Présentation du domaine d'application

2.1 Introduction

De nos jours, il est devenu primordial que les utilisateurs des services en ligne (exemple : services de vente, bibliothèques en ligne,..etc.) accèdent à l'information qu'ils souhaitent, et qu'ils soient satisfaits des réponses obtenues pour qu'ils restent fidèles au service. Lorsqu'il s'agit d'un forum d'entraide en ligne, les traces des utilisateurs en termes de questions, de réponses, de commentaires, de votes, etc., sont des sources très riches permettant après exploitation d'améliorer la qualité du service. Parmi les exploitations utiles de ces services figure l'analyse de l'activité des utilisateurs afin de mieux comprendre la façon dont ils interagissent avec le système.
Certains systèmes mettent à la disposition des analystes intéressées une quantité de leurs propres données pour leurs analyses, comme StackExchange. Généralement ces données sont en format tabulaire, la représentation de ces données sous forme de graphes (les bases de données orientées graphes) permet une richesse de représentation des interactions entre les utilisateurs ce qui impose des opportunités d'analyses plus riches et orientées, à l'instar des analyses des interaction dans les réseaux sociaux.
Dans ce chapitre , nous allons présenter les systèmes d'entraide et un modèle commun pour la représentation de leurs données. Ensuite, nous allons présenter un système particulier d'entraide, à savoir StackExchange que nous utiliserons dans notre travail. Enfin, nous allons illustrer quelques possibilités d'analyse orientées graphes dans le système de StackExchange.

2.2 Forums d'entraide

L'évolution de la connaissance scientifique ne cesse de croître. Les gens, étudiants, chercheurs ou autres font face à des situations où ils ont besoin de l'aide dans un domaine ou un sujet qui l'intéresse. L'évolution de l'informatique, et entre autres, l'intelligence artificielle ainsi que l'apparition de l'Internet ont promu des systèmes de questions/réponses (Q/R)(appelés aussi forums d'entraide).
Le concept du forum d'entraide s'articule au-tout de trois questions principales (Figure 2.1). Dans ce qui suit, nous détaillons chacune de ces questions.

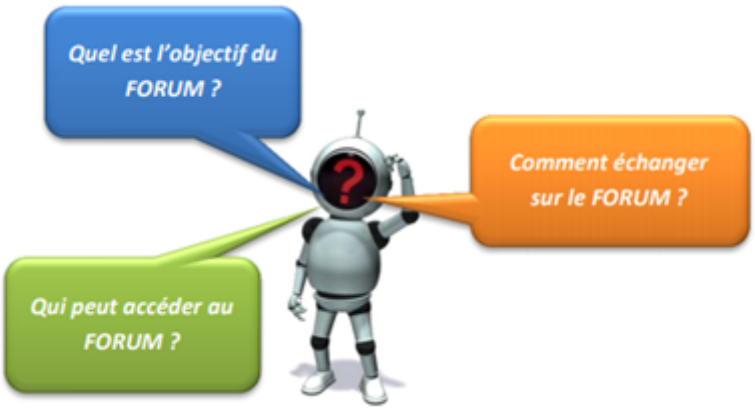

FIGURE 2.1: Le forum en trois questions [29]

2.2.1 C'est quoi un forum d'entraide ?

Les forums dits d'entraide foisonnent sur le Web. Ils touchent des domaines nombreux et variés : santé, technologie, éducation etc. Leur point commun est, de manière générale, la constitution d'un fil de discussion à partir d'une requête/demande et la/les réponses éventuelle(s) d'autres participants [26].
Les forums d'entraide sont implémentés sous forme de plateformes permettant la discussion ainsi que les échanges d'information dans le but de s'aider mutuellement, de manière réciproque, spontanée et gratuite entre des internautes (des communautés en ligne) pour tenter de surmonter les difficultés. généralement sans spécialiser dans un domaine spécifique [6].
En ce qui concerne l'expérience client, un forum d'entraide est un type de support en libre-service où les clients peuvent trouver une solution à leurs problèmes sans avoir à contacter un agent du service client. Comme dit le proverbe, le meilleur nombre de contacts est zéro. Les clients veulent résoudre eux-mêmes les problèmes [12].

2.2.2 Utilité des forums d'entraide

La recherche simple dans Google ou Wikipédia ne répond pas toujours à nos besoins d'apprentissage et de connaissance. Parfois on ne comprend jamais les gens qui répondent ou qui posent ces questions ni leurs intentions ou motivations. Les gens qui développent les sites d'entraide en ligne visent à régler ce problème par :

- L'augmentation de niveau de questions/réponses significative en ligne dans un domaine précis,
- Développement de systèmes de qualité qui permettent le contrôle des utilisateurs et par l'utilisateur, et la mise d'un système de reconnaissance à ceux qui posent/répondent à des meilleures publications (posts), badges, réputation, droits d'action sur le site, etc,
- Développement de sites extrêmement efficaces, tant dans la pertinence des questions posées que dans la lisibilité et la qualité des réponses obtenues (dans chaque domaine précis il y a des experts). [33]

2.2.3 Types de forums d'entraide

Il y a trois (3) types de forums [29] :

- **Forum standard** pour utilisation générale (le plus utilisé) : un forum ouvert, où chacun peut entamer une nouvelle discussion à tout instant.
  - Une seule discussion simple et un seul sujet de discussion sur lequel chacun peut s'exprimer (ne peut pas être utilisé avec des groupes séparés). Ce type de forum est des forums thématiques dans lesquels vous souhaitez canaliser les échanges.
  - Ou bien chaque personne commence une seule discussion : chaque étudiant ne peut entamer qu'une seule discussion, à laquelle chacun peut répondre.
- **Forum questions-réponses :** d'utilisateurs doivent poster un message avant de pouvoir consulter et répondre aux questions et messages des autres participants.
- **Forum standard** affiché comme un blog : un forum ouvert, où chacun peut entamer une nouvelle discussion à tout instant. Les sujets de discussion sont affichés sur une page, avec un lien « Discuter sur ce sujet » pour y répondre.

2.2.4 Classification des forums de Questions-Réponses

Le service ou forum de questions-réponses en ligne peut être classé en plusieurs types en termes de qui répond aux questions et comment les services maintiennent et contrôlent la qualité des informations. Des études ont classé le service de questions-réponses en ligne en trois catégories :

A. **Services de référence numérique :**
connus également comme référence virtuelle, il est une version étendue du service de référence traditionnel où les utilisateurs de la bibliothèque peuvent obtenir de l'aide des bibliothécaires à identifier les matériaux désirés. L'avantage vient du fait que les utilisateurs peuvent avoir accès à tout moment sans contraintes physiques. Les services actuels de ce type comprennent l'interrogation du bibliothécaire IPL de la bibliothèque publique Internet et les éducateurs du bureau de référence.

B. **Services d'experts :**
Les organisations autres que les bibliothèques offrent un service Q&R où les experts donnent une réponse aux questions de domaine spécifique. Certains sont payants (par exemple, NetWellness (http ://www.netwellness.org), tandis que d'autres offrent-service gratuit (par exemple, PickAnswer (http ://pickanswer.com).

C. **Questions-réponses sociales (questions-réponses communautaires) :**
Dans les services de Q&R social, tout utilisateur dans la communauté peut poser ou répondre aux questions. A partir du moment où une question peut recevoir l'attention de beaucoup de gens contrairement au service d'experts, les interrogateurs sont susceptibles de bénéficier de la sagesse des foules. L'ensemble des réponses faites par un grand nombre de personnes dépasse souvent des réponses d'experts en questions et réponses service traditionnel. Des études ont prouvé qu'un forum de questions-réponses sociales ,un système a fini par une meilleure qualité des réponses que celles des services de référence bibliothèque en comparant les services existants. [70].

2.2.5 Motivation de l'utilisation d'un forum de Questions-Réponses

Plus précisément, deux types de motivations poussent les gens à participer aux questions et réponses en ligne : Pourquoi les gens posent-ils des questions et répondent-ils à des questions ?

A. **Motivation à poser des questions :** Bien qu'il existe d'autres moyens de répondre aux besoins en informations, davantage de personnes ont posé des questions dans les services de questions / réponses en ligne. Qu'est-ce qui motive les gens à profiter de ces systèmes ? Et qu'attendent les utilisateurs de l'utilisation des systèmes ? En se basant sur les statistiques établies par Choi, Erik [23], les cinq facteurs les plus significatifs des comportements de demande dans ces statistiques sont :
   - Apprentissage ; auto-éducation par l'acquisition d'informations,
   - S'amuser en posant une question,
   - Demander des conseils ou des opinions pour prendre des décisions,
   - Trouver des informations pertinentes,
   - Acquérir un sentiment de sécurité grâce à la connaissance.

B. **Attentes des demandeurs :** La prochaine étape concernant le cycle de comportement des questions et des réponses consiste pour les demandeurs à évaluer les réponses pour voir si les réponses répondent à leurs attentes par rapport aux besoins d'information. Dans la même enquête examinée par Choi, Erik [23] le facteur le plus recherché était «Informations supplémentaires ou alternatives (4.03 / 5)», suivi de «Informations exactes et complètes». Il est intéressant de noter que les facteurs liés à l'information elle-même ont dépassé le «soutien social et émotionnel (2,47 / 5)» qui est lié aux besoins affectifs.

C. **Motivation à répondre** Les comportements de réponse ont suscité la curiosité des chercheurs, car il n'y aura pas de compensation explicite contre l'activité. voici les motivations les mieux classées basées sur les statistiques de Raban, Daphne R. ; Harper, F. Maxwell [53] :
   - Amélioration de la réputation,
   - Plaisir d'aider les autres,
   - Réciprocité,
   - Satisfaction,
   - Confirmation,
   - Intention de continuation.

2.2.6 Exemple des forums d'entraide en ligne

Parmi les sites d'entraide en ligne Q/R, qui sont les plus populaires dans le Web nous citons :

A. **Quora :**Le premier système de Q/R est Quora. Il s'agit avant tout d'une entreprise sur le web qui permet à ses utilisateurs de créer, d'éditer et d'organiser des questionsréponses. Le site organise les questions-réponses par sujets et permet aux utilisateurs de collaborer. La maison mère, Quora Inc., est localisée à Mountain-View, en Californie. Le site, fondé en juin 2009, et rendu disponible au public le 21 juin 2010, a atteint les 100 millions de visiteurs uniques par mois en mars 2016. [9]

B. **Yahoo ! Questions/Réponses (en anglais Yahoo ! Answers) :** ce site, couramment connu sous le sigle Yahoo ! Q/R, est un service communautaire collaboratif dans lequel les membres proposent des réponses aux questions posées par d'autres. Il s'agit d'un service du portail Yahoo ! qui a été lancé en décembre 2005 aux États-Unis, en 2006 en France. [15]

C. **Stack Exchange :** est un réseau de sites anglophones de questions et réponses à édition collaborative, chacun traitant d'un thème particulier. Deux exemples sont les sites StackOverflow qui est le plus important et populaire du réseau Stackexchange et dédié à la communauté des développeurs et des programmeurs, et AskUbuntu (en), qui offre des réponses aux problèmes rencontrés sur le système d'exploitation Ubuntu. Au 06 septembre 2020, le réseau regroupe 176 sites et plus de 13 millions d'utilisateurs. [10]

	Stackexchange	**Yahoo ! Answers**	**Quora**
Nature des questions posée par les utilisateurs	- privilégient les questions spécifiques et objectivement responsables aux questions de discussion large et ouverte	- large et ouverte	- large et ouverte
Nature des réponses aux Questions	- Les réponses doivent être tout aussi précises et appuyer leurs arguments avec des arguments et des sources (ce qui est positive)	-encourager des discussions et des opinions plus ouvertes	-encourager des discussions et des opinions plus ouvertes (mais ce n'est pas une chose négative)
Reglements dans le site	- Les sites Stackexchange ont par leur conception un objectif très étroit et des règles extrêmement strictes en matière de sujet	-moins strictes (mais pas faible)	-moins strictes (mais pas faible)
Domaines dans le site	- Stackexchange est composé de plusieurs secteurs : plusieurs sites, un domaine de connaissance chacun	-horizontal plusieurs catégories	Quora est horizontal : un site, tous les domaines de la connaissance
Tags	-oui on peut taguer une question(les tags est une table dans le schéma)	-oui	-oui on peut mais par commentaire
Classement(Ranking)	-oui on peut faire des classements pour des fins d'analyse	-oui on peut faire des classements pour des fins d'analyse	-oui on peut faire des classements pour des fins d'analyse
API	-oui	-oui	-Actuellement non

Table 2.1: Comparaison entre SE, Quora, Yahoo !answers. [33]

Le tableau ci-dessous (Table 2.2) nous permet de dégager les composants communs aux sites Web que nous venons de citer : les utilisateurs peuvent poser des questions, répondre à des questions, éditer des questions ou des réponses, commenter un ou plusieurs posts (un post correspond à une question, une réponses, etc.), identifier des questions, voter un post positivement (Up vote) ou négativement (Down vote) ou sur un score positivement ou négativement, gagner en réputation. Ces points communs sont généralement les éléments qui composent les systèmes d'entraide en ligne Q/R.

2.3 Modèle de données de forum d'entraide

2.3.1 Description des données

Stackexchange a mis au public une quantité de ses propres données à des fins d'analyse. La disponibilité des données de Stackexchange via le SEDE (StackExchange Data Explorer) nous a encouragées à l'utiliser comme référence de base pour le reste de notre travail. L'une des principales approches de la collecte de données de StackExchange consiste à utiliser la plate-forme par l'organisation StackExchange. L'utilisateur doit spécifier et exécuter la requête SQL qui à son tour renvoie les données qui peuvent être téléchargées en local au format CSV (valeurs séparées par des virgules). Cependant, la plate-forme Data Explorer impose une limite de taille de données stricte car elle renvoie au plus 50 000 enregistrements par requête. et cela va être la méthode que nous allons utiliser dans notre travail.

Dans qui suit nous présentons une description générale des entités de base et des relations entre ces entités :

2.3.2 Descriptions des Datasets

Comme nous avons mentionné plus tôt Stackexchange a mis à la disposition du public ses données en libre téléchargemet. Le SEDE (StackExchange Data Explorer) de StackExchange nous a encouragés à le choisir comme source de données principale vu que les données de StackExchange sont complètes et suffisantes et en plus sont des données réelles appropriés pour les besoins d'analyses. Dans le tableau 2.2, nous présentons une description générale des entités de site Stackexchange.

Data-set	Description
Posts	les publications des utilisateurs.
Users	les utilisateurs.
Comments	Les commentaires des utilisateurs.
PostHistory	l'historique d'un post
PostHistoryTypes	types d'historique des postes.
PostLinks	les liens vers un post.
PostNotices	avis sur un post (annotations).
PostNoticeTypes	Types d'avis sur un post.
PostsWithDeleted	Similaire à Post mais, incluant les postes supprimés
PostTags	les tags d'un post
PostTypes	les types des postes.
PostFeedback	Collections des up/down votes des 'Users' anonymes votants un post, ces informations n'ont pas un effet actuel (pas de réputation, etc.) c'est juste pour les statistiques.
ReviewRejectionReasons	Examiner les motifs de rejet pour les ' suggestededits '
ReviewTaskResults	Résultats de la tache de révision
ReviewTaskResultTypes	Types de Résultats de la tache de révision
ReviewTasks	Tâches de révision.
ReviewTaskStates	Etats de la tache de révision
ReviewTaskTypes	Types de tache de révision
SuggestedEdits	Suggestions de modifications à un utilisateur propriétaire d'un Post.
SuggestedEditVotes	Modifications suggérées à un utilisateur pour un type de vote.
TagSynonyms	Synonymes des 'tags'
Votes	Votes pour les postes.
VoteTypes	Types de votes .
Tags	Tags (balises) des postes.
Badges	Insignes (Bronze, Silver, Gold)
CloseAsOffTopicReasonTypes	Fermeture à cause de 'post' hors sujet
CloseReasonTypes	Types de raison de fermeture d'un Post pour un utilisateur (exemple : 2 = off topic)
FlagTypes	Types de marquage(Le marquage est un moyen de porter un contenu inapproprié à l'attention de la communauté).
PendingFlags	Les marquages en attente de l'acceptation ou le rejet.

TABLE 2.2: Description des données de StackExchange [11] , [2]

2.3.3 Descriptions des relations

Les entités précédentes sont reliées entre elles avec des associations père-fils et aussi des associations qui définissants des rôles (pour plus de details voir le schéma relationnel de StackExchange [11] , [2]), le tableau 2.3 montre la description des associations entre les entités).

N°	Nature de l'Association	Les entités	Description de l'Association
A1	Posts.OwnerUserId-> Users	Posts, Users	L'utilisateur propriétaire de Post
A2	Posts.OwnerDisplay Name ->Users	Posts, Users	le nom de l'utilisateur propriétaire de Post
A3	Comments.UserId->Users	Comments, Users	l'utilisateur propriétaire de commentaire.
A4	Comments.UserDisplay Name->Users	Comments, Users	le nom de L'utilisateur propriétaire de commentaire
A5	Comments.PostId->Posts	Comments, Users	le Post concerné par le commentaire.
A6	PostHistory.PostHistory TypeId->PostHistoryTypes	PostHistory, PostHistoryTypes	le type de 'PostHistory' concerné.
A7	PostHistory.PostId->Posts	PostHistory, Posts	L'historique d'un Post
A8	PostHistory.UserId->Users	PostHistory, Users	l'historique d'un Post qui concerne un utilisateur
A9	PostHistory.UserDisplay Name->Users	PostHistory, Users	Nom d'utilisateur relié avec l'historique d'un Post
A10	PostLinks.PostId->Posts	PostLinks, Posts	Liens vers le Post
A11	PostNotices.PostId->Posts	PostNotices, Posts	Notifications d'un Post
A12	PostNotices.Deletion UserId->Users	PostNotices, Users	
A13	PostNotices.PostNotice TypeId->PostNoticeTypes	PostNotices, PostNoticesTypes	Les types de notification
A14	PostNotices.Owner UserId->Users	PostNotices, Users	Utilisateur propriétaire de la notice.
A15	PostsWithDeleted.Post TypeId->Posts	PostsWithDeleted, Posts	le type de Post.
A16	PostWithDeleted.Last EditorUserId->Users	PostsWithDeleted, Users	le dernier utilisateur qui a édité le post.
A17	PostsWithDeleted. OwnerDisplayName->Users	PostsWithDeleted, Users	Le nom de propriétaire.
A18	PostsWithDeleted. OwnerUserId->Users	PostsWithDeleted, Users	l'utilisateur propriétaire de Post
A19	PostTags.PostId->Posts	PostTags, Posts	le Post Taguée.
A20	PostTags.TagId->Tags	PostTags, Tags	le Tag relié au Post.
A21	PostFeedback.PostId-> Posts	PostFeedback, Posts	le 'Feedback' de post
A22	PostFeedback.Vote TypeId- >VoteTypes	PostFeedback, VoteTypes	Le mode de vote relier au 'Feedback' (2=UpMod 3 = DownMod)
A23	ReviewRejectionReasons .PostTypeId->PostTypes	ReviewRejectionReasons, PostTypes	pour des raisons qui s'appliquent uniquement aux PostTypes : Wiki (5) ou Excerpt (6), sinon null
A24	ReviewTaskResults. RejectionReasonId-> ReviewRejectionReasons	ReviewTaskResults, ReviewRejectionReasons	Pour les 'suggested edits' répertorié dans ReviewRejectionreason. ReviewTaskResultTypeId : [2 = Approve (suggested edits) 3 = Reject (suggested edits)]
A25	ReviewTaskResults. ReviewTaskId- >ReviewTasks	ReviewTaskResults, ReviewTasks	La tache a révisé.

A26	ReviewTaskResults. ReviewTaskResultTypeId-> ReviewTaskResultTypes	ReviewTaskResults, ReviewTaskResultTypes	Pour connaitre le type de résultat de la tache de révision.
A27	ReviewTasks.PostId-> Posts	ReviewTasks, Posts	Le post concerné par les Taches de révision
A28	ReviewTasks.Suggested EditId- >SuggestedEdits	ReviewTasks, SuggestedEdits	Pour les 'SuggestedEdit' Qui ont leur propre numérotation pour des raisons d'historisation Note : SuggestedEdits est le : ReviewTaskTypeId N°1
A29	ReviewTasks.ReviewTask TypeId- >ReviewTaskTypes	ReviewTasks, ReviewTaskTypes	Le Type de la tache de Révision, exemple : 1 = Suggested Edit 2 = Close Votes 5= Late answer
A30	ReviewTasks.Review TaskStateId ->Review TaskStates	ReviewTasks, ReviewTaskStates	L'état de la tache de révision, 1 = Active 2 = Completed 3 = Invalidated
A31	ReviewTasks.Completed ByReviewTaskId-> ReviewTaskResults	ReviewTasks, ReviewTaskResults	Id associe a ReviewTaskResults qui stock le résultat (la sortie) d'une révision terminée
A32	SuggestedEdits.PostId-> Posts	SuggestedEdits, Posts	Post a éditer
A33	SuggestedEdits.Owner UserId- >Users	SuggestedEdits, Users	Utilisateur propriétaire de post a édité.
A34	SuggestedEditVotes. VoteTypeId- >VoteTypes	SuggestedEditVotes, VoteTypes	Modifications suggérée pour le type de vote (2 types de vote) : 2 = Approve (technically UpMod) 3 = Reject (technically DownMod).
A35	SuggestedEditVotes. UserId- >Users	SuggestedEditVotes, Users	L'utilisateur concerné par l'édition de vote
A36	SuggestedEditVotes. TargetUserId ->Users	SuggestedEditVotes, Users	L'utilisateur ciblé par le vote.
A37	SuggestedEditVotes. SuggestedEdi tId-> SuggestedEdits	SuggestedEditVotes, SuggestedEdits	Les modifications Suggérées pour un Post.
A38	TagSynonyms.Source TagName- >Tags	TagSynonyms, Tags	Le nom synonyme de nom de tag d'origine. Exemple : CSharp
A39	TagSynonyms.Target TagName- >Tags	TagSynonyms, Tags	Le nom de Tag originaire. Exemple : C#
A40	TagSynonyms.Owner UserId- >Users	TagSynonyms, Users	le créateur de Synonyme de Tag. Note : pour crée un TagSynonyms, il faut avoir plus de 2500 en réputation, un utilisateur avancé.
A41	TagSynonyms.Appro vedByUserId- >Users	TagSynonyms, Users	L'utilisateur qui Valide le Tagsynonym. Il peut être un modérateur.
A42	Votes.VoteTypeId-> VoteTypes	Votes, Votetypes	Le type de vote. Exemple : 1 = AcceptedByOriginator 2 = UpMod (upvote) 3=DownMod (Downvote) 10 = Deletion
A43	Votes.PostId->Posts	Votes, Posts	Le post ciblée par le vote

A44	Votes.UserId->Users	Votes, Users	'UserId' est présent juste quand le VotetypeId est dans (5, 6, 7,8), -1 quand l'utilisateur est supprimé. 5 = Favorite 6 = Close (Close votes are only stored , in table : PostHistory) 7 = Reopen 8= BountyStart (UserId and BountyAmount will also be populated)
A45	Badges.UserId->Users	Badges, Users	L'utilisateur gagne les badges pour chaque activité qui fait dans le site
A46	PendingFlags.PostId ->Posts	PendingFlags, Posts	Le post en attente de marquage.
A47	PendingFlags.Flag TypeId->FlagTypes	PendingFlags, FlagTypes	Le Type de marquage qui est utilise Dans l'attente de l'accepte ou le rejet ou, Autres
A48	PendingFlags.Close ReasonTypeId-> closeReasontypes	PendingFlags, CloseReasonTypes	Le marquage en attente est, Post ferme pour une raison (le typeId).
A49	PendingFlags.CloseAsO ffTopicReasonTypeId->Close AsOffTopicReasonTypes	PendingFlags, CloseReasonTypess	Le marquage en attente est, Post ferme pour une raison (le typeId)

TABLE 2.3: Description des associations de StackExchange [11] , [2]

2.3.4 Schéma de données de Stackexchange

La figure 2.2 représente le schéma de données de Stackexchange, ce schéma s'applique à tous les sous-réseaux dans tous les sites stackexchange (Data Science, Stackoverflow ...) :

Figure 2.2 : Schéma de données de Stackexchange A3

2.4 Fouille de graphes dans les forums de questions-réponses.

La fouille de données ou l'analyse d'une masse de données, permet d'extraire des connaissances invisibles au but de prédire l'évolution future ou de créer et suivre des stratégies, ainsi d'améliorés le rendement.
Cette analyse peut être effectuée par quatre approches comme présenté dans le premier chapitre (Analyse de centralité, Analyse de la communauté, Analyse de chemin, Analyse de connectivité), selon les buts et les besoins ainsi que la nature des données.
D'après la nature de forum d'entraide utilisé et son objectif, nous pouvons viser plusieurs axes principaux :

- identification des acteurs importants (les rôles joués par les individus en fonction de leurs liens) et des experts,
- extraction de communautés,
- prédiction de liens entre les acteurs de système,
- diffusion de l'information,
- recommandations et confiance,
- recherche dans les réseaux des forums d'entraides (amélioration des algorithmes).

Dans ce qui suit nous allons détailler certains des éléments sus-cités :

2.4.1 Identification des acteurs importants et des experts

L'identification des acteurs et des experts au sein d'un réseau d'entraides en ligne est motivée par le nombre important de données sur les liens entre les différentes interactions entre les utilisateurs.
La connaissance sur les relations entre les utilisateurs nous permet aussi de faire des classifications à partir des liaisons et la nature de liaisons qu'un utilisateur entretient avec les autres utilisateurs et non pas seulement les données de ces utilisateurs.

2.4.1.1 Identification des acteurs importants

Cette importance peut être mesurée par diverses mesures basées sur la centralité des nœuds, exactement d'un utilisateur :

- par rapport aux degrés : déterminer les utilisateurs les plus actifs ou populaires, par le calcul de nombre des réponses postées ou reçus que ces réponses soient correctes ou incorrectes, ou le nombre des commentaires postés ou d'autres interactions.
- par rapport à la proximité (nœuds proches) : trouver l'utilisateur le plus proches des autres, qui commente un grand nombre d'acteurs relativement au nombre total des commentaires.

2.4.1.2 Identification des experts

Une identification des experts sur un domaine est également possible ; elle est basée sur une analyse d'informations issues des votes sur les réponses proposées. On peut atteindre nos fins par un classement des individus selon le nombre des «Upvotes» ou «Downvotes». Cette approche consiste à déduire certaines propriétés d'un nœud à partir de ses liens et parmi les algorithmes, nous allons utiliser le plus connu qui est "PageRank".

2.4.2 Extraction de communautés

Il est possible d'utiliser d'algorithmes de détection des groupes, ou bien d'analyse de l'état du réseau ou de la communauté en fonction de la densité de connexion entre les utilisateurs de « StackExchange » ou les relations entre les balises des publications (Tags). Aussi on peut faire l'évaluation de ces groupes et leur façon de groupement ou partitionnement ainsi que leur tendance à se renforcer ou à se séparer.

2.4.3 Prédiction de liens

La prédiction des liens nous permet de déduire des associations futures entre les utilisateurs, des liens cachés, en fonction des interactions existant entre eux et de leurs proximités que ce soit par des réponses ou les commentaires ou d'autres interactions.

2.4.4 Diffusion de l'information

On peut connaitre les limites de diffusion de l'information en identifiant les utilisateurs intermédiaires et leur quantité d'influence sur le flux d'informations, qu'ils peuvent-être servir de pont entre les communautés des utilisateurs.

2.5 Conclusion

Au cours de ce chapitre ; nous avons abordé le concept des forums d'entraide, qui est une solution pour surmonter les difficultés qui rencontre l'utilisateur dans un domaine spécifique. Nous avons présenté la définition d'un forum d'entraide, ses types, et ses motivations. Aussi, nous avons défini les sites d'entraide en ligne Q/R les plus populaires dans le Web. Nous avons fait une comparaison entre Stackexchange, Yahoo ! Answers et Quora pour faire sortir les différents éléments entre eux. Nous avons étudié l'exploration des données Q/R de Stackexchange et enfin nous avons présenté quelques exemples d'analyse de graphes dans les forums de questions-réponses. Ces analyses ont pour objectif d'extraire des informations utiles et des connaissances ou extraire des savoirs à partir de la masse de données de notre base, par des méthodes et des algorithmes qui seront présenté dans le chapitre suivant.

Chapitre III

Application des algorithmes de fouille de graphe aux forums d'entraide

3.1 Introduction

Dans ce chapitre, nous présentons les algorithmes de fouille de graphes et leurs familles appliqués au forum d'entraide en ligne. Ces algorithmes sont la base de système d'analyse, nous permettant d'atteindre nos fins et d'extraire les connaissances cachés et invisibles. Dans ce qui suit, nous détaillons chaque algorithme et nous montrons comment l'appliquer au système d'analyse. Notons que les algorithmes que nous détaillons dans ce chapitre sont eux appartenant à la bibliothèque Graph Data Science de l'éditeur Neo4j.

3.2 Les algorithmes de fouille de graphe utilisés et leurs familles

Les algorithmes existent dans l'un des trois niveaux de maturité [43] :

- **Qualité de production :** Indique que l'algorithme a été testé en termes de stabilité et d'évolutivité. Les algorithmes de ce niveau sont préfixés par **gds.<algorithm>**.
- **Bêta :** Indique que l'algorithme est candidat au niveau de qualité de production. Les algorithmes de ce niveau sont préfixés par **gds.beta.<algorithm>**.
- **Alpha :** Indique que l'algorithme est expérimental et peut être modifié ou supprimé à tout moment. Les algorithmes de ce niveau sont préfixés par **gds.alpha.<algorithm>**.

3.2.1 Détection de communauté

Une communauté est un groupe de personnes, d'objets ou de concepts qui entretiennent des liens privilégiés parce qu'ils ont des affinités particulières, ou présentent des caractéristiques similaires, ou encore partagent des centres d'intérêts, etc. Au sens du graphe, une communauté est un cluster de nœuds qui sont fortement liés entre eux, et faiblement liés avec les nœuds situés en dehors de la communauté.

La bibliothèque Neo4j Graph Data Science (GDS) comprend plusieurs algorithmes de détection de communauté, mais selon l'utilité de chacun et notre objectif ainsi que le niveau de maturité, nous avons utilisé les algorithmes suivants :

3.2.1.1 Louvain

C'est un algorithme qui appartient au niveau -Qualité de production-, sa méthode est basée sur la maximisation de modularité , où la modularité [1] quantifie la qualité d'une affectation de nœuds aux communautés. Louvain se caractérise par limite de résolution et une scalabilité bien meilleure. L'algorithme de Louvain est un algorithme de clustering hiérarchique, qui fusionne récursivement les communautés en un seul nœud et exécute le clustering de modularité sur les graphes condensés, mais peut également s'exécuter sur des graphes pondérés, prenant en compte les poids de relation donnés lors du calcul de la modularité . [45]

A- L'algorithme

Etape 1 :
1- chaque nœud du graphe est affecté à sa propre communauté.
2- pour chaque nœud « i », on calcule le changement de modularité occasionné par la suppression de « i » de sa propre communauté et son déplacement dans la communauté de chacun des voisins « j » de « i » (« i » est placé dans la communauté qui entraine la plus grande augmentation de la modularité. Si aucune augmentation n'est possible, « i » reste dans sa communauté d'origine).
Ce processus est appliqué de manière répétée et séquentielle sur tous les nœuds jusqu'à ce qu'aucune augmentation de la modularité ne se produise.
Etape 2 :
Tous les nœuds d'une même communauté sont regroupés, et un nouveau graphe est construit où les nœuds sont les communautés de la phase précédente. [66]

B- Syntaxe d'exécution

Call gds.louvain.stream (configuration : Map)
YIELD
nodeId : Integer,
communityId : Integer,
intermediateCommunityIds : Integer [] [45]

C- Configuration

Dans le tableau 3.1, nous présentons une configuration générale pour l'exécution de l'algorithme :

1. Elle mesure la différence entre le nombre d'arêtes internes à la communauté et l'espérance de cette valeur dans le modèle de configuration

Nom	Type	Valeur par défaut	Optionnel	Description
nodeProjection	String, String [] ou Map	Null	Oui	Projection de nœud utilisée pour la création de graphes anonymes via une projection native.
relationProjection	String, String [] ou Map	Null	Oui	Projection de relation utilisée pour la création de graphe anonyme une projection native.
nodeProperties	String, String [] ou Map	Null	Oui	Propriétés du nœud à projeter lors de la création d'un graphe anonyme.
relationProperties	String, String [] ou Map	Null	Oui	Propriétés de la relation à projeter lors de la création d'un graphe anonyme.

TABLE 3.1: Configuration de l'algorithme louvain [45]

3.2.1.2 Composants fortement connectés

Un ensemble est considéré comme un composant fortement connecté s'il existe un chemin dirigé entre chaque couple de nœuds au sein de l'ensemble. L'algorithme permet de calculer la décomposition d'un graphe à un ensemble des composantes fortement connexes. Cet algorithme appartient au niveau -Alpha-. [48]

A- L'algorithme

Une application classique de l'algorithme de recherche en profondeur d'abord, un parcours de graphe qui commence à un nœud donné et explore autant que possible le long de chaque branche avant de revenir en arrière.
1- Marquer v comme sommet déjà visité.
2- Pour toutes les arêtes dirigées de v à un sommet w, **si** le sommet w n'est pas marqué comme déjà visité, **alors** ajouter w au composant de v et aller à 1 pour w.
3- commençant une nouvelle recherche en profondeur chaque fois que la boucle atteint un sommet qui n'a pas déjà été ajouté à un composant trouvé trouvé précédemment. [65]

B- Syntaxe d'exécutionL'algorithme

CALL gds.alpha.scc.stream (configuration : Map)
YIELD nodeId, compoinent [48]

C- Configuration

Dans le tableau 3.2, nous présentons une configuration générale pour l'exécution de l'algorithme :

Nom	Type	Valeur par défaut	Optionnel	Description
nodeProjection	String, String [] ou Map	Null	Oui	Projection de nœud utilisée pour la création de graphes anonymes via une projection native.
relationProjection	String, String [] ou Map	Null	Oui	Projection de relation utilisée pour la création de graphe anonyme une projection native.
nodeProperties	String, String [] ou Map	Null	Oui	Propriétés du nœud à projeter lors de la création d'un graphe anonyme.
relationProperties	String, String [] ou Map	Null	Oui	Propriétés de la relation à projeter lors de la création d'un graphe anonyme.

TABLE 3.2: Configuration de l'algorithme Composants fortement connectés [48]

3.2.1.3 Propagation d'étiquettes

L'algorithme de propagation d'étiquettes (Label Propagation Algorithm - LPA) est un algorithme rapide, appartenant au niveau -Qualité de production-, et détecte ces communautés en utilisant uniquement la structure du graphe comme guide, et ne nécessite pas de fonction objective prédéfinie ou d'informations préalables sur les communautés. [44]

A- L'algorithme

1- Chaque nœud est initialisé avec une étiquette de communauté unique (un identifiant).
2- Ces étiquettes se propagent à travers le réseau.
3- À chaque itération de propagation, chaque nœud met à jour son étiquette à celle à laquelle appartient le nombre maximum de ses voisins. Les liens sont rompus de manière arbitraire mais déterministe.
4- LPA atteint la convergence lorsque chaque nœud a l'étiquette de la majorité de ses voisins.
5- LPA s'arrête si la convergence ou le nombre maximal d'itérations défini par l'utilisateur est atteint. [44]

B- Syntaxe d'exécution

CALL gds.labelPropagation.stream(configuration : Map)
YIELD
nodeId : Integer, communityId : Integer [44]

C- Configuration

Dans le tableau 3.3, nous présentons une configuration générale pour l'exécution de l'algorithme :

Nom	Type	Valeur par défaut	Optionnel	Description
nodeProjection	String, String [] ou Map	Null	Oui	Projection de nœud utilisée pour la création de graphes anonymes via une projection native.
relationProjection	String, String [] ou Map	Null	Oui	Projection de relation utilisée pour la création de graphe anonyme une projection native.
nodeProperties	String, String [] ou Map	Null	Oui	Propriétés du nœud à projeter lors de la création d'un graphe anonyme.
relationProperties	String, String [] ou Map	Null	Oui	Propriétés de la relation à projeter lors de la création d'un graphe anonyme.
maxIterations	Entier	Dix	Oui	Nombre maximal d'itérations à exécuter.
nodeWeightProperty	Chaîne	Null	Oui	Nom de propriété du nœud contenant le poids. Doit être numérique.
relationWeightProperty	String	Null	Oui	Le nom de propriété qui contient le poids.
seedProperty	String	n/a	Oui	Utilisé pour définir le jeu initial d'étiquettes (doit être un nombre).

TABLE 3.3: Configuration de l'algorithme propagation d'étiquettes [44]

3.2.2 Centralité

Les algorithmes de centralité sont utilisés pour déterminer l'importance de nœuds distincts dans un réseau. Nous avons utilisé les algorithmes de centralité suivants :

3.2.2.1 Page Rank (Classement)

Le Page Rank est un algorithme qui appartient au niveau -Qualité de production- et qui mesure l'influence ou l'importance des nœuds dans un graphe orienté, il est calculé en fonction du nombre de relations entrantes et de l'importance des nœuds sources correspondants.
Le Page Rank est défini dans le document Google d'origine comme une fonction qui résout l'équation suivante modularité [46] :

$$\mathbf{PR(A) = (1\text{-}d) + d\ (PR(T1)/C(T1) + \ldots.. + PR(Tn)/C(Tn))}$$

Où,

- **A :** une page qui possède des pages (T1, T2,...,Tn) auxquelles elle pointe (c.-à-d. sont des citations).
- **d :** un facteur d'amortissement qui peut être réglé entre 0 et 1, il est généralement réglé sur 0,85.
- **C(A) :** est défini comme le nombre de liens sortant de la page A.

A- L'algorithme

1. initialisez toutes les composantes de **V** à **1/N** (**N** étant le nombre des nœuds du graphe et **V** le vecteur de répartition sur le graphe)
2. calculez **V'= V*M**, (où **M** est la matrice représentant le graphe)
3. copiez le contenu de **V'** dans **V**.
4. reprenez en 2 sauf si les deux vecteurs **V** et **V'** sont très proches, dans ce cas l'algorithme est terminé. [3]

B- Syntaxe d'exécution

Call gds.pageRank.stream (configuration : Map)
YIELD
nodeId : Integer,
score : float [46]

C- Configuration

Dans le tableau 3.4, nous présentons une configuration générale pour l'exécution de l'algorithme :

Nom	Type	Valeur par défaut	Optionnel	Description
nodeProjection	String, String [] ou Map	Null	Oui	Projection de nœud utilisée pour la création de graphes anonymes via une projection native.
relationProjection	String, String [] ou Map	Null	Oui	Projection de relation utilisée pour la création de graphe anonyme via une projection native.
nodeProperties	String, String [] ou Map	Null	Oui	Propriétés du nœud à projeter lors de la création d'un graphe anonyme.
relationProperties	String, String [] ou Map	Null	Oui	Propriétés de la relation à projeter lors de la création d'un graphe anonyme.
Facteur d'amortissement	Flotte	0.85	Oui	Le facteur d'amortissement du calcul du Page Rank
maxIterations	Entier	20	Oui	Nombre maximal d'itérations à exécuter.
relationWeightProperty	String	Null	Oui	Le nom de propriété qui contient le poids.

TABLE 3.4: Configuration de l'algorithme Page Rank [46]