, car elles ne renvoient des résultats que pour les modèles de graphe qui existent dans le référentiel. Prenons l'exemple suivant : les consommateurs peuvent souhaiter utiliser des moteurs de recommandation qui fonctionnent simultanément sur plusieurs domaines. Par exemple, un utilisateur qui a déclaré qu'il aime certains éléments d'un domaine peut également aimer recevoir des suggestions d'un autre domaine (c'est-à-dire pour la récupération inter-domaines). Dans le cas où un profil utilisateur contient des informations de rétroaction pour les films, il serait souhaitable que le système puisse générer des suggestions de films basées sur ces données. Bien que l'approche consistant à interroger des graphiques RDF puisse être utile pour ce scénario de recommandation (car elle peut identifier les objets correspondants en fonction de déclarations de type d'entité ou d'informations de lien typé), il se peut également qu'il n'y ait pas de liens directs d'un film préféré vers un film approprié dans le graphique (RDF) Resource Description Framework.

Chapitre 1

Les systèmes de recommandation : vue d'ensemble

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 propositions de cours. Ce qui rend la tache 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 filtrage qui permettent la recommandation. Nous enchainons en présentons les différentes mesures de similarité qui permettent aux systèmes de recommandation de faire l'appariement entre les concepts. En fin, nous terminons en citant les limites et inconvénients des systèmes de recommandation ainsi que quelques principaux travaux dans le domaine.

1.2. HistoriQUe

Les systèmes de recommandation ont été utilises afin de faire face au problème de surcharge et de richesse d'informations disponibles notamment a 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 » [1] peut être considère 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 distributions basée sur les groupes d'intérêt.

Quelques années plus tard, un certain nombre de systèmes académiques de recommandation ont vu le jour en 1994 et en 1995, tels que le système de recommandation d'articles d'actualités et de films développe par "GroupLens"

[2] et le système de recommandation de musique "Ringo" propose par [3]. Ces deux systèmes sont également bases sur le filtrage collaboratif, des

livres [2], des vidéos, des films, des pages Web, des articles de nouvelles Usenet et des liens Internet.

Par la suite, avec l'essor de l'Internet et des applications Web, il y a eu un engouement pour les systèmes de recommandation qui se sont développes dans différents domaines d'applications. Nous pouvons en citer :

- Les systèmes de recommandation de films, tels que : Mobiles et Eachmovie.
- Les systèmes de recommandation de livres (Bookcrossing).
- Les systèmes de recommandation de musique (LastFM6).
- Les systèmes de recommandation d'articles d'actualités.
- Les systèmes de recommandation de blagues.
- Les systèmes de recommandations introduits sur des sites e-commerce (Amazon).
- Les systèmes de recommandation de restaurants.
- Les systèmes de recommandation intègres aux Extranets documentaires (l'Extranet documentaire du Crédit Agricole).
- Les systèmes de recommandations intègres aux moteurs de recherche (le moteur de

recherche d'AOL).

- Les systèmes de recommandations implémentent sur des sites de recrutement (Job-Finder).Les systèmes de recommandations de citations bibliographique.

Pour tous les systèmes de recommandation développes jusqu'au nos jours, la collecte de données relatives aux utilisateurs et/ou aux items, représente une phase clé dans le processus de personnalisation. La section qui suit décrit en détails la typologie de données exploitables par les systèmes de recommandation ainsi que les enjeux lies a leur collecte.

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 [4] 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) [5], est un filtre de flux entrant d'information de façon personnalisée pour chaque acteur. Autrement dit, dans un 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'usager et l'item. L'« usager » 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 utilise pour designer ce que le système recommande aux usagers.

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

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

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. Par exemple, dans la figure 1.1, la fonction de décision du système traite le flux entrant de document pour suggérer à l'utilisateur, en consultant son profil, les documents qu'il préfère. A son tour, l'utilisateur doit fournir des évaluations c'est-à-dire évaluer fréquemment les recommandations, pour que le système comprenne mieux

ses besoins en information, et lui fournisse par conséquent de meilleures nouvelles recommandations.

Les trois parties suivantes constituent un système de recommandation :

- Les producteurs : Ce sont ceux qui vont permettre de faire les recommandations, ils "fourniront" les données pour.
- Le module de calcul : Il s'agit de l'algorithme en lui même. En entrée il y a toutes les données et la demande et en sortie les différentes recommandations.
- Le consommateur : C'est celui qui demande la recommandation.

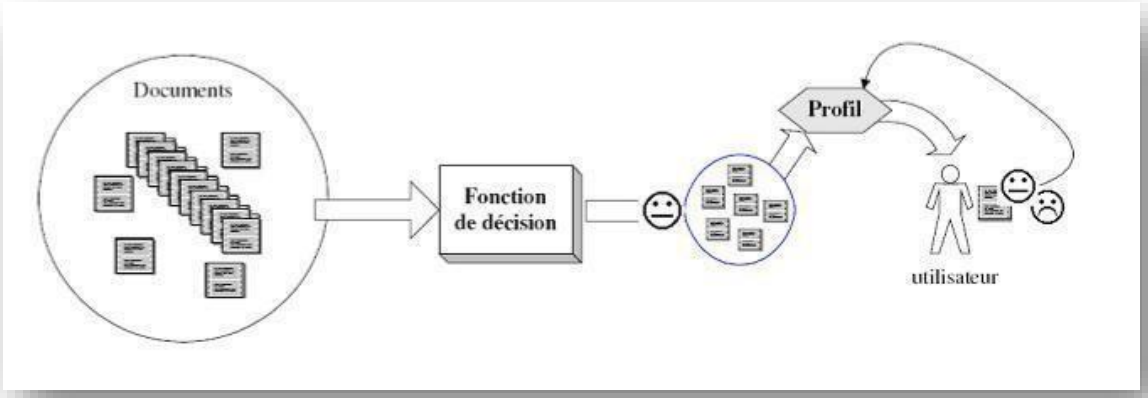


Figure 1.1. Schéma général du filtrage d'information

1.4. Classification des systèmes de recommandation

Il existe plusieurs classifications des systèmes de recommandations (Figure 1.2) :

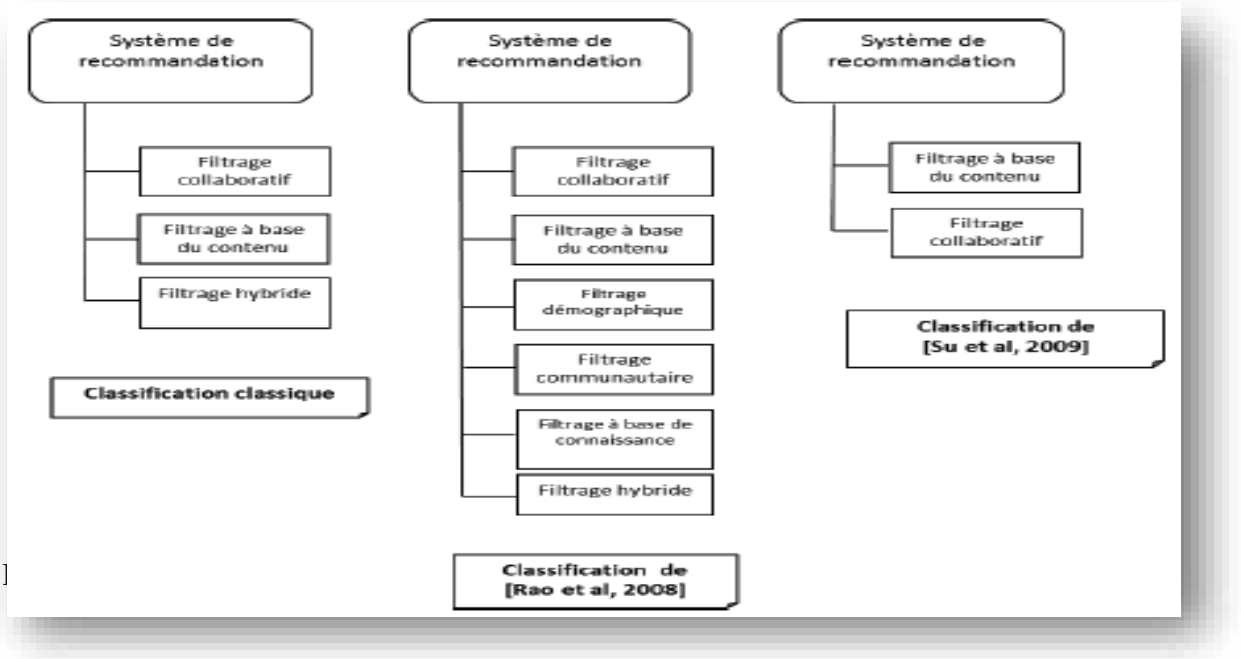
La classification classique : cette classification de [6] est reconnue par trois types de filtrage ; un filtrage collaboratif(CF), un filtrage base sur le contenu(CBF) et le filtrage hybride.

La Classification de [7] : elle est utilisée dans les systèmes collaboration. Ils

proposent une sous- classification qui comprend les techniques hybrides les classer dans les méthodes de collaboration hybrides. [Su et al, 2009] classent filtrage collaboratif en trios categories :

- Approches CF a base de mémoire : pour K-plus proches voisins.
- Approches FC base sur un modèle englobant une variete de techniques telles que: clustering, les réseaux bayesiens, factorisation de matrices, les processus de décision de Markov.
- CF hybride qui combine une technique recommandation CF avec un ou plusieurs autres methodes.

La classification de [8] : c'est une classification en fonction de la source d'information utilisée.



1.5. Différents types de recommandation

Il existe trois grandes approches de filtrage : basé sur le contenu, collaboratif et 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.

1.5.1. Recommandation basé sur le contenu

1.5.1.1. Définitions

Le filtrage basé sur le contenu (Content-based Filtering) [5], 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 des centres d'intérêts. Par

exemple, le profil peut contenir une liste des 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 cognitifs, bases contenu est qu'ils permettent d'associer des documents a 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. Afin de recommander par exemple des films a 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 recommandes a l'utilisateur, sont les plus similaires (En terme d'attribut) aux films consultes 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 base sur le critère thématique uniquement, absence d'autres facteurs comme la qualité scientifique, le public vise, l'intérêt porte par l'utilisateur, etc.
- Les difficultés a recommander des documents multimédia (images, vidéos, etc.) et ceci a cause de la difficulté a indexer ce type de documents, c'est en fait la même problématique dont souffrent les systèmes de recherche.
- Problème de démarrage a 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 ou l'utilisateur fournit des textes exemples.

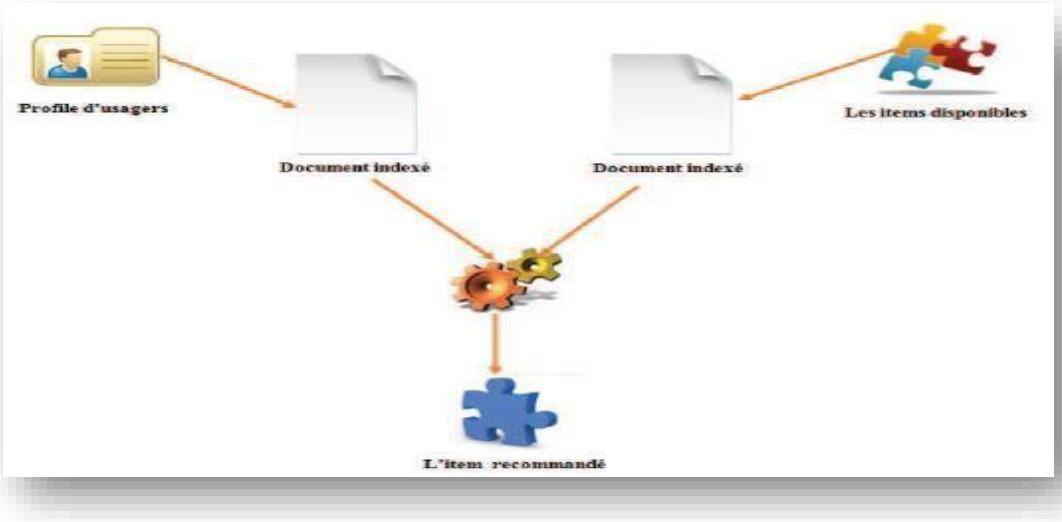


Figure 1.3. Recommandation base sur le contenu.

On distingue deux types de recommandation base sur le contenu : recommandation base sur les mots clefs et recommandation base sur la sémantique.

1.5.1.2. 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, l ivre, etc.) et un profile 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'article : 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.

Similarités 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 (le même) 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

1.5.2.1. Définition

Le filtrage collaboratif (Collaborative Filtering « CF ») 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.

De plus, l'utilisateur est capable de découvrir divers domaines intéressants, car le principe du filtrage collaboratif ne se fonde absolument pas sur la dimension thématique des profils, et n'est pas soumis à l'effet « entonnoir ».

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. 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 données ci-dessous :

1.5.2.2.1. Evaluation des recommandations

Selon le principe de base du filtrage collaboratif, les utilisateurs doivent fournir leurs évaluations sur des documents afin que le système forme les communautés. Evaluer une recommandation peut se faire de façon explicite ou implicite, comme suit :

- Explicite : L'utilisateur donne une valeur numérique sur une echelle donné (par exemple de 1 à 5, ou de 1 à 10, etc.), ou bien, une valeur qualitative de satisfaction, par exemple, mauvaise, moyenne, bonne et excellente.
- Implicite : Le système induit la satisfaction de l'utilisateur à travers ses actions. Par exemple, le système estimera qu'une recommandation supprimée correspond à une évaluation tés mauvaise, alors qu'une recommandation imprimée ou sauvegardée peut être interprétée comme une bonne évaluation.

1.5.2.2.2. Formation des communautés

Le processus de formation des communautés est le noyau d'un système de filtrage collaboratif. Pour chaque utilisateur, le système doit calculer sa communauté, généralement cela se fait par la proximité des évaluations des utilisateurs. Pour ce faire, on peut calculer, dans un premier temps, la proximité entre un utilisateur donné et tous les autres. Ensuite, et afin de créer

contrairement la communauté de l'utilisateur, on applique la méthode des voisins les plus proche en utilisant un seuil pour le niveau de proximité ou un seuil pour la taille maximale de la communauté, en raison de sa performance et sa précision.

1.5.2.2.3. ProdUCtion des recommandations

Dans ce derniers 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.

1.5.2.2.4. Profils et commUnaUTés

Ici, nous discutons les profils bases sur l'historique des évaluations des utilisateurs, ainsi que les communautés, qui sont les deux facteurs clés d'SFC. Le problème de la surcharge d'information peut être pallié par la personnalisation de l'accès aux informations, en utilisant des profils représentant des intérêts relativement stables des utilisateurs. En d'autres termes les profils des utilisateurs sont utilisés comme des critères persistant dans la recherche d'information

a-) Profil de l'Utilisateur :

Le profil utilisateur est composé de prédicats pondérés. Le poids d'un prédicat exprime son intérêt relatif pour l'utilisateur. Il est spécifié par un nombre réel compris entre 0 et 1. Le profil s'enrichit progressivement

au fur et à mesure que l'utilisateur évalue des documents reçus. Outre les informations d'identification de base (par exemple, l'identifiant ou des éléments d'état civil), le profil de l ' utilisateur peut regrouper des informations très diverses selon les besoins.

Parmi celles-ci, nous pouvons citer :

- Des caractéristiques personnelles pouvant influencer fortement l'interaction (âge, sexe, ….)
- Les intérêts et les préférences générales de l'utilisateur relatives à la tâche à accomplir, qui permettent une adaptation à ses attentes.
- Qualité. Cette dimension contient tous les facteurs reflétant les préférences relatives à la qualité de l'information, comme la disponibilité de données, la concision, le style et la structure du document, etc. Dans cette dimension, nous nous intéressons en particulier à diversité de l'information.
- Sécurité. La dimension de sécurité dans le contexte du filtrage collaboratif, est le niveau de confidentialité concernant tous les autres critères.
- Un historique des interactions avec le service, qui peuvent permettre de modéliser les habitudes comportementales.

b-) Communautés

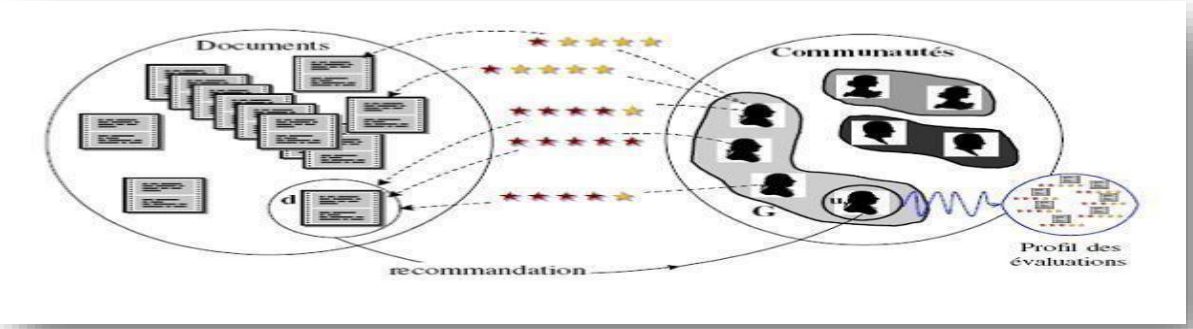
La notion de communauté dans un système de filtrage collaboratif est définie comme le regroupement des utilisateurs en fonction de l'historique de leurs évaluations, afin que le système calcule des recommandations. Selon cette

optique, les profils sont un facteur interactif, alors que les communautés sont considérées comme un facteur interne du système.

1.5.2.3. Exemple

Dans la figure 1.4, nous schématisons le principe du filtrage collaboratif. On suppose que l'on a des communautés formées par la proximité des évaluations des utilisateurs. Le document d sera recommandé à l'utilisateur u, car ce document est apprécié par la communauté G où se l'utilisateur.

Figure 1.4. Principe général du filtrage collaboratif



1.5.3. 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 [9].

Plus généralement, les systèmes hybrides gèrent des profils d'utilisateurs

orientes contenu, et la comparaison entre ces profils donne lieu a la formation de communautés d'utilisateurs permettant le filtrage collaboratif. La meilleure description des méthodes hybrides a été faite par [4].Alors, selon Burke on peut distinguer sept façons de combiner les méthodes traditionnelles : Pondération (Weighted)

Une méthode hybride qui combine la sortie d'approches distinctes, utilisant, par exemple, une combinaison linéaire des scores de chaque technique de recommandation. *Commutation (Switching)*

C'est une technique qui permet de faire le choix d'un modèle de recommandation parmi plusieurs, en se basant sur plusieurs critères. La détermination de la technique appropriée dépend de la situation. Le système se doit alors de définir les critères de commutation, ou les cas ou l'utilisation d'une autre technique est recommandée. Ceci permet au système de connaitre les points forts et les points faibles des techniques de recommandation qui le constituent.

Technique mixte (Mixed)

Dans cette approche, le recommandeur ne combine pas, mais augmente la description des ensembles de données, en prenant en considération les estimations des utilisateurs et la description des items. La nouvelle fonction de prédiction doit faire face aux deux types de descriptions et permetd'éviter les problèmes poses par le filtrage collaboratif, a savoir, le démarrage a froid.

Combinaison de caractéristiques (Features combination)

Dans un hybride base sur la combinaison de caractéristiques, les données provenant de techniques collaboratives sont traitées comme une caractéristique, et une approche basée sur le contenu est utilise sur ces

données *Cascade.*

La cascade implique un processus étape par étape. Dans ce cas, une technique de recommandation est appliquée en premier, produisant un ensemble de candidats potentiels.

Puis, une deuxième technique raffine les résultats obtenus dans la première étape. Cette méthode a pour avantage que si la première technique génère peu de recommandations, ou si ces recommandations sont ordonnées afin de permettre une sélection rapide, la deuxième technique ne sera plus utilisée.

Augmentation de caractéristiques (Feature augmentation)

L'augmentation de caractéristiques est semblable a la cascade, mais dans ce cas-la les résultats obtenus (le classement ou la classification) de la première technique sont utilisées par le deuxième comme une caractéristique ajoutée.

Méta niveau (Meta-level)

Dans un hybride base sur méta niveau, une première technique est utilisée, mais différemment que la précédente méthode (augmentation de caractéristiques), non pas pour produire de nouvelles caractéristiques, mais pour produire un modèle. Et dans la deuxième étape, c'est le modèle entier qui servira d'entrée pour la deuxième technique [10].

Comme nous l'avons déjà mentionné précédemment, au cœur de la plupartdes systèmes de recommandation, nous trouvons 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. Comme les profils utilisateurs et les descripteurs de contenu sont souvent modélisés avec des vecteurs de mots clés pondérés, seules les mesures

vectorielles comme Cosinus et corrélation de Pearson sont utilisées. Or, l'avènement du web sémantique et le développement des ontologies ont mis à notre disposition une panoplie de mesures de similarité sémantiques qui peuvent compléter les mesures vectorielles. Nous avons jugé important de présenter quelques mesures de similarité des plus connues. La section suivante montre une classification de ces mesures.

1.6. Classification des approches de mesure de similarité

Dans cette classification, nous distinguons quatre grandes catégories de mesures de similarité.

1.6.1. Approches basées sur l'espace vectoriel

Dans le domaine de la recherche de l'information, les modèles de l'espace vectoriel sont largement adoptés, on parlera alors de similarité numérique. Ces approches [11] utilisent un vecteur caractéristique, dans un espace dimensionnel, pour représenter chaque objet et calculent la similarité numérique en se basant sur la mesure de cosine ou la corrélation de Pearson. Parmi les approches citées dans la littérature on peut citer :

1.6.4.1. Similarité de Cosine

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.

$$Sim\,(X,Y) = \cos\,(X,Y) = \frac{X * Y}{\|X\|_2 * \|Y\|_2}$$

La mesure de Cosine [11] quantifie donc la similarité numérique entre les deux vecteurs, comme le cosinus de l'angle entre les deux vecteurs.

1.6.4.2. *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 :

$$\text{Sim (x,y)} = r_p = \frac{\sum_{i=1}^{N}(x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}}$$

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:

est la moyenne de X

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

est la moyenne de Y

$$\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$$

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 [12] est basée sur le principe suivant : Etant donnée une ontologie formée d'un ensemble de nœud et un nœud racine R (Figure 1.5). 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.

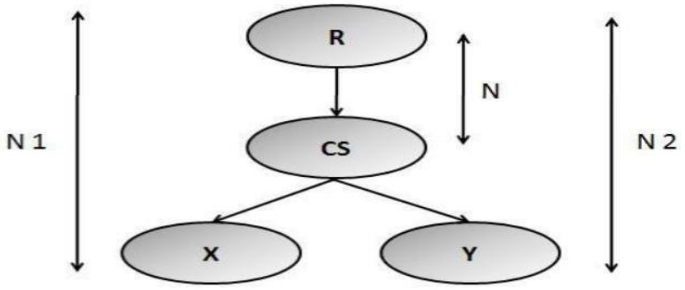


Figure 1.5. Exemple d'un extrait d'une ontologie

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

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

1.6.1.2. 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

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

«is-a». Pour calculer la similarité de concepts dans une ontologie, on doit calculer le nombre des arcs minimums qui les séparent. Cette mesure [11], 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. Il présente ainsi une mesure utilisant une métrique, distance(c1, c2), qui indique le nombre d'arcs minimum à parcourir pour aller d'un concept c1 à un concept c2.

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 : $\boldsymbol{E(c) = -log(Pc)}$

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'ne 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 [13] 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 1.5, le PPG des concepts X et Y

est le concept CS.

$$Sim\ (c1, c2) = Max\ [CI\ (PPG\ (c1, c2))]$$ Où $$CI = -log\ (P\ (c))$$

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 [14]. Cette mesure est légèrement différente de celle de Resnik :

$$Sim\ (a, b) = \frac{2 * CI\ (PPG(a, b))}{CI\ (a) + CI\ (b)}$$

1.6.2.3. 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 [14]. Cette mesure est légèrement différente de celle de Resnik :

$$Sim\ (a, b) = \frac{2 * CI\ (PPG(a, b))}{CI\ (a) + CI\ (b)}$$

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 [15]

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 es définie par l'inverse de la distance sémantique.

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

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

$$Distance\,(c1, c2) = CI\,(c1) + CI\,(c2) - (2.CI\,(PPG\,(c1, c2)))$$

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 [11]. 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 :

$$Sim\,(X, Y) = -Log\left(\frac{CD(X, Y)}{2 * M}\right)$$

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.

1.7.Avantages et inconvénients des systèmes de recommandation

Le tableau 1.1 résume les forces et faiblesses des méthodes traditionnelles utilise par les systèmes de recommandation, en l'occurrence le Filtrage Collaboratif (FC), le Filtrage Démographique (FD), le Filtrage à Base de Contenu (FBC), et le Filtrage à base de données communautaires.

Adaptabilité : Au fur et à mesure que la base de données des évaluations augmente, la recommandation devient plus précise.

Nouvel utilisateur : un nouvel utilisateur qui n'a pas encore accumule suffisamment d'évaluations ne peut pas avoir de recommandations pertinentes.

Nouvel item : un item doit avoir suffisamment d'évaluations pour qu'il soit pris en considération dans le processus de recommandation.

Démarrage à froid : le démarrage a froid est un problème pour les nouveaux utilisateurs qui commencent a jouer avec le système, parce que le système ne dispose pas d'assez d'informations a leur sujet. Si le profil d'utilisateur est vide, il doit consacrer une somme d'efforts à l'aide du système avant d'obtenir une récompense (les recommandations utiles).

D'autre part, quand un nouvel item est ajoute a la collection, le système doit avoir suffisamment d'informations pour être en mesure de recommander cet item aux utilisateurs.

Techniques	Avantages	Inconvénients
Filtrage démographique	N'exige aucun historique d'estimations.	• Problème de confidentialité. • Utilisateur avec un goût unique. • Nouvel Item.
Filtrage à base de données communautaire	Adaptabilité : la qualité croit avec le nombre d'amis.	• Nouvel utilisateur. • Nouvel item.
Filtrage à base du contenu	• Pas besoin d'une large communauté d'utilisateurs pour pouvoir effectuer des recommandations. • Une liste de recommandations peut être générée même s'il n'y a qu'un seul utilisateur. • La qualité croit avec le temps. • Pas besoin d'information sur les autres utilisateurs. • Prendre en considération les gouts uniques[37] des utilisateurs.	• L'analyse du contenu est nécessaire pour faire une recommandation. • Problème de recommandation des images et de vidéos en absence de Métadonnées. • Nécessité du profil d'utilisateur.
Filtrage collaboratif	• Ne demande aucune connaissance sur le contenu de l'item ni sa sémantique. • La qualité de la recommandation peut être évaluée. • Plus les nombre d'utilisateurs est grand plus la recommandation est meilleure.	• Démarrage à froid. • Nouvel Item. • Nouvel utilisateur. • Problème de confidentialité. • La complexité : dans les systèmes avec un grand nombre d'items et d'utilisateurs, le calcul croit linéairement.

Table 1.1 – **Les avantages et les inconvénients des techniques de recommandations.**

1.8. ConcLUsion

Dans ce chapitre, nous avons d'abord, présenté la notion des systèmes de recommandation, en détaillons les trois approches les plus utilisées, à savoir, l'approche FNC, FC et hybride. Ensuite, nous avons défini la notion de profil utilisateur. Nous avons également passé en revue les différentes classes de mesures de similarité utilisées par les SRs pour faire le matching 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.

Chapitre 2

Les graphes de connaissances

1. Introduction

Les graphes de connaissances ont émergé comme une abstraction convaincante pour organiser les connaissances structurées du monde sur Internet et un moyen d'intégrer les informations extraites de plusieurs sources de données. Les graphes de connaissances ont également commencé à jouer un rôle central dans l'apprentissage de la machine en tant que méthode pour intégrer des connaissances mondiales, comme une représentation des connaissances ciblées pour les connaissances extraites et pour expliquer ce qui est appris.

Notre objectif ici est d'expliquer la terminologie de base, les concepts et l'utilisation des graphes de connaissances de manière simple à comprendre. Nous n'avons pas l'intention de donner ici une enquête exhaustive sur le passé et le travail actuel sur le sujet des graphes de connaissances.

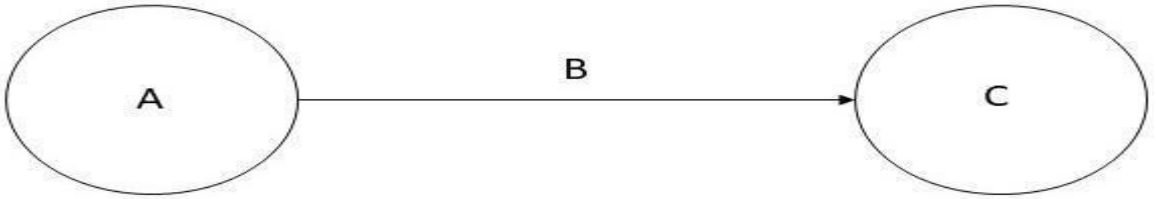
Nous commencerons par définir des graphes de connaissances, certaines applications qui ont contribué à la récente surtension de la popularité des graphes de connaissances, puis d'utiliser des graphes de connaissances dans l'apprentissage de la machine. Nous conclurons cette note en résumant ce qui est nouveau et différent de l'utilisation récente des graphes de connaissances.

2. Définition du graphe de connaissances

Un graphe de connaissances est un graphe marqué dirigé dans lequel les étiquettes ont des significations bien définies. Un graphe marqué dirigé est composé de nœuds, de bords et d'étiquettes. Tout peut agir comme un nœud, par exemple, des personnes, de la société, de l'ordinateur, etc. Un EDGE relie une paire de nœuds et capture la relation d'intérêt entre eux, par exemple, une relation d'amitié entre deux personnes, la relation client entre une entreprise et une personne ou une connexion réseau entre deux ordinateurs. Les étiquettes capturent la signification de la relation, par exemple la relation d'amitié entre deux personnes.

Plus formellement, étant donné un ensemble de nœuds n, et un ensemble d'étiquettes L, un graphe de connaissances est un sous-ensemble du produit croisé N
× l × n. Chaque membre de cet ensemble est appelé triple et peut être visualisé comme

indiqué ci-dessous.



La représentation graphe dirigée est utilisée de différentes manières en fonction des besoins d'une application. Un graphe dirigé tel que celui dans lequel les nœuds sont des personnes et les bords capturent

La relation d'amitié est également appelée graphe de données. Un graphe dirigé dans lequel les nœuds sont des classes d'objets (par exemple, livre, manuel, etc.), et les bords capturent la relation de sous-classe, sont également appelés taxonomie. Dans certains modèles de données, A est appelé sujet, B est appelé prédicat et C est appelé objet.

De nombreux calculs intéressants sur les graphes peuvent être réduits pour naviguer sur le graphe. Par exemple, dans un graphe de connaissances de l'amitié, pour calculer les amis d'un ami d'une personne A, nous pouvons naviguer dans le graphe de connaissances à partir de A à tous les nœuds b connectés par une relation étiquetée comme ami, puis à tous les nœuds C relié par l'ami relationnelle à chaque B.

Un chemin dans un graphe g est une série de nœuds (v1, v2, ..., vn) où pour tout i ∈ n avec $1 \leq i < N$, il y a un bord de VI à VI + 1. Un chemin simple est un chemin sans nœuds répétés. Un cycle est un chemin dans lequel le premier et les derniers nœuds sont les mêmes. Habituellement, nous ne sommes intéressés que par les chemins dans lesquels l'étiquette de bord est la même pour chaque paire de nœuds. Il est possible de définir de nombreuses propriétés supplémentaires sur les graphes (par exemple, des composants connectés, des composants fortement connectés) et de fournir différentes manières de traverser les graphes (par exemple, chemin le plus court, chemin hamiltonien, etc.).

3. Applications récentes des graphes.

Il existe de nombreuses applications de graphe de connaissances à la fois dans la recherche et l'industrie. Au sein de l'informatique, il existe de nombreuses utilisations d'une représentation graphe dirigée, par exemple, des graphes de flux de données, des diagrammes de décision binaires, des diagrammes d'État, etc. pour

notre discussion ici, nous avons choisi de se concentrer sur deux applications concrètes qui ont conduit à une augmentation récente En popularité des graphes de la connaissance: organiser des informations sur Internet et l'intégration des données.

3.1 graphe pour organiser des connaissances sur Internet.

Nous expliquerons l'utilisation d'un graphe de connaissances sur le Web en prenant l'exemple concret de Wikidata. Wikidata sert de stockage central pour les données structurées de Wikipedia. Pour montrer l'interaction entre les deux et motiver l'utilisation du graphe de connaissances Wikidata, tenez compte de la ville de Winterthur en Suisse, qui a une page à Wikipedia. La page Wikipedia pour Winterthur répertorie ses jumeaux Twin: deux sont à Switerzland, une en République tchèque et une en Autriche. La ville de l'Ontario en Californie dispose d'une page Wikipedia intitulée Ontario, Californie, répertorie Winterthur comme ville sœur. Les relations de la ville sœur et de la ville jumeaux sont identiques et réciproques. Ainsi, si une ville a est une ville sœur d'une autre ville b, alors B doit être une ville sœur d'A. Cette inférence doit être automatique, mais que ces informations sont indiquées en anglais à Wikipedia, il n'est pas facile de détecter cette divergence . En revanche, dans la représentation de Winkthur de Wikidata, il existe une relation appelée organe administratif jumelé qui énumère la ville de l'Ontario. Comme cette relation est symétrique, la page Wikidata de la ville de l'Ontario comprend automatiquement Winterthur. Ainsi, lorsque le graphe de connaissances Wikidata sera entièrement intégré à Wikipedia, de telles divergences disparaîtront naturellement.

Wikidata inclut des données de plusieurs fournisseurs indépendants, par exemple la bibliothèque du Congrès qui publie des données contenant des informations sur Winterthur. En utilisant l'identifiant Wikidata pour Winterthur, les informations publiées par la bibliothèque du Congrès peuvent être facilement liées à des informations disponibles sur d'autres sources. Wikidata facilite la création de tels liens en publiant les définitions de relations utilisées dans Schema.org.

Le vocabulaire des relations dans Schema.org nous donne au moins trois avantages. Premièrement, il est possible d'écrire des requêtes qui s'étendent sur plusieurs jeux de données qui n'auraient pas été possibles autrement.

Un exemple d'une telle requête est: afficher sur une carte les villes de naissance des personnes décédées à Winterthour? Deuxièmement, avec une telle capacité de requête, il est possible de générer facilement structuré

boîtes d'information au sein de Wikipedia. Troisièmement, les informations structurées renvoyées par requêtes peuvent également apparaître dans les résultats de la recherche désormais une fonctionnalité standard pour les principaux moteurs

de recherche.

Une version récente de Wikidata comptait plus de 80 millions d'objets, avec plus d'un milliard de relations entre ces objets. Wikidata établit des connexions sur plus de 4872 catalogues différents en 414 langues différentes publiées par des fournisseurs de données indépendants. Selon l'estimation récente, 31% des sites Web et plus de 12 millions de fournisseurs de données publient des annotations Schema.org utilisent actuellement le vocabulaire de Schema.org.

Observons plusieurs caractéristiques essentielles du graphe e de connaissances Wikidata. Premièrement, il s'agit d'un graphe de l'échelle sans précédent et est le plus grand graphes de connaissances disponible aujourd'hui. Deuxièmement, il est créé conjointement par une communauté de contributeurs. Troisièmement, certaines des données de Wikidata peuvent provenir d'informations extraites automatiquement, mais elles doivent être facilement comprises et vérifiées conformément aux politiques éditoriales Wikidata. Quatrièmement, il existe un effort explicite pour fournir des définitions sémantiques de noms de relation différents par le vocabulaire de Schema.org. Enfin, le principal cas d'utilisation de conduite pour Wikidata est d'améliorer la recherche sur le Web. Même si Wikidata a plusieurs applications l'utilisant pour des tâches analytiques et de visualisation, mais son utilisation sur le Web continue d'être l'application la plus convaincante et la plus comprise.

3.2 Graphes pour l'intégration des données dans les entreprises

L'intégration des données est le processus de combinaison des données de différentes sources et de fournir à l'utilisateur une vue unifiée des données. Une grosse fraction de données dans les entreprises réside dans les bases de données relationnelles. Une approche de l'intégration des données repose sur un schéma global qui capture les interrelations entre les éléments de données représentés dans toutes ces bases de données. La création d'un schéma global est un processus extrêmement difficile car il existe de nombreuses tables et attributs; Les experts qui ont créé ces bases de données ne sont généralement pas disponibles; Et en raison du manque de documentation, il est difficile de comprendre la signification des données. En raison des défis de la création d'un schéma mondial, il est pratique de passer cette question et de convertir les données relationnelles en une base de données avec le schéma générique de triples, c'est-à-dire un graphe de connaissances. Les mappages entre les attributs sont créés au besoin, par exemple, en réponse à la résolution de questions commerciales spécifiques et peuvent elles-mêmes être représentées dans un graphe de connaissances. Nous illustrons ce processus en utilisant un exemple concret.

De nombreuses institutions financières souhaitent créer un graphe de connaissances de la société qui combine les données client internes avec les données sous licence de tiers. Certains exemples de jeux de données tiers tels que Dunn & Bradstreet, S & P 500, etc. Un exemple d'utilisation d'un graphe d'entreprise consiste à évaluer le risque tout en prenant des décisions de prêt. Les données externes contiennent des informations telles que les fournisseurs d'une entreprise. Si une entreprise passe par des difficultés financières, elle augmente le risque d'attribuer un prêt aux fournisseurs de cette société. Pour combiner cette donnée externe avec les données internes, il faut relier les schémas externes avec le schéma de la société interne. En outre, les noms de société utilisés dans les sources externes doivent être liés aux identificateurs clients correspondants utilisés par les institutions financières. Tout en utilisant une approche graphe de connaissances en matière d'intégration des données, la détermination de ces relations peut être retardée jusqu'à ce qu'elles soient réellement nécessaires.

4. Graphes dans l'intelligence artificielle

Les graphes de connaissances, appelés réseaux sémantiques, ont été utilisés comme une représentation pour l'intelligence artificielle depuis les premiers jours du domaine. Au fil des ans, des réseaux sémantiques ont été évolus dans différentes représentations telles que des graphes conceptuels et des logiques de description. Pour capturer des connaissances incertaines, des modèles graphe probabilistes ont été inventés.

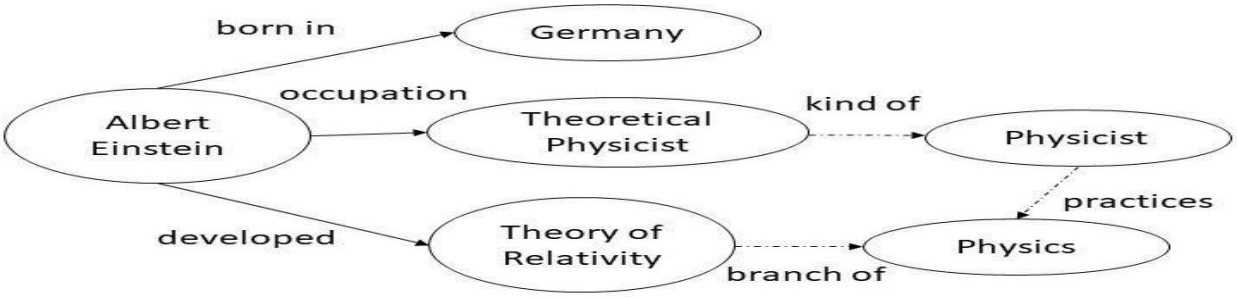
Orthogonal à la représentation des connaissances, un défi central dans l'AI est un goulot d'étranglement de la connaissance, c'est-à-dire comment capturer les connaissances dans la représentation choisie de manière économiquement évolutive. Les premières approches reposaient sur l'ingénierie des connaissances. Les efforts visant à automatiser les portions de génie des connaissances ont conduit à des techniques telles que l'apprentissage inductif et la génération actuelle d'apprentissage de la machine.

Par conséquent, il est naturel que les graphes de connaissances soient utilisés comme une représentation de choix pour stocker automatiquement les connaissances acquises. Il existe également un intérêt croissant pour tirer parti des connaissances du domaine exprimées dans des graphes de connaissances pour améliorer l'apprentissage de la machine.

1.1 Graphes comme sortie de l'apprentissage de la machine

Nous examinerons la manière dont les graphes sont utilisés comme une représentation de sortie cible pour le traitement des langues naturelles et les algorithmes de vision informatique.

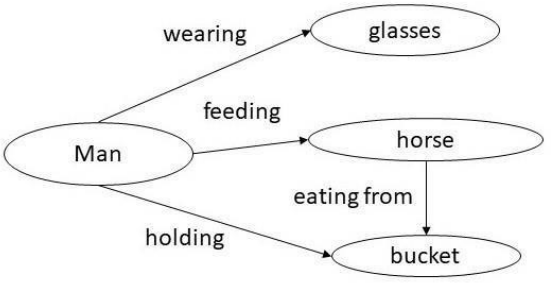
L'extraction de l'entité et la relation extraction du texte sont deux tâches fondamentales dans le traitement des langues naturelles. Les informations extraites de plusieurs parties des besoins de texte doivent être corrélées et les graphes fournissent un support naturel pour accomplir un tel objectif. Par exemple, de la phrase indiquée à gauche, nous pouvons extraire les entités Albert Einstein, l'Allemagne, le physicien théorique et la théorie de la relativité; et les relations nées dans, profession et développées. Une fois que cet extrait de graphe est incorporé dans un graphe plus grand, nous obtenons des liens supplémentaires (montrés par des bords en pointillé) tels qu'un physicien théorique est une sorte de physicien qui pratique la physique et que la théorie de la relativité est une branche de la physique.



elativité

t-à-dire

re leurs relations. Comprendre des scènes permettrait d'applications importantes telles que la recherche d'images, la réponse à la question et les interactions robotiques. De nombreux progrès ont été accomplis ces dernières années vers cet objectif, y compris la classification de l'image et la détection d'objet.



Par exemple, à partir de l'image indiquée ci-dessus, un système de compréhension d'image doit produire un graphe montré à droite. Les nœuds du graphe sont les sorties d'un détecteur d'objet. Les recherches actuelles sur la vision informatique se concentrent sur les techniques de développement pouvant en déduire correctement les relations entre les objets, tels que l'homme tenant un seau et l'alimentation des chevaux du seau, etc. Le graphe montré à droite est un exemple de graphe de connaissances, et conceptuellement, cela aime les graphes de données que nous avons introduits plus tôt.

4.2 Graphes comme entrée à l'apprentissage de la machine

Les modèles populaires d'apprentissage des machines profondes s'appuient sur une entrée numérique qui nécessite d'abord que toutes les structures symboliques ou discrètes soient d'abord converties en une représentation numérique. Les embarquements qui transforment une entrée symbolique en un vecteur de chiffres ont émergé comme une représentation de choix pour la saisie des modèles d'apprentissage de la machine. Nous allons expliquer ce concept et sa relation avec des graphes de connaissances en prenant l'exemple des embarcations de mots et des embarcations de graphe

Les embarcations de mots ont été développées pour calculer la similitude entre les mots. Pour comprendre le mot embarcation, nous considérons l'ensemble de phrases suivant.

J'aime les graphes de la connaissance. J'aime
les bases de données.
J'aime courir.

Dans le jeu de phrases ci-dessus, nous compterons à quelle fréquence un mot apparaît à côté d'un autre mot et enregistre les comptes dans une matrice présentée ci-dessous. Par exemple, le mot que je présente à côté du mot comme deux fois, et à côté de Word, profitez-en une fois, et donc, ses comptes pour ces deux mots sont respectivement 2 et 1, et 0 pour tout autre mot. Nous pouvons calculer les comptes pour d'autres mots de la même manière pour remplir la table.

comptes	je	aimer	prendre plaisir	connaissances	graphes	Bases de données	Fonction Nement	.
Je	0	2		0	0	0	0	0
aimer	2	0		1	0	1	0	0
prendre plaisir	1	0		0	0	0	1	0
connaissances	0	1		0	1	0	0	0
graphes	0	0		1	0	0	0	1
bases de données	0	1		0	0	0	0	1
fonctionnement	0	0		0	0	0	0	1
.	0	0		0	1	1	1	0

Table 2.1. constitue une matrice qui est souvent appelée compte de cooccurrence de mots

Le tableau ci-dessus constitue une matrice qui est souvent appelée compte de cooccurrence de mots. Nous disons que la signification de chaque mot est capturée par le vecteur de la ligne correspondant à ce mot. Pour calculer la similitude entre les mots, nous pouvons simplement calculer la similitude entre les vecteurs qui leur correspondent. En pratique, nous sommes intéressés par un texte pouvant contenir des millions de mots et une représentation plus compacte est souhaitée. Comme la matrice ci-dessus est clairsemée, nous pouvons utiliser des techniques d'algèbre linéaire (par exemple, décomposition de la valeur singulière) pour réduire ses dimensions. Le vecteur résultant correspondant à un mot est connu sous le nom d'incorporation de mots. Parole typique Les intégres utilisées s'appuient aujourd'hui sur des vecteurs de longueur 200. Il existe de nombreuses variations et extensions de l'idée de base présentée ici. Les techniques existent pour apprendre automatiquement les embarcations de mots pour tout texte donné.

L'utilisation d'embarcations de mots a été constatée pour améliorer les performances de nombreuses tâches de traitement des langues naturelles, y compris l'extraction de l'entité, l'extraction de relation, l'analyse, la récupération de

passage, etc. L'une des applications les plus courantes de Word Embedings est l'achèvement automatique des requêtes de recherche.

Les embarcations de mots nous donnent un moyen simple de prédire les mots susceptibles de suivre la requête partielle qu'un utilisateur a déjà saisi.

Comme un texte est une séquence de mots et que les embarcations de mots calculent des co-occurrences de mots, nous pouvons afficher le texte sous forme de graphe dans lequel chaque mot est un nœud, et il y a un bord dirigé entre chaque mot et un autre mot qui suit immédiatement. Graphe Embeddings Généralisez cette notion pour la structure générale du réseau. L'objectif et l'approche, cependant, reste la même chose: représenter chaque nœud dans un graphe par un vecteur, de sorte que la similitude entre les nœuds puisse être calculée comme une différence entre leurs vecteurs correspondants. Les vecteurs pour chaque nœud sont également appelés graphes embarquant.

Pour calculer des embarcations graphe, nous définissons une méthode pour encoder chaque nœud dans le graphe dans un vecteur, une fonction pour calculer la similitude entre les nœuds, puis optimiser la fonction de codage. Une fonction de codage possible consiste à utiliser une promenade aléatoire du graphe et de calculer les comptes de coordination des nœuds sur le graphe donnant une matrice similaire à la co-occurrente Nombre de mots dans le texte. Il existe de nombreuses variations de cette méthode de base pour calculer des embarcations graphes.

Nous avons choisi d'expliquer les embarcations de graphe en expliquant d'abord des embarcations de mots car, car il est facile de les comprendre, et leur utilisation est un lieu commun. Les embarcations graphes sont une généralisation du mot embarcation. Ils sont un moyen de saisir les connaissances du domaine exprimées dans un graphe de connaissances dans un algorithme d'apprentissage de la machine. Les embarcations de graphe ne font pas induire une représentation des connaissances, mais sont un moyen de transformer la représentation symbolique en une représentation numérique à la consommation par un algorithme d'apprentissage de la machine.

Une fois que nous avons calculé des embarcations graphes, ils peuvent être utilisés pour une variété d'applications. Une utilisation évidente pour les embarcations graphes calculées à partir d'un graphe d'amitié consiste à recommander de nouveaux amis. Une tâche plus avancée implique une prédiction de liaison (c.-à-d. La probabilité d'un lien entre deux nœuds), etc. La prédiction de liaison dans un graphe de la société pourrait être utilisée pour identifier de nouveaux clients potentiels.

5. Résumé

Les graphes sont une construction fondamentale de mathématiques discrètes et disposent d'applications dans tous les domaines de l'informatique. Les utilisations les plus notables des graphes de la représentation des connaissances et des bases de données ont pris la forme de graphe de données, de taxonomies et d'ontologies. Traditionnellement, de telles applications ont été conduites par une conception descendante. Comme un graphe de connaissances est un graphe marqué dirigé, nous sommes en mesure de tirer parti de la théorie, des algorithmes et des implémentations de systèmes de graphe plus généraux en informatique.

Une augmentation récente de l'utilisation de graphe de connaissances pour organiser des informations sur Internet, l'intégration des données et une cible de sortie pour l'apprentissage de la machine, est principalement entraînée de manière d'une manière de revers. Pour organiser des informations sur le Web, et dans de nombreuses applications d'intégration de données, il est extrêmement difficile de proposer une conception de haut en bas d'un schéma. Les applications d'apprentissage de la machine sont motivées par la disponibilité des données et ce qui peut être utilement inféré de celui-ci. Les utilisations ascendantes des graphes de connaissances ne diminuent pas la valeur d'une conception de haut niveau du schéma ou d'une ontologie. En effet, le projet Wikidata exploite des ontologies pour assurer la qualité des données et la plupart des projets d'intégration des données de l'entreprise préconisant la définition du schéma au besoin.

Les applications d'apprentissage des machines bénéficient également de manière significative avec l'utilisation de la riche ontologie pour rendre les déductions des informations apprises même si une ontologie mondiale ou un schéma n'est pas nécessaire au début.

Les embarcations de mots et les graphes-intégres tirent à la fois une structure de graphe dans les données d'entrée, mais elles sont nécessairement plus générales que les graphes de connaissances en ce sens qu'il n'y a pas de besoin implicite ni explicite pour un schéma ou une ontologie. Par exemple, les embarcations de graphe peuvent être utilisées sur le réseau définies par échange de messages entre les nœuds d'Internet, puis utilisés dans des algorithmes d'apprentissage de la machine pour prédire les nœuds roux. En revanche, pour le graphe Wikidata, les graphes de connaissances dans les entreprises et dans la représentation de la production des algorithmes d'apprentissage de la machine, un schéma ou une ontologie peut jouer un rôle central.

Nous concluons en observant que la surtension récente d'intérêt pour les graphes de connaissances est principalement tirée par les exigences d'analyse de plusieurs applications professionnelles convaincantes. Les graphes de connaissances dans ces

applications peuvent certainement bénéficier des travaux classiques sur les techniques de conception de la représentation supérieure et, en fait, nous envisageons que les deux finissent par la convergence.

Comment créer un graphe de connaissances?

1. Introduction

Il est possible de commencer avec un graphe de connaissances sans conception initiale de son schéma et peupler à la fois son schéma et ses instances pendant le processus de création. Au degré de conception initiale d'un graphe de connaissances est pratique, elle peut améliorer considérablement son utilité. Un tel conception consiste à faire un choix approprié des nœuds, des étiquettes de nœud, des propriétés des nœuds, des relations et des propriétés de relation.

L'entrée de la population de graphe de connaissances peut provoquer une ou plusieurs sources consistant en des données structurées, des données semi- structurées, du texte gratuit ou des images, ou une création directe de l'entrée humaine. Lorsque nous travaillons avec des sources de données structurées et semi- structurées, nous devons effectuer une tâche de mappage de schéma (c.-à-d. Relier le schéma dans la source d'entrée avec le schéma du graphe de connaissances) et l'enregistrement de la tâche de liaison (c.-à-d. Relier de nouvelles instances avec le pré -exister des instances dans le graphe de connaissances). Ces mêmes tâches sont également confrontées lors de l'intégration des données avec la seule différence que les données intégrées sont exprimées dans un modèle de données graphe. Lorsque nous travaillons avec les sources non structurées, nous devons résoudre les problèmes d'extraction d'informations de l'extraction de l'entité et de l'extraction de la relation.

Le choix des méthodes utilisées dans la population de graphe de connaissances dépend de l'échelle du problème et de la précision souhaitée. Si un graphe de connaissances doit être utilisé sur l'échelle Web pour obtenir une récupération d'informations, la précision n'a pas besoin d'être parfaite, et il est infaisable d'utiliser la vérification humaine pour chaque triplement du graphe. Si un graphe de connaissances doit être utilisé dans une entreprise où la précision doit être presque parfaite, la vérification humaine est essentielle même si elle est effectuée juste avant que les informations soient utilisées. Comme la précision est toujours souhaitée, quelle que soit l'entreprise ou les paramètres de www, d'assurer la rentabilité et l'évolutivité, il existe un emphases sur le crowdsourcing et d'autres méthodes à faible coût d'obtention d'une contribution humaine.

Dans ce chapitre, nous nous concentrerons sur la conception de schéma de graphe

de connaissances. Dans les deux chapitres suivants, nous discuterons des problèmes qui se posent dans la population d'un graphe de connaissances à partir de données structurées, c'est-à-dire les problèmes de mappage de liaisons et de schémas de schéma, ainsi que des problèmes qui se posent tout en remplissant du texte, c'est-à- dire l'extraction de l'entité et l'extraction de relation.

2. Conception de graphe de connaissances

Les modèles de données de propriété et RDF ont à la fois un ensemble de problèmes de conception dont certains sont communs à travers les deux, tandis que d'autres sont uniques. Par exemple, les deux modèles doivent utiliser une réification pour des situations qui ne peuvent pas être directement modélisées à l'aide de Thecles. Un modèle RDF doit adopter un système d'iris qui n'est pas nécessaire pour les graphes de propriété. Dans un modèle de graphe de la propriété, nous devons décider si une valeur doit être représentée en tant que propriété ou en tant que nœud, tandis que cette distinction n'est pas cessée dans un modèle RDF. Dans cette section, nous présenterons une vue d'ensemble de ces problèmes de conception confrontés à chacun de ces deux modèles.

2.1 Conception d'un graphe RDF

Les lignes directrices de création de graphe de connaissances pour les données RDF sur le www sont appelées principes de données liées telles que décrites ci- dessous.

1. Utilisez Uris comme noms pour les choses.
2. Utilisez HTTP URI afin que les gens puissent rechercher ces noms.
3. Lorsque quelqu'un recherche une URI, fournissez des informations utiles, en utilisant les normes (RDF, SPARQL).
4. Inclure des liens vers d'autres URI afin qu'ils puissentdécouvrir plus de choses.

Nous considérerons plus en détail chacune de ces directives.

2.1.1 Utilisez URI comme noms pour les choses

Pour publier un graphe de connaissances sur le www, nous devons d'abord identifier les éléments d'intérêt de notre domaine. Ce sont les choses dont les propriétés et les relations que nous voulons décrire dans le graphe. Dans la terminologie web, tous les éléments d'intérêt sont appelés ressources. Les ressources sont de deux types: ressources d'information et ressources sans information. Toutes les ressources que nous trouvons sur le www traditionnel, telles que des documents, des images et

d'autres fichiers multimédia, sont des ressources d'information. Mais beaucoup de choses que nous voulons dans notre graphe de connaissances ne sont pas: des personnes, des produits physiques, des lieux, des protéines, des concepts scientifiques, etc. En règle générale, tous les "objets du monde réel" qui existent en dehors du www sont non- ressources d'information.

Les éditeurs de graphes de connaissances doivent construire les URI à partager de manière à être simples, stables et gérables. Les URI mnémoniques courtes ne se casseront pas aussi facilement lorsqu'ils sont envoyés dans des courriels et sont généralement plus faciles à retenir. Une fois que nous avons configuré une URI pour identifier une certaine ressource, il devrait rester ainsi le plus longtemps possible. Pour assurer la persistance à long terme, il est préférable de garder des morceaux et des morceaux spécifiques à la mise en œuvre tels que ".php" et ".asp" hors des URIS. Enfin, les URI devraient être définies de manière à pouvoir être entièrement gérée par l'éditeur.

2.1.2 Utilisez HTTP URI afin que les gens puissent rechercher ces noms

Nous identifions les ressources en utilisant des identificateurs de ressources uniformes (URI). Nous nous limitons à utiliser uniquement HTTP URIS et à éviter d'autres schémas d'URI tels que des noms de ressources uniformes (URN) et des identifiants d'objet numériques (DOI).

Le processus de recherche de noms est appelé la déséroférience de l'URI. Lorsque nous avons une désarférence une URI pour un objet d'information, nous nous attendons à obtenir la représentation de son état actuel (par exemple, un document texte, une image, une vidéo, etc.), mais lorsque nous avons une désespectation une ressource sans information, nous pouvons obtenir Sa description dans RDF exprimée dans une notation XML.

2.1.3 Lorsque quelqu'un recherche une URI, fournissez des informations utiles à l'aide de RDF et de SPARQL

Lorsque quelqu'un recherche une URI, le fournisseur devrait renvoyer un graphe de connaissances dans le RDF. Les données doivent réutiliser des vocabulaires normalisés pour nommer l'iris utilisé pour décrire les données RDF. Plusieurs vocabulaires utiles sont disponibles pour décrire des catalogues de données, des

organisations et des données multidimensionnelles, telles que des statistiques sur le Web. Un effort open source appelé schema.org publie la communauté créée des vocabulaires open source pour une utilisation ouverte sur le Web. Nous considérons quelques exemples de telles vocabulaires.

Les données RDF suivantes décrivent un extrait de la structure organisationnelle du bureau du Cabinet britannique.

@prefixuk_cabinet:
<http://reference.data.gov.uk/id/department/> uk_cabinet:co rdf:type
org:Organization

uk_cabinet:co skos:prefLabel "Cabinet Office"

uk_cabinet:co org:hasUnit uk_cabinet:cabinet-office-communications
uk_cabinet:cabinet-office-communications rdf:type org:OrganizationUnit

uk_cabinet:cabinet-office-communications skos:prefLabel "Cabinet Office
Communications"

uk_cabinet:cabinet-office-communications org:hasPost uk_cabinet:post_246

uk_cabinet:post_246 skos:prefLabel "Deputy Director, Deputy Prime
Minister's Spokesperson"

Dans les données ci-dessus, le premier triple utilise la classe Org: organisation de l'ORGANISATION AOUTOLOGY. Le deuxième triple utilise la relation Skos: préflabel tiré de l'ontologie Skos. Skos signifie un système d'organisation de connaissances simple et fournit quelques relations couramment utiles telles que SKOS: préflabel pour décrire des données. Dans ce cas, Skos: Preflabel nous permet simplement d'associer une étiquette de texte avec UK_CABINET: CO. Le troisième triple utilise la relation Org: Hisunit de l'ontologie de l'organisation pour décrire une unité au sein du bureau du Cabinet britannique. Les deux triples suivants font des affirmations supplémentaires sur cette unité. Le sixième triple utilise l'org: le hasopp de la relation décrit une position avec dans un département et les deux derniers triples donnent des informations supplémentaires sur cette position.

Il n'est peut-être pas toujours possible de trouver des vocabulaires préexistants pouvant être utilisés pour créer un ensemble de données RDF. Si la création d'un nouveau vocabulaire devient nécessaire, il faut s'assurer qu'il est documenté, auto- décrivant, a une stratégie de version de version, est définie dans plusieurs langues et est publiée par une source de confiance afin que les URI utilisées y persistent

pendant une longue période. de temps. Nous disons qu'un vocabulaire est auto- décrivant si chaque propriété ou chaque terme a une étiquette, une définition et un commentaire défini.

2.1.4 Inclure des liens vers d'autres URI afin qu'ils puissent découvrir plus de choses

Bien que la publication des données utilisant RDF, il faut fournir des liens vers d'autres objets afin que son utilité augmente. Il peut y avoir trois types de liens: liens relationnels, liens d'identité et liens de vocabulaire. Nous examinerons un exemple de chacun de ces types de liens.

Relation Links Pointez sur des choses associées dans d'autres sources de données telles que d'autres personnes, des lieux ou des gènes. Par exemple, les liens de relation permettent aux personnes de pointer vers des informations de base sur l'endroit où ils vivent ou des données bibliographiques sur les publications dont ils ont écrit. Dans le triple ci-dessous, nous montrons un lien dans lequel une personne d'un ensemble de données est affirmée à être basée à proximité d'un emplacement géographique spécifié à l'aide d'une URI dans un autre ensemble de données.

@prefix big: <http://biglynx.co.uk/people/> @prefix dbpedia: <http://dbpedia.org/resource/> big:dave-smith foaf:based_near dbpedia:Birmingham

Liens d'identité Point dans les alias Uri utilisés par d'autres sources de données pour identifier le même objet réel ou abstrait. Les liens d'identité permettent aux clients de récupérer d'autres descriptions sur une entité et de servir une fonction sociale importante car elles permettent d'exprimer différentes vues du monde sur le www des données. Il s'agit d'une pratique standard d'utiliser le type de lien http://www.w3.org/2002/07

/ hibou # Samas pour affirmer que deux alias URI se réfèrent à la même ressource. Par exemple, si Dave Smith

maintiendrait également une page d'accueil de données privée à part les données que Big Lynx publie sur lui,

Il pourrait ajouter un lien http://www.w3.org/2002/07/owl#sameas sur sa page d'accueil de données privée, indiquant que l'URI utilisée pour lui faire référence dans ce document et l'URI utilisé par Big Lynx se référer à la même entité du monde réel. Un triple capturant ces informations est indiquée ci-dessous.

@prefix ds: <http://www.dave-smith.eg.uk> @prefix owl: <http://www.w3.org/2002/07/owl> @prefix big: <http://biglynx.co.uk/people/> ds:me owl:sameAs big:dave- smith

Les liens de vocabulaire pointent des données aux définitions des termes de vocabulaire utilisés pour représenter les données, ainsi que de ces définitions aux définitions de termes connexes dans d'autres vocabulaires. Les liens de vocabulaire rendent des données auto-descriptives et permettent aux applications de données liées pour comprendre et intégrer des données sur les vocabulaires. Dans le lien de vocabulaire indiqué ci-dessous, la catégorie PetiteRediumEnterprise définie par Biglynx est définie comme une sous-classe de la société de classe dans l'ontologie DBPEDIA. En faisant un tel lien, il est possible de récupérer diverses affirmations sur la société de classe à partir de DBPEDIA et de les utiliser avec la classe petiteMediumEnterprise.

@prefix dbpedia: <http://dbpedia.org/ontology/> big:sme#SmallMediumEnterprise rdfs:subClassOf dbpedia:Company

2.2 Conception d'un graphe de propriété

La conception d'un graphe de propriété implique de choisir des nœuds, des étiquettes de nœuds, des propriétés du nœud, des bords et des propriétés de bord. Les questions de base de la conception sont de déterminer de manière à modéliser une information comme une propriété, une étiquette ou un objet distinct; quand introduire des propriétés de relation; et comment gérer des relations d'arité plus élevées. Nous illustrerons le processus de fabrication de ces choix en utilisant des exemples.

2.2.1 Choisir des nœuds, des étiquettes et des propriétés

Dans un modèle de graphe de la propriété, les nœuds représentent généralement des entités dans le domaine. Si nous voulions représenter des informations sur les personnes, nous allons créer un nœud de chaque personne (par exemple, John) et associer la personne étiquette avec ce nœud.

Il existe plusieurs considérations dans la fabrication d'autres choix d'étiquettes de nœuds, de propriétés du nœud et de bords. Ces considérations comprennent: la naturalité des étiquettes, que les étiquettes puissent changer sur une période de temps, la performance de la requête d'exécution et la cardinalité des valeurs.

Pour illustrer le choix de modéliser une information comme une étiquette, une propriété ou un objet distinct, envisagez la tâche de représenter le sexe d'une

personne. Nous avons trois moyens potentiels de capturer ces informations: nous pouvons créer: mâle et: femme comme étiquettes et les associez avec les nœuds de la personne; (2) Nous pouvons créer une propriété appelée "genre" et l'associer avec les nœuds de la personne et lui permettre d'avoir les valeurs "mâle" et "femme"; (3) Nous pouvons créer un objet de genre, l'associer à une personne utilisant une relation HAS_GENDER et lui donner une propriété appelée "nom" qui peut prendre "homme" et "femme" comme des valeurs.

Les étiquettes dans un modèle de graphe de propriété sont utilisées pour collecter des nœuds dans des ensembles. Tous les nœuds étiquetés avec la même étiquette appartiennent au même ensemble. Les requêtes peuvent fonctionner avec ces ensembles au lieu de tout le graphe, ce qui facilite l'écriture des requêtes et plus efficace. Un nœud peut être étiqueté avec un nombre quelconque d'étiquettes, y compris aucun, rendant les étiquettes une addition facultative au graphe. En tant que des groupes d'étiquettes, des nœuds dans un ensemble, il peut être considéré comme une classe. La question de savoir si une nouvelle étiquette peut être retraitée si vous souhaitez introduire une nouvelle classe?

Création de nouvelles classes masculines et femelles vs introduisant une propriété de nœud "genre" qui peut prendre deux valeurs de "mâle" et "féminin" capture les mêmes informations. En général, chaque fois qu'une phrase naturelle dans la langue est fréquemment utilisée dans un domaine, il s'agit d'un candidat à introduire comme une classe tant que l'adhésion de la classe ne change pas avec le temps. Comme certaines implémentations optimisent la récupération basée sur l'utilisation d'étiquettes, l'utilisation d'étiquettes peut entraîner des performances rapides sur les requêtes qui doivent filtrer les résultats en fonction de l'adhésion de la classe. Si l'adhésion de la classe change avec le temps, ni une étiquette ni une valeur de propriété nœud n'est un choix approprié, et nous devons utiliser une relation. Nous considérerons cela dans la section suivante.

2.2.2 Quand introduire des relations entre les objets

Pour des situations pouvant être modélisées soit à l'aide d'une propriété de nœud, soit en introduisant un objet et une relation distincts, il y a au moins deux considérations différentes. La première de ces considérations a été introduite dans la section précédente: l'adhésion à la classe change avec le temps. La deuxième considération survient lorsque nous souhaitons réaliser une meilleure performance de la requête. Nous considérerons ces situations plus en détail ensuite.

Continuant l'exemple de la section précédente, lorsque le sexe d'une personne pourrait changer sur une période de temps, notre seul choix est de saisir les

informations comme un objet sexiste distinct qui est lié à la personne utilisant la relation HAS_GENDER. Nous pouvons ensuite associer une propriété relationnelle avec la relation HAS_GENDER qui indique la durée de la durée pour laquelle une valeur particulière du genre tient. La création d'un nœud de genre distinct conduirait toutefois à un grand nombre de bords qui gaspillent que pour la plupart des personnes que le genre ne change pas. Dans une telle situation, une combinaison des deux solutions pourrait être souhaitée où pour la plupart des personnes, le genre est représenté comme une valeur de la propriété du nœud, mais pour une petite fraction de personnes, elle est représentée comme une valeur de propriété relative à une relation à une relation distincte Noeud de genre.

Considérons une situation où une meilleure performance de la requête est une considération clé. Supposons que nous souhaitiions modéliser des films et leurs genres. Dans une conception, pour un noeud de type film, nous pouvons introduire un
«genre» de propriété qui peut prendre des valeurs telles que «Action», «SCIFCI», etc. dans une autre conception, nous pouvons introduire un nouveau genre de type nœud qui a une Nœud A Propriété "Nom" qui peut prendre des valeurs telles que S "Action", "SCIFCI". Nous allons ensuite relier un noeud de type film avec un nœud de genre de type à l'aide de la relation HAS_GENRE. En général, nous pouvons associer plus d'un genre avec un film. Supposons que nous souhaitiions à interroger ces films qui ont au moins un genre commun. Dans la première solution dans laquelle nous utilisons la propriété du nœud "Genre", cette requête serait indiquée dans Cypher comme suit:

```
MATCH (m1:Movie), (m2:Movie)
WHERE any(x IN m1.genre WHERE x IN m2.genre)
AND m1 <> m2
RETURN m1, m2
```

Lorsque nous modélisons le genre comme objet séparé, la même requête peut être indiquée comme suit:

```
MATCH (m1:Movie)-[:has_genre]->(g:Genre), (m2:Movie)-[:has_genre]->(g)
WHERE m1 <> m2 RETURN m1, m2
```

Dans la deuxième requête ci-dessus, nous pouvons utiliser plus directement les motifs de graphe et, dans certains moteurs graphe, cette requête a une performance d'exécution plus rapide en raison de l'indexation des relations.

Par conséquent, dans ce cas, il faut choisir entre les deux designs en fonction du type de requêtes cela sera attendu.

2.2.3 Quand introduire des propriétés de la relation

Nous avons déjà vu un exemple d'une propriété associée à une relation pour faire face aux situations lorsque la relation change avec le temps. Parmi les autres situations dans lesquelles il est logique d'introduire des propriétés avec une relation incluent l'association de poids ou de la confiance en une relation ou d'associer la provenance ou d'autres métadonnées avec une relation.

Certains moteurs graphes n'exposent pas en fonction des propriétés de la relation. Si l'affaire d'utilisation est telle que la majeure partie de l'évaluation de la requête puisse être effectuée sans utiliser les propriétés de la relation, elle n'est requise que pour le filtrage final des résultats, on peut ne pas payer un panneau de performance significative en raison du manque d'indexation. Si l'accès aux propriétés de la relation est au cœur de la performance de la requête, il est préférable de réifier la relation que nous discuterons dans la section suivante.

2.2.4 Manipulation des relations non binaires

Nous avons souvent besoin de modéliser des relations qui ne sont pas binaires. Un exemple commun d'une telle relation est la relation entre les objets A, B et C captures que c est comprise entre A et B. Une approche standard de capture de telles relations d'arité plus élevées dans un graphe est la réification.

Nous avons déjà discuté de la rétification dans le contexte du RDF, mais cette technique est également utile et souhaitable pour les graphes de propriété. Pour capturer la relation entre la relation, nous introduisons un nouveau type de nœud, entre_relationhip qui a deux propriétés: has_Object (avec des valeurs A et B) et HAS_BETWEEN_OBJECT (avec valeur C). Nous pouvons utiliser la réification pour des relations avec n'importe quelle arité en créant un nouveau type de nœud pour la relation et en introduisant des propriétés du nœud pour les différents arguments de cette relation.

2. Résumé

Dans ce chapitre, nous avons considéré la conception du modèle de données graphe pour les graphes RDF et Property. La conception des modèles de données concerne de manière à savoir s'il s'agit de réinfinir une relation, de manipuler des relations non

binaires, etc., sont courantes dans l'ensemble du RDF et des modèles de données de graphe de propriété. Le choix de l'utilisation d'une propriété vs une relation est unique au modèle de données de graphe de la propriété. Le modèle RDF fournit des directives explicites sur l'utilisation de l'IRIS, la réutilisation des vocabulaires existantes et des liens entre les vocabulaires. Même si les considérations de liaison de données ne font pas partie intégrante du modèle de graphe de la propriété, mais leur utilisation peut rendre un système de graphe de propriété plus utile dans l'intégration des données.

Comment créer un graphe de connaissances à partir de données?

1. Introduction

Les grandes organisations génèrent de nombreuses données internes et consomment également des données produites par des fournisseurs tiers. De nombreux fournisseurs de données obtiennent leurs données en traçant des sources non structurées et investissent des efforts importants dans la fourniture d'une forme structurée à utiliser par d'autres. Pour utiliser efficacement ces données externes, il doit être lié aux données internes de la société. Cette intégration de données permet de nombreux cas d'utilisation populaire tels que 360 Vue d'un client, d'une détection de fraude, d'une évaluation des risques, d'une approbation de prêt, etc. Pour ce chapitre, nous aborderons le problème de la création d'un graphe de connaissances en intégrant les données disponibles à partir de sources structurées. Nous examinerons le problème de l'extraction des données provenant de sources de données non structurées dans le chapitre suivant.

Lors de la combinaison de données de multiples sources dans un graphe de connaissances, nous pouvons entreprendre une conception initiale de schéma comme indiqué dans le chapitre précédent. Nous pouvons également commencer par aucun schéma car il est simple de charger des données externes comme triple dans un graphe de connaissances. La conception du schéma initial est généralement entraînée

par le cas d'utilisation d'entreprise spécifique que l'on souhaite aborder. Dans la mesure où un tel schéma initial existe, nous devons déterminer comment les éléments de données d'une nouvelle source de données doivent être ajoutés au graphe de connaissances. Ceci est généralement appelé problème de cartographie de schéma. En plus de relier les schémas des deux sources, nous devons également faire face à la possibilité qu'une entité dans les données entrantes (par exemple, une société) peut déjà exister dans notre graphe de connaissances. Le problème de déduire si les deux entités des données peuvent être la même entité mondiale réelle est connue sous le nom de problème de liaison des enregistrements. Le problème de liaison des enregistrements se pose également lorsque des fournisseurs de données tiers envoient de nouveaux flux de données et que notre graphe de connaissances doit être tenu à jour avec le nouveau flux de données.

Dans ce chapitre, nous discuterons des approches actuelles pour résoudre le mappage de schéma et les problèmes de liaison des enregistrements. Nous allons décrire les algorithmes de pointe et discuterons de leur efficacité sur les problèmes actuels de l'industrie.

2. Cartographie de schéma

Le mappage de schéma suppose l'existence d'un schéma qui sera utilisé pour stocker de nouvelles données provenant d'une autre source. Le mappage de schéma définit ensuite les relations et les attributs de la base de données d'entrée correspond aux propriétés et relations dans le graphe de connaissances. Il existe des techniques pour les mappages de schéma de bootstring. Les mappages de schéma boots au niveau peuvent être corrigés par une intervention humaine.

Nous allons commencer notre discussion sur la cartographie du schéma en décrivant certains des défis et faire valoir que l'on devrait être préparé à l'éventualité que ce processus sera largement manuel et plus rentable. Nous décrivons ensuite une approche pour spécifier des mappages entre le schéma de la source d'entrée et le schéma du graphe de connaissances. Nous conclurons la section en examinant certaines des techniques pouvant être utilisées pour la cartographie de schéma de bootstrap.

2.1 Défis dans la cartographie du schéma

Les principaux défis de la cartographie automatisation des schémas sont les suivants: (1) difficile à comprendre le schéma (2) la complexité des mappages et (3) l'absence de données de formation disponibles. Nous discutons ensuite de ces défis plus en détail.

Les schémas de base de données relationnels commerciaux peuvent être énormes composés de milliers de relations et de dizaines de milliers d'attributs. Parfois, les

noms de relation et d'attributs n'ont pas de sémantique (par exemple, segment1, segment2) qui ne prête pas à traiter la prédiction automatisée réaliste des mappages.

Les mappages entre le schéma d'entrée et le schéma dans le graphe de connaissances ne sont pas toujours simples de mappage unique. Les mappages peuvent impliquer des calculs, appliquer une logique commerciale et prendre en compte des règles spéciales pour la manipulation des situations telles que les valeurs manquantes. Il devient un bon ordre de s'attendre à ce que tout processus automatique indique de tels mappages complexes.

Enfin, de nombreuses solutions de cartographie automatisées s'appuient sur des techniques d'apprentissage de la machine nécessitant une grande quantité de données de formation pour fonctionner efficacement. À mesure que les informations de schéma, par définition, sont beaucoup plus petites que les données elles-mêmes, il est irréaliste de s'attendre à ce que nous aurons déjà un grand nombre de mappages de schéma disponibles contre lesquels un algorithme de cartographie pouvait être formé.

2.2 Spécification du mappage de schéma

Dans cette section, nous examinerons une approche possible de spécifier des mappages entre les sources de données d'entrée et un schéma graphe de la connaissance cible. Nous prendrons un exemple dans le domaine de la batterie de cuisine. Nous pouvons imaginer différents fournisseurs fournissant des produits qu'un site de commerce électronique voudra peut-être regrouper et faire de la publicité à ses clients. Nous considérerons deux exemples de sources, puis introduisons le schéma du graphe de connaissances auxquels nous définirons des mappages.

We show below some sample data from the first data source in a relational table called cookware. It has four attributes: name, type, material, and price.

Cookware			
Name	type	material	Price
c01	skillet	cast iron	50
c 02	saucepan	st eel	40
c 03	skillet	st eel	30
c 04	saucepan	aluminium	20

Table 2.2. répertorie les produits d'un fabricant.

La deuxième base de données indiquée ci-dessous répertorie les produits d'un fabricant. Dans ce cas, il y a plusieurs tables, une pour chaque attribut de produit. La table en nature spécifie le type de chaque produit. La table de base spécifie si chaque produit est fabriqué à partir d'un métal corrostible (aluminium ou inoxydable), un métal non corrosible (fer ou acier) ou autre chose que métal (verre ou céramique). La table revêtue spécifie les produits qui ont des revêtements antiadhésives. Le tableau des prix donne le prix de vente. Il n'y a pas d'informations matérielles. La société a choisi de ne pas fournir d'informations sur le métal utilisé dans chaque produit. Notez que la table revêtue n'a que des valeurs positives; Les produits sans revêtements antiadhésives ne sont laissés laissés inhabitués.

id	Value
m01	Skillet
m02	Skillet
m03	Saucepan
m04	Saucepan

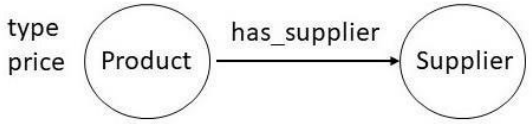
id	value
m01	corrosible
m02	noncorrosible
m03	noncorrosible
m04	nonmetallic

id	value
m01	60
m02	50
m03	40
m04	20

id	value
m01	yes
m02	yes

Table 2.3. schéma souhaité pour le graphe de connaissances

Supposons que le schéma souhaité pour le graphe de connaissances exprimé en tant que graphe de propriété est comme indiqué ci-dessous. Nous avons deux types de noeuds différents: un pour les produits et l'autre pour les fournisseurs. Ces deux nœuds sont connectés par une relation appelée has supplier. Chaque nœud de produit a des propriétés "Type" et "Prix".



Pour spécifier les mappages et pour illustrer le processus, nous utiliserons une triple notation de sorte qu'un processus similaire soit applicable, que nous utilisions un modèle de données RDF ou de graphe de propriété pour le graphe de connaissances. Pour un graphe de connaissances RDF, nous devrons créer un IRIS qui est un processus orthogonal pour relier les deux schémas, et donc omis d'ici. Les triples souhaités dans le graphe de connaissances cible sont énumérés ci-dessous.

knowledge graph		
subject	predicate	Object
c01	type	Skillet
c01	price	50
c01	has_supplier	vendor_1
c02	type	Saucepan
c02	price	40
c02	has_supplier	vendor_1
c03	type	Skillet
c03	price	30
c03	has_supplier	vendor_1
c04	type	Saucepan
c04	price	20
c04	has_supplier	vendor_1
m01	type	Skillet
m01	price	60
m01	has_supplier	vendor_2
m02	type	Skillet
m02	price	50
m02	has_supplier	vendor_2
m03	type	Saucepan
m03	price	40
m03	has_supplier	vendor_2
m04	type	Saucepan
m04	price	20
m04	has_supplier	vendor_2

Table 2.4. knowledge graph

Tout langage de programmation de choix pourrait être utilisé pour exprimer les mappages. Ici, nous avons choisi d'utiliser Datalog pour exprimer les mappages. Les règles ci-dessous sont simples. Les variables sont indiquées en utilisant des lettres majuscules. La troisième règle introduit le Vendor_1 constant pour indiquer la source des données.

Connaissation_graphe (ID, Type, Type): - Cookware (ID, Type, Matériel, Prix) Knowleward_GRraph (id, prix, prix): - Cookware (ID, type, matériel, prix) Knowleward_graph (ID, HAS_SUPPLIER, Vendor_1): - batterie de cuisine (ID, type, matériel, prix)

Ensuite, nous considérons les règles de mappage de la deuxième source. Ces règles sont très similaires aux règles de mappage de la première source, sauf que, maintenant, les informations proviennent de deux tables différentes dans les

données source.

Connaissation_graphe (ID, Type, Type): - type (ID, type) Knowleown_graph (ID, prix, prix): - Prix (ID, prix) Knowleowne_graph (ID, HAS_SUPPLIER, VENDOR_2): - type (ID, type)

En général, il n'a peut-être pas de sens de réutiliser les identifiants des bases de données source, et on peut souhaiter créer de nouveaux identifiants à utiliser dans le graphe des connaissances. Dans certains cas, le graphe de connaissances peut déjà contenir des objets équivalents à ceux des données importées. Nous examinerons cette question dans la section sur le lien record.

2.3 Cartographie de schéma de bootstrapping

Comme indiqué à la section 2.1, une approche entièrement automatisée de la cartographie de schéma est confrontée à de nombreuses difficultés pratiques. Il existe un travail considérable sur le bottage des mappages de schéma basés sur une variété de techniques et les validant à l'aide de l'entrée humaine. Les techniques de bootstress pour le mappage de schéma peuvent être classées dans les catégories suivantes: correspondance linguistique, correspondance basée sur des instances et correspondance en fonction des contraintes. Nous examinerons des exemples de ces techniques suivant.

Les techniques linguistiques peuvent être utilisées sur le nom d'un attribut ou sur la description du texte d'un attribut. La première et la plus évidente approche consiste à vérifier si les noms des deux attributs sont égaux. On peut avoir une plus grande confiance en une telle égalité si les noms étaient des iris. Deuxièmement, on peut canonaliser les noms en les traitant à travers des techniques telles que la stemming, puis la vérification de l'égalité. Par exemple, grâce à ce traitement, nous pourrons peut-être conclure le mappage de CNAME au nom du client. Troisièmement,

on pourrait rechercher le mappage basé sur des synonymes (par exemple, voiture et automobile) ou hypernymatiques (par exemple, livre et publication). Quatrièmement, nous pouvons vérifier la cartographie basée sur des substrings communs, la prononciation et la façon dont les mots sonnent. Enfin, nous pouvons faire correspondre les descriptions des attributs via des techniques de similitude sémantique. Par exemple, une approche consiste à extraire des mots-clés de la description, puis à vérifier la similitude entre eux à l'aide des techniques que nous avons déjà répertoriées.

En appariement basé sur les instances, on examine le type de données existant. Par exemple, si une valeur d'attribut particulière contient des dates, elle ne peut correspondre qu'à ces attributs contenant des valeurs de date. De nombreux types de données standard peuvent être déduits en examinant les données.

Dans certains cas, le schéma peut fournir des informations sur les contraintes. Par exemple, si le schéma spécifie qu'un attribut particulier doit être unique pour une personne et doit être un nombre, il s'agit d'une correspondance potentielle pour les attributs d'identification tels qu'un numéro d'employé ou un numéro de sécurité sociale.

Les techniques considérées ici sont inexactes et peuvent donc être utilisées uniquement pour bootstrap le processus de cartographie de schéma. Tous les mappages doivent être vérifiés et validés par des experts humains.

Comment se rapportent des graphes de la connaissance à AI?

1. Introduction

Dans ce chapitre de conclusion, nous discuterons de différentes manières dans lesquelles les graphes de connaissances se croisent avec l'intelligence artificielle (AI). Comme nous l'avons noté dans le chapitre d'ouverture, l'utilisation de graphes dirigés étiquetés pour la représentation des connaissances existe depuis les premiers jours d'AI. Notre objectif de discussion dans ce chapitre concerne l'utilisation de graphes de connaissances dans les développements récents. Par conséquent, nous avons choisi trois thèmes pour une élaboration plus poussée: des graphes de

connaissances en tant que lit d'essai pour les algorithmes d'AI, une nouvelle zone spécialisée de la science des données graphes et des graphes de connaissances dans le contexte plus large de la vision ultime de l'AI.

2. Graphes de connaissances en tant que lit de test pour la génération de courant AI algorithmes

Les graphes de connaissances ont une relation à deux voies avec AI algorithmes. D'une part, les graphes de connaissances permettent de nombreuses applications AI actuelles et, de l'autre, de nombreux algorithmes AI actuels sont utilisés pour créer les graphes de connaissances. Nous examinerons cette synergie symbiotique dans les deux sens.

Les assistants personnels, les systèmes recommandés et les moteurs de recherche sont des applications qui présentent un comportement intelligent et ont des milliards d'utilisateurs. Il est maintenant largement accepté que ces applications se comportent mieux s'ils puissent tirer parti des graphes de KnowEldge. Un assistant personnel utilisant un graphe de connaissances peut faire plus de choses. Un système de recommandation avec un graphe de connaissances peut faire de meilleures recommandations. De même, un moteur de recherche peut retourner de meilleurs résultats lorsqu'il a accès à un graphe de connaissances. Ainsi, ces applications fournissent un contexte convaincant et un ensemble d'exigences pour les graphes de connaissances aient un impact sur les offres de produits immédiats.

Pour créer un graphe de connaissances, nous devons absorber les connaissances de plusieurs sources d'information, aligner les informations, distiller des pièces de connaissances clés de la mer d'information et la mine de la connaissance de la sagesse qui influencerait le comportement intelligent. Les techniques d'IA jouent un rôle important à chaque étape de la création et de l'exploitation des graphes de connaissances. Pour extraire des informations provenant de sources, nous avons considéré des techniques d'extraction d'entité et de relation. Pour aligner des informations sur plusieurs sources, nous avons utilisé des techniques telles que la cartographie de schéma et la liaison avec l'entité. Pour distiller les informations extraites, nous pouvons utiliser des techniques telles que le nettoyage des données et la détection annamitique. Enfin, pour extraire la sagesse du graphe, nous avons utilisé des algorithmes d'inférence, une réponse de la question de la langue naturelle, etc.

Par conséquent, les graphes de connaissances permettent aux systèmes IA, qui fournissent une motivation et un ensemble de exigences pour eux. Les techniques d'AI alimennent également notre capacité à créer le graphe de connaissances économiquement et à l'échelle.

3. Graphes de connaissances et science des données graphes

Graphe Data Science est une discipline émergente qui vise à tirer des connaissances en exploitant la structure dans les données. Les organisations ont généralement accès à une énorme quantité de données, mais leur capacité à exploiter ces données a été limitée par une collection de rapports prédéfinis générés à l'aide de ces données. La discipline de la science des données graphes transforme cette expérience en combinant des algorithmes de graphes, des requêtes de graphes et des visualisations dans des produits qui accélèrent considérablement le processus de prise de vue.

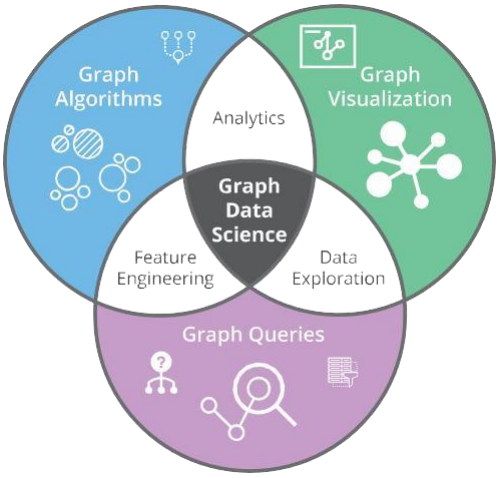


Figure 2.1. Graphe Data Science

Comme nous l'avons vu dans les cas d'utilisation axée sur l'analyse pour l'industrie financière, les entreprises souhaitent exploiter la structure relationnelle de leurs données pour faire des prédictions sur le risque, les nouvelles opportunités de marché, etc. pour faire des prédictions, il est courant d'utiliser des algorithmes d'apprentissage de la machine qui dépendez de l'ingénierie prudente. Alors que les algorithmes d'apprentissage de la machine deviennent en train de devenir une marchandise et peuvent être utilisés comme produits hors tension, il existe une compétence distincte d'ingénierie de fonctionnalités. L'ingénierie de fonctionnalités nécessite une compréhension profonde du domaine ainsi que la compréhension du fonctionnement des algorithmes d'apprentissage de la machine.

C'est cette synergie parmi le système traditionnel graphe et la disponibilité de la machine apprentissage à identifier et à prédire les propriétés relationnelles dans des données, qui a catalysé la création de la sous-discipline de la science des données graphes. En raison des cas d'utilisation élevée d'impact possibles grâce à la science

des données graphes, elle devient une compétence logicielle de plus en plus recherchée dans l'industrie aujourd'hui.

4. Graphes de connaissances et objectifs à long terme de l'AI

Les premiers travaux de l'AI ont porté sur une représentation explicite des connaissances et ont lancé le domaine des graphes de connaissances par le biais de représentations telles que des réseaux sémantiques. Au fur et à mesure que le champ évolué, des réseaux sémantiques ont été formalisés et ont conduit à plusieurs générations de langues de représentation telles que la description logique, programmes logiques et modèles graphes. Parallèlement au développement de ces langues, un défi tout aussi important de la création des connaissances dans le formalisme choisi a été abordé. Les techniques de création de connaissances ont varié de l'ingénierie de la connaissance, de l'apprentissage inductif et des méthodes d'apprentissage plus récentes.

Pour réaliser la vision de l'AI, une représentation explicite des connaissances d'un domaine correspondant à la compréhension humaine et permet de raisonner avec elle est essentielle. Bien que dans certaines tâches de performance telles que la recherche, la recommandation, la traduction, etc., la compréhension humaine et la précision ne sont pas des exigences difficiles, mais il existe de nombreux domaines dans lesquels ces exigences sont essentielles. Des exemples de tels domaines comprennent la connaissance de la loi sur les calculs d'impôt sur le revenu, la connaissance d'un domaine pour l'enseignement à un étudiant, une connaissance d'un contrat de sorte qu'un ordinateur puisse l'exécuter automatiquement, etc. En outre, il est de plus en plus reconnu que pour de nombreux situations où nous

Peut atteindre un comportement intelligent sans représentation explicite des connaissances, le comportement doit encore être explicable afin que les humains puissent comprendre la justification de celui-ci. Pour cette raison, nous croyons qu'une représentation explicite est essentielle.

Il y a un récit dans la communauté de la recherche que l'ingénierie du savoir n'a pas échoue et que le traitement des langues naturelles et l'échelle des méthodes d'apprentissage de la machine. Ces revendications sont basées sur une caractérisation incorrecte des tâches traitées par le traitement des langues naturelles. Par exemple, en utilisant des modèles de langue, on peut être capable de calculer la similitude de mots, mais le modèle de langue ne nous donne aucune information sur la raison de cette similitude. En revanche, lorsque nous utilisons une ressource telle que WordNet pour calculer la similitude de mots, nous savons exactement la base de cette similitude. Un modèle de langue pourrait avoir une échelle réalisée, mais au coût de la compréhensibilité humaine de ses conclusions.

Le succès des méthodes d'échelle Web dépend crucieusement de l'entrée humaine sous forme d'hyperliens, cliquez sur Données ou des commentaires explicites de

l'utilisateur. Tirer parti de ces méthodes évolutives et automatisées pour créer des graphes de connaissances pouvant être compréhensibles et les utiliser pour obtenir un comportement intelligent répond vraiment à la manière dont un système AI devrait fonctionner.

Il est connu qu'une simple représentation graphe marquée est insuffisante pour de nombreuses tâches de performance souhaitées de l'AI. C'était précisément la raison pour développer des formalismes de représentation plus expressifs. En raison de la nécessité de remédier à l'économie et à l'ampleur de la création de cette représentation, les formalismes expressives sont moins couramment utilisés, mais cela n'implique pas que les problèmes de ce formalisme ont été résolus par les nouvelles méthodes d'apprentissage et de PNL. Certains exemples de tels problèmes incluent la conscience de soi, le raisonnement de la communication, le raisonnement à base de modèles, la conception expérimentale, etc. Un système de protection de soi peut reconnaître et exprimer les limites de ses propres connaissances. La compréhension de la communication du monde donne une capacité système à reconnaître évidemment des situations absolument absurde, par exemple une pièce de monnaie comportant une date de 1800 B.C., n'est pas une vraie pièce de monnaie. Les modèles de langue courante peuvent générer des phrases qui ne sont logiques qu'à une certaine longueur, mais elles ne manquent pas d'un modèle global du récit pour générer des textes plus connectés. Création de programmes AI pouvant maîtriser un domaine, formuler une hypothèse, concevoir une expérience et analyser ses résultats est un défi hors de portée de l'un des systèmes de génération de courant.

4. Résumé

Nous avons considéré trois manières différentes que les travaux sur les graphes de connaissances se croisent avec AI: comme un lit d'essai pour évaluer l'apprentissage des machines et les algorithmes de la PNL, en tant que facilitateur de la discipline émergente de la science des données graphiques, ainsi qu'un ingrédient de base pour réaliser le long terme Vision de l'AI. Nous espérons que ce volume inspirera beaucoup pour tirer parti de ce qui est possible grâce à la création évolutive des graphes de connaissances et à leur exploitation. Et pourtant, nous ne devrions pas laisser passer une vision à long terme de la création de représentations expressives et compréhensibles humaines pouvant également être créées de manière flagrante.

Chapitre 3

Conception et Réalisation

3.1. Introduction :

Ce chapitre est consacré à la partie conception et réalisation de notre application, qui consiste à recommander des films à un utilisateur en se basant sur les graphes de connaissances. Nous présentons quelque diagrammes, l'ontologie utilisée, l'environnement de travail choisi tout en présentant les langages et les outils utilisés ainsi que les copies d'écran de chaque étape.

3.2 . Modèle de données :

Dans cette section, nous présentons l'ensemble des définitions relatives aux concepts de base que nous manipulons dans ce chapitre.

Préférence : Une préférence est une formule qui permet d'hiérarchiser un ensemble d'objet par rapport aux intérêts et aux besoins d'un utilisateur

Profil utilisateur: Un utilisateur U1 qui a une grande préférence pour les films de fiction peut attribuer à ce prédicat une pondération (poids) =5 par exemple.

Descriptif de contenu : Le module Création de Profil films a pour tâche de créer le descriptif de films. Cela correspond, dans notre application, aux catégories associées aux films. Un film peut avoir plusieurs catégories.

3.7. Conception de notre application :

Le langage de modélisation que nous avons utilisé est Unified Modeling Language (UML). La fonction d'UML consiste à spécifier, visualiser, construire et documenter un système informatique.

3.3.1. Diagramme d'activité

Un diagramme d'activité fournit une vue du comportement d'un système en décrivant la séquence d'actions d'un processus. Les diagrammes d'activité sont similaires aux organigrammes de traitement de l'information, car ils montrent les flux entre les actions dans une activité.

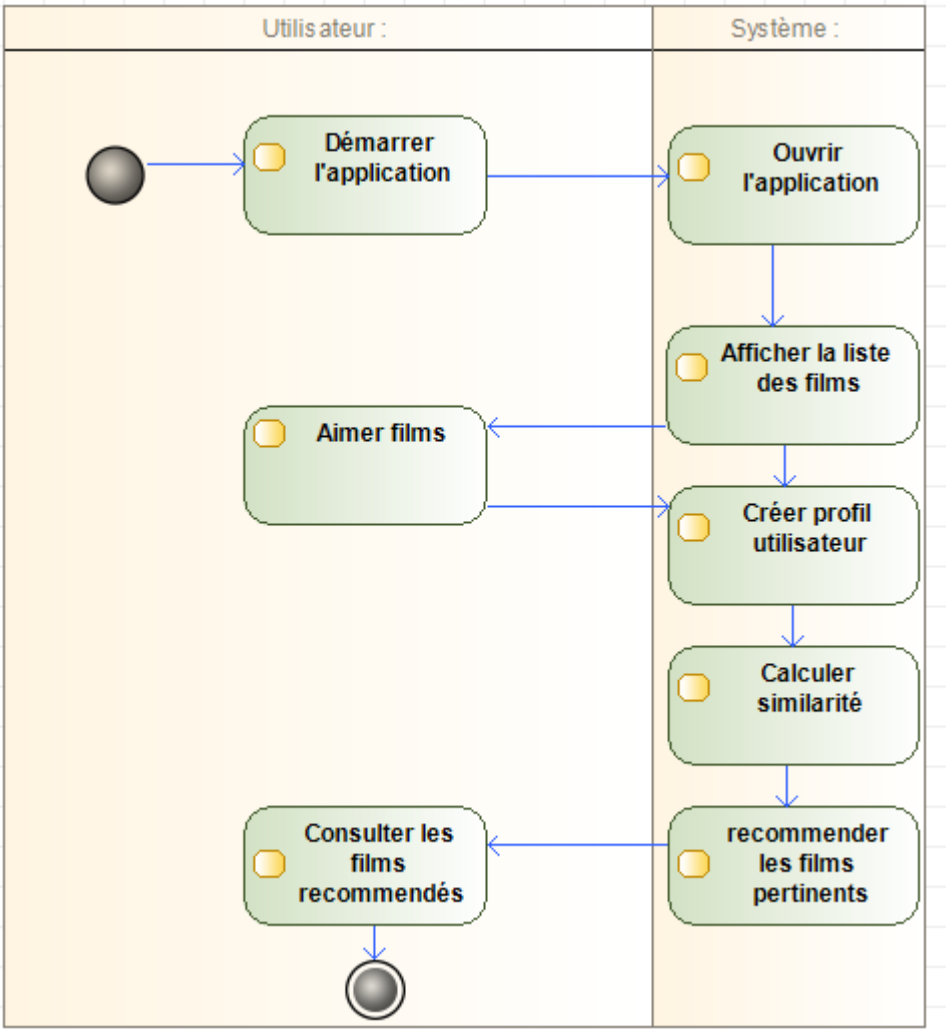


Figure3.1 : Diagramme d'activité

3.3.2. Diagramme des cas d'utilisation

Les diagrammes de cas d'utilisation (DCU) sont des diagrammes UML utilisés pour une représentation du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet, mais pour le développement, les cas d'utilisation sont plus appropriés.

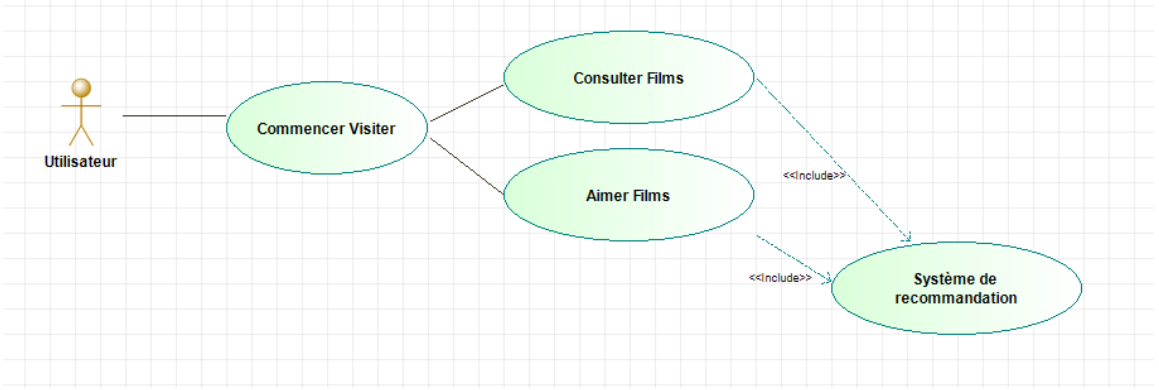


Figure 3.2 : Diagramme des cas d'utilisation

3.3.3. Diagramme de séquence :

Les diagrammes de séquence présentent la coopération entre différents objets. Les objets sont définis et leur coopération est représentée par une séquence de messages entre eux.

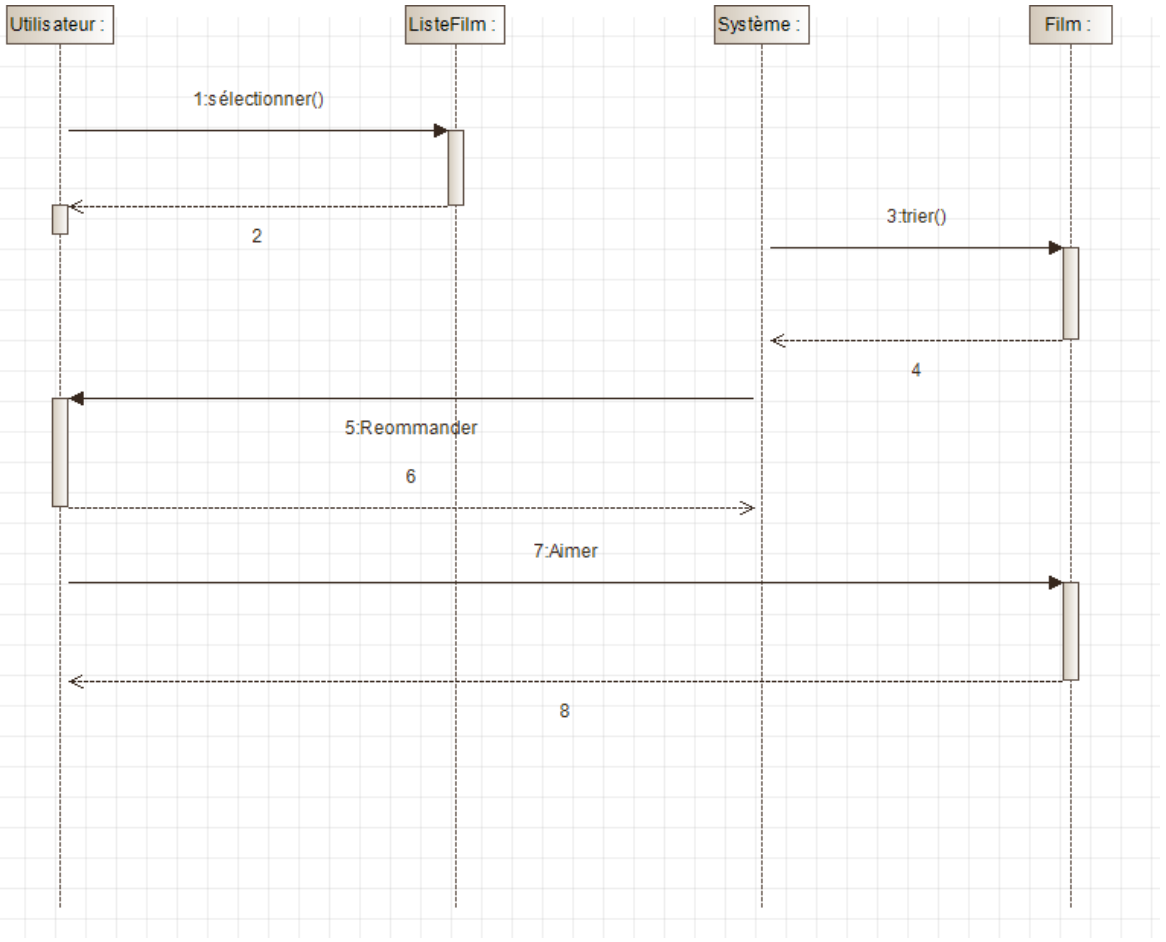


Figure 3.3 : Diagramme de séquence

3.3.4. Diagramme de collaboration

Un diagramme de collaboration montre les relations entre les objets jouant les différents rôles. Toutefois, ce diagramme ne contient aucune notion de temps.

L'utilisateur ait évalué un certain nombre de films, le module « Création profil utilisateur » construit son profil à partir de son historique et le renvoi au module « Matching ». Le module «Matching » calcule la similarité entre le profil utilisateur et les différents films, pour ainsi renvoyer les films pertinents au module «

Présentation ». Le module « Présentation » classe les films par ordre décroissant selon leurs degrés de similarités et les affiche pour les utilisateurs. La figure illustre l'échange d'informations en les différents modules.

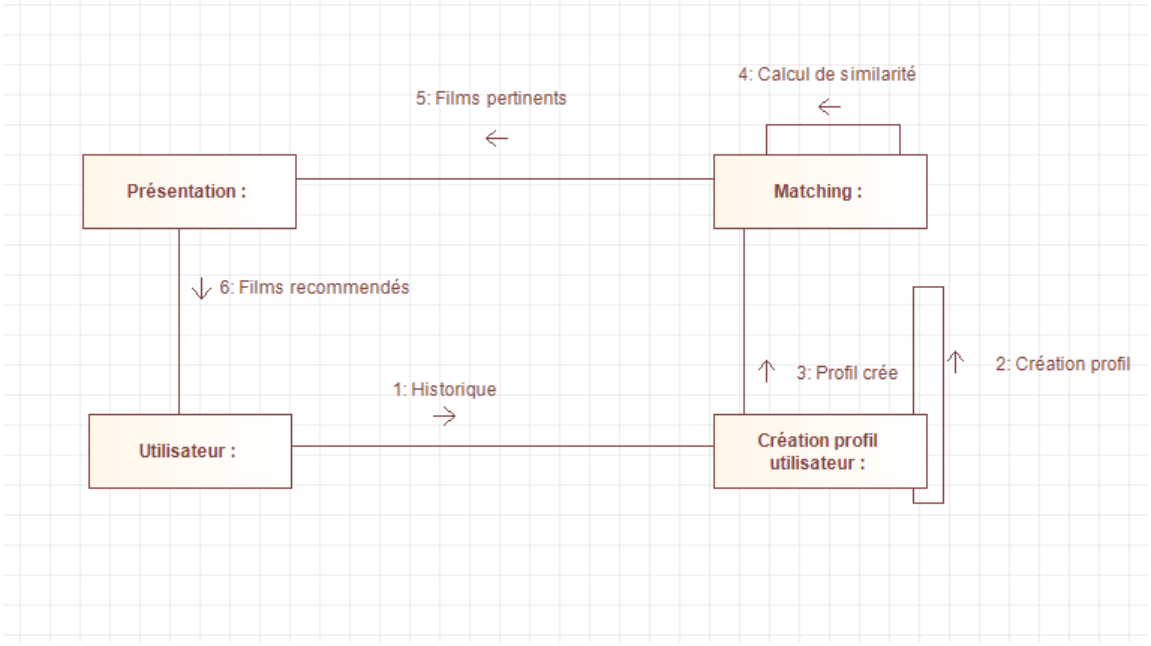


Figure 3.4 : Diagramme de collaboration

3.3.5. Diagramme de Classes

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation. Il s'agit d'une vue statique, car on ne tient pas compte du facteur temporel dans lecomportement du système. Le diagramme de classes modélise les concepts du domaine d'application ainsi que les concepts internes créés de toutes pièces dans le cadre de l'implémentation d'une application. Chaque langage de Programmation orienté objet donne un moyen spécifique d'implémenter le paradigme objet (pointeurs ou pas, héritage multiple ou pas, etc.), mais le diagramme de classes permet de modéliser les classes du système et leurs relations indépendamment d'un langage de programmation particulier.

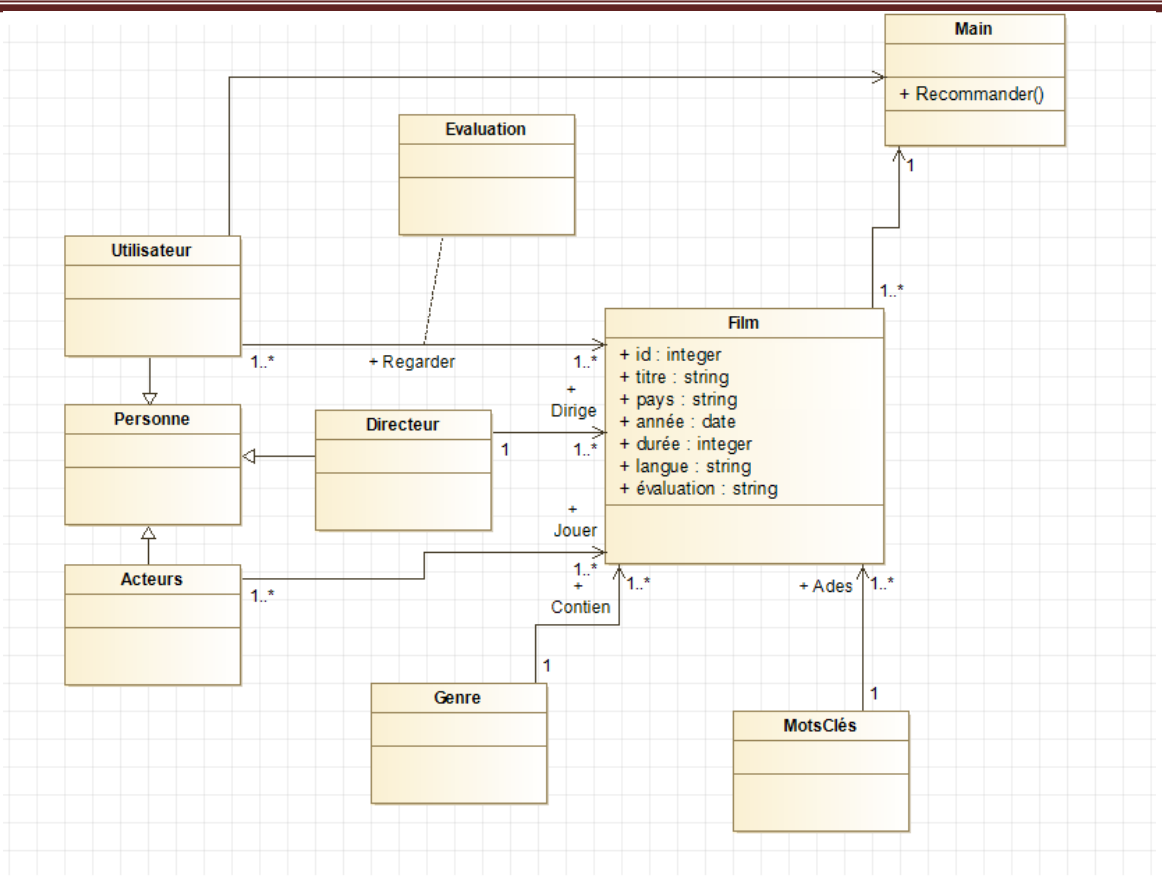


Figure3.5 : Diagramme de classes

Dans la section suivante, nous présentons notre approche.

3.4.Description de notre approche :

Notre système est composé de 4 étapes :

1) Exploration et traitement des données.

2) Création d'un profil utilisateur.

3) Calcul de score de chaque film en fonction de sa correspondance avec le profil utilisateur

4) recommandation des films qui ont eu des scores les plus élevés.

3.5… Exploration et traitement des données :

Le projet utilise un ensemble de données obtenu à partir de bien-connu site Web imdp.com .

Les données ont été importées dans un fichier .json qui à son tour été convertien tripple database pour pouvoir y accéder avec SPARQL pour l'évaluation et le traitement des données avant leur utilisation dans la recommandation.

3.6. Evaluations des films:

Chaque film est définie par un Identifiant, Titre , ville , mots_clés, l'année, directeur, les acteurs, la durée, les genres, Num_votes, , évaluation, Lien .


```
{
  "country": "USA",
  "keywords": {
    "keyword_3": "marine",
    "keyword_2": "future",
    "keyword_5": "paraplegic",
    "keyword_4": "native",
    "keyword_1": "avatar"
  },
  "gross": "760505847",
  "year": "2009",
  "imdb_id": "tt0499549",
  "director": "James Cameron",
  "runtime": "178",
  "language": "English",
  "title": "Avatar",
  "actors": {
    "actor_1": "CCH Pounder",
    "actor_3": "Wes Studi",
    "actor_2": "Joel David Moore"
  },
  "genres": "Action|Adventure|Fantasy|Sci-Fi",
  "num_votes": "886204",
  "content_rating": "PG-13",
  "imdb_rating": "7.9",
  "imdb_link": "http:\/\/www.imdb.com\/title\/tt0499549\/?ref_=fn_tt_tt_1"
},
```


Les utilisateurs :

Chaque utilisateur est définie par : genre et liste des films préfères .

3.7. Création d'un profil utilisateur :

L'utilisateur choisis ses films et genres préférés .

Pour chaque film choisis par l'utilisateur le système le compare avec la totalité des films existant en se basant sur les titres, les acteurs, les genres, l'année, les directeurs, langage, le pays , l'évaluation et les mots_clés des films.

3.8. Calcul de score de chaque film en fonction de sa correspondance avec le profil utilisateur

Titres : Si le film cible partage une partie de titre de film , le système accorde beaucoup de points par exemple : si vous avez choisis le film « HARRY POTTER 1 » le système vous recommande le film « HARRY POTTER 2 ».

Acteurs : Pour chaque films nous calculons le nombre des acteurs similaires * point supplémentaire (par exemple 5 points pour deux films ayant un acteur en commun, 10 points pour 2 acteur en commun …..) .

Genres : Nous utilisons le même principe utilisé par les acteurs.

Année : Plus les dates des productions des films sont proches plus c'est mieux.

Directeur : 5 points pour tout films ayant des directeurs similaires.

Langage : 10 Points pour des films utilisant le même langage.

Pays : 5 points pour tout films ayant le même pays.

Evaluation : La priorité à la recommandation des films ayant des scores plus élevés.

Mots-clés : La recommandation des films dépend de nombre des mots clés en commun : pour chaque mot clé en commun un point supplémentaire est ajouté.

3.9. Recommandation des films qui ont eu des scores les plus élevés.

Ps : la note calculée à partir ses 9 critères sera stocké dans un tableau.

Une liste des films ordonnée basé sur la note calculée sera recommandé à l'utilisateur.

Contribution à la recommandation des films par on graphe de connaissance :

() {

Entrée : utilisateurs, films ; L'utilisateur

accède à l'application.

2. L'utilisateur donne ses préférences.

3. Début.

4. Création de l'ensemble des règles ou prédicats.

5. Pour chaque élément de on graphe de connaissance film faire

i. Comparer l'intérêt de l'utilisateur avec les éléments de l'ontologie Film (calcul similarité)

ii. Si l'intérêt de l'utilisateur est similaire au contenu de l'ontologie

• ajouter le Film à la liste des films recommandés

6. Fin Pour.

7. Recommander l'élément le plus proche à l'utilisateur.

8. Si l'utilisateur veut une autre recommandation.

i. Passez à (2)

9. Fin.

Exemple

Démarrer la recommendation
Le film selectionner **Harry Potter** and **the Chamber of Secrets**
Score =0
Comparer avec le film Fantasia 2000
Score =0
: Pour chaque film nous calculons le nombre des acteurs similaires * point supplémentaire
Alors le score = 0+ 0
Genres : Nous utilisons le même principe utilisé par les acteurs. Mm genre alors score = 0 +5
Les dates des productions des films <3 alors score= 5+3 Mm langue alors score = 8 +10
Pas le mm paye alors score = 18 -3
Evaluation de film Harry Potter and the Chamber of Secrets = 7.4
L'Evaluation de film Fantasia 2000 = 7.3
Alors score = 15 + 0
Pour chaque mot clé en commun 10 points sont ajouté.

Le film selectionner **The Legend of Zorro**
Score =0
Comparer avec le **film Fantasia 2000**
Score =0
: Pour chaque film nous calculons le nombre des acteurs similaires * point supplémentaire
Alors le score = 0+ 0
Genres : Nous utilisons le même principe utilisé par les acteurs. Mm genre alors score = 0 +5
Les dates des productions des 5<films <10 alors score= 5+1 Pas le mm langauge alors score = 6 -5
Mm paye alors score = 1 +5
Evaluation de film The Legend of Zorro = 5.9
L'Evaluation de film Fantasia 2000 = 7.3 Alors score = 6 - 1
Pour chaque mot clé en commun 10 points sont ajouté.

3.10. Implémentation

3.10.1.Base de données utilisé

3.10.1.1.DBpedia

est un projet universitaire et communautaire d'exploration et extraction automatiques de données dérivées de Wikipédia. Son principe est de proposer une version structurée et normalisée au format du web sémantique des contenus de Wikipedia. DBpedia vise aussi à interconnecter Wikipédia avec d'autres ensembles de données ouvertes provenant du Web des données : DBpedia a été conçu par ses auteurs comme l'un des « noyaux du Web émergent de l'open data »1 (connu également sous le nom de Web des données) et l'un de ses possibles points d'entrée.

3.10.1.2.L'Internet Movie Database IMDB

(littéralement « Base de données cinématographiques d'Internet »), abrégé en IMDb, est une base de données en ligne sur le cinéma mondial, sur la télévision, et plus secondairement les jeux vidéo. IMDb restitue un grand nombre d'informations concernant les films, les acteurs, les réalisateurs, les scénaristes et toutes personnes et entreprises intervenant dans l'élaboration d'un film, d'un téléfilm, d'une série télévisée ou d'un jeu vidéo. L'accès aux informations publiques est gratuit. Un service payant, IMDbPro, donne accès aux informations supplémentaires susceptibles d'intéresser les professionnels. Créé le 17 octobre 1990 par l'Anglais Col Needham, c'est un site visité, en 2010, par plus de 57 millions d'usagers uniques chaque mois2, ce qui le plaçait au 39e rang des sites les plus visités au monde.

3.11. Outils de développement :

3.11.1. NetBeans IDE :

NetBeans IDE est un environnement de développement intégré (EDI), permet également de supporter différents autres langages, comme java, C, C++, JavaScript, XML, Groovy, PHP et HTML de faon native ainsi que bien d'autres (comme Python ou Ruby) par l'ajout de greffons. Il comprend toutes les caractéristiques d'un IDE

moderne (éditeur en couleur, projets multi-langage, refacturant, éditeur graphique d'interfaces et de pages Web). NetBeans constitue par ailleurs une plateforme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)). L'IDE NetBeans s'appuie sur cette plateforme.

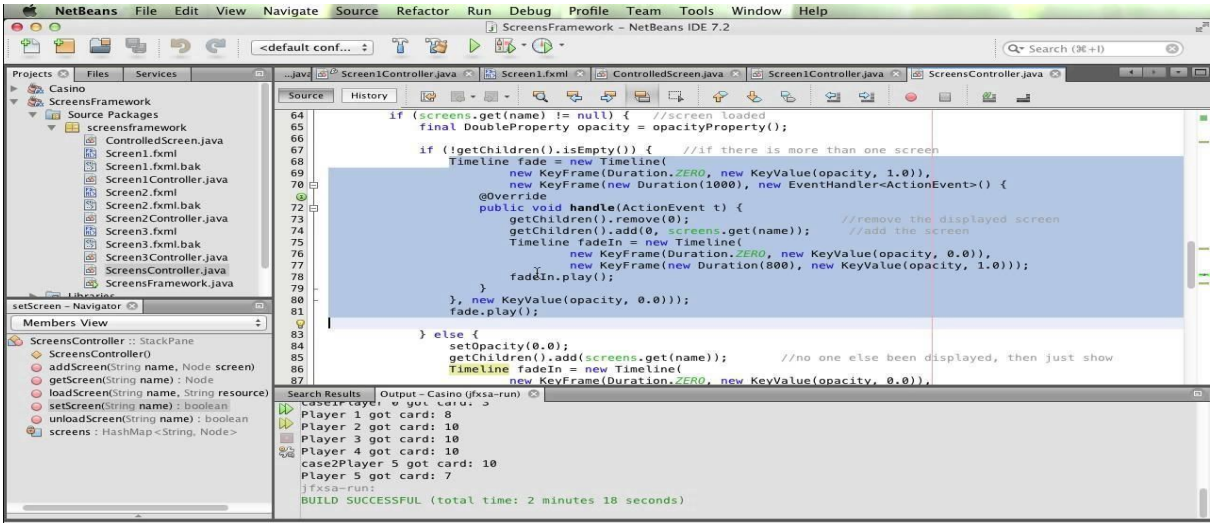


Figure3.6 fenêtre de programmation Sur Netbeans

3.12. Langage de programmation (Java)

Java est un langage de programmation et une plateforme informatique qui ontété créés par Sun Microsystems en 1995. Beaucoup d'applications et des sites Web ne fonctionnent pas si Java n'est pas installé et leur nombre ne cesse de croître chaque jour. Java est rapide, sécurisé et fiable. Des ordinateurs portables aux centres de données, des consoles de jeux aux super ordinateurs scientifiques, des téléphones portables à Internet, la technologie Java est présente sur tous les fronts.

C'est un langage de programmation à usage général, évolué et orienté objet dont la syntaxe est proche du C.

Ses caractéristiques ainsi que la richesse de son écosystème et de sa communauté lui ont permis d'être très largement utilisé pour le développement d'applications de types très disparates. Java est notamment largement utilisée pour le développement d'applications d'entreprises et mobiles.

Environnement Java

Java est un langage interprété, ce qui signifie qu'un programme compilé n'est pas

directement exécutable par le système d'exploitation mais il doit être interprété par un autre programme, qu'on appelle interpréteur.

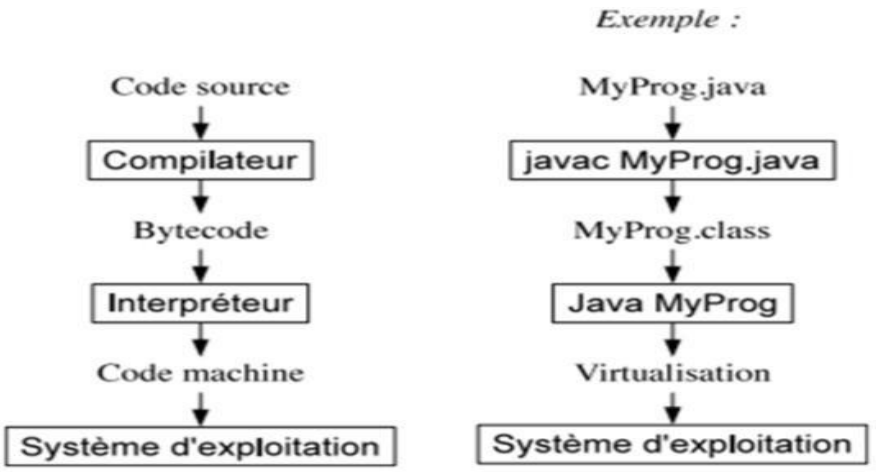


Figure3.7 architecture exécutable Code java

3.12.1. JavaFX :

JavaFX Intégration :

JavaFX est une bibliothèque graphique intégrée dans le JRE et le JDK de Java. Oracle la décrit comme « The Rich Client Platform », c'est-à-dire qu'elle permet de réaliser des interfaces graphiques évoluées et modernes grâce à de nombreuses fonctionnalités, telles que les animations, les effets, la 3D, l'audio, la vidéo, etc. Elle a de plus l'avantage d'être dans le langage Java, qui permet de réaliser des architectures avec des paradigmes objet, et aussi de pouvoir utiliser le typage statique. Dans ce premier tutoriel, nous allons voir ensemble un rapide historique de la bibliothèque pour ensuite découvrir les fondamentaux qui sont les classes « Stage
», « Scene », « Application » et le « threading » associé, pour finir nous verrons les
« Node » avec un exemple d'utilisation du « scene graphe ». Cette présentation ne fait pas dans le bling-bling, même si JavaFX est doué pour cela, en préférant se focaliser sur les concepts primordiaux d'une telle bibliothèque.

Bien comprendre ces basiques vous aidera bien à commencer pour ensuite pouvoir faire des interfaces de qualité et peut-être spectaculaires

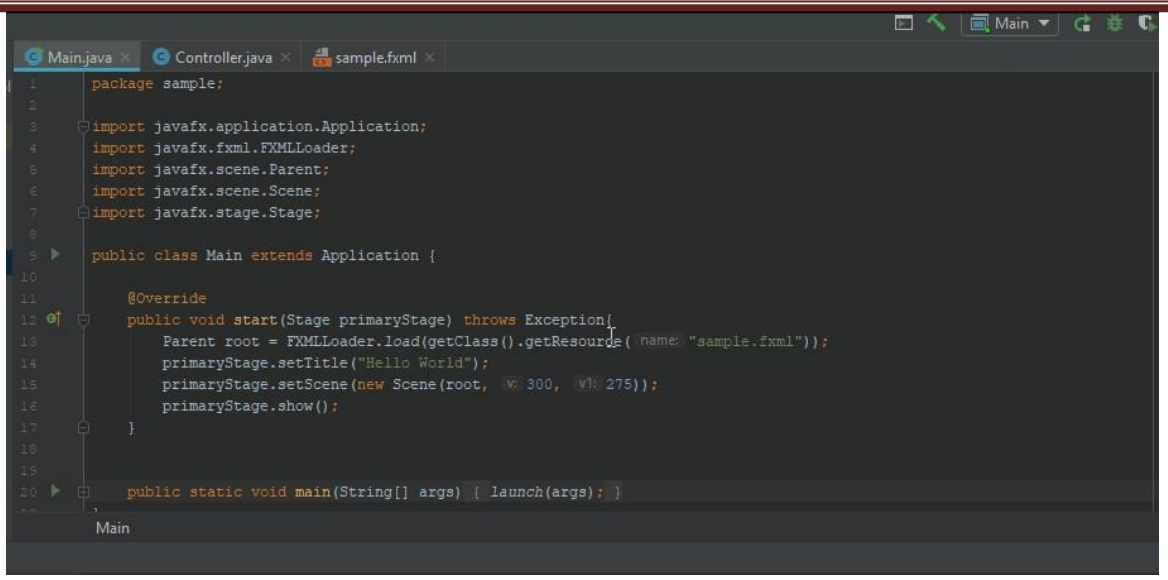


Figure3.8 : projet Java FX Main

3.12.2. Scene Builder

JavaFXSceneBuilder (Scene Builder) vous permet de concevoir rapidement des interfaces utilisateur d'application JavaFX en faisant glisser un composant de l'interface utilisateur d'une bibliothèque de composants de l'interface utilisateur et en le déposant dans une zone d'affichage du contenu. Le code FXML de la mise en page de l'interface utilisateur que vous créez dans l'outil est automatiquement généré en arrière-plan.

Scene Builder peut être utilisé comme un outil de conception autonome, mais il peut également être utilisé avec des IDE Java pour que vous puissiez utiliser l'IDE pour écrire, construire et exécuter le code source du contrôleur que vous utilisez avec l'interface utilisateur de votre application. Bien que SceneBuilder soit plus étroitement intégré à l'EDINetBeans, il est également intégré aux autres EDI Java décrits dans ce document. L'intégration vous permet d'ouvrir un document FXML à l'aide de Scene Builder, d'exécuter les exemples Scene Builder et de générer un modèle pour le fichier source du contrôleur.

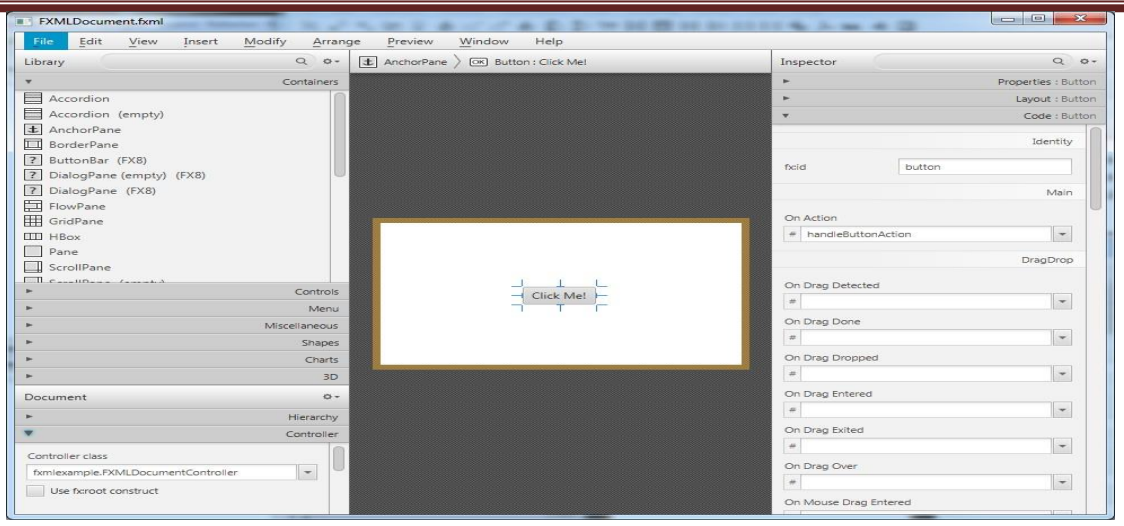


Figure3.9 : Utilisation Java FX Scene Builder

Derby Apache

Derby Apache est une base de données relationnelle open-source entièrement développée en Java par la fondation Apache.

Derby a la particularité de pouvoir être utilisé comme gestionnaire de base de données embarqué dans une application Java. Ce qui rend inutile l'installation et la maintenance d'un serveur de base de données autonome. A l'inverse Derby supporte aussi un mode de fonctionnement client-serveur.

3.12.3. JENA

Notre ontologie est implémentée en langage OWL (ontology Web Language), or les fichiers OWL sont inexploitables en état brut car leur structure est très complexe. Donc pour pouvoir l'exploiter il nous a fallu un « traducteur » capable de traduire les balises et la sémantique véhiculée par le fichier OWL en objet manipulable par des programmes. L'outil disponible qu'on a pu avoir est L'API JENA. Cet outil est développé par une équipe de la firme HP (Hewlett Packard) dans le cadre du Projet HP « Labs Semantic Web Programme » qui a pour but de réaliser un outil d'exploitation des fichiers OWL. JENA est développé entièrement en Java, elle donne aux programmes la possibilité d'exploiter le contenu des fichiers RDF et OWL (extraction du contenu sémantique de ces derniers).

Langages utilisés

3.12.4. SPARQL : SPARQL est un acronyme récursif pour SPARQL Protocol and RDF Query Language. Comme son nom l'indique, SPARQL est un terme général deux en un, c'est-à-dire qu'il représente à la fois un protocole et un langage de requête. Ainsi, comme on peut le deviner, le langage de requête SPARQL est un langage dont la syntaxe ressemble à SQL pour l'interrogation des graphes RDF par filtrage. Les caractéristiques du langage comprennent des modèles de base conjonctifs et des filtres de valeurs, notamment. Le protocole SPARQL est une méthode pour l'invocation distante des requêtes SPARQL. Elle spécifie une interface simple qui peut être prise en charge via HTTP ou SOAP et qu'un client peut utiliser pour émettre des requêtes SPARQL sur des points d'accès SPARQL. Le langage de requête SPARQL et le protocole SPARQL sont des produits du W3C et sont donc normalisés.

Implémentation de l'application :

Interfaces de l'application :

Dans ce qui suit, nous allons présenter quelques interfaces de notre application.

3.13. Interface d'accueil

L'utilisateur doit choisir les films, les genres selon son envie. Après cela, il doit cliquer sur <<Entrer>> pour voir les films RECOMMANDÉS par l'application.

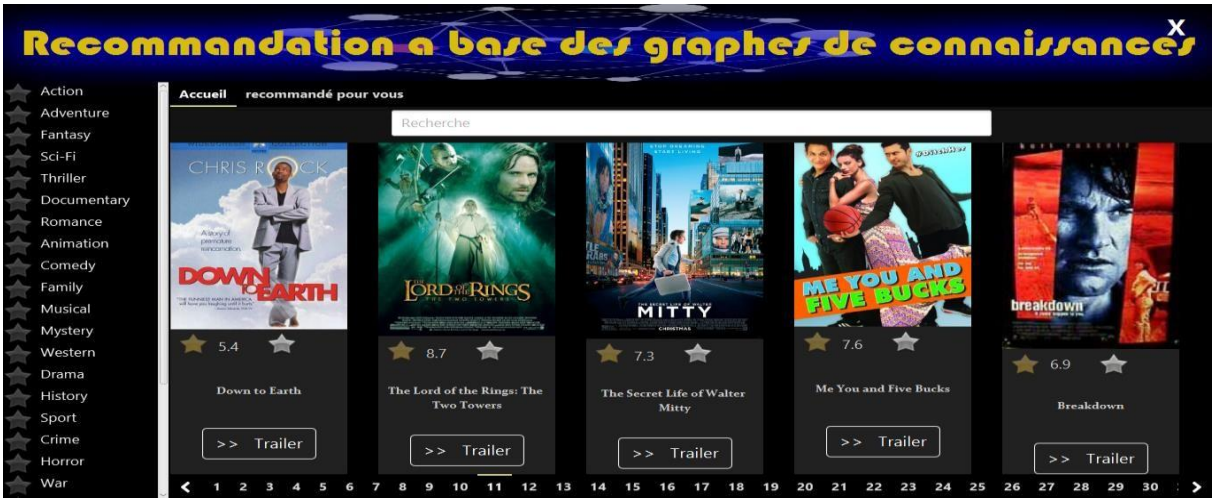
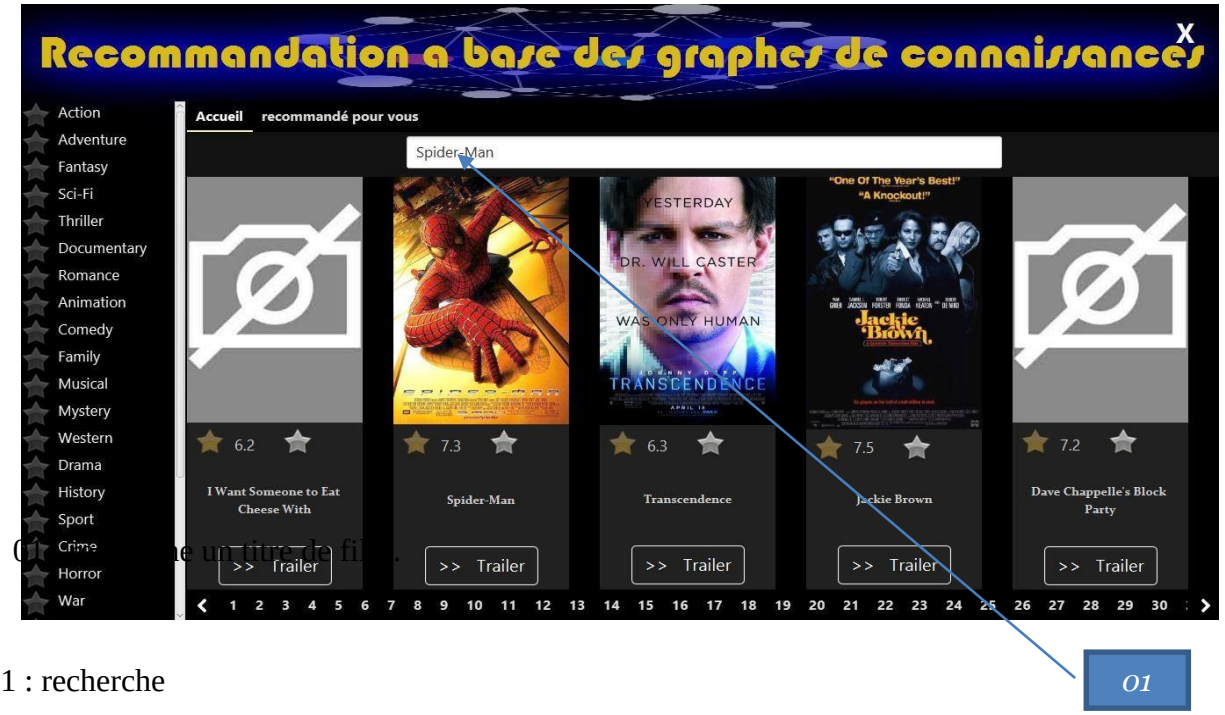


Figure 3.10 : Interface d'accueil pour l'application

Exemple :



01 : recherche un titre de film.

01

Recommandé pour vous Dernièrement, nous aurons l'interface présentée ci- dessous qui illustre la liste des films recommandés.

nous pouvons voir les films qu' on a basés sur.

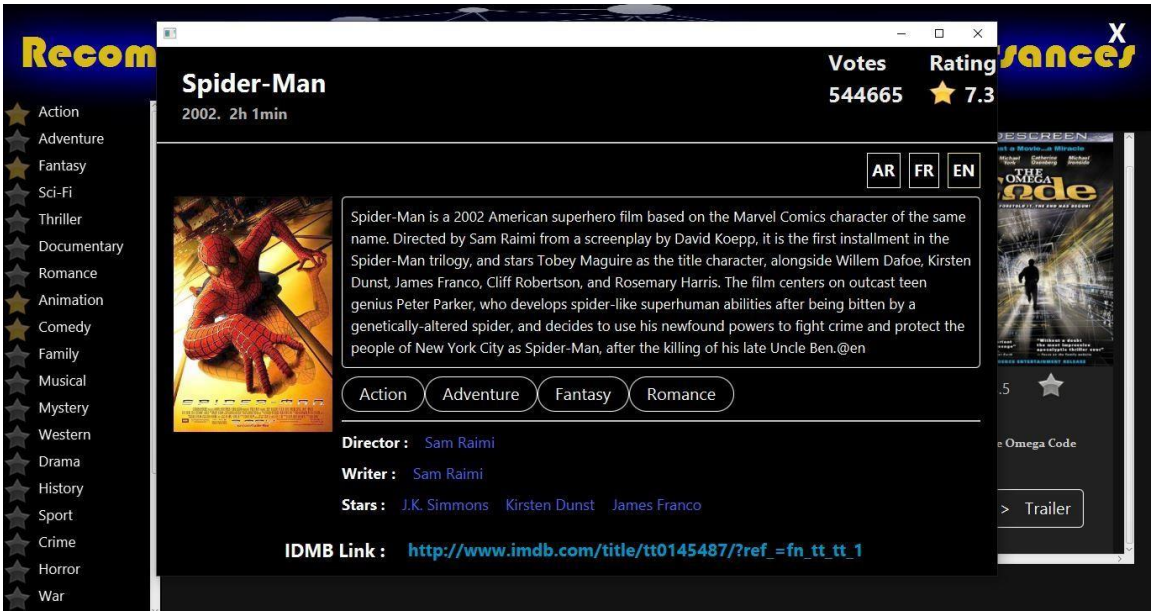
02 : coché les genres pour faire la recommandation.

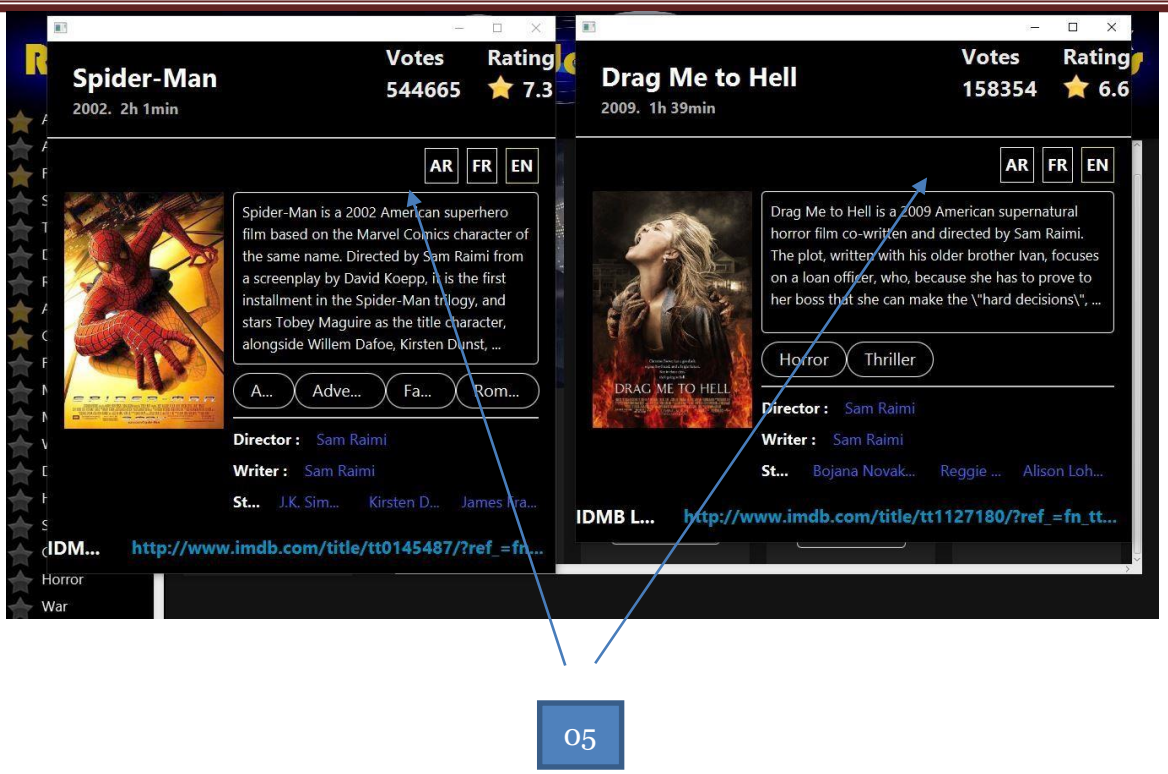
02

03 : les films Qu 'on basés sur.

04 : les films recommenders.

Figure : Interface de recommandation.





05

05 : traduire : Traduire le texte en anglais ou français ou arabe.

3.15.Conclusion

Ce chapitre a abordé l'aspect de l'implémentation de notre application. Dans ce chapitre, nous avons exposé et présenté les différentes phases suivies pour la conception et la réalisation de notre système de recommandation de films à base d'ontologie. En effet, nous avons explicité le modèle de données utilisé, la méthode de création des profils utilisateur et les descripteurs de films. Ensuite nous avons présenté les aspects techniques utilisés dans notre travail. Nous avons commencé par présenter les aspects d'implémentation qui sont utilisés dans notre travail. L'implantation repose essentiellement sur le langage JAVA avec NetBeans comme environnement de développement, Jena API pour exploiter et manipuler l'ontologie, le langage OWL pour la représentation de l'ontologie. Ensuite, nous avons présenté et commenter les différentes parties de notre application. Avec les premiers tests notre système a réalisé des résultats encourageants.

4. Conclusion

Depuis quelques années, les systèmes de recommandation ont une place particulièrement importante dans le marketing en ligne. Grâce à eux, des entreprises du e-commerce ont pu se différencier de leurs concurrents, faciliter la vie des clients actuels et atteindre leurs clients potentiels.

Selon les stratégies des entreprises, plusieurs techniques de recommandation sont intégrées pour adapter les besoins. Comme nous avons pu le voir dans l'article, ces méthodes ont différents avantages et inconvénients, aucune solution ente eux ne peut donc répondre à toutes les problématiques.

En réalité, les entreprises utilisent plusieurs approches et les combinent pour avoir une meilleure recommandation, évaluée par certains critères prédéfinis dans leur contexte ainsi que leurs objectifs.

Si le fonctionnement d'un système de recommandation est assez simple, sa mise en place est toutefois compliquée. Ces difficultés résident dans certains aspects comme la collection et la sélection de données pertinentes, la taille et la qualité de données, la rareté des données, la construction de profils utilisateurs, la prédiction pour des nouveaux profils d'utilisateurs ou de nouveaux produits.

Conclusion

خـــلاصـــة

في السنوات الأخيرة، كانت أنظمة التوصية مهمة بشكل خاص في التسويق عبر الإنترنت. تمكنت شركات التجارة الإلكترونية من التفريق عن أنفسهم من منافسيهم، لتسهيل حياة العملاء الحاليين والوصول إلى عملائهم المحتملين.

وفقا لاستراتيجيات العمل، يتم دمج العديد من تقنيات التوصية في تكييف الاحتياجات. كما رأينا في المقال، فإن هذه الأساليب لها مزايا وعيوب مختلفة، لا يوجد حل لا يمكن أن تلبي جميع القضايا.

في الواقع، تستخدم الشركات نهجا متعددة والجمع بينها للحصول على توصية أفضل، وتقييمها من قبل بعض المعايير المحددة مسبقا في سياقها وكذلك أهدافها.

إذا كانت عملية التوصية بسيطة للغاية، فإن تنفيذها معقد. هذه الصعوبات تكمن في بعض الجوانب مثل جمع واختيار البيانات ذات الصلة، وحجم ونوعية البيانات، وندرة البيانات والبناء تعريف المستخدم، والتنبؤ التشكيلات الجانبية للمستخدم جديدة أو منتجات جديدة.

5. Bibliographie

[1] Malone, T., Brobst, S., Cohen, S., Grant, K., and Turbak, F. (1987). Intelligent information des systemes de partage. In *Communications of the ACM*, volume 30, pages 390–402.

[2] Resnick, P. and Varian, H. (1997). Recommender systems. In Communications of the ACM, volume 40, pages 56–58.

[3] Maes, P. and Shardanand, U. (1995). Social information filtering: algorithmsfor automating“word ofmouth”. Inthe SIGCHI conference on Humanfactorsin computing systems, Denver, Colorado, United States. ACM Press/Addison-Wesley Publishing Co.

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

[5] A. T. NGUYEN. COCoFil2 : Un nouveau système de filtrage collaboratif basé sur le modèle des espaces de communautés. Thèse. Université joseph Fourier Grenoble I. Novembre 2006.

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

[8] Rao, N. and Talwar, V. (2008). Application domain and functional classification of recommender systems a survey. Desidoc journal of library and information technology, 28(3)
:17–36.

[9] Nguyen, A. T. (2006). COCoFil2 : Un nouveau système de filtrage collaboratif basé sur le modèle des espaces de communautés. PhD thesis, universite Joseph Fourier- Grenoble I.

[10] Arnautu, O. R. (2012). Mures : Un systeme de recommandation de musique. Master’s thesis, La Faculte des arts et des sciences Universite de Montreal.

[11] T. Slimani, B. Ben Yaghlane et K. Mellouli. Une extension de mesure de similarité entre les concepts d’une ontologie. 4rth International Conference: Sciences of Electronic, Technologies of Information and Telecommunications. Mars 2007.

[12] Z. Wu et M. Palmer. Verb semantics and lexical selection. Conference. In Proceedings of the 32nd Annual Meeting of the Associations for Computational Linguistics. Pages 133-138. 1994.

[13] P. Resnik. Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. Journal of Articial Intelligence Research 11. Pages 95-130. 1999.

[14] D. Lin. An Information-Theoretic Definition of similarity. In Proceedings of the Fifteenth International Conference on Machine Learning. Pages 296 - 304. 1998.

[15] J. Jiang et D. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In Proceedings of the 10Th International Conferenceon Research in Computational Linguistics, Taiwan. 1998.

[16] Neches, R; Fikes, R E; Finin, T; Gruber, T R; Senator, T; Swartout, W R. (1991). Enabling technology for knowledge sharing. AI Magazine , 12 (3), 36–56.

[17] Gruber, T. R. (1993a). A translation approach to portable ontology specification. Knowledge Acquisition , 5 (2), 199–220.

[18] Borst, W. N. (1997). Construction of Engineering Ontologies. Centre for Telematica and Information Technology, University of Tweenty, Enschede, The Netherlands.

[19] Studer, R., Benjamins, V. R., & Fensel, D. (1998). Knowledge Engineering: Principles and Methods. IEEE Transactions on Data and Knowledge Engineering , 25 (1-2), 161-197.

[20] Guarino, N. (1998). Formal Ontology in Information Systems. In N. Guarino (Ed.), 1st International Conference on Formal Ontology in Information Systems (FOIS'98) (pp. 3-15). IOS Press.

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