



HAL
open science

Modèles neuronaux de recommandation basés sur les folksonomies

Tahar-Rafik Boudiba

► **To cite this version:**

Tahar-Rafik Boudiba. Modèles neuronaux de recommandation basés sur les folksonomies. Sciences de l'information et de la communication. Université Paul Sabatier - Toulouse III, 2022. Français. NNT : 2022TOU30256 . tel-04057677

HAL Id: tel-04057677

<https://theses.hal.science/tel-04057677v1>

Submitted on 4 Apr 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *25/11/2022* par :

Boudiba Tahar-Rafik

Modèles neuronaux de recommandation basés sur les folksonomies

JURY

PR. LYNDA TAMINE	Université Paul Sabatier UT3	Président
PR. AMEL BOUZEGHOUB	Telecom SudParis	Rapporteuse
PR. PATRICE BELLOT	Université Aix-Marseille	Rapporteur
PR. MOHAND-SAÏD HACID	Université Claude Bernard Lyon1	Examinateur
PR. MOHAND BOUGHANEM	Université Paul Sabatier UT3	Directeur de thèse
PR. TAOUFIQ DKAKI	Université Paul Sabatier UT3	Co-directeur de thèse

École doctorale et spécialité :

MITT : Image, Information, Hypermédia

Unité de Recherche :

Institut de Recherche en Informatique (UMR 5505)

Directeur de Thèse :

Mohand Boughanem, Dr. Taoufiq Dkaki

Rapporteurs :

Amel Bouzeghoub et Patrice Bellot

Modèles neuronaux de recommandation basés sur les folksonomies

Tahar-rafik Boudiba

Novembre 2022

Manuscript submitted for the doctor of science degree
PhD student : Tahar-Rafik Boudiba
Supervisor : Professor Mohand Boughanem & Dr. Taoufiq Dkaki
UNIVERSITÉ TOULOUSE ©Novembre 2022

© Novembre 2022 - Tahar-Rafik Boudiba
Contact me for any comments and corrections : Tahar-rafik.Boudiba@irit.fr
Institut de Recherche en Informatique de Toulouse, UMR 5505 CNRS,
Université Toulouse III Paul Sabatier,
118 route de Narbonne,
F-31062 Toulouse CEDEX 9

REMERCIEMENTS

J'adresse mes plus sincères remerciements à mes rapporteurs Monsieur *Patrice Bellot*, Professeur à l'Université d'Aix-Marseille et Madame *Amel Bouzeghoub*, Professeur à Telecom SudParis, pour avoir évalué ce manuscrit et pour la pertinence de leurs retours. Ensuite, je tiens également à remercier Madame *Lynda Tamine*, Professeur à l'Université Paul-Sabatier UT3 et Monsieur *Mohand-Saïd HACID*, Professeur à l'Université Claude Bernard Lyon 1, d'avoir participé à mon jury de thèse en enrichissant la qualité du débat. Enfin, je remercie chaleureusement mes directeurs de thèse : Professeur *Mohand Boughanem* et Docteur *Taoufiq Dkaki* pour la qualité de leurs encadrements, leurs disponibilités, leurs conseils avisés et le soutien infaillible dont ils ont fait preuve à mon égard. Je remercie aussi Monsieur *Gilles Hubert*, directeur de l'équipe IRIS qui m'a accueilli et intégré au sein de l'équipe ainsi que tous les permanents en particulier : Madame *Karen Pinel-Sauvagnat*, Monsieur *Guillaume Cabanac* et Monsieur *José Moréno*.

Du plus profond de mon cœur, je dédie ce modeste travail :

À toi Papa, merci de m'avoir accompagné, merci d'avoir cru en moi quand le monde entier doutait, merci de t'être démené pour élever deux super gamins, merci de nous avoir appris les échecs, les mathématiques et la lecture ... puisses-tu reposer en paix.

À ma mère vers qui je me suis très souvent tourné dans mes moments de déprime, je t'exprime mon amour inconditionnel et ma gratitude, merci pour tes prières.

À mon épouse *Lila*, merci pour ta patience, le soutien dont tu as fait preuve pendant la durée de cette thèse, merci pour tout le bonheur et la douceur que tu m'apportes au quotidien.

À mon grand frère *Mohamed-Ameziane*, à tous les moments d'enfance passés à tes côtés. Puissent nos liens fraternels se consolider et se pérenniser.

Aux enseignants qui ont fait de moi ce que je suis : Madame *F.Amirouche*, Monsieur *R.Ahmed-Ouamar* et Monsieur *S.Khemliche*.

Merci à mes amis de l'IRIT : *Malik, Damien, Antoine, Lila, Morgan, Nicolas, Raphaël, Luis, Maxime, Maël, Mira, Jesus, Alexis, Olivier, Alexandre, Eya et Paul*.

Enfin, mes remerciements vont aux personnels de l'IRIT et de l'université Toulouse III, particulièrement à Madame *Catherine Stasiulis*, Madame *Chantal Morand* et à Madame *Martine Decalmès*.

*«Sometimes it's the people no one imagines anything of who do the things that no one can
imagine.»*

— [Alan Turing]

ABSTRACT

The works presented in this thesis are mainly based on folksonomies : a collaborative data structure that emerged with the Web 2.0. The increased need for collaborative services to index, classify and retrieve information in various forms, the huge amount of data generated as well as the heterogeneity of the data sources has favored the emergence of processes by which users qualify online contents, notably by associating tags (or descriptors) to them. Such processes produce data that can be modeled by hypergraphs of tripartite graphs representing folksonomies and constituting the set of tags assigned by users to resources. Since these tags are user-generated, they form a new set of personal data that reflects users' interests and preferences for any resources they have tagged. Such tags are used in various information retrieval tasks, especially those based on user profiles such as personalized information retrieval and recommendation.

These profiles are built using different approaches : as a simple list of weighted keywords, vectorial, temporal or based on *clustering*. It is in this context that we approached our first contribution. We were able to implement an unsupervised method for the construction of folksonomy profiles. We found that these profiles, which we represented in the form of evolving interest clusters, were able to better describe users and resources efficiently. They have been successfully implemented in a recommendation process and then evaluated among classical vector profile construction methods. This helped us to better understand the diverse and evolving nature of user interests associated with social annotations. Specifically, by exploiting the normalized degree of user preference propagated by tags.

Our works then turned to the study of recent work in representation learning, which exploits neural models to enrich profiling methods with continuous representations of users and items. We found that such models have significantly improved the performance of classical recommendation systems. This led us to consider enriching these neural models by associating user and item representations with tag embeddings. These tag embeddings that we extracted from pre-trained neural language models allowed us to formalize neural collaborative filtering models. This is the context of our second contribution. After exploiting these representations (embeddings) through different neural architectures, we have highlighted which neural models allow the extraction of more precise characteristics and provide information on the contextual semantics of tags. Finally, the integration of these representations within these neural architectures, also allowed us to address other underlying issues and which are related in particular to the way to effectively include the neighborhood of tag representations in a neural collaborative filtering model for recommendation. In this sense, we have evaluated the accuracy of the predictions of users' notes from annotation activity history and then determined which models lead to better performances compared to classical neural collaborative filtering approaches that do not necessarily integrate this type of representation.

Keywords : Folksonomies, information retrieval, profiles, neural network models, tags, clustering, recommendation

RÉSUMÉ

Les travaux présentés dans cette thèse s'articulent principalement autour des *folksonomies* : une structure de données collaborative ayant émergé avec le Web 2.0. Les besoins accrus des services collaboratifs pour indexer, classer et retrouver l'information sous diverses formes, l'immensité des données engendrées ainsi que l'hétérogénéité des sources de données, ont favorisé l'apparition de processus par lesquels des utilisateurs qualifient des contenus en ligne, notamment, en leur associant des tags (ou des descripteurs). De tels processus produisent des données pouvant être modélisées par des hypergraphes, des graphes tripartites représentant les *folksonomies* et constituant l'ensemble des tags assignés par les utilisateurs à des ressources. Du fait que ces tags soient générés par les utilisateurs, ils forment un nouvel ensemble de données personnelles qui reflètent les intérêts ainsi que les préférences des utilisateurs à l'égard de ressources qu'ils ont annotées. Ces tags sont utilisés dans différentes tâches de recherche d'informations, plus particulièrement dans celles qui se basent sur des profils d'utilisateurs telles que la recherche d'informations personnalisées et la recommandation.

Ces profils sont construits en exploitant différentes approches : sous la forme de simple liste de mots-clés pondérés, vectorielles, temporelles ou encore regroupés sous la forme de *clusters*. C'est dans ce contexte que nous avons abordé notre première contribution. Nous avons mis en œuvre une méthode non supervisée pour la construction de profils *folksonomiques*. Nous avons constaté que ces profils que nous avons représentés sous la forme de clusters d'intérêts évolutifs étaient plus aptes à décrire efficacement les utilisateurs et les ressources. Ils ont d'ailleurs été implémentés avec succès dans un processus de recommandation, puis comparés parmi des méthodes de construction de profils vectoriels classiques. Cela nous a aidés à mieux appréhender la nature diverse et évolutive des intérêts de l'utilisateur véhiculés par les annotations sociales. Plus précisément, en exploitant le degré de préférence normalisé de l'utilisateur propagé par les tags et biaisé par une fonction temporelle.

Notre travail s'est ensuite porté sur l'étude des récents travaux en apprentissage de représentations qui exploitent notamment des modèles neuronaux pour enrichir les méthodes de profilage grâce à des représentations continues d'utilisateurs et d'items. En constatant que de tels modèles ont permis d'améliorer de manière significative les performances des systèmes de recommandation classiques. Cela nous a poussé à envisager d'enrichir ces modèles neuronaux en y associant des représentations d'utilisateurs et d'items intégrant des plongements lexicaux ou *embeddings* de tags. Ces *embeddings* de tags que nous avons extraits à partir de modèles de langues neuronaux pré-entraînés nous ont permis de formaliser des modèles de filtrage collaboratif neuronaux. C'est dans ce cadre que s'inscrit notre deuxième contribution. Nous avons mis en évidence à l'issue de l'exploitation de ces représentations (*embeddings*) à travers différentes architectures neuronales, quels étaient les modèles neuronaux permettant l'extraction de caractéristiques plus précises et renseignant sur la sémantique contextuelle des tags. Enfin, l'intégration de ces représentations au sein de ces architectures neuronales, nous a

aussi permis d'aborder d'autres problématiques sous-jacentes et qui sont liées notamment au moyen d'inclure efficacement le voisinage des représentations de tags dans un modèle de filtrage collaboratif neuronal pour la recommandation. Dans ce sens, il a été question d'évaluer la précision des prédictions des notes d'utilisateurs à partir d'historique d'activité d'annotation puis de déterminer quels modèles conduisent à de meilleures performances par rapport aux approches neuronales de filtrage collaboratif classiques n'intégrant pas forcément ce type de représentation.

Mots clés : Folksonomies, Recherche d'information, profils, modèles neuronaux, tags, clustering, recommandation

PUBLICATIONS

Our ideas and contributions have already been published in the following scientific publications :

Conférences

1. Boudiba, T. and Dkaki, T. (2022) Tag-based embeddings for neural collaborative filtering. In CIRCLE 2022 : Joint Conference of the Information Retrieval Communities in Europe, Samatan, Gers, France, 2022.
2. Boudiba, T. and Dkaki, T. (2022) Exploring contextualized tag-based embeddings for neural collaborative filtering. In ICAART : 14th International Conference on Agents and Artificial Intelligence, 2022. ACM SIGAI.
3. Boudiba Tahar-rafik & Ahmed-Ouamer Rachid. Approche temporelle pour la génération de profil folksonomiques. In : INFORSID : INFormatique des ORganisations et Systèmes d'Information et de Décision. 35e édition - Toulouse 2017.

Journal

1. Boudiba, T. R., Ahmed-Ouamer, R. (2018). Approche temporelle pour la génération personnalisée de profils folksonomiques. Document numerique, 21(1), 91-114.

TABLE DES MATIÈRES

Contexte général	25
1 Annotations sociales	27
2 Multiples centres d'intérêts des usagers des folksonomies	28
3 Modèles de représentation neuronaux pour la recommandation	29
4 Problématiques et questions de recherche	30
5 Contributions	31
6 Organisation du manuscrit	31
Folksonomies en RI	33
1 FOLKSONOMIES EN RI	35
1 Concepts de base en recherche d'information	36
1.1 Indexation	36
1.2 Indexation classique	37
1.3 Indexation sociale	38
1.4 Appariement	39
1.5 Modèles de recherche d'information	40
1.6 Évaluation des systèmes de recherche d'information	41
1.7 Principe de Folksonomies d'utilisateurs en RI	43
1.7.1 Définitions formelles	44
1.7.2 Folksonomie : Définition formelle	44
1.7.3 Personomie : Restriction de folksonomies	44
2 Représentation des tags	44
3 Tâches de RI associées aux folksonomies	46
3.1 Personnalisation de l'accès à l'information	46
3.2 Classification et topic modeling	47
3.3 Recommandation	48
4 Conclusion	50
Modèles de représentation de folksonomies	51
2 MODÈLES DE REPRÉSENTATIONS DE FOLKSONOMIES	53
1 Introduction	53
2 Caractéristiques et propriétés remarquables	54
3 Modèles de représentation de tags issus de folksonomies	56
3.1 Représentations de tags en sac de mots	56
3.1.1 Représentations de tags dans un espace vectoriel	57
3.1.2 Représentation de tags issues de modèles de langues	59
3.2 Représentation de tags en graphe	61
3.3 Représentation temporelle de tags	63
3.4 Représentations distribuées de mots et de tags	65
3.4.1 Représentations distribuées des mots	65
3.4.2 Représentation distribuée agrégée	66
3.4.3 Représentation distribuée de tags	66

	3-4.4	Représentation contextuelle de tags	69
4		Discussion	69
5		Conclusion	71
Modèles de recommandations basés sur les tags			73
3		MODÈLES DE RECOMMANDATION BASÉS SUR LES TAGS	75
1		Introduction à la recommandation	75
	1.1	Notations et exemple introductif	76
2		Folksonomies pour la recommandation	78
	2.1	Approches basées sur le contenu	80
	2.2	Approches basées sur le filtrage collaboratif	82
	2.2.1	Approches basées sur le voisinage de représentations tags	83
	2.2.2	Exploitation de profils folksonomiques pour le filtrage collaboratif	84
	2.3	Approches de Filtrage collaboratif basées sur l'apprentissage au- tomatique	86
	2.3.1	Exploitation des annotations sociales dans des approches de filtrage collaboratif basées sur l'apprentissage automa- tique	88
3		Modèles de réseaux de neurones pour la recommandation	89
	3.1	Filtrage collaboratif basé sur des modèles neuronaux <i>end-to-end</i> .	90
	3.2	Modèles neuronaux <i>end-to-end</i> de recommandation basés sur l'ap- prentissage de représentation de graphes	92
	3.3	Modèles neuronaux de recommandation basés sur l'apprentissage de représentations	95
	3.4	Modèles neuronaux pour la recommandation basés sur des repré- sentations denses de tags	97
4		Limitations	99
5		Discussion	101
6		Conclusion	102
Statistiques de la collection			105
4		DESCRIPTION DE LA COLLECTION	107
1		Introduction	107
2		Analyse de la collection <i>MovieLens</i>	107
3		Statistiques descriptives	108
	3.1	MovieLens-10M	108
	3.2	MovieLens-20M	108
	3.3	MovieLens-25M	110
	3.4	Plongement lexicaux de tags avec <i>word2vec</i> et visualisation à l'aide de <i>t-SNE</i>	110
	3.5	Tableau récapitulatif	113
Contribution I			115
5		MULTIPLÉS CENTRES D'INTÉRÊTS D'UTILISATEURS ÉVOLUTIFS	117
1		Introduction	117
2		Motivations	117

2.1	Profilage des centres d'intérêts multiples des utilisateurs	119
3	Description générale de l'approche	120
3.1	Approche détaillée	122
3.2	Signature de clusters	123
4	Expérimentations	124
4.1	Résultats et discussions	126
4.1.1	Impacte du degré de préférence de l'utilisateur	126
4.1.2	Impact du clustering	127
4.1.3	Évaluation du classement personnalisé des ressources par utilisateur	128
4.1.4	Évaluation du classement de pertinence	128
4.1.5	Évaluation globale	129
4.2	Conclusion partielle	130
5	Variante temporelle de profils aux multi-centres d'intérêts évolutifs	130
6	Conclusion	134
Contribution II		137
6	APPROCHES NEURONALES BASÉES SUR LES FOLKSONOMIES POUR LA RE- COMMENDATION	139
1	Introduction	139
2	Représentations des entrées des modèles neuronaux de filtrage collaboratif	140
2.1	Représentation en sac de mots dans l'espace des tags	140
2.2	Représentation dans un espace dense statique de tags	140
2.3	Représentation dans un espace dense contextualisé	142
2.4	Représentation du voisinage dans un espace dense	144
3	Modèles de filtrage collaboratif neuronal basés sur les tags	146
3.1	Filtrage collaboratif neuronal à base de tags exploitant une archi- tecture en MLP	146
3.2	Filtrage collaboratif neuronal à base de tags exploitant une archi- tecture en autoencoder	147
3.3	Filtrage collaboratif neuronal en graphe exploitant le voisinage de représentations continues de tags	148
4	Expérimentation	149
4.1	Métriques d'évaluation	150
4.2	Paramètres d'apprentissage	151
4.2.1	Hyper-paramètres	151
4.3	Extraction d'embeddings de tags à partir de modèles pré-entraînés	151
5	Évaluation et comparaison avec d'autres approches	153
6	Résultats et discussion	154
6.1	Impacts des architectures neuronales	154
6.2	Impacts de l'exploitation des embeddings contextels de tags	155
6.3	Impact de la fonction d'agrégation sur les modèles d'apprentissages	156
7	Conclusion	157
Conclusion générale		159
7	CONCLUSION GÉNÉRALE	161
1	Synthèse des contributions	161

2	Perspectives	163
	Annexe I : Apprentissage automatique & réseaux de neurones	165
I	ANNEXE I : APPRENTISSAGE AUTOMATIQUE	167
1	Introduction	167
1.1	Réseaux de neurones formels	167
1.2	Modèles de réseaux de neurones artificiels	168
1.3	Architecture de réseaux de neurones en Autoencoder	170
2	Algorithme d'apprentissage	170
2.1	Apprentissage supervisé de réseaux de neurones	170
2.2	Fonction de coût	171
2.2.1	Application en TALN : Représentations distribuées des mots	171
2.3	Apprentissage non supervisé : Clustering	174
2.4	Application : Structures communautaires dans les systèmes de tagging	175
2.4.1	Clustering pour la détection de communautés	176

LISTE DES FIGURES

Figure 1.1	À l'issu du processus d'indexation, la représentation de la requête ainsi que celle de l'ensemble des documents sont mises en correspondance pendant l'appariement et un score de pertinence est calculé permettant de classer les résultats selon le degré de similarité requête/document.	36
Figure 1.2	Figure illustrant l'évolution du processus d'indexation au fil des rééditions du Web. Les folksonomies se situent dans le contexte de l'indexation sociale.	37
Figure 1.3	Exemple illustratif représentant les documents, les tags et les utilisateurs dans une folksonomie.	39
Figure 1.4	Figure illustrant les domaines d'application exploitant les folksonomies pour différentes tâches de RI.	43
Figure 1.5	Capture extraite de la page principale du site web d'hébergement de vidéos et média social <i>YouTube</i> . Différents tags apparaissent dans l'interface de recherche.	46
Figure 2.1	Cette figure met en valeur les performances des différents <i>embeddings</i> (coût en mémoire/complexité) et les caractéristiques propres à l'exploitation de chaque type d' <i>embeddings</i>	54
Figure 2.2	Cette figure illustre l'évolution du nombre total de tags distincts N , dans <i>del.icio.us</i> . La courbe que l'on retrouve dans <i>Cattuto et al., [46]</i> représente N en fonction du temps τ . La loi de puissance qui décrit la fonction $N(\tau)$ indique un exposant $\sigma \approx 0.8$, le graphe confirme la croissance du nombre N de tags distincts en fonction du temps. Cela donne aussi un aperçu de la taille du vocabulaire engendré.	55
Figure 2.3	Nuage de tags extraits de la collection <i>Movielens-10M</i> , la dimension de chaque tag est proportionnelle à sa fréquence.	61
Figure 2.4	Extrait de la méthode de visualisation de <i>Halpin et al., [90]</i> à partir d'un graphe de corrélation entre tags et en fixant le terme " <i>complexity</i> " comme nœud central. Les tags sont extraits de la collection <i>del.icio.us</i>	62
Figure 2.5	Modèle de plongement lexical de tags et de contenu textuel d'un document dans un espace vectoriel commun pour la recommandation personnalisée de tags	67
Figure 3.1	Matrice $n \times m$ matrice des notes d'utilisateurs $r_{i,j}$	77
Figure 3.2	Comparaison entre utilisateurs selon la note qu'ils attribuent pour un même ensemble d'items.	78

Figure 3.3	Cette figure illustre les différentes dimensions associées aux tags (dimension temporelle, affective, communautaire). Elle souligne la manière dont ces dimensions peuvent être exploitées lors de processus de recherche, notamment, pour le classement des résultats de recherche et de recommandations personnalisées, calculées entre représentation d'utilisateurs et représentation de ressources.	79
Figure 3.4	Filtrage de contenus dans les systèmes de tagging	81
Figure 3.5	Filtrage collaboratif dans les systèmes de tagging	82
Figure 3.6	Framework de filtrage collaboratif neuronal simple comme explicité dans [95]	91
Figure 3.7	Architecture de <i>Autorec</i> illustré pour l'ensemble des notes des items représenté sous la forme d'un vecteur $r_{(i)}$ contenant des notes à prédire (nœuds en gris). Ces vecteurs sont projetés dans un espace latent de faible dimension puis sont reconstruits dans l'espace de sortie pour prédire les notes manquantes en vue de processus de recommandation avec $u \in U = \{1..m\}$ l'ensemble des utilisateurs et $i \in I = \{1..n\}$, l'ensemble des items	92
Figure 3.8	Framework de filtrage collaboratif des graphes neuronaux (NGCF) comme explicité dans [22]	93
Figure 3.9	Cette illustration explicite la génération de l'embedding du nœud u_1 . La représentation dense de ce nœud tient compte du voisinage auquel ce nœud est associé ($K=2$).	94
Figure 3.10	Architecture en MLP permettant l'apprentissage de de représentations de profils d'utilisateur et d'items et le calcul de la similarité entre ces profils	96
Figure 3.11	Illustration de l'approche de <i>Zuo et al.</i> , [254] utilisant le filtrage collaboratif neuronal en exploitant un autoencodeur empilés à l'aide de plusieurs couches cachées.	97
Figure 3.12	Illustration du modèle neuronal TRSDL [132], d'une part la modélisation de l'utilisateur à travers un réseau de neurones récurrents et d'une autre part la modélisation des items à travers un réseau de neurones profonds exploitant la moyenne des embeddings de tags extraits de <i>word2vec</i>	99
Figure 4.1	Page principale de la plate-forme de recommandation <i>MovieLens</i>	107
Figure 4.2	Distribution du nombre de <i>tokens</i> (mots) composant un tag dans <i>MovieLens-10M</i>	109
Figure 4.3	Distribution du nombre de <i>tokens</i> (mots) composant un tag dans <i>MovieLens-20M</i>	109
Figure 4.4	Distribution du nombre de <i>tokens</i> (mots) composant un tag dans <i>MovieLens-25M</i>	110
Figure 4.5	<i>MovieLens-10M</i> : Les tags sont représentés à droite {'good', 'animation', 'nudity', 'thriller', 'comic', 'bachelor', 'crime', 'hero', 'book', 'fatality', 'overrated', 'pixar', 'Disney', 'reality', 'oscar', 'war', 'Marvel'} et leurs correspondance avec des mots similaires plongé dans un espace à k-dimension	112
Figure 5.1	Illustration de la personomie P_u	121

Figure 5.2	Étapes générales de notre approche de personnalisation pour la génération de profils d'utilisateurs. D'abord les tags sont pondérées selon le degré de préférence de l'utilisateur par rapport au tag noté $v_{i,t}$, ensuite le graphe 1-mode est généré par projection du graphe TR_u sur l'ensemble des ressources de l'utilisateur R_u . Après avoir procédé à un clustering sur l'ensemble des noeuds de ce graphe, des signatures propres à chaque cluster sont extraites et comparées à l'ensemble des tags qui apparaît dans chaque ressource. À l'issue de cette comparaison les ressources sont cassées selon les préférences de l'utilisateur. Au final, le profil de l'utilisateur est un cluster unique ou un ensemble de clusters contenant une liste de ressources classées par ordre de préférence.	123
Figure 5.3	NDCG@10 à NDCG@100 pour les approches illustrées par l'exploitation de DPN_{Tu} et Yeung et al., [237] $\rho = 20\%$ $\rho = 40\%$. . .	129
Figure 5.4	Précisions à $p@10$ et $p@20$ des approches Yeung et al., [237], DPN_{Tu} et $DPNTmp_u$ lorsque $\rho = 40\%$	132
Figure 5.5	Comparaison des approches parmi différentes valeurs de la mesure $nDCG$ pour une signature de cluster idéale fixée à $\rho = 40\%$.	133
Figure 6.1	Construction des <i>embeddings</i> de tags $\vec{T}_{(u \times i)}$ à partir du modèle de langue pré-entraîné Word2vec	141
Figure 6.2	Construction des <i>embeddings</i> contextualisés de tags $\vec{T}_{(u \times i)}$ à partir de $BERT_{base}$ (12 couches cachées avec la dimension de chaque couche cachée à 768 et 12 têtes d'attentions). La sortie de chacun des 12 encodeurs peut être considérée comme une représentation vectorielle de la séquence passée en entrée. La pertinence de cette représentation est assurée par le mécanisme d'attention mis en œuvre par les encodeurs.	144
Figure 6.3	Agrégation des représentations vectorielles de noeuds au voisinage $k = 2$ dans l'espace dense \mathbb{E}_{w2v} et \mathbb{E}_{Bert}	145
Figure 6.4	Filtrage collaboratif neuronal basé sur une architecture en MLP extrait de l'article Boudiba & Dkaki [27]	147
Figure 6.5	Approche de filtrage collaboratif basée sur un modèle neuronal avec une architecture en autoencoder extrait de l'article Boudiba & Dkaki [27]	147
Figure 6.6	Approche basée sur le filtrage collaboratif de graphes neuronaux ayant comme caractéristiques des <i>embeddings</i> de tags étendus à un certain voisinage.	148
Figure 6.7	En haut de cette figure, nous avons présenté la distribution des erreurs de chaque modèle neuronal. En bas, nous avons donné les scores de validation des modèles après 20 epochs.	155
Figure I.1	Architecture MLP pour de la régression	169
Figure I.2	Architectures CBOW et Skip-gram de <i>Word2vec</i>	172
Figure I.3	Illustration des voisinages de noeuds dans un graphe tel que défini dans [211]	174
Figure I.4	Graphe d'utilisateurs de tags <i>Movielens-10M</i> Number of nodes : 11610 Number of edges : 55484 Average degree : 9.5580	176

Figure I.5 Algorithme de Détection de communautés (*Newman*) via clustering sur le graphe d'utilisateurs d'après le nombre de leurs tags en communs lors d'activités d'annotation sur *Movielens10M*
Number of nodes : 11610 Number of edges : 55484 Average degree : 9.5580 Nombre de communautés détectées : 80 177

LISTE DES TABLES

Table 1.1	Tableau représentant les schémas de pondération tf-idf absolus et normalisés. N est le nombre de documents dans un corpus documentaire et df_i est le nombre de documents dans lequel le terme t_i apparaît. $tf(t_i, d_j)$ est la fréquence d'occurrence du i -ème terme dans le document d_j . $ D $ est la longueur du document D . $avgdl$ correspond à la longueur moyenne des documents, k_1 et b sont des paramètres qui prennent respectivement les valeurs 1.2 et 0.75.	38
Table 2.1	Modèles de représentation de tags classés par tâche et par travaux de recherche	57
Table 2.2	Ce tableau synthétise les différents modèles de pondération exploités dans des approches classiques de personnalisation et de recommandation basées sur les folksonomies	58
Table 3.1	Tableau de Notations	77
Table 4.1	Statistical details of the 10M, 20M and 25M collections from MovieLens	113
Table 5.1	Rappel des différents schémas de pondération associés aux tags, dans ce tableau un document peut aussi bien faire référence à une ressource qu'à un item.	118
Table 5.2	Limitations des schémas de pondération tf^*iuf et iuf pour un utilisateur u_i	119
Table 5.3	Illustration de la limitation associée aux poids des arêtes dans l'approche de <i>Yeung et al.</i> , [237]	120
Table 5.4	Comparaison de notre approche avec la baseline <i>Yeung et al.</i> , [237]	126
Table 5.5	Ce tableau compare les précisions pour différentes valeurs des signatures ρ parmi toutes les ressources d'un utilisateur, parmi l'ensemble des ressource de la collection. Comparaison avec <i>Cai et al.</i> , [33].	127
Table 5.6	NDCG@10 et NDCG@20 pour différentes valeurs des signatures ρ . Comparaison avec [237]	129
Table 5.7	Synthèse des valeurs NDC@10, comparaison avec [237]. Les valeurs sont calculées parmi toutes les ressources d'un utilisateur, parmi l'ensemble des ressource de la collection pour différentes valeurs de ρ	130
Table 5.8	Précisions pour différentes valeurs des signatures ρ . Comparaison <i>Yeung et al.</i> , [237] et DPN_{Tu}	132
Table 5.9	Précisions globales pour différentes valeurs des signatures ρ . Les valeurs sont calculées parmi toutes les ressources d'un utilisateur, parmi l'ensemble des ressource de la collection. Comparaison <i>Yeung et al.</i> , [237].	132

Table 5.10	Le tableau présente les scores de NDCG@10 à NDCG@100 en comparaison avec les approches <i>Yeung et al.</i> , [237], DPN_{Tu} et DPN_{Tmp-Tu} [29].	133
Table 5.11	Dans ce tableau, les résultats avec (★) sont statistiquement significatifs comparés aux baselines.	134
Table 6.1	Notations détaillées	141
Table 6.2	Tableau récapitulant les scores RMSE, MAE et ndcg@10 des approches présentées	156
Table I.1	Tableau récapitulatif de fonctions d'activation populaires	168
Table I.2	Quelques caractéristiques des modes d'apprentissage supervisé/non supervisé	171
Table I.3	Fonctions de coût utilisées pour la régression	171

ACRONYMES

- ACP** Analyse en composantes principales
- BM₂₅** Okapi BM₂₅ mesure de pondération impliquant le classement en fonction de la fréquence des termes apparaissant dans chaque document indépendamment des relations pouvant exister en termes.
- BERT** Bidirectional Encoder Representations from Transformers
- CF** Collaborative Filtering ou Filtrage collaboratif
- DCG** Discounted Cumulative Gain
- DNN** Deep Neural Network
- ELMo** Embeddings from Language Models
- GloVe** Global Vectors for Words Representations
- IDF** Inverse Document Frequency
- LSA** Latent Semantic Analysis ou Analyse Sémantique Latente
- LSTM** Long Short-Term Memory
- LDA** Latent Dirichlet allocation ou Allocation de Dirichlet latente
- MAE** Erreur absolue moyenne qui correspond à la moyenne arithmétique des valeurs absolues des écarts, en anglais Mean Average Error.
- MLP** Multi-layer perceptron ou perceptron multi-couches
- NCF** neural collaborative filtering
- MLP** Perceptron Multi-Couches ou Multi-Layer perceptron
- NTF** Fréquence de termes normalisés ou Normalized term frequency
- NDCG** Normalized Discounted Cumulative Gain
- ODP** Open Directory Project
- PRP** Probability Ranking Principle
- RNN** Réseau de neurones récurrents ou Recurrent Neural Networks
- RMSE** Racine de l'erreur quadratique moyenne ou Root-Mean-Square Error .
- RI** Recherche d'information
- RSV** Retrieval Status Value
- SRI** Système de Recherche d'Information
- SVD** Singular Value Decomposition ou décomposition en valeurs singulières
- TALN** Traitement Automatique du Langage Naturel
- TF** Term Frequency
- t-SNE** t-distributed Stochastic Neighbor Embedding
- VSM** Vectorial space model
- Word2Vec** Ensemble de modèles utilisé pour le plongement lexical de mots, le *word embedding*.

CONTEXTE GÉNÉRAL

«In Social Media the “squeaky wheel” gets the oil. You have to put yourself out there, to find people who will relate or even debate with you, depending on what you are looking for.»

— Jessica Northey

1 Annotations sociales

Dans les années 2000, le développement de plateformes sociales, telles que *Twitter*¹, marque un tournant majeur dans les pratiques du Web. Une multitude d'utilisateurs génèrent des contenus personnalisés qui sont relayés, commentés, partagés et annotés. Face à ce foisonnement d'informations, le traitement des contenus a pris une nouvelle dimension, dans laquelle le tag joue un rôle de premier ordre. Il permet de décrire, de classer, et de catégoriser ces contenus multimédias (textes, images, vidéos). Ainsi, depuis plus d'une décennie, l'étude des annotations sociales est devenue un sujet prépondérant de recherche en RI.

C'est avec le succès de certains sites Web comme *Del.icio.us*² ou *Flicker*³ que le concept d'annotation sociale ou tag s'est popularisé. Depuis l'intégration du premier système de *tagging* collaboratif sur le site *Delicious*⁴ en 2003, celui-ci a connu d'importantes évolutions sur un plan technique et théorique. Progressivement, d'autres sites de partage de contenus multimédias, tels que *Youtube*⁵ et *Dailymotion*⁶ en 2005, ont développé ce procédé [55]. L'exploitation des annotations sociales ou tags par ces deux sites a permis d'élargir les méthodes de navigation et de partage classique, étendant le concept d'indexation subjective à plusieurs domaines. Dans le domaine de la gestion des productions scientifiques, la plateforme de bookmarking (système de signets social) *Bibsonomy*⁷, intègre en 2006, les fonctionnalités des systèmes de tagging pour identifier les thèmes des publications scientifiques. Les chercheurs associent un ou plusieurs tags à chaque publication. Dans le domaine des réseaux sociaux et des microblogging, le site Web social *Pinterest*⁸ en 2007 et la plate-forme de microblogage *Tumblr*⁹ en 2010, ont exploité les annotations d'utilisateurs pour améliorer la description d'images pour faciliter la navigation et la recherche. Dans le domaine de la recommandation, les plateformes comme *MovieLens*¹⁰ exploitent les annotations sociales d'utilisateurs ayant décrit des ressources mises en ligne avec des tags, pour inférer des recommandations personnalisées. *Last.fm*¹¹ offre la possibilité aux utilisateurs d'associer un mot-clé à un artiste, un album ou une chanson et génère des recommandations musicales pour chaque utilisateur. *Netflix*¹² utilise un système de classification de contenus basé sur les descriptions

1. <https://twitter.com/>

2. <https://Del.icio.us/>

3. <https://Flicker.com/>

4. <https://del.icio.us/>

5. <https://www.youtube.com>

6. <http://www.dailymotion.com>

7. <https://www.bibsonomy.org/>

8. <https://www.pinterest.com/>

9. <https://www.Tumblr.com/>

10. <https://movielens.org/>

11. <https://www.last.fm/>

12. <https://www.netflix.com/fr/>

de leurs propres employés.

Nos travaux portent sur les annotations sociales, ils se situent précisément dans le domaine de la recommandation personnalisée de contenus. Ils s'intéressent à la mise en place de modèles permettant d'intégrer des tags dans des processus de recommandation. Les annotations permettent de mettre en valeur de riches univers sémantiques, de nouvelles formes d'accès, et la mise en valeur des ressources en ligne [55, 88]. L'évolution de tels systèmes est pérennisée grâce à la participation de groupes d'utilisateurs, de communautés et d'utilisateurs individuels, qui organisent leurs besoins en information. Comme les utilisateurs annotent différents types de ressources, cela implique naturellement qu'ils possèdent des centres d'intérêt multiples. Nous décrivons dans ce qui suit les deux axes de recherche que nous avons abordés, la notion de centres d'intérêt multiples et les modèles neuronaux pour la recommandation.

2 Multiples centres d'intérêts des usagers des folksonomies

Les systèmes d'annotation collaborative sur le Web sont généralement centrés sur le partage de ressources en ligne. Ces systèmes permettent aux utilisateurs d'annoter des ressources en employant des tags et offrent un moyen puissant d'organisation, de recherche et de partage de différents types de ressources sociales.

Les tags générés par des utilisateurs décrivent leurs centres d'intérêt, ils fournissent de riches informations pouvant être utilisées pour représenter les profils des utilisateurs ou de ressources. Dans ce sens, plusieurs travaux [162, 38, 41] ont exploité ces profils dans un contexte de recherche d'informations personnalisées. Dans ces travaux, le profil de l'utilisateur (ou de la ressource) est souvent représenté comme un vecteur de tags pondérés. Chaque tag est pondéré en utilisant un modèle de pondération tel que : tf (nombre de fois qu'un tag donné est assigné par un utilisateur à des ressources), ou des variantes de ce modèle, $tf-idf$, $BM25$.

Ces approches supposent que les tags les plus fréquents sont les plus à même de décrire un profil d'utilisateur ou de ressource. Cependant, l'utilisation de fréquences de termes absolues fera augmenter anormalement le poids d'utilisateurs plus actifs [34]. Pour pallier ce problème, plusieurs travaux se sont intéressés à la prise en compte de l'activité de l'utilisateur pour inférer les intérêts de celui-ci, en proposant d'ajuster ou de normaliser les poids des tags en fonction des activités de l'utilisateur. De plus, la plupart de ces travaux supposent que le profil de l'utilisateur est représenté par un seul ensemble de tags populaires (fréquents). Cependant, dans les systèmes d'annotation collaborative, les utilisateurs sont libres d'utiliser tout type de vocabulaire (tags) pour annoter différents types de ressources. Cela aboutit à l'utilisation, d'un large éventail de tags qui est en relation avec une multitude de ressources liées elles aussi à un large éventail de thématiques (topics) [237].

Dès lors, l'utilisation d'un seul ensemble de tags (ou vecteur pour représenter un profil) ne permet pas de prendre en compte la diversité des centres d'intérêts de l'utilisateur. Pour faire face à cette limitation, certains travaux [166, 200, 78, 77] proposent de

rassembler les tags en clusters qui reflètent les préférences de l'utilisateur pour un topic d'intérêt particulier. Les méthodes de partitionnement (clustering) sont souvent basées sur la similarité entre vecteurs de tags. Plus précisément, ces méthodes [78, 14, 199] calculent des similarités entre deux tags représentés sous la forme de vecteurs pondérés d'utilisateurs ou de ressources utilisant les modèles de pondération basés sur tf , tf^*idf ou BM_{25} .

Cependant, bien que ce type d'approche offre une perception plus large du profil de l'utilisateur, dans la mesure où, les centres d'intérêts sont plus diversifiés et mieux adaptés à représenter le large éventail des intérêts de l'utilisateur, elle ne considère pas le contexte lié à l'activité de ce dernier, ni l'évolution des intérêts de l'utilisateur dans le temps [240, 139]. De plus, ces approches ne fournissent pas les moyens d'identifier les tendances en matière d'intérêts et de communautés en examinant à la fois l'activité des utilisateurs sur les différents tags ainsi que l'activité des tags à travers les différents utilisateurs.

De ce fait, la dimension temporelle de l'annotation a été étudiée dans des travaux qui intègrent des données temporelles lors de la création de profils. Ces données sont quantifiées à l'aide d'une stratégie de pondération visant à attribuer un poids à chaque ressource en se basant sur les informations que fournissent les tags comme la popularité, à partir de données d'horodatage (timestamp) ou d'une combinaison des deux. Cependant, les travaux [109, 20, 2] exploitant ce type d'information ne nous renseignent pas sur le moment à partir duquel un contenu devient obsolète.

3 Modèles de représentation neuronaux pour la recommandation

Dans les systèmes collaboratifs, les ressources sont souvent associées à des items auxquels des utilisateurs assignent différents tags. Les principales approches pour la recommandation dans ces systèmes se basent sur le filtrage du contenu, en exploitant le contenu des ressources ou les tags qui leur sont associés, ou sur le filtrage collaboratif qui exploite les similarités entre vecteurs d'utilisateurs et vecteurs de ressources. Parmi les techniques de filtrage collaboratif [242, 188], la factorisation matricielle [96, 118] est l'une des approches les plus populaires, cette technique permet la projection d'utilisateurs et d'items dans un espace latent partagé. Les travaux exploitant ces techniques [68, 95, 50, 68] utilisent un vecteur de caractéristiques latent pour représenter un utilisateur ou un item et modélisent l'interaction utilisateur/item par le produit scalaire de ces deux vecteurs. Cependant, le produit scalaire est linéaire, et peut être insuffisant pour capturer la nature complexe des interactions entre utilisateurs et items.

Récemment, les modèles neuronaux [254, 82, 105, 127] ont démontré leurs performances dans les tâches d'apprentissage supervisées et non supervisée. Leurs capacités à intégrer des sources multiples d'informations et à tirer profit d'importantes masses de données peuvent être appliquées aux systèmes de recommandation. Ces modèles visent à apprendre des représentations jointes d'utilisateurs et d'items, grâce à des transformations linéaires et des fonctions non-linéaires qui permettent la prise en compte d'in-

teractions plus complexes entre utilisateurs et items. Les travaux décrivant ces modèles [50, 196, 245] projettent simultanément des représentations d'utilisateurs et d'items dans un espace vectoriel commun. En outre, ils exploitent les informations sémantiques issues de plongements lexicaux de tags ou « embeddings » pour inférer des représentations plus significatives d'utilisateurs et d'items. Le but d'une telle manœuvre est de parfaire l'apprentissage de ces représentations pour la recommandation grâce notamment à l'utilisation de méthodes d'apprentissage automatique.

Cependant, peu de ces travaux parviennent à intégrer à la fois les tags et des représentations jointes d'utilisateurs et d'items au sein de modèles neuronaux pour la recommandation. Du fait de l'ambiguïté de certains tags, mais aussi, de la tendance du vocabulaire engendré par les tags à manquer de contextes, en particuliers quand le tag se compose d'un mot unique.

4 Problématiques et questions de recherche

Nous avons mis en évidence certaines limites relatives à l'utilisation des tags en RI, que nous tenterons de lever en répondant, tout au long de ce manuscrit, aux questions de recherche suivantes :

- Comment mieux qualifier les intérêts ou l'importance des tags au regard de l'activité des personnes qui les ont produits et du moment où ils ont été produits ?
- Comment prendre en compte efficacement les multiples centres d'intérêts des utilisateurs ? A travers le temps ?

Plus précisément, proposer des recommandations de qualité requiert une compréhension profonde des caractéristiques des items et des préférences de l'utilisateur. Cela nécessite de capturer des associations non-triviales et complexes entre utilisateurs et items. Dans ce sens, certains travaux modélisent ces relations à travers les triplets utilisateur, tags, item représentés sous forme de graphe et abordent la recommandation comme une tâche relative à la prédiction de liens entre utilisateurs et items. Cependant, ces travaux ne tiennent pas compte des informations textuelles (tags) associées aux entités (utilisateurs et items), et ne permettent pas d'étendre ces informations à travers les voisinages auxquels sont associés de telles représentations.

- Quelle architecture neuronale efficace permettrait de représenter conjointement les triplets utilisateur, tags, item dans un espace de représentation adapté à la recommandation ?
- Comment agréger des représentations de tags associées à des utilisateurs et à des items en tenant compte du voisinage induit par les représentations sous forme de graphe ?

5 Contributions

Notre première contribution s’inscrit dans le cadre de la découverte des multiples intérêts que porte l’utilisateur à des items dans les plateformes d’annotations collaboratives.

- Nous avons proposé une approche visant à générer des profils folksonomiques. Ces profils sont des clusters correspondant aux différents centres d’intérêts de l’utilisateur qui tiennent compte aussi de l’évolution de ses préférences.
- Nous avons également proposé un schéma de pondération qui prend en compte les aspects qualitatifs et temporels d’une folksonomie. Le modèle de pondération utilisé a été défini grâce à une méthode d’apprentissage automatique non supervisée : Le clustering. Cette méthode nous a permis d’extraire des ressources classées selon l’ordre de préférence de l’utilisateur et d’apprécier son évolution dans le temps.

Notre deuxième contribution porte sur l’exploitation des représentations continues d’utilisateurs et d’items. Ces représentations ont été extraites à partir de modèles de langue pré-entraînés avant d’être intégrés à différents modèles neuronaux de filtrage collaboratif exploitant différentes architectures. Parmi ces modèles, nous nous sommes particulièrement intéressés au moyen de prendre en compte les relations de voisinage lors de la construction de représentations d’utilisateurs et d’items. Nous soutenons que l’intégration de telles représentations dans un modèle collaboratif permettrait la prise en compte de la préférence de l’utilisateur pour un item en fonction de tags d’utilisateurs ou d’items voisins. En outre, nous avons proposé et testé plusieurs approches pour la prise en compte de différents espaces de représentation incluant les utilisateurs, les items ainsi que les représentations denses d’utilisateurs et d’items.

- Nous avons étudié l’apprentissage de représentations issues de triplets utilisateur, tags, item pour la recommandation à travers différents modèles neuronaux.
- Nous avons mis en œuvre une fonction d’agrégation qui a été appliquée au voisinage de représentations denses d’utilisateurs et d’items à travers un modèle neuronal étendu et basé sur les graphes.
- Nous avons testé l’efficacité des modèles générés en les comparant aux modèles de la littérature.

6 Organisation du manuscrit

Notre manuscrit s’organise en sept chapitres. Une annexe recensant certains concepts inhérents aux modèles d’apprentissage automatique clôt ce travail. Les chapitres se déclinent comme suit :

— **Chapitre I : Folksonomies & RI**

Ce chapitre décrit les concepts fondamentaux en recherche d'informations et introduit la notion d'indexation sociale. Il fait état des nombreux domaines d'application des folksonomies.

— **Chapitre II : Modèles de représentations de folksonomies**

Ce chapitre est dédié aux principaux modèles de représentations de tags issus des folksonomies. Il propose une vue d'ensemble de ces représentations en explicitant leurs caractéristiques ainsi que les tâches sur lesquelles elles se sont le plus illustrées à travers la littérature.

— **Chapitre III : Modèles de folksonomies pour la recommandation**

Ce chapitre aborde les modèles de folksonomies pour la recommandation. Il contient une synthèse non exhaustive de l'état de l'art à ce sujet et discute de manière détaillée certaines approches de la littérature.

— **Chapitre IV : Étude/Analyse de la collection**

Ce chapitre décrit les statistiques détaillées de la collection *MovieLens*, nous ayant permis de mettre en œuvre nos contributions.

— **Chapitre V : Apports des tags pour la construction de profils folksonomiques**

Ce chapitre détaille notre première contribution relative à la construction de profils d'utilisateurs aux multiples centres d'intérêt évolutif. Il présente en outre les différents résultats obtenus à l'issue de cette approche.

— **Chapitre VI : Apports des folksonomies dans les modèles de filtrage collaboratif neuronaux**

Ce chapitre explicite notre deuxième contribution relative à la construction de modèles neuronaux pour la recommandation tenant compte explicitement du vocabulaire des tags via des représentations denses d'utilisateurs et d'items. Il présente les résultats de nos évaluations et leur comparaison avec d'autres approches de la littérature.

— **Chapitre VII : Conclusion générale**

Ce dernier chapitre synthétise nos contributions en apportant des réponses globales aux problématiques abordées dans ce manuscrit.

FOLKSONOMIES EN RI

«We are all now connected by the Internet, like neurons in a giant brain.»

— Stephen Hawking

Chapitre 1

FOLKSONOMIES EN RI

LA recherche d'informations (RI) est une discipline issue des sciences de l'information et de l'informatique. La RI porte sur des modèles de recherche permettant de fournir à l'utilisateur une liste triée de documents pertinents répondant à sa requête. La majorité des systèmes de RI représente les documents et les requêtes par une liste de mots clés pondérés. C'est ce qu'on appelle une représentation en sac de mots ou *Bag of Word* (BoW). La sélection de documents pertinents revient à calculer un degré de similarité entre ces représentations. Ce degré de similarité est au cœur des modèles de RI. On y distingue différentes classes. Les principaux modèles de recherche d'information, booléen [184], vectoriel [187] et probabiliste [170, 172] se basent dans leur majorité sur une mise en correspondance lexicographique [6] qui correspond à une simple comparaison entre les mots clés de la requête et ceux des documents. Cette mise en correspondance exacte entre termes posent des problèmes de disparité de vocabulaire (défaut d'appariement) communément nommé en anglais *vocabulary mismatch*.

Dés lors, les modèles classiques de RI ont étendu la notion de similarité en y incluant des représentations capables de mieux prendre en compte les relations sémantiques entre les termes. Dans ce contexte, différents travaux de recherche notamment ceux issus du traitement automatique des langues (TAL) ont suggéré l'exploitation de matrices de co-occurrences pour représenter le vocabulaire dans un espace restreint formé de vecteurs sémantiques denses. L'approche la plus illustre est LSA «Analyse Sémantique Latente» (*Latent Semantic Analysis*), elle se base sur la Décomposition en Valeurs Singulières (SVD) (*Singular Value Decomposition*) pour la réduction de dimensionnalités. D'autres méthodes issues de la factorisation matricielle telles que Latent Dirichlet Allocation [227] (LDA), ou encore *Bi-Lingual Topic Model* (BLTM) ont été proposées et appliquées avec succès pour la mise en correspondance des représentations sémantiques [25, 101, 73, 79].

Récemment, des modèles de représentations neuronales permettent de décrire le sens d'un mot par les contextes dans lesquels ce mot apparaît le plus fréquemment. Ces modèles découlent de représentations de mots dans des espaces multidimensionnels. Le plus populaire est le modèle neuronal *Word2Vec* proposé par Mikolov et al. [152], d'autres modèles de représentations sémantiques tel que GloVe [167] ont démontré leurs capacités à capturer des patterns linguistiques en les identifiant comme relations linéaires entre vecteurs de mots [164]. Depuis 2019, des modèles de représentations contextualisés basés sur les *transformers* tel que BERT sont de plus en plus utilisés en TALN en général et en RI particulièrement.

À l'issue de ce chapitre, nous présentons les notions de base de la RI en détaillant certaines caractéristiques propres aux SRI et les mesures utilisées pour les évaluer. Nous expliciterons brièvement le principe de l'indexation social avant de souligner les avantages de l'association de mots-clés ou tags à des documents. Nous présenterons également diverses tâches de RI associées à l'utilisation de folksonomies d'utilisateurs.

1 Concepts de base en recherche d'information

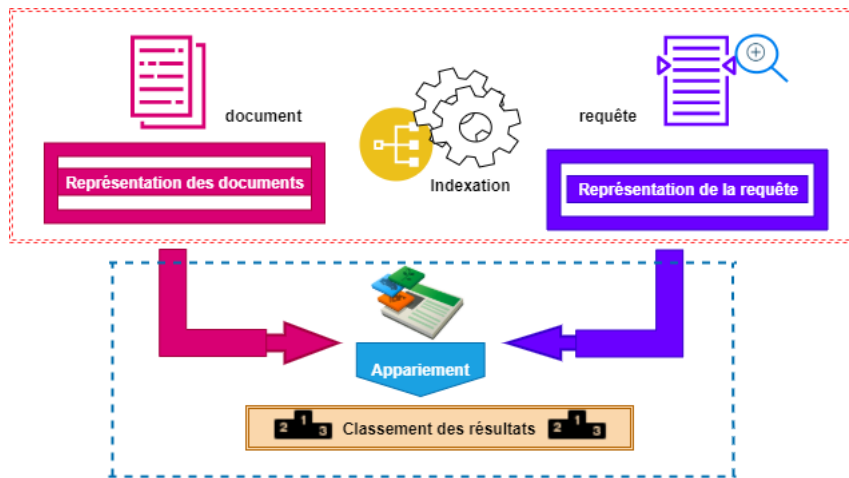


FIGURE 1.1 – À l’issue du processus d’indexation, la représentation de la requête ainsi que celle de l’ensemble des documents sont mises en correspondance pendant l’appariement et un score de pertinence est calculé permettant de classer les résultats selon le degré de similarité requête/document.

Différents processus ont été mis en œuvre pour garantir un accès efficace à l’information. Leur objectif premier est de sélectionner des ressources pertinentes associées à la requête d’un utilisateur. Cette requête peut prendre différentes formes, tout comme le type de ressources retournées. En effet, les utilisateurs peuvent exprimer leur besoin en information textuellement, il est aussi possible que la requête soit une image, ou qu’elle soit associée à une commande à reconnaissance vocale, ou dans certains cas, elle peut être implicite [153]. Les ressources retournées sont tout aussi diverses et variées comme le contenu textuel des documents, pages Web ou le contenu multimédia des vidéos ou des images. Ces résultats sont souvent classés par ordre de pertinence. Ce classement peut prendre en compte, autre le contenu du document, l’historique des recherches antérieures de l’utilisateur, sa localisation, les variations temporelles de l’information à rechercher ou encore les différents contextes en relation avec la requête. Sur le plan pratique, les documents et les requêtes sont transformés en une représentation interne. Ces représentations sont ensuite appariées entre elles pour calculer un score ou un degré de similarité (de pertinence ou d’appariement). Ces documents sont ensuite classés par ordre décroissant de ces scores. La figure 1.1 illustre le processus de recherche classique en RI. Afin d’améliorer les résultats de la recherche, le système peut être doté d’un mécanisme d’amélioration et de raffinement de la requête par reformulation [15].

1.1 Indexation

Le processus d’indexation a évolué au fil des rééditions du Web comme l’illustre la figure 1.2. Il consiste dans un premier temps à extraire pour chaque document les termes caractérisant son contenu informationnel, puis à les exploiter de sorte à fournir un accès plus rapide aux documents. L’objectif étant de permettre la sélection des documents

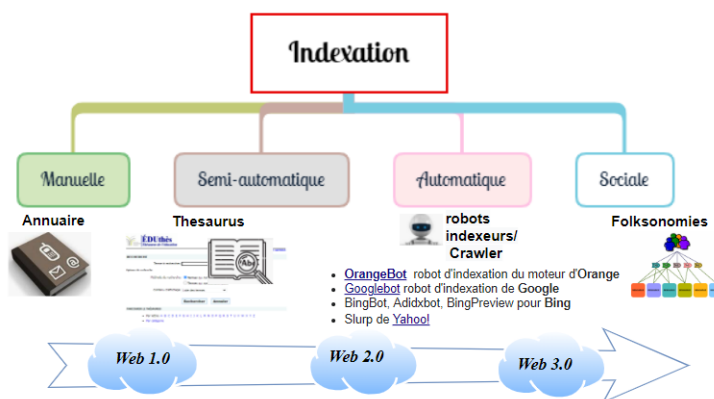


FIGURE 1.2 – Figure illustrant l'évolution du processus d'indexation au fil des rééditions du Web. Les folksonomies se situent dans le contexte de l'indexation sociale.

contenant les termes (mots-clés) de la requête fournie au préalable. L'ensemble de tous les termes d'indexation constitue le lexique de l'indexation ou vocabulaire. L'indexation peut être : manuelle[190], semi-automatique[142, 107, 10] ou automatique[183, 144, 137]. L'indexation peut être aussi un processus collaboratif social dans lequel les utilisateurs assignent des mots-clés à des ressources. L'indexation sociale [102, 78, 103] est apparue avec le Web 2.0 et a facilité l'indexation de contenus non-textuels. Dans ce qui suit, nous donnons brièvement les bases de l'indexation classique avant d'introduire le principe de l'indexation sociale qui est à l'origine des structures de données en folksonomies.

1.2 Indexation classique

De manière générale, l'indexation est basée sur l'analyse des documents en vue de l'extraction des termes (mots-clés simples ou composés) représentatifs de leurs contenus informationnels. Elle comprend une série de traitements automatisés appliqués aux documents, mais aussi aux requêtes et repose sur les étapes suivantes : l'extraction des mots, l'élimination des mots vides[171, 173], la normalisation [75, 74] et la pondération[56]. À l'issue de l'indexation, chaque document sera représenté par une liste de mots clés avec leur fréquence (un sac de mots). Les documents et les termes sont stockés dans une structure de données commune nommée "index inversé" [186, 117, 250] qui permet la sélection rapide des documents comportant un terme donné. Les termes des documents n'ayant pas tous la même importance, on associe à chaque terme un poids combinant la fréquence d'occurrence dans un document, notée tf ainsi que sa fréquence inverse notée idf . L'utilisation de la fréquence pour mesurer l'importance d'un terme est fondée sur les corollaires linguistiques de Zipf [249] et Luhn [137], qui établissent une relation entre la fréquence des termes et leur pouvoir discriminant, c'est-à-dire leur importance dans la participation au sens relatif d'un texte [212]. Le produit de ces deux fréquences *i.e.* $tf \times idf$ [203], utilisées avec des formules diverses, sert à indiquer l'importance d'un terme dans un document. De nombreuses déclinaisons ont été formulées pour atténuer des problématiques liées à la longueur des documents, quelques-unes d'entre elles incluant des normalisations logarithmiques sont décrites dans le tableau 1.1, où les schémas de pondération classiques et normalisés y sont dé-

crits. Il est aussi possible de définir de manière probabiliste la mesure idf , dans ce cas, idf peut être approximer dans le cadre d'un modèle d'indépendance binaire ou BIR pour *Binary Independence Model* avec $\frac{\log(N-df_i)}{df_i} \simeq \log \frac{N}{df_i} = idf$, cette variante est exploitée dans le cadre du modèle BM_{25} [179].

Schéma de pondération $tf-idf$			
Fréquences	Fréquence de terme	Fréquence inverse	idf
Fréquence absolue	$tf(t_i, d_j)$	fréquence inverse	$\log(\frac{N}{df_i})$
Normalisation logarithmique	$1 + \log(tf(t_i, d_j))$	fréquence inverse lissé	$\log(1 + \frac{N}{df_i})$
Double Normalisation 0.5	$0.5 + 0.5 \frac{tf(t_i, d_j)}{\max(tf(t_i, d_j))}$	fréquence inverse max	$\log(1 + \frac{\max(df_i)}{df_i})$
BM_{25}	$\frac{(k_1+1) \times tf(t_i, d_j)}{k_1 \times (1-b + b \times \frac{ D }{avgdl}) + tf(t_i, d_j)}$	fréquence inverse probabiliste	$\log(\frac{N-df_i}{df_i})$

TABLE 1.1 – Tableau représentant les schémas de pondération $tf-idf$ absolus et normalisés. N est le nombre de documents dans un corpus documentaire et df_i est le nombre de documents dans lequel le terme t_i apparaît. $tf(t_i, d_j)$ est la fréquence d'occurrence du i -ème terme dans le document d_j . $|D|$ est la longueur du document D . $avgdl$ correspond à la longueur moyenne des documents, k_1 et b sont des paramètres qui prennent respectivement les valeurs 1.2 et 0.75.

1.3 Indexation sociale

L'indexation sociale repose sur une pratique collaborative décrite à travers des systèmes impliquant l'exploitation d'une relation ternaire ($u \times d \times t$) entre l'utilisateur u , le document d et l'index t , qu'on appelle aussi tag et qui est représenté textuellement, par un mot-clé [194, 197, 60, 43]. Elle se rapporte aussi à une méthode d'annotation de ressources en ligne sur le Web en utilisant des termes issus d'un vocabulaire non contrôlé [72, 112]. Les utilisateurs y jouent le rôle d'indexeurs et la relation ternaire est pérennisée grâce au produit de la relation impliquant les ressources décrites par l'utilisateur et les tags (mots-clés) utilisés pour la description de ces ressources [143, 72]. Dans une telle configuration nous pouvons affirmer qu'utilisateurs et ressources sont inter-connectés dans un environnement collaboratif grâce à l'ensemble des index textuels (tags) assignés par ces utilisateurs à ces ressources. Ainsi, les ressources annotées avec les mêmes tags donnent lieu à d'autres types de relations comme l'illustre la figure 1.3.

Nous pouvons intuitivement affirmer que les documents *Doc 3* et *Doc 4* sont liés par une relation car ils partagent un tag en l'occurrence *roman*. Cette relation peut être assimilée à une thématique. Dans la figure 1.3, ces documents sont aussi reliés à des utilisateurs, on remarque qu'ils sont liés les uns aux autres lorsqu'ils utilisent les mêmes tags ou lorsqu'ils annotent les mêmes documents. Les utilisateurs *user1* et *user2* annotent le même document *doc1* en utilisant le même tag *wiki*. Cela peut signifier que ces utilisateurs sont aussi liés par une relation. Ainsi, l'étendue de la communauté des utilisateurs ou des ressources peut être illustrée quantitativement avec des scores de similarité entre utilisateurs ou entre ressources, par exemple selon l'ensemble des tags communs utilisés par une communauté d'utilisateurs ou le nombre de ressources ayant été annotées par le même ensemble de tags pour une communauté de ressources. Les

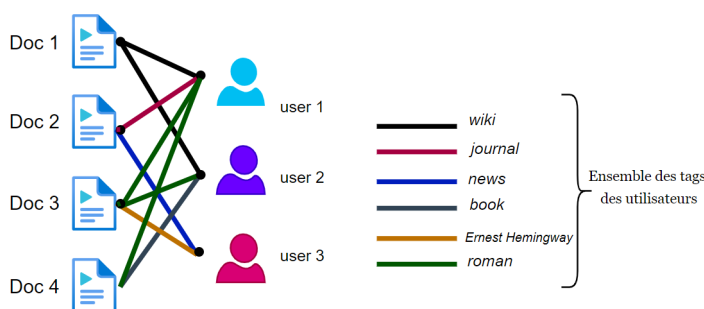


FIGURE 1.3 – Exemple illustratif représentant les documents, les tags et les utilisateurs dans une folksonomie.

ressources peuvent être indexées avec un ou plusieurs tags. Ces tags sont à leur tour utilisés comme index avec une fréquence d'attribution qui peut varier selon l'entité que l'on considère. La fréquence d'un tag peut simplement se baser sur la fréquence de l'utilisation de ce tag pour décrire une ressource ou sur la fréquence de l'utilisation de ce tag par un utilisateur donné. Elle peut être en relation avec la fréquence avec laquelle cet utilisateur annote indépendamment des ressources. Elle peut aussi se rapporter à une ressource spécifique en considérant la fréquence avec laquelle cette ressource a été annotée indépendamment des utilisateurs. *Doc 3*, par exemple, est annoté avec deux tags différents, cependant le tag "roman" a été utilisé deux fois et le tag "Ernest-Hemingway" une fois. De cette manière, selon la fréquence à laquelle les tags sont attachés à une ressource, nous pouvons obtenir des distributions de tags spécifiques à la ressource. De manière analogue, il est possible de déterminer des distributions de tags spécifiques à l'utilisateur [104]. De nombreux travaux [41, 219] ont été axés sur l'utilisation de tags comme moyen d'indexation des ressources sur le Web pour améliorer la qualité de la recherche en personnalisant l'accès à l'information. Dès lors l'utilité de cette méthode d'indexation personnelle s'est étendue à d'autres tâches de recherche d'information, notamment dans le domaine de la recommandation qui est notre objet d'étude principal dans cette thèse [49, 233].

1.4 Appariement

L'appariement est un processus qui se déclenche à chaque nouvelle requête de l'utilisateur [212]. Le cadre théorique de ce processus fait état d'un modèle de représentation des requêtes et des documents ainsi que d'une fonction d'appariement (ou fonction de correspondance) :

- Le modèle de représentation décrit la manière dont le besoin de l'utilisateur (exprimé par une requête) et les documents sont représentés formellement. Il dépend de la fonction d'appariement utilisée.
- La fonction d'appariement détermine la manière dont la pertinence d'un document par rapport à une requête est calculée. Cette fonction est déterminée par un modèle de RI.

L'appariement requête/document génère un score de pertinence reflétant le degré de similarité entre la requête et le document. Ce score se traduit par une valeur communément appelée $RSV(Q, D)$ (Retrieval Status Value), où Q représente une requête et D un document. Les fonctions de mesure de pertinence sont déterminées en s'appuyant sur des modèles de RI tels que : modèle booléen[185], modèle vectoriel [186], modèle probabiliste [172]. Ces modèles fournissent un cadre théorique pour interpréter et traduire la notion de pertinence. Certains documents parmi les résultats retournés par le SRI peuvent potentiellement satisfaire les besoins de l'utilisateur. Il s'agit des documents pertinents. Un système parfait ne doit retourner que des documents pertinents, en rejetant les non-pertinents. De nos jours, les systèmes retournent généralement une liste classée de documents, dans lesquels les documents qui sont les plus à même d'intéresser les utilisateurs, sont placés en tête de liste. De plus, l'ordre dans lequel les documents susceptibles de répondre à la requête sont retournés est important. En effet, l'utilisateur consulte généralement les premiers documents retournés (les 10 ou 20 premiers). D'une manière générale, l'appariement document-requête et le modèle d'indexation permettent de caractériser et d'identifier un modèle de recherche d'information [100, 8].

1.5 Modèles de recherche d'information

De manière générale, les modèles de RI se déclinent en quatre grandes classes de modèles, nous donnons ici un bref aperçu, et détaillons les plus communément utilisés en RI :

- Le modèle booléen fait partie des modèles ensemblistes. Il considère les documents comme des ensembles de termes associés à des variables booléennes. Ces variables prennent la valeur "vrai" si un terme est présent dans le document et la valeur "faux" dans le cas contraire. La requête est représentée sous forme de suites de termes séparés par les opérateurs logiques **AND**, **OR** ou **NOT**. Les documents pour lesquels cette expression est vérifiée sont retournés à l'utilisateur. Ce modèle est dit exact car il ne renvoie aucun document quand il ne répond que partiellement à la demande et ne permet pas de classer les résultats [212]. Cependant il a été étendu (*extended boolean model*) [185] pour prendre en compte les poids des termes et autoriser un classement, donnant lieu aux modèles basés sur les ensembles flous (*fuzzy set model*) [184, 7, 86] ou sur les espaces p -normes [70].
- Les modèles vectoriels [184], font appel à des méthodes algébriques de représentation de documents et de requêtes. Les documents et les requêtes sont représentés par des vecteurs dont les coordonnées sont calculées dans l'espace des termes. La pertinence d'un document vis-à-vis d'une requête est perçue comme étant la similarité des vecteurs associés [7, 31].
- Les modèles probabilistes [144], utilisent la théorie des probabilités pour aborder la notion de pertinence et de non-pertinence d'un document vis-à-vis d'une requête à travers notamment, le "principe de classement probabiliste" (*Probability*

Ranking Principle – PRP) [212]. Le premier modèle probabiliste a été proposé par Maron et Kuhns [144] au début des années 60. Le principe de base consiste à présenter les résultats d'un SRI dans un ordre basé sur la probabilité de pertinence d'un document vis-à-vis d'une requête. Parmi les modèles les plus courants et les plus performants de nos jours, le modèle *BM25* [177] ou encore les modèles basés sur les modèles de langue [6].

- Les modèles neuronaux sont issus de la théorie connexionniste [122], ils emploient des réseaux de neurones artificiels. Ils ont la particularité de prendre en compte les dépendances séquentielles entre les termes de la requête en s'appuyant sur les caractéristiques du texte brut [153]. Ces modèles se basent sur l'apprentissage des représentations de textes bruts issus de la requête et du contenu des documents. Il a été établi que de tels modèles apportaient plus de précision lors de la mise en correspondance du vocabulaire des requêtes et celui des documents [253]. Ils nécessitent cependant un volume de données important afin d'être entraînés efficacement. De nos jours, *Google BERT*¹, modèle neuronal développé vers la fin de l'année 2018 par *Google AI*², a été déployé dans le cadre de la plus importante mise à jour d'algorithme effectué par le moteur de recherche *Google* au cours de ces cinq dernières années.

1.6 Évaluation des systèmes de recherche d'information

Les utilisateurs de SRI sont confrontés aux choix multiples s'offrant à eux parmi une liste de résultats classés et retournés par le système pour une requête formulée q . Les mesures d'évaluation communément utilisées en RI sont la précision et le rappel. Elles se basent sur la comparaison entre ordre de pertinence des documents retournés par le système et l'ordre de pertinence idéale des documents qui peut être associé à des jugements de pertinence extraits à partir de données comportementales d'utilisateurs comme par exemple leurs clics. Ces mesures sont le plus souvent calculées pour un rang donné k et sont moyennées à travers toutes les requêtes de l'ensemble test. R se réfère donc au top- k résultats retournés par le système. On suppose que la valeur de pertinence est associée à un label binaire (label pertinent et label non-pertinent).

La précision mesure la capacité d'un SRI à sélectionner des documents pertinents parmi le nombre total de documents retournés et classés, que l'on nomme R_q pour une requête q_i . Elle peut être calculée à un rang donné k , auquel cas la précision $P@k$ est déterminée parmi les k résultats retournés, ainsi $P@10$ correspond au nombre de résultats pertinents parmi les 10 premiers documents retournés par le système. La précision moyenne $AP(q_i)$ est donnée dans l'équation 1.1 en supposant que $P@k(q_i) = \sum_{j=1}^k \frac{rel(j)}{k}$ avec $rel(k)$ la fonction qui prend la valeur 1 si le document au rang k est pertinent et 0 sinon :

1. BERT est une architecture de réseaux de neurones utilisant des Transformers en anglais BERT fait référence à *Bidirectional Encoder Representations from Transformers* proposé par Devlin et al., [61].

2. *Google AI* est une filiale du groupe Google dédiée à l'intelligence artificielle.

$$AP(q_i) = \frac{\sum_{k=1}^{|R_{q_i}|} (P@k(q_i) \cdot rel(k))}{\sum_{k=1}^{|R_{q_i}|} rel(k)} \quad (1.1)$$

La précision moyenne ou MAP (*Mean average precision*) est la précision moyenne sur l'ensemble des requêtes Q . La MAP est généralement utilisée lorsque les jugements de pertinence sont binaires, bien que des variantes utilisant des jugements gradués ont été proposées [178]. La mesure de la MAP est formalisée selon l'équation 1.2 :

$$MAP = \frac{\sum_{i=1}^{|Q|} AP(q_i)}{|Q|} \quad (1.2)$$

Cependant, d'autres mesures sont utilisées pour évaluer les SRIs. Certaines dont la mesure du $nDCG$ (*Normalized Distinct Cumulative Gain*) ou gain cumulé actualisé et normalisé [108] qui permet d'intégrer un rang k auquel sont associés le document pertinent et son degré de pertinence. Il formalisé selon la fraction :

$$nDCG@k = \frac{DCG_k}{IDCG_k} \quad (1.3)$$

Où, $IDCG_k$ (*Ideal Distinct Cumulative Gain*) décrit le classement de pertinence idéal pour un utilisateur, il est défini d'après l'équation suivante :

$$IDCG_k = \sum_{i=1}^{|rel_{(k)}|} \frac{rel(i)}{\log_2(i+1)} \quad (1.4)$$

Où $rel_{(k)}$ est la liste classée de documents par ordre de pertinence jusqu'au rang k et $rel(i)$ la pertinence graduelle du résultat à la position i . Le DCG_k quant à lui est calculé selon la valeur de pertinence réduite (pénalisée) de manière logarithmique proportionnelle à la position du résultat cumulé au rang k . Le DCG_k est décrit dans l'équation suivante :

$$DCG_k = \sum_{i=1}^k \frac{2^{rel(i)} - 1}{\log_2(i+1)} \quad (1.5)$$

Il est à noter que pour un algorithme de classement ou *ranking* idéale alors $DCG_k = IDCG_k$ et cela entraîne $nDCG@k = 1$.

D'autres mesures comme le rang réciproque moyen [223] (MRR pour *Mean Reciprocal Rank*) permet de calculer la réciproque du rang auquel est associé le premier document pertinent retourné pour une requête donnée q_i . Cette réciproque est égale à 1 si un document pertinent est retourné au $rang_1$, $\frac{1}{2}$ si un document pertinent est retourné au $rang_2$ et $\frac{1}{n}$ au $rang_n$. Cette mesure est moyennée sur l'ensemble des requêtes Q , elle est formalisée suivant l'équation 1.6 où $rang_i$ dénote le rang du premier document pertinent pour la requête q_i :

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rang_i} \quad (1.6)$$

1.7 Principe de Folksonomies d'utilisateurs en RI

En 2004, *CiteULike* s'est fait connaître comme étant le premier système de *tagging* collaboratif en ligne où les utilisateurs pouvaient annoter des articles scientifiques. Par la suite, en 2005, *Confoto*³, un service incluant des représentations de méta-données en RDF⁴ offrait la possibilité aux utilisateurs de parcourir différents types de contenus scientifiques en ligne. En fonction des annotations de chacun, le système propose des recommandations personnalisées. *Confoto* a été lauréat du concours *Semantic Web challenge*. Depuis, de nombreux axes de recherche se sont développés autour de cette technique collaborative, en particulier, dans les plateformes de recommandation où la notion d'informations personnelles est associée à l'utilisation de folksonomies donnant lieu à différents champs d'application comme l'illustre la figure 1.4.

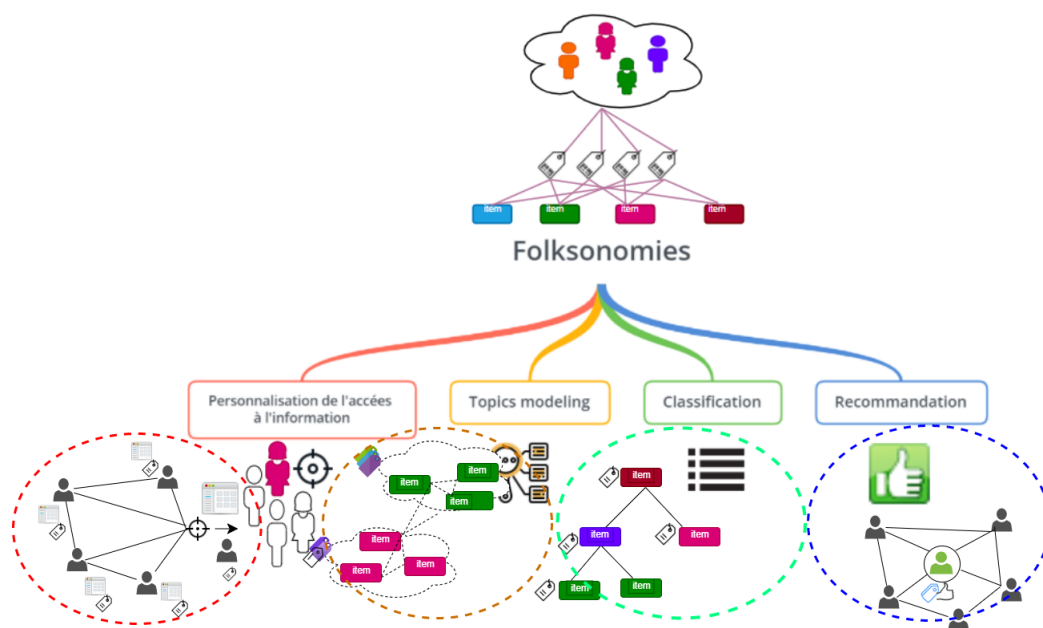


FIGURE 1.4 – Figure illustrant les domaines d'application exploitant les folksonomies pour différentes tâches de RI.

Le terme «Folksonomie» est dérivé du néologisme «*folksonomy*», issu de la formulation du mot «*folks*» (les gens) avec le terme «*taxonomie*» (ou *taxinomie*, qui est un terme faisant référence à la science des lois de la classification). Ce terme fait référence à un système de classification collaboratif, qui permet aux usagers de décrire librement des contenus sur le Web en leur attribuant des mots clés (*tags*). L'utilisateur choisit les mots clés sans obéir à une terminologie prédéfinie. Une folksonomie d'utilisateur se compose du triplet, *tags*, utilisateurs et ressources. Les *tags* inclus dans une folksonomie d'utilisateur reflète une forme d'intelligence collective [42]. Les folksonomies jouent un rôle majeur en tant que moyen d'indexation collaboratif, notamment, pour faciliter la recherche, la catégorisation et la navigation dans un volume important de

3. <http://confoto.org/>

4. Resource Description Framework décrit formellement des ressources Web ainsi que leurs méta-données pour un traitement automatique. Développé par le W3C, RDF est le langage de base du Web sémantique

ressources, simplifiant ainsi l'accès et la recherche de ressources en ligne. La principale raison de leur adoption immédiate sur le Web, provient du fait qu'aucune compétence spécifique n'est requise pour y participer [103]. Dès lors, l'une des premières étapes visant à intégrer formellement des folksonomies dans les modèles de recherche d'information classiques a été de considérer un ensemble de mots-clés représentant une ressource au même titre qu'une requête. Cet ensemble de tags permet de retourner des ressources ou des documents ayant été annotés au préalable ou de catégoriser des résultats de recherche.

1.7.1 Définitions formelles

Différents formalismes ont été proposés pour représenter de manière précise une folksonomie. Nous formulons dans ce qui suit la définition d'une folksonomie d'utilisateurs. Cette définition peut donner lieu à une description plus concise d'un utilisateur du Web. Aussi, une personomie d'utilisateur correspond à son identité construite au fil de ses sessions antérieures et ses préférences préalablement déduites de la folksonomie.

1.7.2 Folksonomie : Définition formelle

Formellement, une folksonomie F peut être défini comme un 4-tuples $F = (U, T, R, A)$ [103], où U est l'ensemble des utilisateurs qui annotent un ensemble de ressources R avec $U = \{u_1, u_2, \dots, u_m\}$, où chaque u_i est un utilisateur; T est l'ensemble de tags qui inclus le vocabulaire exprimé par la folksonomie $T = \{t_1, t_2, \dots, t_l\}$; R est l'ensemble des ressources annotées par les utilisateurs $R = \{r_1, r_2, \dots, r_n\}$; $A = (u_i, t_l, r_n) \in U \times T \times R$, (u_i, t_l, r_n) est le triplet formalisant l'assignement par l'utilisateur u_i de la ressource r_n avec le tag t_l .

1.7.3 Personomie : Restriction de folksonomies

Une personomie peut être définie comme étant la restriction de l'ensemble généré par une folksonomie à un utilisateur donné. Elle regroupe donc les annotations d'un utilisateur. Nous la définissons comme proposé par *Hotho et al.*, [103], soit \mathcal{P} une personomie d'un utilisateur donné $u \in U$ et décrit comme une restriction de \mathcal{F} pour un utilisateur u i.e., $P_u = (T_u, R_u, I_u, A_u)$ avec $I_u = \text{left}\{(t, r) \in T \times R \mid (u, t, r) \in A, T_u = \pi_1(I_u), R_u = \pi_2(I_u)$, où π_i dénote la projection sur la $i^{\text{ème}}$ dimension.

Une fois ces définitions établies, nous présentons le cadre théorique permettant de modéliser les activités d'annotations dans les environnements collaboratifs. Pour cela nous abordons les différentes représentations qui ont popularisé l'exploitation des tags dans diverses tâches de RI.

2 Représentation des tags

L'apparition des folksonomies comme nouvelle forme de donnée personnelle a favorisé l'adoption de représentation de tags intrinsèquement liée aux utilisateurs ayant annotés, mais aussi aux ressources annotées. Cela a permis aux folksonomies d'être

exploitées dans de nombreuses tâches de recherche d'information. Par conséquent, cela a contribué à la diversité des représentations de tags. En premier lieu, les modèles de représentation en sac de mots BoW comme la fréquence ($tf_{u_m(t_i)}$ qui est exprimée par le nombre de fois que le tag t_i a été annoté par l'utilisateur u_m) ont été considérés. Puis, ces modèles de représentations se sont déclinés, au fur et à mesure de leur intégration dans des tâches plus variées, en d'autres modèles de représentations. Nous synthétisons ces modèles comme suit :

- Les représentations vectorielles étendent la représentation en sac de mots des tags dans un espace vectoriel dans le but d'induire des mesures de similarités entre représentations d'utilisateurs et représentations de ressources [219, 12, 162, 103]. Ce modèle de représentation a été popularisé notamment grâce aux possibilités de faire correspondre des vecteurs pondérés de tags à des profils d'utilisateurs ou de ressources.
- Les représentations probabilistes s'appuient sur des estimations de vraisemblance entre tags d'utilisateurs et ressources d'utilisateurs annotées dans le but d'estimer la pertinence d'un tag [219]. Dans le cas d'un modèle de langue, l'estimation de cette vraisemblance peut être calculée en considérant l'ensemble des tags pouvant apparaître dans une requête [42]. Ce type de représentation a été principalement introduit dans les modèles de personnalisation de l'accès à l'information et de recommandation [234, 64, 34].
- Les représentations de tags basées sur le regroupement (*clustering*) sont particulièrement exploitées à travers l'étude de la co-occurrences entre tags, ou entre groupes définis pour organiser les tags similaires en groupes ou *clusters* [14, 237, 199, 121]. En d'autres termes, au lieu de représenter des tags individuellement, ils sont regroupés par clustering pour former des clusters de tags partitionnées selon des caractéristiques comme la fréquence [14] la proximité sémantique [234] ou encore les ressources en communs [237]. Ce type de représentations a été largement utilisé pour des tâches de classification/catégorisation et de recommandation.
- Les représentations temporelles de tags permettent de décrire l'évolution des intérêts des utilisateurs [248]. Les travaux décrivant ces représentations [2, 248, 109] exploitent des fonctions temporelles qui simulent l'atténuation des intérêts d'utilisateurs exprimés à travers des valeurs temporelles attribuées aux tags et couvrant une période de temps définie. De telles représentations ont permis l'amélioration de la pertinence des systèmes de personnalisation à la recherche ainsi que des systèmes de recommandation.
- Les représentations de tags distribuées (denses ou continues) [18] sont issues de modèles de langues neuronaux, elles se basent sur des vecteurs denses représentés dans des espaces multidimensionnels qui exploitent des distributions de termes qui varient en fonction de fenêtres contextuelles [167, 114, 152]. Le «*word embedding*» ou "plongement lexical" découle d'une méthode d'apprentissage de représentation de mots utilisée en traitement automatique des langues naturelles

(TALN) a fortement impacté les représentations de tags dans la mesure où les techniques communes de plongement lexical ont été appliquées avec succès aux tags d'utilisateurs. Ces techniques ont prouvé leur utilité dans diverses tâches en particulier pour l'extraction de relations sémantiques [21] ou encore pour la recommandation [161, 114].

3 Tâches de RI associées aux folksonomies

À travers les différentes représentations de tags, les systèmes de *tagging* ont façonné la manière dont les utilisateurs interagissent et consomment l'information. Différentes activités peuvent être effectuées par les utilisateurs de ces systèmes. Ces activités intègrent souvent une série de tâches de RI spécifiques à l'exploitation des folksonomies. De nos jours, ces tâches correspondent à diverses solutions techniques prenant la forme de *tags* visibles sur les interfaces de sites tel que *YouTube*⁵ comme le souligne la figure 1.5.

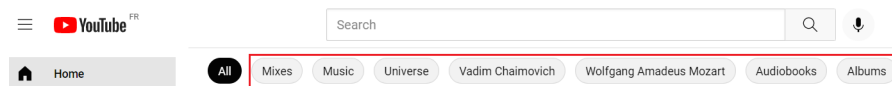


FIGURE 1.5 – Capture extraite de la page principale du site web d'hébergement de vidéos et média social *YouTube*. Différents tags apparaissent dans l'interface de recherche.

Ci-après nous détaillons ces tâches en présentons d'abord l'implication des folksonomies dans des processus d'accès personnalisé à l'information. Puis, en considérant certaines propriétés descriptives, nous faisons état des techniques et des méthodologies utilisées pour mener certaines tâches de classification, mais aussi certaines tâches permettant de structurer automatiquement l'information sous la forme de thèmes ou de sujets d'intérêts pertinents. Enfin, nous abordons l'importance de la prise en compte des données sociales et notamment celles extraites des folksonomies dans les systèmes de recommandation qui est l'objet principal de notre recherche.

3.1 Personnalisation de l'accès à l'information

Les méthodes de recherche et d'accès aux ressources en ligne sont longtemps restées principalement focalisées sur la correspondance lexicographique entre la requête et les descriptions de ressources. Bien qu'il puisse y avoir différentes correspondances entre les besoins d'utilisateurs exprimés à travers des termes dans une requête, les utilisateurs peuvent avoir un besoin d'informations différent correspondant aux préférences de chacun. Les ressources classées dans le même ordre de pertinence ne correspondent pas forcément aux besoins souvent très spécifiques de chaque utilisateur. Les approches personnalisées permettent de remédier à cela en proposant un classement des ressources qui tient compte des spécificités des utilisateurs. Différentes approches ont été proposées à des fins de personnalisation en amont du classement des résultats de recherche [33, 219, 42, 163, 233, 40], ou en étendant la représentation de la requête via des pro-

5. <https://www.youtube.com/>

cessus d'expansion de requête en aval [12, 13, 192]. Ces approches de personnalisation ont exploité les tags pour la construction de profils d'utilisateurs ou des profils de ressources. Les profils de ressources se basent principalement sur la popularité qui est associée à la fréquence relative d'apparition des tags dans une ressource [219]. Les profils d'utilisateurs se basent sur l'activité d'annotation de l'utilisateur qui peut faire référence au nombre de fois que ce dernier a annoté une ressource donnée [42]. Il peut être aussi plus générique et faire référence aux fréquences de l'ensemble des tags utilisés indépendamment des ressources [33]. D'autres aspects peuvent être intégrés lors de la construction de profils d'utilisateurs et de ressources, les centres d'intérêts des utilisateurs par rapport aux ressources par exemple [233] ou la dimension temporelle de l'annotation garantissant la fraîcheur du contenu à l'utilisateur [109]. De plus, ces profils peuvent être enrichis par des caractéristiques individuelles propres aux usagers (activité en ligne [64], historique de navigation [207], groupes et communautés [215]). De cette manière, l'accès personnalisé à l'information pour un utilisateur implique la création et la mise en correspondance de profils d'utilisateurs et de profils de ressources en fournissant :

1. Une représentation de l'usager en tenant compte de ces interactions(tags) [53].
2. Une représentation de la ressource à travers les tags que les utilisateurs expriment à son égard [33].

3.2 *Classification et topic modeling*

Si l'on considère les propriétés descriptives des tags utilisés comme descripteurs de sujets d'intérêts possédant différents niveaux de spécificités [234], alors le processus de sélection de tags pertinents représentant le mieux une page web s'apparente à une tâche de classification parmi les catégories représentées par ces tags. Contrairement à l'exploitation traditionnelle des contenus documentaires qui repose principalement sur l'analyse des termes d'un document, dans les systèmes d'annotation collaborative ces contenus sont portés par différents utilisateurs à travers différents types de ressources. Cela a permis de mieux décrire les ressources non-textuelles et a contribué à l'utilisation des folksonomies comme moyen de classification supervisée de ressources sous la forme de catégories. Les travaux dans ce sens [252, 81], se sont focalisés sur l'exploitation de fréquences de tags issus de différents schémas de pondération pour catégoriser des ressources associées à des pages Web. L'idée est de regrouper des ressources similaires (pages Web) partageant des caractéristiques communes(tags) en catégories identifiées par les tags les plus populaires. En se basant sur le modèle vectoriel classique, ces travaux décrivent l'attribution de catégories à des pages Web en comparant des vecteurs de tags associés à ces pages Web à des vecteurs de tags représentant chaque catégorie. De cette manière, il a été possible de démontrer que les annotations sociales sont dans certaines conditions de meilleures représentations que les contenus mêmes des ressources [252]. Cependant, lorsque les catégories ne sont pas connues, et que les tags employés pour la catégorisation dépendent principalement du vocabulaire employé par différents utilisateurs alors certaines inconsistances peuvent apparaître en raison de la terminologie personnelle et des motivations diverses des utilisateurs [83], qui réduisent la capacité des tags à classer [66]. Certains travaux ont exploité des approches non-supervisées issues de modèles probabilistes tel que LDA (Latent Dirichlet Allocation) [25]

permettant de manière générale d'identifier des sujets (*topics*) en regroupant à partir de caractéristiques observables les ressources associées aux tags qui sont formalisés sous forme de distribution de Dirichlet [25, 120]. Le but étant d'apprendre les thèmes représentés dans chaque ressource à travers la distribution de tags initialement représentés en sac de mots et associés ensuite à chaque thème. Cette distribution de probabilités permet au modèle de décrire l'attribution de thèmes à une ressource en supposant qu'elle soit décrite par un ensemble suffisant de tags. Chaque ressource couvre un certain nombre de thématiques dans des proportions variables. De même, chaque tag a une distribution associée à chaque thème. De cette manière, chaque thème est décrit par la probabilité $p(\text{ressource}|\text{topic})$ que la ressource soit assignée au thème et par $p(\text{topic}|\text{tag})$ la probabilité que le thème soit assigné au tag.

3.3 *Recommandation*

Les premiers systèmes de recommandation qui ont intégré les tags dans leurs processus de recommandation ont d'abord considéré les annotations sociales d'utilisateurs sous la forme de matrices d'interactions entre utilisateurs et items de manière implicite (une matrice binaire dont les éléments sont de valeur 1 si un utilisateur a annoté un item et 0 sinon). Ces matrices ont été mises en place pour palier les problèmes de parcimonie qui subsiste à la mise en œuvre de modèles pour la recommandation, en complétant notamment les valeurs nulles des matrices implicites de feedback [236]. Cependant, face à la richesse sémantique et à la possibilité d'inférer plus de connaissances à partir d'informations personnelles, les tags ont été ensuite introduits au moyen d'approches vectorielles classiques exploitant les similarités entre profils d'utilisateurs et profils de ressources [219, 34, 162, 198, 219]. Aussi, des problématiques récurrentes liées à la polysémie et à la redondance engendrées par la prise en compte de volume massif de données ont été observées. D'autres travaux [63, 148, 2] ont considéré d'autres types de caractéristiques comme la temporalité de l'interaction [2], ou l'appartenance communautaire de l'annotation [63] dans le but d'améliorer les descriptions des profils et d'augmenter leurs pertinences. Ces contenus sont généralement associés en complément des données d'annotations déjà existantes, par combinaison linéaire [136]. Les tags ont été ensuite, progressivement introduits dans des approches d'apprentissage de représentation pour la recommandation [134] notamment, grâce à l'émergence de méthodes d'apprentissage supervisées comme celles basées sur la factorisation de matrices [118] permettant de représenter des utilisateurs et des ressources dans un même espace latent, décrivant ainsi de la même façon les caractéristiques de la ressource et les préférences de l'utilisateur [174]. En particulier, cette approche a permis, d'obtenir de meilleures représentations des données, et d'autre part, de prendre en compte certains biais comme par exemple le fait qu'un utilisateur peut être plus sévère qu'un autre lorsqu'il est question d'apprécier une ressource donnée. Aussi, d'autres informations contextuelles associées à l'utilisateur comme la position géographique, l'âge et le sexe de l'utilisateur mais aussi à la ressource comme la popularité et la fraîcheur peuvent compléter l'espace des représentations de l'utilisateur ou de la ressource. Récemment les approches neuronales pour la recommandation [235, 138] ont permis d'intégrer d'avantages la complexité des interactions entre utilisateurs et ressources, mais aussi des contenus associés à ces interactions grâce à des modèles pouvant capturer des caractéristiques non-linéaires dans le

comportement de certains usagers ainsi que des représentations abstraites des contenus informationnelles associés aux utilisateurs et aux ressources.

4 Conclusion

Dans ce chapitre, nous avons tout d'abord présenté les notions de base de la RI, nous avons ensuite introduit la notion de folksonomie d'utilisateurs, ainsi que quelques définitions formelles avant de parcourir les différentes tâches en relation avec l'exploitation des folksonomies dans le domaine de la RI. Nous avons d'abord fait état des techniques de personnalisation à la recherche et à la navigation. Ces techniques ont exploité des mesures de similarité classiques étendues aux utilisateurs ayant annoté des tags similaires et aux ressources ayant été annotées par les mêmes utilisateurs. Cela a donné lieu à des méthodes de comparaison entre profils d'utilisateurs et profils de ressources. Cependant, face à la nature non-structurée du vocabulaire issus des tags, d'autres moyens ont été mis en œuvre, notamment grâce à des méthodes d'analyse statistique permettant d'organiser les tags d'utilisateurs en groupes (clusters) homogènes, selon une ou plusieurs caractéristiques communes. Cela a favorisé la découverte de nouvelles connaissances qui ont permis de mesurer l'intérêt que portent les usagers pour des ressources représentées au sein de clusters. Par ailleurs, l'apparition de processus permettant la recommandation automatique de tags pour la description de ressources en ligne a incité certains travaux à exploiter des modèles probabilistes tels que *LDA* en favorisant l'utilisation des tags pour la découverte de thématiques latentes. Cela a eu pour résultat la formulation quantitative au sein de ces modèles de l'association entre tags et ressources exploitée pour recommander l'ensemble de tags décrivant au mieux une ressource.

Enfin, parmi ces différentes tâches, nous avons prêté une attention particulière au domaine de la recommandation. Nous avons discuté de la possibilité d'appréhender les tags comme descriptions informationnelles des contenus en ligne, auquel cas, les approches standards de recommandation basées sur le contenu sont le principal point de départ pour la construction de potentielles extensions des modèles classiques. Nous avons ensuite abordé le fait que les tags peuvent être considérés comme dimension supplémentaire de la matrice d'interaction utilisateur-ressource et dans ce cas, les approches collaboratives ont servi de base au développement de modèles de recommandation plus performants.

Dans le chapitre suivant, nous présentons certaines caractéristiques observables du vocabulaire extrait de collections des folksonomies comme *del.icio.us* avant de détailler les différents modèles de représentation de folksonomies. Parmi ces modèles, l'étude de certaines propriétés inhérentes à la fréquence des tags ou à leurs temporalités, au choix de l'espace à modéliser (point de vue de l'utilisateur ou point de vue de la ressource) ou encore à la prise en compte de tags sous la forme de graphe d'interactions a donné lieu à plusieurs types de représentations où le tag y occupe une place prépondérante.

MODÈLES DE REPRÉSENTATION DE FOLKSONOMIES

*«Folksonomies are part of a new generation of tools for the retrieval, deployment,
representation and production of information, commonly termed Web 2.0»*

— Isabella Peters

Chapitre 2

MODÈLES DE REPRÉSENTATIONS DE FOLKSONOMIES

1 Introduction

COMME abordé lors du précédent chapitre, les tags permettent le partage de connaissances et de ressources connexes. Ces tags qui évoluent en temps réel offrent une facilité de recherche et de navigation à moindre coût, car aucun traitement n'est nécessaire pour structurer la donnée. Ils constituent également un socle pour certains systèmes de classification sociale de ressources [251, 81] et de recommandation personnalisée [162, 219, 33, 65]. En effet, ces tags sont exploités par ces systèmes pour représenter aussi bien les ressources que les utilisateurs (profils). Cette liberté d'annotation et d'indexation conduisent naturellement à la redondance et à la multiplication de tags associés à des mots d'argot, des néologismes, des allusions ou des références pouvant entraîner de l'ambiguïté [204, 38, 78]. D'autant plus que les représentations de tags utilisées pour ces tâches sont souvent limitées aux modèles en sac de mots qui sont comme nous l'avons vu dans le chapitre précédent sujets à la disparité et à la synonymie. Des approches récentes, basées sur des représentations distribuées de tags (denses ou continues) permettent en partie d'atténuer le problème de disparité du vocabulaire. Cependant, l'activité de l'utilisateur et la temporalité associée à cette activité doivent aussi être prise en compte. En effet, un tag assigné aujourd'hui par un utilisateur devrait avoir un impact différent, aussi bien dans la représentation de l'utilisateur que de la ressource, qu'un tag utilisé il y a quelques temps par cet utilisateur. D'autre part, les approches de représentations continues (denses) emploient des algorithmes de *word embedding* comme *word2vec* [152] ou *GloVe* (*Global Vectors for word representation*) [167]. En effet, certains travaux [161, 253, 225] se sont intéressés à l'apprentissage des représentations de tags pour saisir davantage la sémantique et le contexte de certains tags. D'autres travaux [132, 225] se sont focalisés sur la possibilité d'agréger des représentations distribuées de tags à partir de modèles pré-entraînés comme *word2vec*. Toutefois, les *embeddings* extraits à partir de ces modèles sont statiques et servent surtout à décrire des utilisateurs ou des ressources dans l'espace des représentations induit par la projection des tags dans un espace multidimensionnel [114, 222]. Des modèles de langues neuronaux contextualisés pré-entraînés tels que *BERT* [61] (*Bidirectional Encoder Representations from Transformers*), *RoBERTa* [135] (*Robustly Optimized BERT Pretraining Approach*) ou encore *Electra* [54] (*Efficiently Learning an Encoder that Classifies Token Replacements Accurately*), permettent de saisir le contexte d'un mot. À l'inverse des *embeddings* statiques issus de modèles comme *word2vec* ou *GloVe* [167], les *embeddings* considérés grâce à ce type d'architecture neuronale sont contextuels. La figure 2.1 illustre l'exploitation des *embeddings* issus de différents modèles de langues pré-entraînés.

Dans ce chapitre, nous présentons les propriétés de tags issus de folksonomies d'utilisateurs avant de mettre en exergue les différents modèles de représentations. Ces modèles ont été décrits rétrospectivement et détaillés à travers divers travaux de recherche.

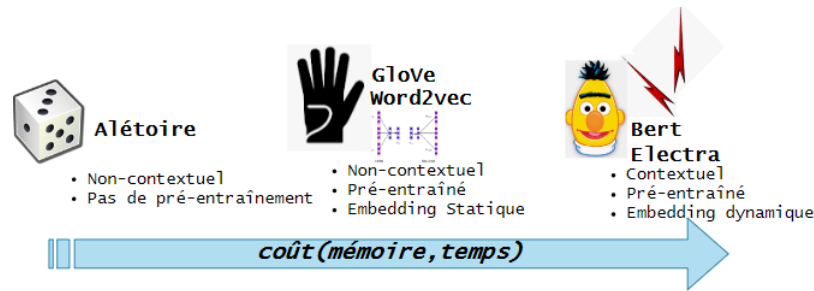


FIGURE 2.1 – Cette figure met en valeur les performances des différents *embeddings* (coût en mémoire/complexité) et les caractéristiques propres à l'exploitation de chaque type d'*embeddings*.

2 Caractéristiques et propriétés remarquables

De nombreux travaux se sont focalisés sur l'étude des propriétés relatives aux folksonomies en présentant des analyses issues d'études statistiques. Ces analyses renseignent sur la nature des processus cognitifs, souvent non-triviaux, dans les modèles d'interaction complexes caractérisant l'annotation collaborative [46, 90, 45, 43, 60].

Parmi ces travaux, *Dellschaft et Staab* [60] ont préconisé la prise en compte de deux composantes essentielles pour comprendre le comportement des utilisateurs lors de sessions d'annotations, i.e. les connaissances de base de l'utilisateur et l'influence des attributions de tags antérieurs. Ils ont intégré ces deux paramètres dans un modèle dynamique probabiliste génératif de folksonomies. Le modèle utilisé est issu du modèle de *Yule–Simon* [47] faisant référence à une loi de probabilité discrète où les fréquences relatives du n -ième mot le plus fréquent dans un corpus textuel est inversement proportionnel à la puissance typique de n . Le but étant de démontrer les propriétés caractéristiques des tags de la même manière que si on considérait du contenu textuel. Les résultats de leur étude ont révélé deux manières fondamentales par lesquelles un utilisateur peut assigner des tags : (1) il peut imiter une attribution précédente (2) il peut choisir un mot de son vocabulaire actif qui est lié au contenu de la ressource. Dans leur étude le taux d'imitation à l'attribution des tags était compris entre 60% et 90%.

Cattuto et al., [46, 45] ont étudié de manière empirique l'historique des activités d'annotations d'utilisateurs des plate-formes de tagging *del.icio.us* et *Flicker* dans le but d'examiner la dynamique collaborative portée par l'utilisation de tags dans de tels environnements. L'analyse a fourni un moyen de sonder l'aspect social et comportemental à l'origine de l'augmentation caractéristique de la taille du vocabulaire. Cela a été possible en analysant les données d'annotations de *del.icio.us* au cours d'une période de temps s'étalant sur 3 ans. La courbe (Figure 2.2) indique le nombre total de tags distincts $N(\tau)$ présents dans le système au temps τ . Cette valeur donne lieu à une ligne droite sur le graphique log-log. Ceci est d'autant plus intéressant que les données représentées sur la Figure 2.2 traversent un intervalle de temps couvrant presque tout l'historique de *del.icio.us* : la tendance en loi de puissance sur le graphique est nettement visible et évolue avec le nombre d'utilisateurs actifs et celui des ressources annotées. Les résultats des travaux de *cattuto et al.*, [46, 45], font état d'une loi de puissance caractérisée par

une augmentation exponentielle du nombre de tags distincts du vocabulaire. Le plus remarquable dans les résultats de cette étude est le fait qu'il ne semble pas y avoir de discontinuités associées à la croissance de la taille de ce vocabulaire. On pourrait s'attendre à ce que le nombre de tags distincts se stabilise au fil des multiples sessions des utilisateurs mais cela n'est pas le cas. À ce sujet, les auteurs ont aussi permis de mettre en évidence des vocabulaires locaux intrinsèquement liés à un temps imparti en étudiant dans leurs expérimentations pour un nombre de ressources populaires donné (le top 10 des ressources les plus annotées du système) le nombre d'utilisateurs ayant assigné au moins un tag à ces ressources en fonction du temps intrinsèque, autrement dit : le nombre total de tags qui leur ont été attribués. Ce vocabulaire local décrit une croissance linéaire homogène, il a aussi été constaté qu'il n'y avait pas de différences observables mesurées par rapport au rang (popularité) des ressources sélectionnées à ce niveau. D'autres analyses statistiques découlant des mêmes travaux révèlent cependant une décroissance exponentielle associée aux tags contenant plus de deux mots parmi le corpus de tags. Dans ce contexte, les auteurs assurent que le nombre moyen de tags est une caractéristique intrinsèque du système qui ne dépend pas fortement de la taille du corpus de tags considéré.

Ces observations ont donné lieu à une multitude de questions de recherche, comme la possibilité que la croissance observée du vocabulaire dans le contexte d'un seul utilisateur peut être corrélée à un phénomène collectif voir collaboratif. Où qu'au contraire, cette croissance ne fait que refléter les processus cognitifs complexes (intégrant la sémantique) au niveau de chaque utilisateur. Il a aussi été question de déterminer quels étaient les éléments clés de l'interaction utilisateur-système qui conduit aux comportements observés.

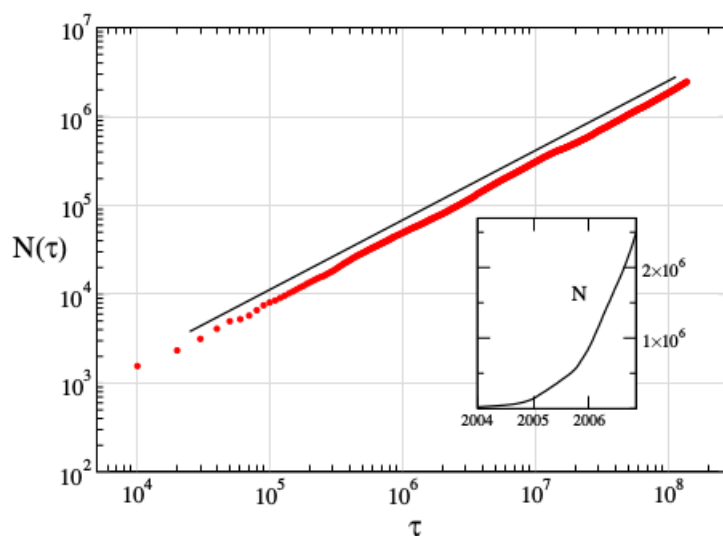


FIGURE 2.2 – Cette figure illustre l'évolution du nombre total de tags distincts N , dans *del.icio.us*. La courbe que l'on retrouve dans *Cattuto et al.*, [46] représente N en fonction du temps τ . La loi de puissance qui décrit la fonction $N(\tau)$ indique un exposant $\sigma \approx 0.8$, le graphe confirme la croissance du nombre N de tags distincts en fonction du temps. Cela donne aussi un aperçu de la taille du vocabulaire engendré.

En mettant l'accent sur les tags comme étant des entités dynamiques, *Cattuto et al.*, [43] se sont par la suite intéressés aux systèmes de tagging d'un point de vue sémiotique, c'est-à-dire l'étude de leurs significations, de leur représentation, ainsi que de leurs interprétations. En outre, ils ont été en mesure de déterminer comment le vocabulaire local de l'utilisateur, provoque l'émergence de caractéristiques observables à l'échelle de la collection. Pour ce faire, les auteurs ont examiné l'évolution des relations entre un tag donné et un ensemble de tags co-occurents. Ces travaux ont inspiré *Halpin et al.*, [90] qui ont exploré la possibilité de rassembler des tags similaires en se basant sur leurs fréquences d'apparition locales ou globales et/ou de co-occurrences.

Au fur et à mesure de l'intégration de fonctions permettant d'automatiser la représentation et la gestion des méta-données, la nature des folksonomies a été mieux appréhendé. Ainsi, l'émergence de modèles de représentation plus élaborés a favorisé le développement de travaux de recherche exploitant de manière plus rigoureuse le contenu informationnel et relationnel entre entités impliquées, c'est-à-dire, entre utilisateurs, tags et ressources. Afin d'apporter plus de précisions quant à la manière dont a été appréhendé ces différentes représentations de tags dans des tâches concrètes de RI, nous les avons détaillé dans la section suivante.

3 Modèles de représentation de tags issus de folksonomies

Dans une folksonomie, chaque élément peut être représenté distinctement dans son propre espace de représentation. Un tag peut être représenté dans l'espace engendré par l'ensemble de tous les tags, sous forme vectorielle en encodage *One hot*¹. Une ressource ou un utilisateur peut être représenté dans l'espace des tags engendré par les utilisateurs qui annotent les ressources ou par l'ensemble des ressources annotées. Chaque élément d'une folksonomie peut également être représenté dans l'espace d'un autre élément, un tag dans l'espace des utilisateurs ou des ressources ou un utilisateur et une ressource dans l'espace des tags. Pour modéliser les différentes relations existant entre ces éléments, différentes représentations ont été illustrées à travers de nombreux travaux de recherche. Elles peuvent être spécifiques à certaines tâches telles que la personnalisation de l'accès à l'information, l'expansion de requête, la classification ou la recommandation. Les représentations de folksonomies ont été synthétisées selon les tâches pour lesquelles elles ont été adoptées dans la littérature, le tableau 2.1 illustre les différentes représentations de tags à travers différentes tâches et travaux de recherche connexes.

3.1 Représentations de tags en sac de mots

Les représentations de tags en sac de mots (*Bag Of Words BoW*) sont dérivées des multiples travaux en recherche d'information abordant la représentation de documents à travers l'ensemble des mots les composants. Le modèle *Bag Of Words* permet de représenter les contenus informationnels de ressources décrites par des utilisateurs de

1. L'encodage *One-hot* consiste à encoder une variable représentant n catégories par n variables binaires de telle manière que la i -ème variable binaire représente la i -ème catégorie

Représentations	Tâches associées		
	Classification & Topic modeling	Recommandation	Personnalisation & expansion de requête
Sac de mots	[81, 252, 43, 90]	[116, 38, 38]	[234, 64, 121, 162]
Vectorelles	[162, 219, 65]	[233, 63]	[38, 33]
Modèle de langue	[119, 243, 219]	[192, 119, 240]	[208, 133, 11, 12, 13]
Graphe et Clustering	[83, 199, 78]	[200, 14, 237, 78]	[20, 60, 199]
Temporelles	[248, 207]	[20, 248, 139]	[2, 109, 29, 139]
Denses (continues)	[128, 57, 254]	[127, 131, 92, 161, 105]	[253, 85, 50]

TABLE 2.1 – Modèles de représentation de tags classés par tâche et par travaux de recherche

manière normalisée, selon la fréquence avec laquelle une ressource a été annotée par un ou plusieurs utilisateurs. Il s’agit donc d’une représentation où chaque tag se voit affecter le nombre de fois qu’il a annoté une ressource.

L’utilisation du modèle en sac de mots peut s’illustrer dans un cadre vectoriel classique. Il peut être étendu en considérant le nombre de fois qu’une ressource ait été annotée par un tag spécifique. Dans ce cas, le BoW considère les tags composants les ressources indépendamment les uns des autres. Ce modèle est aussi associée au calcul de la co-occurrence entre paires de tags, ou d’ensemble de tags co-occurrent avec un tag particulier. Dans un espace vectoriel classique, des similarités de type cosinus sont calculées entre ensemble de tags [90, 97, 60]. Ces ensembles de tags décrivant des utilisateurs ou des ressources peuvent aussi être considérés comme des requêtes permettant de retourner des résultats de recherche personnalisée. Dès lors, le BoW a été décliné selon différents schémas de pondérations tels que : tf [219], $tf*idf$ [36] ou BM25 [219] qui ont été largement utilisés dans des travaux [163, 162, 219] visant principalement à décrire des utilisateurs ou des ressources par un ensemble (bag) de tags issu de différents schémas de pondération pour des tâches de personnalisation de l’accès à l’information mais aussi dans le cadre de recommandation de contenus annotés en ligne. Le tableau 2.2 liste quelques schémas de pondération communément utilisés en RI.

3.1.1 Représentations de tags dans un espace vectoriel

La caractéristique principale d’une folksonomie tient au fait que les utilisateurs peuvent annoter des tags ; ces tags sont associés à des ressources et les ressources sont liées aux utilisateurs. Cette caractéristique a été exploitée dans le cadre du modèle vectoriel de base [183]. Sur le plan pratique, ce modèle a été adapté aux folksonomies pour inférer des représentations vectorielles d’utilisateurs et de ressources parmi les interactions tags-utilisateurs et tags-ressources. Dans un espace vectoriel, ces interactions peuvent être représentées sous la forme de matrices, soit respectivement $\mathbb{U} = (u_{i,j}) \in \mathbf{R}^{t \times u}$ et $\mathbb{R} = (r_{i,j}) \in \mathbf{R}^{t \times r}$. Les interactions tags-utilisateurs peuvent être définies sommairement pour un corpus de t tags et de u utilisateurs selon $\mathbb{U} = (u_{i,j})$ où $u_{i,j} = 1$ si l’utilisateur j a annoté avec le tag i et 0 sinon. L’ensemble des vecteurs colonnes $\overrightarrow{u_{(*,j)}}$ défini

Notation	référence	formule	mention
$tf_{u_m}(t_l)$	User tag frequency	$tf_{u_m}(t_l) = count(t_l, u_m)$	Le nombre de fois que le tag t_l est utilisé par l'utilisateur u_m
$tf_{r_k}(t_l)$	Resource tag frequency	$tf_{r_k}(t_l) = count(t_l, r_k)$	Le nombre de fois que le tag t_l a été utilisé pour annoter la ressource r_k
$idf(t_l)$	Resource inverse tag frequency	$idf(t_l) = \log \frac{N}{n_r(t_l)}$; N : nombre de ressources. $n_r(t_l)$: nombre de ressources ayant été annotées avec le tag t_l .	Nombre de fois que le tag t_l a été utilisé dans toute la collection
$iuf(t_l)$	user inverse tag frequency	$iuf(t_l) = \log \frac{M}{n_u(t_l)}$ M : nombre d'utilisateurs. $n_u(t_l)$ nombre d'utilisateurs ayant annotés avec le tag t	Nombre de fois que le tag t_l est utilisé parmi tous les utilisateurs U
$Bm25_{u_m}(t_l)$	$Bm25$	$iuf(t_l) \cdot \frac{tf_{u_m} \cdot (k_1 + 1)}{tf_{u_m} + k_1(1 - b + b \cdot u_m \cdot avg(u_m))}$ b et k_1 sont deux paramètres standards de valeurs resp 2 et 0,75. $avg(u_m)$: Le nombre moyen d'utilisateurs	Représentation d'un tag t_l parmi l'ensemble des utilisateurs en $Bm25$
$Bm25_{r_k}(t_l)$	$Bm25$	$idf(t_l) \cdot \frac{tf_{r_k}(t_l) \cdot (k_1 + 1)}{tf_{r_k}(t_l) + k_1(1 - b + b \cdot r_k \cdot avg(r_k))}$ b et k_1 sont deux paramètres standards de valeurs resp. 2 et 0,75. $avg(r_k)$: Le nombre moyen de ressources	Représentation d'un tag t_l parmi l'ensemble des ressources en $Bm25$

TABLE 2.2 – Ce tableau synthétise les différents modèles de pondération exploités dans des approches classiques de personnalisation et de recommandation basées sur les folksonomies

l'ensemble des représentations des utilisateurs. De manière similaire, les relations entre tags et ressources peuvent tout aussi être formalisées via la matrice $\mathbb{R} = (r_{i,j})$ pour le même corpus de t tags et une collection de r ressources, avec $r_{(i,j)} = 1$ si la ressource j a été annotée avec le tag i et 0 sinon. Chaque vecteur ligne $\vec{r}_{(i,*)}$ de la matrice \mathbb{R} est un

vecteur dans l'espace induit par les tags dont la somme des éléments non-nuls représentent le nombre de fois que les ressources de la collection ont été annotées avec le tag $tag_{(i)}$. Ces représentations qui induisent des vecteurs de tags dans un espace vectoriel ont été la base d'une série d'études [39] visant à démontrer la corrélation existante entre les distributions des termes d'une requête et les distributions des tags de ressources retournées par cette requête. Cela a motivé le développement d'approches d'expansion sémantique de requêtes basées sur des relations sémantiques entre tags ou des proximités sociales entre utilisateurs [1]. Cela équivaut à privilégier les ressources annotées par des représentations similaires d'utilisateurs lors du classement des résultats de recherche comme l'ont suggéré les travaux de *Bender et al.*, [17]. Dans ce sens, les travaux de *Abbasi et al.*, [1] en plus d'exploiter les termes d'une requête dans le modèle vectoriel classique, ont associé aux tags présents parmi la liste des termes de la requête des vecteurs de ressources extrait à partir de la matrice d'interaction tag-ressource. Autrement dit, cela revient à attribuer plus de pertinence aux ressources ayant été annotées avec des termes(tags) apparaissant parmi l'ensemble des termes figurant dans une requête. Ainsi, pour une requête représentée sous forme vectorielle \vec{q} , la similarité entre le vecteur \vec{q} et le vecteur de ressource \vec{r} est calculée avec la similarité du cosinus de la manière suivante :

$$Sim_{cosinus}(\vec{q}, \vec{r}) = \frac{\vec{q} \cdot \vec{r}}{|\vec{q}| |\vec{r}|} \quad (2.1)$$

Si le vecteur de requête \vec{q} est égal au vecteur de ressource \vec{r} , alors leur similarité sera égale à 1 et si les vecteurs \vec{q} et \vec{r} n'ont pas de termes en commun, leur similarité en cosinus sera égale à 0.

Ces représentations vectorielles ont été exploitées à différents niveaux, au niveau de l'utilisateur, sous la forme de profil d'utilisateur afin de mieux appréhender ses besoins. Du côté de la ressource, sous la forme de profil de ressource pour mieux la décrire [33, 38]. Ces représentations ont été étendu selon différents schémas de pondération attribuée aux tags et parmi les entités considérées (fréquence de tags par utilisateurs ou par ressources). Ces profils ont fait l'objet d'études approfondies qui ont permis de mettre en valeur différentes méthode pour la modélisation des profils. Cela a motivé les chercheurs à proposer des systèmes de recommandation personnalisés répondant aux besoins de chaque utilisateur selon son profil.

3.1.2 Représentation de tags issues de modèles de langues

Ces représentations sont issues du modèle de langue introduit en RI par *Ponte et Croft* [172], qui calcule la probabilité qu'une requête puisse être générée à partir d'une ressource. Les modèles de langues ont été adaptés aux folksonomies en considérant une ressource et le ou les tag(s) la décrivant. À partir d'un ensemble de requêtes que l'on associe à un ensemble de tags et d'une collection de ressources, le modèle de langue octroie à un tag la possibilité de classer des ressources en fonction de leur probabilité d'apparition [243]. Cela peut être traduit selon la théorie Bayésienne par $p(t|r)$ qui décrit la probabilité qu'un tag t soit attribuer à une ressource donnée r et peut être estimée comme suit :

$$p(t|r) = \frac{p(r|t) \cdot p(t)}{p(r)} \quad (2.2)$$

$p(r|t)$ est estimée selon la fréquence d'apparition de la ressource r parmi l'ensemble des ressources annotées par le tag t . $p(t)$ est la probabilité a priori du tag t . Il est trivial d'affirmer que si le tag t apparaît à des fréquences plus élevées, cela conduit à une plus grande probabilité. Dans le cas où la ressource r fait parti de l'ensemble des ressources annotées alors $p(r|t)$ est estimée grâce à la fréquence d'occurrence $tf_r(t)$ (par maximum de vraisemblance) qui représente le nombre de fois que la ressource r a été annotée avec le tag t . Sinon, l'une des solutions afin d'éviter le score de probabilité nulle résultant du fait que la ressource ne fasse pas partie de l'ensemble des ressources annotées est de considérer leurs contenus textuels représentés grâce à un modèle de langue. Pour un modèle de langue unigram la ressource r peut être représentée sous la forme d'un sac de mots $W_r = \{w|w \in r\}$. En respectant l'hypothèse d'indépendance des mots, l'estimation de la probabilité de la ressource r pour un tag t donné est estimée avec :

$$P(r|t) = \prod_{w \in W_r} P(w|t) \quad (2.3)$$

Dans l'équation 2.3, $P(w|t)$ représente la vraisemblance du mot w apparaissant dans le contenu textuel de la ressource r pour un tag t donné.

En introduisant la composante comportementale liée à l'utilisateur u , alors $P(t|r, u)$ estime la probabilité pour cet utilisateur qu'un tag t soit attribué à une ressource r suivant l'équation :

$$P(t|r, u) = \frac{P(r|t, u) \cdot P(t|u)}{P(r|u)} \quad (2.4)$$

$P(r|t, u)$ correspond à la vraisemblance de la ressource r pour un tag donné t et un utilisateur donné u . $P(t|u)$ est la probabilité conditionnelle a priori du tag t pour un utilisateur donné u . L'équation 2.4 se base sur l'ensemble des interactions passées de l'utilisateur à l'égard des ressources qu'il a annoté. En conséquent, l'équation 2.4 se réécrit de la manière suivante en considérant le contenu de la ressource :

$$P(t|r, u) = \frac{\prod_{w \in W_r} P(w|t, u) \cdot P(t|u)}{P(r|u)} \quad (2.5)$$

Dans l'équation 2.4, si un tag t n'a pas été utilisé par l'utilisateur u alors la probabilité a priori de $P(t|u) = 0$, de même pour $P(r|t, u)$. Les tags sont restreints à l'ensemble de tags que l'utilisateur a déjà utilisés.

Lin et al., [134] ont proposé d'utiliser ces représentations de tags issues de modèles probabilistes pour l'expansion de requête dans une tâche de personnalisation : en tenant compte des co-occurrences des termes identifiées lors d'annotations d'utilisateurs [216] et/ou en se basant sur le calcul de similarité entre représentations de ressources et représentations de tags [12, 13]. Dans leurs travaux, *Bao et al.*, [13] étudient la pertinence de la prise en compte de mesures de pondérations issues de modèles probabilistes tels que *BM25* comme mesures de popularité dans un système de personnalisation à la recherche sur le Web. Ils introduisent deux valeurs de score d'importance, *SocialSimRank* et qui mesurent respectivement la similarité entre les annotations sociales et les requêtes web et *SocialPageRank* qui mesure la popularité d'une page Web. Ils concluent que ces mesures fournissent une meilleures performances que les mesures traditionnelles, telles que la traditionnelle correspondance de termes et l'algorithme *PageRank* [165].

nœuds (tags) sont pondérées par la mesure définie dans l'équation 2.6. La figure 2.4 illustre un exemple de visualisation d'un graphe de corrélation inter-tags, considérant uniquement les corrélations entre termes correspondants au nœud central "complexity".

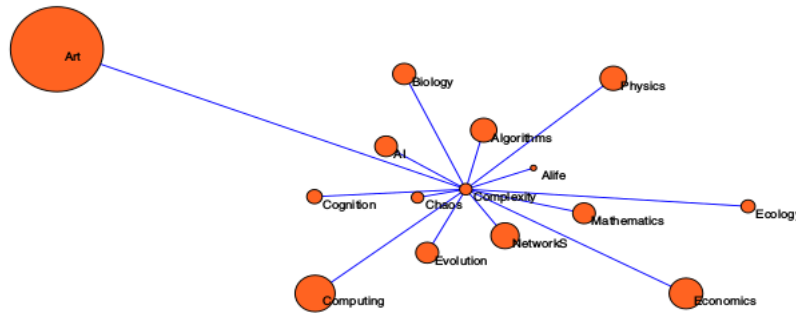


FIGURE 2.4 – Extrait de la méthode de visualisation de *Halpin et al.*, [90] à partir d'un graphe de corrélation entre tags et en fixant le terme "complexity" comme nœud central. Les tags sont extraits de la collection *del.icio.us*

Les modèles de personnalisation et de recommandation qui ont adopté ces types de représentations ont été abordés dans de nombreux travaux [200, 14, 237, 20, 207]. La majorité des approches issues de ces travaux exploitent les structures en graphes tripartis ou bipartis. Par exemple, dans les graphes représentant des utilisateurs, les sommets représentent des ressources qui sont liées entre elles par des tags. De cette manière, un utilisateur est représenté par le graphe de ses activités d'annotation. Cependant, il existe de nombreuses variantes de tels graphes. Cela est le cas avec les graphes de relations sémantiques entre tags qui ont été représentés sous la forme des graphes conceptuels [202] et exploités pour la construction d'ontologies [220, 126, 195] pour formaliser la représentation d'utilisateurs et inférer de nouvelles connaissances à partir du vocabulaire de tags. D'autres approches exploitant des algorithmes de clustering ont été utilisés pour rassembler les tags similaires dans des clusters reflétant les préférences de l'utilisateur à l'égard de thématiques diverses [237, 20, 207]. L'aspect communautaire des systèmes de tagging reste aussi un moyen d'identifier des groupes ou communautés partageant les mêmes centres d'intérêt et reste une problématique inhérente à la détection de communautés dans les réseaux complexes [113]. Il est donc significatif de se focaliser sur l'aspect communautaire des systèmes d'annotations collaboratives lors de la construction de profils d'utilisateurs et/ou de ressources. Des systèmes d'intérêts peuvent être représentés comme étant un ensemble de nœuds (ou sommets pour graphes non orientés), reliés par des arêtes. Cependant, les tags fournis par les utilisateurs sont différents de par leurs natures et de par le degré d'importance que leur accordent les usagers. Différentes motivations sont à l'origine de l'action d'annoter une ressource (partage, expression d'opinions, catégorisation), cela a donc suscité des intérêts divers à travers des domaines distincts.

3.3 Représentation temporelle de tags

Une Folksonomie qui prend en compte la composante temporelle est représentée par le quadruplet (U, R, T, τ) où $u \in U$ représente un utilisateur, $r \in R$ une ressource, $t \in T$ un tag et τ la composante temporelle, celle-ci est communément exploitée sous le format *timestamp*². Une annotation issue de ce quadruplet $a \in A \subseteq U \times R \times T \times \tau$ permet de mettre en évidence les changements comportementaux des usagers de folksonomies à travers leurs activités d'annotation qui déclinent au fil du temps. Dans cette optique, certains travaux [52, 248, 29] ont exploité une fonction Gaussienne permettant de simuler l'atténuation graduelle de l'activité d'annotation passée de l'utilisateur en considérant son activité d'annotation la plus récente ainsi que son activité d'annotation la plus ancienne. Ainsi, $w_{(\tau)}(u, r)$ qui représente le degré de déclin des intérêts de l'utilisateur u pour une ressource r peut être formalisé comme suit :

$$w_{(\tau)}(u, r) = e^{\frac{-\ln 2 \times \tau(u, r)}{hl_u}} \quad (2.7)$$

Où, $\tau(u, r)$ est un entier non négatif dont la valeur est une plage temporelle pouvant être associée à des jours et qui est égal à 0 si cette plage (le nombre de jours) entre la dernière annotation de l'utilisateur u et la date du jour pris comme référence. Ainsi, $\tau(u, r)$ est égale à 1 si l'activité de tagging de ce même utilisateur se déroule dans une période antérieure à une journée. hl_u représente la durée de vie totale de l'ensemble des sessions de l'utilisateur. En d'autres termes et grâce aux caractéristiques de la gaussienne, l'atténuation du degré d'intérêt de l'utilisateur peut être décrite de la manière suivante :

- Si l'utilisateur a une grande durée de vie ($hl_u \gg$), cela correspond à une forte activité d'annotation pendant une longue période de temps et dans ce cas, l'intérêt de cet utilisateur diminue lentement.
- Si l'utilisateur possède une courte durée de vie ($hl_u \ll$), cela correspond à l'atténuation du degré d'intérêt pour une ressource de manière brusque pour cet utilisateur.

Selon l'équation 2.7, quand $\tau(u, r) = hl_u$ alors $w_{(\tau)}(u, r) = 0.5$. Cela veut dire que quand la valeur temporelle $\tau(u, r)$ est moindre, l'annotation est forcément récente, tandis que pour une annotation plus ancienne cette valeur est plus grande. De cette façon, les intérêts actuels des utilisateurs sont mis en avant.

En utilisant ce principe, Zheng et al. [248] sont parvenus à intégrer efficacement les tags ainsi que les informations temporelles qu'ils véhiculent dans un processus de filtrage collaboratif pour la recommandation. Ils ont considéré des représentations de tags pondérées selon l'équation 2.8, qui permet de prendre en compte la popularité de la ressource par rapport au tag qui le décrit et la valeur temporelle attribuée à la ressource au sein de la matrice utilisateur-ressource $M^{m \times n}$ qui fait référence aux notes de m utilisateurs à n ressource. Cette valeur temporelle correspond à l'instant où la

2. Le *timestamp* est un format uniformisé d'horodatage qui fait référence au nombre de secondes écoulées depuis le 1er janvier 1970 à minuit

ressource a été annotée par l'utilisateur. Dans un espace vectoriel, des représentations vectorielles d'utilisateurs sont construites selon ces mesures de pondérations et sont utilisés pour identifier les utilisateurs similaires. En considérant la matrice $M^{m \times n}$, l'élément $M_{u,r}$ de cette matrice est pondéré selon la fréquence des tags de chaque utilisateur, selon la valeur temporelle attribuée à la ressource quand l'utilisateur a annoté cette ressource, ou en combinant ces deux mesures. L'équation 2.8, $w_{t(u,r)}$ correspond à une mesure de pondération globale comprenant l'ensemble des annotations de l'utilisateur u : $w_{t(u,r)} = \sum_{t \in t(u,r)} w_{(u,t)}$, avec $t(u,r)$ qui définit l'ensemble de tags utilisé par l'utilisateur u pour annoter la ressource r .

$$M_{u,r} = \lambda w_{t(u,r)} + (1 - \lambda) w_{(t)}(u,r) \quad (2.8)$$

Le poids $w_{(u,t)}$ de chaque tag annoté par un utilisateur est calculé avec $tf_u(t)$ le nombre de fois que le tag t a été annoté par l'utilisateur u et k est le nombre total de tags que l'utilisateur u a annoté indépendamment des ressources.

Il est donné comme suit :

$$w_{(u,t)} = \frac{tf_u(t)}{\sum_{i=1}^k tf_u(t_i)} \quad (2.9)$$

Il est aussi à noter que dans l'équation 2.9, la somme $\sum_{i=1}^k w_{u,t_i} = 1$, cela signifie le poids du tag est un nombre réel compris entre 0 et 1. Dans l'équation 2.8, λ est un paramètre qui permet d'ajuster le biais induit par l'atténuation de l'intérêt de l'utilisateur pour une ressource dans le temps. Toutefois, d'autres travaux comme ceux de *Abel et al.* [2] ont comparé d'autres stratégies pour représenter les tags d'utilisateurs à travers une fenêtre temporelle permettant leur prise en compte lors de la modélisation d'utilisateurs dans des processus de personnalisation sur Twitter. *Abel et al.*, ont considéré les utilisateurs comme étant associés à un ensemble de concepts pondérés dans un espace vectoriel. Ces concepts se basent sur la fréquence d'occurrences d'entités relatives à ces concepts tels que les *hashtags* qui sont combinés à une distance représentant l'information temporelle entre le tweet le plus récent et le plus ancien et faisant référence à l'entité auquel est relié ce concept. En outre, ils ont étudié l'impact de la dynamique temporelle de ces représentations sur l'exactitude du processus de personnalisation. Les auteurs ont considéré deux types d'utilisateur : les utilisateurs actifs qui interagissent pendant une longue période de temps et ceux qui le sont moins, et qui interagissent de manière moins régulière.

L'exploitation des représentations temporelles de tags a favorisé le développement de représentations d'utilisateurs et de ressources plus pertinentes. Cela a conduit à l'obtention de recommandations de contenus plus actualisés et plus en phase avec les intérêts évolutifs de l'utilisateur. Les résultats des différents travaux [248, 2, 52, 29] ayant traité le sujet ont confirmé que la prise en compte de la temporalité associée aux tags lors de la modélisation d'utilisateurs et de ressources pouvait améliorer de manière significative la précision des systèmes de recommandations personnalisées.

3.4 Représentations distribuées de mots et de tags

Le *word embedding* « plongement de mots » ou « plongement lexical », est une méthode de représentation de mots issue des domaines du traitement automatique des langues naturelles (TALN) et de l'apprentissage automatique. L'idée générale est de projeter un ensemble de mots d'un vocabulaire de taille donnée dans un espace vectoriel continu où les vecteurs de ces mots sont de taille équivalente à la dimension de l'espace. Il s'agit de méthodes qui consistent à apprendre les représentations des mots à partir d'une fenêtre d'occurrences, qui fait référence au contexte à prendre en considération. Nous détaillons dans l'annexe I, quelques techniques d'apprentissage automatique exploitées dans le cadre de la représentation distribuée des mots. Ces techniques reposent sur l'exploitation d'architecture en réseau de neurones comme dans le modèle *word2vec*[152]. Elles se sont aussi déclinées sous la forme d'algorithme d'apprentissage non supervisé permettant d'obtenir des représentations vectorielles de mots comme dans GloVe [167] ou encore à travers une méthode de représentation des mots en graphe qui optimise une fonction objective conçue pour préserver à la fois les structures locales et globales du graphe comme dans [211].

Ces méthodes de représentation ont été appliquées avec succès aux tags et ont permis de découvrir certaines caractéristiques pouvant être prises en compte en vue d'enrichir le contenu informationnel de certaines ressources en ligne [161]. Dans ce qui suit, nous présentons les modèles de représentations distribuées de mots en explicitant les plus courant, avant de parcourir les travaux adaptant ce type de représentations aux tags d'utilisateurs.

3.4.1 Représentations distribuées des mots

Ces représentations tirent leurs origines du modèle de langue neuronal introduit par *Bengio et al.*, [18]. Il s'agit d'un modèle de langue probabiliste dans lequel les probabilités des mots sont calculées grâce à une architecture neuronale qui apprend simultanément les représentations distribuées (plongements lexicaux) et la fonction de probabilité correspondant aux fenêtres contextuelles des mots. L'approche de *Mikolov et al.*, [152] se base sur ce principe et propose deux variantes d'un algorithme : il s'agit de *word2vec*. Le modèle *continuous bag-of-words* (sac de mots continu CBOW) a pour objectif de prédire un mot à partir de ses mots voisins, le modèle *skip-gram* a pour objectif de prédire les mots du contexte à partir d'un mot central. D'autres méthodes de représentations de mots en fonction de leurs contextes ont fait leurs apparitions. C'est le cas de *GloVe* (*Glove global vectors for word representation*) [167], qui est un modèle de représentation qui se base sur des matrices de co-occurrences mot-contexte. *Pennington et al.*, [167] ont proposé de combiner contexte local et global à travers une matrice de co-occurrence globale pour l'apprentissage de représentations de mots. Parmi les méthodes de plongements qui ont été proposées dans la littérature, certaines ont exploitées des représentations de mots sous la forme de graphes pondérés. C'est le cas de l'algorithme *LINE* [211] (*Large-scale Information Network Embedding*). L'algorithme s'appuie sur la notion de proximité entre les *embeddings* générées à partir des nœuds d'un graphe de co-occurrence. Le contexte général d'un mot est encodé en fonction du voisinage d'un nœud du graphe. *LINE*

associe à chaque nœud du graphe, une représentation vectorielle des proximités au voisinage de ce nœud.

3.4.2 Représentation distribuée agrégée

Il est possible de combiner des représentations distribuées des mots d'une séquence donnée en utilisant une fonction linéaire permettant de les agréger. Dans ce contexte, les approches dites *agrégées* font référence à l'utilisation d'un réseau de neurones qui calcule le vecteur de représentation d'un texte à partir de l'agrégation des vecteurs de représentation des mots qui le compose. Les approches dans ce sens [224, 247] utilisent une simple moyenne des représentations distribuées. Cela revient à accorder la même importance à tous les mots. Il est tout aussi possible de calculer la représentation latente d'un texte (*sentence embedding*) [115, 5]. En ce sens, Kenter et al., [115] ont proposé un modèle (*Siamese CBOW*) qui permet d'apprendre des représentations de mots et de les agréger grâce à une fonction qui emploie la moyenne. Pendant la phase d'apprentissage, le modèle optimise des plongements lexicaux pour calculer le vecteur de représentation du texte. Arora et al., [5] ont décrit un modèle entraîné pour représenter des corpus de textes. Leur modèle se base sur le calcul de la moyenne pondérée des vecteurs distribués des mots d'une séquence. La représentation moyenne est ensuite modifiée à l'aide d'une analyse en composantes principales (ACP) ou d'une décomposition en valeurs singulières (DVS) pour réduire l'espace de représentation du texte, autrement dit, réduire les dimensionnalités. Afin de ne pas accorder la même importance à tous les mots du texte pendant la phase d'agrégation, certains travaux dont ceux de Vulic et al. [224] et Zamani et al., [239] ont proposé d'exploiter des pondérations pour accorder d'avantage d'importance à certains mots. Cependant, les longues séquences de mots peuvent aboutir à des représentations qui manquent de précision d'un point de vue sémantique. Dès lors, d'autres approches ont été développées en construisant directement une représentation latente du texte sans utiliser les représentations des mots qui le compose. Il s'agit des approches dites non-agrégées, dont la plus populaire est sans doute *ParagraphVector* [123]. Cette méthode est issue de *word2vec* [149] et permet la construction de représentation latente d'un texte. Elle se compose de deux modèles, *Distributed Memory* (PV-DM) et *Distributed Bag of Words* (PV-DBOW). A défaut de considérer une combinaison des représentations des mots qui le compose, le modèle *ParagraphVector* estime une séquence de mots associée à une phrase, un paragraphe ou un document, comme une unité indivisible. Le modèle PV-DBOW adopte la même architecture que celle du Skip-Gram, où la donnée d'entrée est un paragraphe plutôt qu'un mot. Le modèle PV-DM présente quant à lui une architecture identique à celle du CBOW, mais est étendue à un paragraphe dans la couche d'entrée et une matrice de représentations de documents.

3.4.3 Représentation distribuée de tags

Dans une folksonomie, contrairement aux contenus textuels classiques, l'extraction du contexte d'un tag peut être appréhendé de différentes manières. C'est ainsi que, Cattuto et al., [209, 44] ont suggéré différentes méthodes de ces représentations. Ces méthodes reposent sur l'analyse des caractéristiques d'utilisateurs ayant annotés en utilisant un même tag et l'examen des relations pouvant exister entre ressources annotées

avec un tag particulier. Ainsi, face au succès qu'ont connu les modèles de représentations distribuées en *RI* et en *TALN*, certains travaux de recherche dont ceux de *Niebler et al.*, [161] ont discuté l'applicabilité d'algorithmes tels que : *word2vec*[152], *GloVe*[167] et *LINE* [211] pour la construction de plongements lexicaux de tags. Niebler et al., ont estimé la qualité sémantique des représentations distribuées de tags produites par l'exploitation de ces algorithmes. Les résultats des travaux de *Niebler et al.*, ont encouragé les travaux sur l'apprentissage des représentations de tags. Récemment, des axes de recherche se sont focalisés sur l'apprentissage de représentations de tags pour la recommandation de contenus [92, 87, 131]. L'idée de base est d'enrichir les représentations de tags qui sont simultanément apprises en y joignant d'autres caractéristiques propres aux ressources telles que leurs contenus textuels ainsi que les tags les décrivant dans un espace vectoriel commun. C'est dans ce contexte que *Kataria et al.* [114] ont exploité un modèle neuronal intégrant une architecture à deux couches. Pour un corpus M donné contenant $\{d_1, d_2 \dots d_M\}$ documents, la première couche sert à encoder le contexte d'un mot dans un document d_m représenté sous la forme d'une séquence $\{w_{(m,1)} \dots w_{(m,N_m)}\}$ de N_m mots. Le contexte d'un mot est formalisé avec c , la fenêtre contextuelle à sélectionner. À l'inverse, le contexte d'un tag est formalisé dans la seconde couche qui modélise la présence de tags $\{t_{(m,1)}, t_{(m,2)} \dots t_{(m,T_m)}\}$ dans le document d_m grâce à un vecteur en encodage *One-hot* qui formalise la présence ou non du tag $t_m \in T_m$ dans ce document. La figure 2.5 illustre ce modèle.

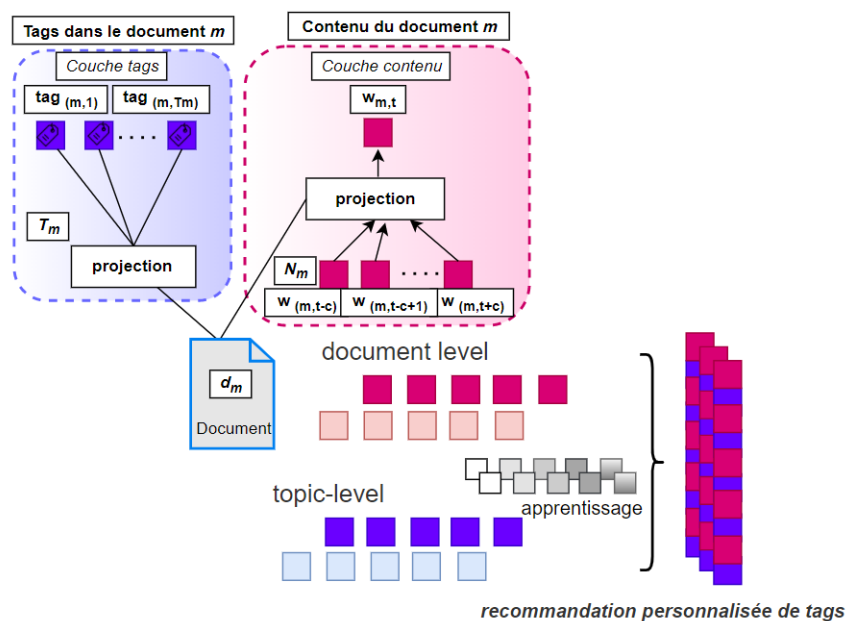


FIGURE 2.5 – Modèle de plongement lexical de tags et de contenu textuel d'un document dans un espace vectoriel commun pour la recommandation personnalisée de tags

De manière formelle la fonction de coût \mathcal{L} à maximiser peut être décrite par l'équation 2.10 :

$$\mathcal{L} = \sum_{d_m \in M} \left\{ \sum_{w_n \in d_m} \{ \log P(w_n | \mathbf{w}_{(n-c, \dots, n+c)}, d_m) \} + \alpha \log P(d_m | \mathbf{w}_{(1, \dots, N_m)}) + \log P(t_{(1, \dots, T_m)} | d_m) + (1 - \alpha) \log P(d_m | t_{(1, \dots, T_m)}) \right\} \quad (2.10)$$

Où α un paramètre permettant la répartition des distributions de la combinaison linéaire. Ainsi, pour aboutir à une prise en compte équitable entre contenu du document d_m et présence de tags dans ce document $\alpha = 0.5$. Dans la figure 2.10, le modèle CBOW est appliqué à la couche du modèle neuronale traitant le contenu des documents. L'architecture Skip-gram peut aussi être utiliser en gardant les mêmes niveaux de performance du modèle [114]. Ce modèle de plongement lexical est inspiré du modèle de Mikolov *et al.*, [123] avec une différence notable due au fait que l'apprentissage des représentations des documents et tags se fait dans les deux directions. Cela veut dire qu'à chaque itération de la descente du gradient, des tâches de classification sont formulées faisant référence à la présence de tags dans le document et à la probabilité d'observer le document pour un ensemble de tags donné. En se basant sur cette architecture, la probabilité d'observer un mot ne dépend pas uniquement du contexte entourant ce mot, mais aussi du document spécifique auquel le mot appartient. De manière plus précise, la probabilité $P(w_n | \mathbf{w}_{(n-c, \dots, n+c)}, d_m)$ est définie :

$$P(w_n | \mathbf{w}_{(n-c, \dots, n+c)}, d_m) = \frac{e^{(\hat{\mathbf{v}}^\top \mathbf{v}_{w_n})}}{\sum_{\mathbf{v}'} e^{(\hat{\mathbf{v}}^\top \mathbf{v}')}} \quad (2.11)$$

Où, $\hat{\mathbf{v}} \propto (\mathbf{v}_{d_m} + \sum_{-c < j < c; j \neq n} \mathbf{v}_{w_j})$. La probabilité d'observer un tag t pour un document d_m donné peut être définie par une fonction de type *softmax* (fonction exponentielle normalisée) exprimée dans l'équation 2.13 :

$$p(t_{(m,i)} | d_m) = \frac{e^{(\mathbf{v}_{t_i}^\top \mathbf{v}_{d_m})}}{\sum_{t'} e^{(\mathbf{v}_{t'}^{\text{top}} \mathbf{v}_{d_m})}} \quad (2.12)$$

De manière similaire, la probabilité d'observer un document d_m pour un tag t donné est aussi définie par une fonction *softmax* :

$$p(d_m | t_i) = \frac{e^{(\mathbf{v}_{t_i}^\top \mathbf{v}_{d_m})}}{\sum_{d'} e^{(\mathbf{v}_{t_i}^\top \mathbf{v}_{d'})}} \quad (2.13)$$

Les représentations vectorielles de documents et de tags sont exploitées conjointement pour l'apprentissage de représentations d'utilisateurs dans le but de satisfaire des tâches de recommandation [114].

Les modèles de représentations distribuées de tags se sont aussi illustrés à travers d'autres approches. En particulier, les approches de *graphe embedding* comme *LINE* [211] qui permettent de saisir certains aspects structurels et topologiques d'un graphe ou autour de chacun de ces sommets ou noeuds *node embedding*. Comme déjà abordé, le graphe embedding peut être perçu comme une fonction de mapping $f : G \rightarrow \{v_0, \dots, v_d\} \in \mathbb{R}^d$, avec $\{v_0, \dots, v_d\}$ un vecteur représentant la topologie d'un graphe $G = (V, E)$. Dans ce contexte, les travaux de Grover *et al.*, [85] ont abordé une extension du

modèle Skip-gram [152] visant à générer des représentations de noeuds à partir d'entités d'un graphe pouvant représenter des relations. Le modèle *node2vec* apprend des représentations vectorielles de noeuds en se basant sur des marches aléatoires configurées à travers des hyper-paramètres. Ces hyper-paramètres sont définis par le nombre et la longueur des marches servant à déterminer le voisinage des noeuds à prendre en considération. Le modèle proposé par *Grover et al.*, [85] a été entraîné grâce un réseau neuronal à couche cachée unique permettant de prédire la probabilité qu'un noeud apparaisse dans une marche aléatoire biaisée qui modifie sa méthode de parcours (en tenant compte d'un parcours en profondeur ou en largeur) au fur et à mesure de la marche. En se basant sur ces éléments et en incluant l'ensemble de tags associé à chaque noeud du graphe, le modèle de représentations distribuées de tags proposé récemment par *Wang et al.*, [225] *Tag2vec* a été entraîné pour apprendre des représentations vectorielles reflétant d'avantages d'informations sémantiques relatives aux tags comme la proximité sociale ou l'appartenance à une communauté.

3.4.4 Représentation contextuelle de tags

L'algorithme de *word embedding* proposé par *Mikolov et al.*, [152] repose sur des réseaux de neurones à deux couches et permet l'apprentissage de représentations vectorielles de mots de telle sorte que les mots qui partagent des contextes similaires soient représentés par des vecteurs à valeurs numériques continues (denses), i.e des vecteurs d'*embeddings*. Cependant, d'autres algorithmes de *word embedding* ont été développés dans le but de mieux capturer la sémantique ainsi que les dépendances syntaxiques entre mots. Ces modèles disposent d'architectures neuronales plus complexes tels que le modèle *ELMo* qui utilise des couches bidirectionnelles LSTM³ [169] permettant d'inclure des informations contextuelles au sein des vecteurs denses pendant la phase d'apprentissage. Récemment, le modèle BERT [61] a comme principale contribution la construction d'*embeddings* de mots en se basant sur une architecture en *Transformers*⁴ permettant l'apprentissage bidirectionnelle d'informations contextuelles. Cela a favorisé l'intégration de contenus textuels bruts au sein des SRIs. Aussi, le contexte que peut véhiculer un tag a pu mieux être abordé. Il a été associé au moyen d'encoder des relations latentes pouvant exister entre tags sous la forme de vecteurs denses. Précisément, ces *embeddings* de tags issues de modèles d'apprentissage neuronal [161, 225] ou extrait à partir de modèles pré-entraînés [114, 132] ont fait l'objet de nombreuses études afin d'être intégrés comme caractéristiques dans des modèles neuronaux de recommandation [127, 254, 30].

4 Discussion

Comme nous l'avons présenté tout au long du chapitre, différentes représentations de tags ont été abordées. Le modèle de représentation en sac de mots a été le plus utilisé à

3. Long short-term memory (LSTM) est une architecture de réseau de neurones artificiels récurrents.

4. L'architecture du Transformer est issue du pattern Encoder-Decoder. La partie encodeur empile une séquence de 6 encodeurs. La partie décodeur dépile une séquence de 6 encodeurs mais prenant chacun, comme entrée supplémentaire, la sortie du 6ème encodeur.

travers la littérature [163, 162, 219]. Il a été surtout d’abord dans un contexte de personnalisation de l’accès à l’information puis s’est étendu à des tâches de recommandations personnalisées. Cependant, l’utilisation de représentations de tags issues de ce type de modèle pose certains problèmes, dûs principalement à l’ambiguïté lexicale comme la polysémie [67]. De telles représentations exigent un traitement lexical rigoureux en amont. Récemment, il a été proposé de représenter les mots à l’aide de vecteurs denses [167, 152]. Dans ce contexte, les travaux sur les méthodes de plongements lexicaux de mots ont directement influencé les travaux abordant des représentations distribuées de tags [225].

Des modèles tels que *word2vec* impliquent un mapping entre le tag et sa fenêtre contextuelle proche. Ce contexte est inexistant dans le cas où le tag serait composé d’un unique token. À l’inverse des contenus textuels traditionnels, les tags peuvent être composés d’un mot unique dépourvu de contexte comme ils peuvent aussi parfois faire référence à une séquence de mots. Toutefois, il est possible d’extraire des représentations à partir de modèles pré-entraînés tels que *word2vec*[149] sur de larges corpus de texte.

Les représentations de mots issues de ces modèles (i.e les modèles pré-entraînés) ont permis un gain de temps considérable au vu du nombre de paramètres à prendre en compte en temps normal lors de la phase d’entraînement. En effet, le paramétrage de la taille de la fenêtre contextuelle (De 5 à 10 selon Mikolov [149]), le nombre de dimensions en sortie (fixée généralement à 300) et le choix de l’algorithme à utiliser (Skip-gram ou CBOW) sont autant de paramètres variables, difficiles à prendre en compte et nécessitent souvent une étude spécifique avant de les mettre en pratique pour des tags d’utilisateurs. Ce cas de figure a été abordé en détail dans les travaux proposés par Niebler *et al.*, [161]. En outre, la problématique des mots rares ou hors-du vocabulaire (*Out Of Vocabulary*) a aussi été soulignée. En effet, l’entraînement de plongements lexicaux exige la présence d’une fréquence minimale de mots du vocabulaire, dans le cas contraire, le modèle attribue une probabilité nulle aux mots qui ne sont pas présents dans l’espace des représentations appris.

Pour lever cette limitation, certains travaux [62] ont simplement remplacé les représentations issues de mots hors-vocabulaire par un vecteur aléatoire. De plus, de récents modèles [140, 58] considèrent des représentations contextuelles de mots, comme *ELMo* [168] et *BERT* [61]. Les représentations issues de ces modèles sont construites à partir d’architectures neuronales spécifiques, en l’occurrence, les *Transformers* qui permettent d’intégrer des informations contextuelles dans les deux sens.

Tout au long de ce mémoire, nous considérons différentes représentations de tags. La plus part de ces représentations sont issues de modèles classiques initialement conçu pour le traitement de contenus textuels dans un espace de modélisation. Cet espace de modélisation est associé à un espace vectoriel comme dans les modèles vectoriels classiques [39, 17], ou à un espace de probabilités, comme dans les modèles de textes probabilistes [238, 12]. Il peut être aussi associé à une fenêtre contextuelle comme dans le modèle *word2vec* [149, 167, 161]. Cependant, la notion de contexte pour un tag peut être abordé autrement que lorsque l’on considère un contenu textuel classique. Même

si l'applicabilité de modèles distribués de mots comme *word2vec* a déjà été prouvé pour des collections de tags [161], d'autres travaux se sont intéressés de plus près à cette notion de contexte en considérant des approches qui abordent le contexte des tags sous la forme de graphes de co-occurrences paramétré grâce à des marches aléatoires et mapés dans un espace vectoriel approprié comme dans *Tag2vec* [225] ou en assimilant la notion de contexte à une proximité de voisinage comme dans *LINE*[211].

En considérant les récentes avancées en matière de TALN, notamment en ce qui concerne les modèles de langues neuronaux contextuels comme *BERT* [61]. Nous avons classifié ces modèles de langues neuronaux d'après la qualité des représentations distribuées qu'ils génèrent. En général, les méthodes neuronales de représentations denses statiques représentent un mot selon un vecteur dense de taille fixe, c'est le cas des modèles *word2vec* et *GloVe*. Les représentations denses générées à partir de ces modèles sont donc statiques. D'autre part, les modèles de langues neuronaux contextuels comme *BERT* génèrent des plongements contextuels en pré-entraînant des *Transformers* [221]. Le modèle Transformer est issu d'un réseau de neurones de type *seq2seq* [205]. Le modèle *seq2seq* dispose d'encodeur qui traite chaque élément d'une séquence d'entrée, capture le contexte et le renvoie au décodeur qui produit la séquence de sortie. Le contexte, l'encodeur et le décodeur ont tendance à être tous deux des RNN [151]. La taille du vecteur de contexte peut être configuré, c'est le nombre d'unités cachées dans l'encoder RNN. Le modèle *BERT* [61] qui est un modèle *seq2seq* [205] a la particularité d'utiliser un mécanisme d'attention [221] sur une fenêtre bi-directionnelle très large ce qui garantit une meilleure prise en compte de l'importance de l'ordre des mots. Les plongements lexicaux issues de ces modèles sont considérés comme étant des représentations denses contextuelles.

5 Conclusion

Dans ce chapitre, nous avons mis en relation l'impact des tags dans les systèmes collaboratifs grâce à l'étude des caractéristiques propres aux tags comme la sémiotique, les systèmes d'intérêts et l'évolution du vocabulaire des folksonomies dans le temps. L'analyse empirique d'historique d'annotation a permis de répondre à certaines problématiques allant de la recherche d'information personnalisée à la recommandation de ressources en ligne.

Pour cela différentes représentations de tags ont été mises en évidence. Celles directement issues des modèles de représentation classiques de RI, vectorielles et probabilistes ont d'abord été mises en exergue. D'autres méthodes de représentation ont été présentées comme celles basées sur les graphes qui abordent l'exploration de tags sémantiquement liés et améliorent en plus l'expérience de l'utilisateur lors de sessions de navigation. Nous avons aussi abordé des techniques de représentations prenant en compte les temporalités des activités d'annotations. Il a été constaté que de telles représentations intégrées à des processus classiques de recommandation permettaient d'améliorer la pertinence des systèmes de recommandation en proposant du contenu actualisé.

Enfin, nous avons détaillé des modèles de représentation issus de récents travaux en TALN. Ces modèles ont suscité plus d'intérêts dans l'état des recherches actuelles, en effet ils permettent d'identifier des mots sémantiquement proches, en se basant sur une fenêtre contextuelle. Les représentations distribuées de tags ont directement été impactées par ces travaux. Cela a conduit au développement de modèles produisant des représentations denses de tags grâce à des approches basées sur l'apprentissage de représentations et/ou en intégrant certaines caractéristiques associées notamment à la modélisation du contexte d'un tag. À ce sujet, il a été établi que le contexte d'un tag pouvait être abordé de différentes manières aussi bien par le biais d'approches basées sur des graphes de co-occurrences de tags, à l'aide de techniques plus élaborées faisant appel à des approches d'apprentissage de représentations au voisinage de tags ou encore à l'aide de modèles neuronaux pré-entraînés.

Nous consacrons le chapitre suivant à une synthèse de l'état de l'art qui se focalise principalement sur l'impact des différentes représentations de tags utilisées dans le but d'inférer des recommandations pertinentes de ressources aux utilisateurs.

MODÈLES DE RECOMMANDATIONS BASÉS SUR LES TAGS

Synthèse de l'état de l'art

Chapitre 3

MODÈLES DE RECOMMANDATION BASÉS SUR LES TAGS

1 Introduction à la recommandation

UNE ressource peut être un document traditionnel comme une page Web, elle peut être issue du Web social. Dans les plates-formes sociales, ces ressources sont le plus souvent associées aux contenus que génèrent les utilisateurs. Ces contenus nommés communément UGCs (*User Generated Contents*), font référence à différents types d'images, vidéos, textes, audio générés en ligne par l'utilisateur. Ils sont directement créés ou modifiés dans les plates-formes d'e-commerces ou de *streaming*. Ces ressources peuvent aussi faire référence à des items que les utilisateurs notent sur une échelle de 0 à 5 comme sur *Netflix* ou de 0 à 10 comme sur *Amazon*¹. Ainsi, les termes items et ressources sont utilisés dans ce manuscrit de manière interchangeable. Les systèmes de recommandations offrent la possibilité de filtrer les différents types de contenus à des usagers en leur proposant des items, similaires à ceux qu'ils ont appréciés, ou susceptibles de les intéresser plus tard. Ces systèmes sont entièrement intégrés à certaines plates-formes et sont adaptés à différents types de contenus : par exemple, *Netflix* propose des recommandations de contenu cinématographique aux utilisateurs à partir du nombre d'utilisateurs et du temps moyens passé sur chaque contenu, *soundcloud*² permet de recommander du contenu audio personnalisé, *Amazon* suggèrent quels types de produits se procurer parmi une sélection qui intéresse potentiellement les utilisateurs. Enfin, *Facebook*³ recommande des liens d'amitié en se basant sur la similarité entre profils d'utilisateurs.

Ces systèmes exploitent, dans leur majorité, des techniques (algorithmes) s'appuyant sur la comparaison entre utilisateurs et certaines caractéristiques de référence. Ces caractéristiques peuvent être issues des contenus mêmes des ressources, on parle alors de *recommandation basée sur le contenu*. Elles peuvent être relatives à l'utilisateur dans son environnement, dans ce cas, il s'agit de *recommandation basée sur le filtrage collaboratif*. Il arrive aussi que ces techniques soient combinées entre elles pour donner des approches dites *hybrides*. Ces techniques suscitent de vifs intérêts au vue du marché que représente l'économie de l'attention, à tel point que *Netflix*⁴ en 2009, offre un prix d'un million de dollars à condition d'améliorer les scores de l'algorithme "Cinematch" déjà en production. Si les systèmes de recommandation ont suscité une telle effervescence, cela vient aussi du fait que l'intégration de tels systèmes aux sites de ventes en ligne permet de booster la diffusion de divers produits et impactent par conséquent les ventes associées aux produits proposés.

1. <https://Amazon.com/>

2. <https://soundcloud.com/>

3. <https://facebook.com/>

4. Le prix de 1 000 000 USD a été décerné à l'équipe Pragmatic Chaos de BellKor qui a surpassé l'algorithme de *Netflix* à hauteur de 10%

Dans une perspective de recherche, les systèmes de recommandation ont été largement étudiés à travers la littérature, différentes approches, techniques et algorithmes ont été exploités. L'élément clé à prendre en compte dans ces systèmes concerne l'analogie entre description des utilisateurs et caractéristiques des ressources qui se traduit principalement par le calcul d'une similarité. De même qu'en recherche d'information où il est question de comparer du contenu textuel entre documents représentés par des vecteurs de termes, dans le domaine de la recommandation, il s'agit de mettre en correspondance des représentations d'utilisateurs et de ressources en utilisant certaines mesures comme la similarité du cosinus ou la mesure de *Pearson* [48]. D'autres mesures comme celle de Spearman [32] permettent d'induire des distances qui renseignent sur la proximité entre utilisateurs et/ou entre ressources.

1.1 Notations et exemple introductif

De manière formelle, en considérant un système non personnalisé S retournant un ensemble de ressources R classées $S(q) \subseteq R$. Si l'on suppose que $\forall q, S(q)$ satisfait une requête q d'un utilisateur, alors le classement suit l'ordre implémenté par le système $[S = r_1 \geq r_2 \dots r_k]$ où $r_i \in R$. En définissant une approche de recommandation personnalisée S' à l'égard d'un utilisateur u , une liste de ressources classées $S'(q, u) \subseteq R$ est proposée en accord avec ses préférences. Formellement cette liste peut être exprimée de la manière suivante : $[S' = r_1 \geq r_2 \dots r_k]$, en définissant la relation d'ordre \geq centrée sur l'utilisateur u par la condition :

$$r_i \geq r_j \iff (u, r_i, q) \geq \text{sim}(u, r_j, q)$$

Dans l'intention de détailler certains concepts fondamentaux associés aux systèmes de recommandation, nous proposons d'illustrer à travers l'exemple introductif suivant l'utilisation des mesures précédemment citées dans un cadre pratique, celui de la prédiction des notes d'utilisateurs. Nous nous baserons sur les notations du tableau 3.1 tout au long de ce chapitre.

Soit une base de données comportant les avis d'utilisateurs d'une plate-forme collaborative spécialisée dans la distribution et l'exploitation d'œuvres cinématographiques. Ces avis sont exprimés sous forme de notes sur une échelle de 0 à 5 cf., figure 3.2. La matrice M représente les notes des utilisateurs pour des ressources (items) exprimées $r_{(i,j)}$ avec $i \in \{1 \dots n\}$ et $j \in \{1 \dots m\}$. La mesure $\text{corr}_{\text{pearson}}(u_1, u_2)$ qui est présentée dans l'équation 3.1, exprime la similarité entre l'utilisateur u_1 et u_2 en fixant $\overline{r_{(u_1)}}$ comme étant la moyenne des notes attribuée par l'utilisateur u_1 et $r_{(u_1,i)}$ la note attribuée à l'item i .

$$\text{corr}_{\text{pearson}}(u_1, u_2) = \frac{\sum_{i \in I} (r_{(u_1,i)} - \overline{r_{(u_1)}})(r_{(u_2,i)} - \overline{r_{(u_2)}})}{\sqrt{\sum_{i \in I} (r_{(u_1,i)} - \overline{r_{(u_1)}})^2} \sqrt{\sum_{i \in I} (r_{(u_2,i)} - \overline{r_{(u_2)}})^2}} \quad (3.1)$$

La mesure de *Pearson* a longtemps été une mesure standard très utilisée dans le domaine de la recommandation, elle permet d'étudier la relation entre deux variables quantitatives [201]. Cette mesure statistique exprime une relation de corrélation qui prend sa valeur de -1 à $+1$. De telle façon, il est possible de proposer une formule permettant de prédire la note de l'utilisateur u_1 à l'égard de la ressource (item) i_4 en

Notations	Détails
$U = \{u_1 \dots u_n\}$	Ensemble des utilisateurs
$I = \{i_1 \dots i_m\}$	Ensemble d'items
T	Ensemble de tags $t_l \in T$
M	Matrice $n \times m$ des notes d'utilisateurs $r_{(u,i)}$
M_{UI}	Matrice utilisateur-item
M_{UT}	Matrice utilisateur-tag
$r_{(u,i)}$	Note attribuée par l'utilisateur u à l'item i , $r \in R$
\mathcal{R}	Ensemble des interactions (notes des utilisateurs ou <i>rating</i>)
$tf_{i(m)}(t_l)$	Nombre de fois que t_l est utilisé pour annoter $i_{(m)}$
p_u	Vecteur latent représentant l'utilisateur u
q_i	Vecteur latent représentant l'item i
$f_{(u)}$	Vecteur de caractéristiques utilisateur u
$f_{(i)}$	Vecteur de caractéristiques utilisateur i
P_u	Matrice de facteurs latents associée aux utilisateurs
Q_i	Matrice de facteurs latents associée aux items
b_u	Moyenne des notes donnée par l'utilisateur u
b_i	Moyenne des notes attribuée à l'item i

TABLE 3.1 – Tableau de Notations









		Items			
		 i_1	 i_2	 i_3	 i_4
Utilisateurs	 u_1	3	2	1	?
	 u_2	4	1	2	5
	 u_3	4	1	3	1
	 u_4	2	1	1	5

FIGURE 3.1 – Matrice $n \times m$ matrice des notes d'utilisateurs $r_{i,j}$

se basant sur la proximité relative des plus proches voisins d'utilisateurs. En notant cet ensemble d'utilisateurs N avec $u_n \in N$, la mesure peut être calculée de la manière suivante en se basant sur les plus proches voisins de u_1 qui sont d'après la mesure de *Pearson*, u_2 et u_4 ($corr_{pearson}(u_2, u_4) = 0.907$) :

$$prediction(u_1, i_4) = \overline{r_{(u_1)}} + \frac{\sum_{u_n \in N} corr_{pearson}(u_1, u_n)(r_{(u_n, i_4)} - \overline{r_{(u_n)}})}{\sum_{i \in N} corr_{pearson}(u_1, u_n)} \quad (3.2)$$

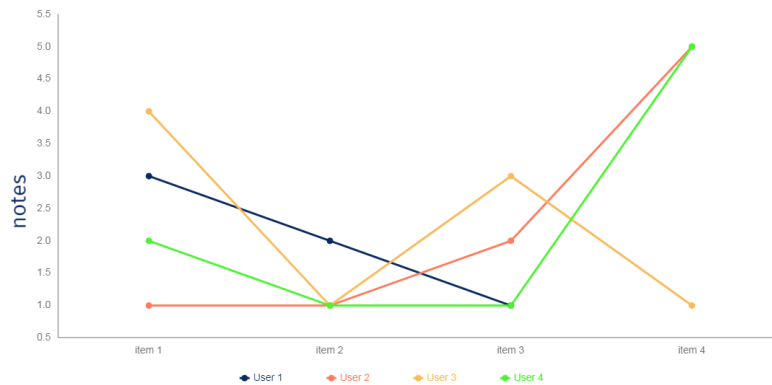


FIGURE 3.2 – Comparaison entre utilisateurs selon la note qu'ils attribuent pour un même ensemble d'items.

Comme le montre la figure 3.2 à travers le comportement de certains utilisateurs, il est fréquent d'observer que certains d'entre eux tendent à attribuer de manière récurrente des notes élevées. D'autres n'attribuent pratiquement jamais de notes élevées. La mesure de *Pearson* exploite les moyennes des notes d'utilisateurs afin de les rendre comparables. Cela nous permet de mettre en évidence une certaine proximité entre utilisateurs, par exemple, entre u_2 et u_4 ou un comportement dissimilaire comme entre u_2 et u_3 .

Ce type de mesures est à la base de certains systèmes de recommandation tant elles ont été exploitées aussi bien dans des processus de filtrage collaboratif qui ne tiennent pas forcément compte de la nature des items que dans des processus de filtrage sur le contenu où au contraire les caractéristiques des items et leur correspondance avec le profil d'utilisateurs ou son voisinage impactent la qualité des recommandations. En outre, nous détaillons dans ce qui suit comment ont été intégrées au sein de ces mesures certaines propriétés issues des tags et qui ont la caractéristique de décrire aussi bien les utilisateurs que les ressources.

2 Folksonomies pour la recommandation

Dans cette section, les modèles de recommandation que nous détaillons formalisent certaines approches issues du filtrage collaboratif. Ces modèles [207, 59, 219] sont initialement basés sur la notion de voisinage et exploitent notamment les tags pour déterminer les voisins les plus proches d'utilisateurs ou de ressources (les plus similaires).

Aussi, il a été prouvé que la prise en compte de tags améliore en termes de performances les modèles de recommandation [118]. Ces modèles [162, 163, 219] ont intégré les tags comme source additionnelle pour calculer la similarité entre utilisateurs et/ou ressources d'un système de recommandation. Aussi, dans de telles approches, un utilisateur et son profil peut être représenté par l'ensemble des items (ressources) qu'il a annoté et l'ensemble des tags qu'il a utilisés. On peut également associer à ces tags le moment où ils ont été assignés. Cette donnée temporelle permet de saisir l'évolution

des centres d'intérêts des utilisateurs.

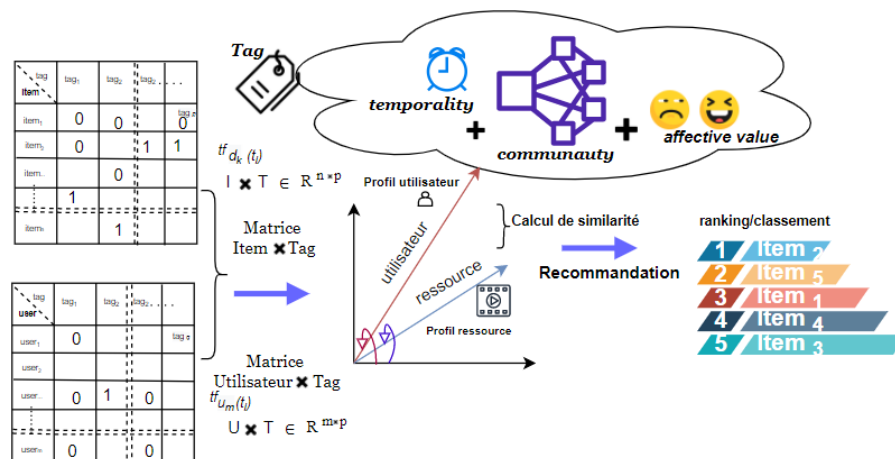


FIGURE 3.3 – Cette figure illustre les différentes dimensions associées aux tags (dimension temporelle, affective, communautaire). Elle souligne la manière dont ces dimensions peuvent être exploitées lors de processus de recherche, notamment, pour le classement des résultats de recherche et de recommandations personnalisées, calculées entre représentation d'utilisateurs et représentation de ressources.

Dans ce qui suit, nous avons aussi fait état de certaines approches basées sur l'apprentissage automatique pour la recommandation [148, 51]. Ces modèles d'apprentissage supervisés sont principalement issus de techniques de factorisation de matrices. Ces techniques s'inscrivent parmi les méthodes de filtrage collaboratif mises en service par les systèmes de recommandation en ligne. D'un point de vue pratique, la factorisation de matrice se base sur l'apprentissage de caractéristiques latentes d'utilisateurs et d'items qui sont exploités pour approximer les valeurs des notes des utilisateurs. Si u_i et v_j sont respectivement les vecteurs latents d'utilisateurs et d'items alors le produit scalaire de ces deux vecteurs peut approximer la note $r_{i,j}$, formellement, $r_{i,j} \approx u_i \cdot v_j$. Ces modèles d'apprentissage pour la recommandation [87, 114] sont capables de généraliser, à partir de données d'apprentissage incluant un ensemble de tags, certaines caractéristiques propres aux activités sociales d'utilisateurs de folksonomies. Récemment, des modèles d'apprentissage profond pour la recommandation ont rendu possible la modélisation de relations non-linéaires entre utilisateurs et items. Certains de ces modèles [235, 138] ont pu apprendre à partir de données d'annotations des représentations abstraites d'utilisateurs et d'items. Ces représentations sont plongées dans un espace de représentation dense commun selon une architecture particulière de réseaux de neurones artificiels.

La synthèse présentée ci-après est principalement axée sur la prise en compte des contenus textuels notamment ceux associés aux tags dans les systèmes de recommandation. Toutefois, nous avons remarqué que les modèles de recommandation classiques se sont limités à l'exploitation du modèle BoW pour intégrer les tags sous différentes formes. Les représentations de tags dans ces modèles sont formalisées sous forme de vecteurs en encodage *One-hot* [235, 138]. L'utilisation de cet encodage engendre le manque de contexte, redondance et polysémie, dû au fait que l'espace des représenta-

tions des tags dans le modèle *BOW* est sémantiquement limité. Récemment, les modèles de représentations issues d’architectures neuronales plus élaborées ont eu pour objectif d’intégrer explicitement le vocabulaire des tags en favorisant l’apprentissage de représentations du triplet $\{\text{utilisateur}, \text{ressource}, \text{tag}\}$ pour des tâches de recommandation [254, 114]. Certaines d’entre eux [87, 92, 132] ont été développés dans le but d’enrichir sémantiquement l’espace des représentations des utilisateurs et d’items et sont parvenus à représenter explicitement les tags au moyen de modèles de langue neuronaux pré-entraînés tel que *word2vec* ou encore *BERT*.

Dans ce qui suit, nous présentons les approches de recommandation basées sur le contenu en abordons le rapprochement entre un item/utilisateur et un cluster formé par un groupement de tags. Ensuite, nous passons en revue les approches basées sur le filtrage collaboratif exploitant exclusivement les annotations ou tags pour des tâches de recommandation. Par la suite, nous présentons les approches basées sur l’apprentissage ainsi que certaines approches hybrides de recommandation. Le reste de la synthèse de l’état de l’art que nous proposons est consacré à l’étude des modèles neuronaux de recommandation que nous classons en deux catégories, les modèles *end-to-end* basés sur le Filtrage collaboratif et les modèles neuronaux basés sur l’apprentissage de représentations.

2.1 Approches basées sur le contenu

Les approches basées sur le contenu se basent dans leur majorité sur des techniques de RI exploitant la similarité textuelle entre la représentation d’un item et la représentation des préférences d’un utilisateur [156, 229]. La tâche traitée par ces approches consiste à déterminer quels items parmi l’ensemble des items coïncident le mieux avec les préférences de l’utilisateur. La manière la plus simple de décrire cet ensemble d’items est d’avoir une liste explicite des caractéristiques de chaque item, on parle alors de profils d’items. De même pour les préférences de l’utilisateur que l’on associe à un profil utilisateur, exprimé sous forme d’une liste d’intérêts. La mise en correspondance entre les caractéristiques des items (profils d’items) et le profil de l’utilisateur peut être mesurée en exploitant différentes techniques basées sur la similarité dans un espace vectoriel [34, 38, 162]. Comme le montre la figure 3.4, les tags sont aussi bien exploités pour enrichir l’espace des représentations décrivant l’item, que pour modéliser l’activité d’annotation de l’utilisateur. L’utilisateur interagit souvent avec le système en sélectionnant un ou plusieurs tags $t_l \in T$ que l’on peut aussi considéré comme une requête. La similarité $S(t_l, i)$ entre un tag t_l et un item i , est calculée entre le nombre de fois que l’item est annoté, soit $tf_{t(i)}$, et $tf_{(t_l, i)}$, le nombre de fois que le tag t_l est utilisé pour annoter l’item i . Cette similarité est décrite comme suit avec T l’ensemble des tags du vocabulaire :

$$S(t_l, i) = \frac{tf_{t(i)}}{\sqrt{\sum_{t_l \in T} (tf_{(t_l, i)})^2}} \quad (3.3)$$

Pour saisir de manière plus pertinente les intérêts des utilisateurs, des approches plus élaborées [209, 199, 78] se sont illustrées dans les systèmes de tagging en exploitant des techniques de regroupement ou *clustering*. Plus précisément, ces travaux [209, 200, 78]

s'intéressent au moyen de calculer l'intérêt d'un utilisateur u pour un item i à partir d'ensembles de tags groupés associés à des clusters d'intérêts $\{c_1, c_2 \dots c_n\} \in C$. Le regroupement en clusters de tags se basent sur des représentations de tags pondérés sur l'ensemble des ressources en utilisant des schémas de pondération comme *tf.idf*.

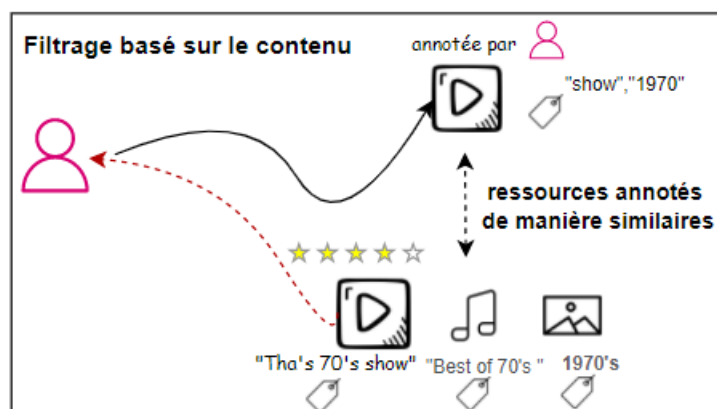


FIGURE 3.4 – Filtrage de contenus dans les systèmes de tagging

$$I(u, i) = \sum_{c \in C} w_{(u,c)} \times w_{(i,c)} \quad (3.4)$$

L'intérêt, $I(u, i)$, de l'utilisateur u par rapport à l'item i est défini par le produit $w_{(u,c)} \times w_{(i,c)}$ qui décrit de manière jointe l'intérêt d'un utilisateur et d'un item par rapport à des groupes de tags (*clusters*) qui caractérisent différents sujets d'intérêts. Cette mesure d'intérêt est calculée selon l'équation 3.4 sur l'ensemble des clusters. $w_{(u,c)}$ exprime le rapport entre le nombre de fois que l'utilisateur u annote un item avec un tag appartenant à ce cluster et le nombre total de tags de cet utilisateur. De manière similaire, $w_{(i,c)}$ détermine le rapprochement entre un item et un cluster décrit comme le rapport entre le nombre de fois que l'item i a été annoté avec un tag appartenant à ce cluster et le nombre total de fois que l'item a été annoté. Ainsi, il est possible de déduire une similarité personnalisée basée sur un ensemble de tags et calculée pour chaque ressource en multipliant la similarité $S(t_l, i)$ de l'équation 3.3 par la pertinence de la ressource pour un utilisateur équation 3.4. Nous désignons cette similarité par :

$$S'(u, t_l, i) = S(t_l, i) \times I(u, i) \quad (3.5)$$

Intuitivement, si les intérêts d'un utilisateur correspondent au sujet d'intérêt d'une ressource, la valeur de $I(u, i)$ sera élevée. Pour un utilisateur donnée u , la similarité $S'(u, t_l, i)$ est calculée pour chaque item et les meilleurs items sont retournés à l'utilisateur comme recommandations. Les auteurs de *Shepisten et al.*, [199] ont expérimenté différentes méthodes de *clustering* et ont montré que le clustering agglomératif conduit à de meilleures performances.

2.2 Approches basées sur le filtrage collaboratif

Ce sont des extensions des modèles de filtrage collaboratif classiques visant à prédire les notes d'utilisateurs à partir d'historique d'interactions passées. La figure 3.5 illustre le filtrage collaboratif dans les systèmes de tagging. Ces modèles ont été détaillés par Ben Schafer *et al.*, [191]. L'idée principale véhiculée par ces algorithmes est d'extraire des caractéristiques à partir de représentations d'utilisateurs similaires et/ou voisins puis de les mettre en œuvre dans un processus d'aide à la décision. Ces représentations servent de base de connaissances au système de recommandation qui proposera un contenu qui aura de grandes chances d'intéresser un utilisateur donné vu qu'il aura intéressé auparavant un utilisateur dont les notes attribuées ont été similaires pour un même corpus d'items. Le système peut aussi exploiter les scores de similarité des utilisateurs/items qui se situent à un certain voisinage [218]. Les tags sont introduits comme information supplémentaire permettant d'estimer quantitativement ce voisinage [240]. Certains modèles de recommandation exploitent ces mêmes scores pour réduire la taille du voisinage à prendre en considération dans le cas où le volume de données des voisins est important.

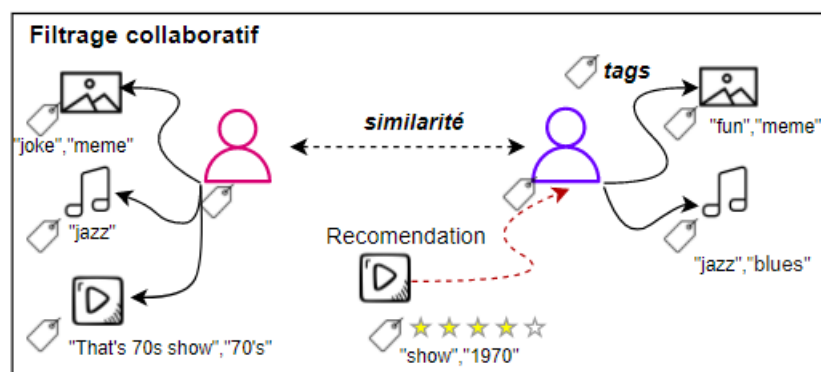


FIGURE 3.5 – Filtrage collaboratif dans les systèmes de tagging

Les travaux de recherche décrits dans [191, 218, 240] ont montré que la prise en compte des tags permet la découverte de similarités socialement plus pertinentes entre utilisateurs et ressources [248]. En effet, nous avons vu que dans ce qui pourrait être une méthode de recommandation classique dans l'exemple de la sous-section 1.1, que la mesure de similarité utilisé pour prédire la note d'un utilisateur spécifique se base exclusivement sur les notes attribuées antérieurement aux items. Cette similarité peut être étendue en prenant compte les tags similaires entre utilisateurs (ou items). En conséquence, les tags peuvent être considérés au même titre que des items supplémentaires dans la matrice utilisateur-item représentant l'attribution de notes à des items par des utilisateurs. Cependant les représentations de tags au sein de ces matrices d'interaction sont souvent implicites (1 si l'utilisateur a annoté 0 sinon). Cela a conduit au développement d'autres approches qui ont considéré les représentations d'utilisateurs et d'items à travers des profils exploitant des fréquences de tags normalisées [33, 42, 41, 163] dans un espace vectoriel. Outre la fréquence, d'autres propriétés peuvent être associées aux tags, par exemple, l'activité de l'utilisateur ou la temporalité associée aux tags qui a

conduit à l'amélioration des méthodes de calcul de similarité entre profils ce qui a amélioré les scores de pertinence du classement des ressources selon l'ordre préférentiel des utilisateurs associés à ces profils. Aussi, les approches de filtrage collaboratif se sont illustrées à travers certains travaux [189, 154, 156] qui exploitent la notion de voisinage en modélisant de différentes manières les profils d'utilisateurs et d'items. Nous détaillons ces travaux de recherche ci-après.

2.2.1 Approches basées sur le voisinage de représentations tags

Comme illustré dans l'exemple de la section 1, il est possible de déduire les notes d'utilisateurs par rapport à des items en se basant sur le comportement d'utilisateurs voisins [154]. Pour résoudre les difficultés liées à la taille des données à traiter, trop importante parfois, pour prendre en compte tous les voisins, des techniques de réduction de la taille du voisinage ont été proposées d'une part en fixant un seuil de similarité minimum entre utilisateurs et d'autre part en réduisant le nombre de voisins aux k plus proches voisins [189]. En conséquent, pour des utilisateurs ayant approximativement notés certains items de la même manière, il est possible de prédire la note de l'utilisateur en fonction de la moyenne des notes de ses voisins suivant l'équation :

$$\hat{r}_{ui} = \frac{1}{|N_i(u)|} \cdot \sum_{v \in N_i(u)} r_{vi} \quad (3.6)$$

Où, $N_i(u)$ est l'ensemble des k -plus proches voisins de l'utilisateur u ayant annoté un item i et r_{vi} la note de l'item i attribuée par l'utilisateur voisin v .

Parmi les approches qui se sont basées sur les voisinages intégrant les tags d'utilisateurs, certaines [189, 156, 154, 193] ont proposé d'inclure des représentations de tags dans le calcul de similarité permettant d'identifier les k - voisins similaires à un utilisateur [156], en partant du principe que deux utilisateurs sont similaires si ils ont annoté des ressources similaires. Cette similarité est aussi calculée en fonction du nombre de tags éventuellement pondérés [156] partagées entre utilisateurs (ou items) représentées le plus souvent sous la forme de vecteurs de tags. D'autres approches [193] ont modélisé les interactions provenant d'annotations sociales entre utilisateurs et items sous la forme de graphe utilisateurs-items liés par autant d'arêtes que de tags entre deux nœuds du graphe. Les auteurs ont identifié les k voisins grâce à une fonction de similarité basée sur le cosinus qui considère la fréquence des tags en commun entre paire de nœuds d'utilisateurs et de ressources en plus du nombre de nœuds en communs partageant des degrés similaires.

Les approches basées sur le voisinage ont formalisé le concept associé à la représentation des intérêts d'un utilisateur à l'aide d'une liste de tags ou plus simplement sous la forme de profil. Dans ce qui suit, nous abordons les approches de recommandation qui intègrent ces profils sous différentes formes. Ces profils d'utilisateurs ou de ressources sont à la base de plusieurs modèles de recommandation. Nous synthétisons dans la sous-section suivante ces modèles ainsi que les différentes techniques de représentation de tags associées à ces modèles.

2.2.2 Exploitation de profils folksonomiques pour le filtrage collaboratif

Nous avons classé les méthodes de profilage exploitées par le filtrage collaboratif pour la recommandation en quatre catégories respectives. Celles représentant les profils sous leurs formes vectorielles [233, 162, 219, 33, 38, 116], celles tirant profit de la dimension communautaire notamment grâce à des méthodes de *clustering* [200, 14, 237, 20, 207], les méthodes représentant les profils grâce à des approches temporelles [28, 109, 2, 20] et enfin les profils se basant sur la dimension sentimentale qui met en œuvre la sémantique des tags dans une perspective d'analyse de sentiment [145, 231, 35]. De manière générale, ces caractéristiques s'ajoutent à l'espace de modélisation du profil de l'utilisateur et de la ressource et peuvent être défini comme étant des dimensions.

- **Profils vectoriels** : Dans ces approches, les profils sont représentés sous la forme de vecteurs de tags pondérés en utilisant des fréquences globales, locales ou une combinaison des deux. La spécificité de ces profils réside principalement dans la manière de pondérer les tags. Une majorité des travaux se basent sur le schéma *tf.idf* où *tf* représente la fréquence du tag au niveau de l'utilisateur (le nombre de fois que l'utilisateur a utilisé le tag), au niveau de la ressource (le nombre de fois que la ressource a été annotée par le tag). De même pour *idf*, il peut être calculé au niveau des utilisateurs, on parle alors de *iuf* [233], ou au niveau de la ressource on se réfère alors à *irf* [63]. Certains auteurs [233, 162] ont suggéré des schémas de pondération plus élaborés comme *BM25*. En s'inspirant de ces travaux, Vallet et al., [219] présentent une approche de recommandation adaptée aux folkonomies d'utilisateurs avec comme schéma de pondération *tf-idf*. Celle-ci est similaire à celle de Xu et al., [233]. Cependant, Vallet et al., éliminent le facteur de normalisation de la longueur du document. Une telle démarche dans le modèle vectoriel vise à pénaliser le score des documents qui contiennent un grand nombre d'informations et qui de ce fait sont susceptibles de correspondre à une requête par hasard. Du point de vue des auteurs, le fait qu'un grand nombre de tags soient connexes est lié à la popularité des ressources. Selon Cai et al., [33] les schémas *tf*, *tf-idf* ou *BM25* utilisés pour pondérer des tags, se révèlent être insuffisant pour indiquer à quel point un utilisateur présente de l'intérêt pour les tags décrivant son profil. De plus, des utilisateurs actifs influencent les fréquences avec lesquelles les ressources ont été annotées, les rendant plus populaires du fait de la tendance disproportionnée des utilisateurs actifs à annoter ces ressources. Cela affecte de ce fait le classement des ressources retournées.
- **Profils basés sur le *clustering*** : Ces approches consistent à rassembler les tags similaires dans des clusters qui reflètent les préférences de l'utilisateur pour différents topics (thématiques). Les modèles de recommandation personnalisés exploitant ce type d'approches ont été abordés dans de nombreux travaux [200, 14, 237, 20, 207]. La question centrale réside dans la manière avec laquelle la similarité entre différents tags est calculée. Shepisten et al., [199, 78] utilisent un algorithme de *clustering* hiérarchique sur l'ensemble des tags associés à un utilisateur. Ils exploitent les clusters générés afin de fournir des recommandations personnalisées. Ils calculent l'intérêt des utilisateurs pour chaque cluster comme étant le rapport entre le nombre de fois qu'un utilisateur annoté une ressource avec un tag se trou-

vant dans un cluster et le nombre total d'annotations par utilisateur. *Begelman et al.*, [14] se basent sur le nombre de tags co-occurent décrivant une même ressource. Un seuil est utilisé afin de déterminer si la co-occurrence des tags indiquent une similarité. *Begelman et al.*, utilisent aussi certains algorithmes de *clustering* pour retrouver des tags sémantiquement liés, ils mettent en évidence que le *clustering* améliore l'expérience utilisateur lors de la navigation dans un immense espace de ressources. *Yeung et al.*, [237] proposent une méthode pour construire des profils folksonomiques en prenant compte la diversité des intérêts des utilisateurs. Ils exploitent un algorithme glouton de *clustering* à partir de graphes représentant les ressources d'un utilisateur. Dans leur analyse sur les données personnelles extraites de *del.icio.us*⁵ et *Flickr*⁶, ils utilisent le nombre de tags communs entre deux ressources comme schéma de pondération avant de construire des profils d'utilisateurs. L'étude révèle que la majorité des utilisateurs possèdent des intérêts multiples.

- **Profils temporels** : La motivation principale de ce type d'approche réside dans le fait que les intérêts des utilisateurs évoluent avec le temps. Certaines approches [2, 248, 109] modélisent cette évolution d'intérêt comme un déclin progressif signe d'un désintéressement temporaire ou comme un déclin brusque indiquant un changement d'intérêt [248]. Les travaux décrivant ces approches [2, 248, 109] exploitent des fonctions temporelles qui simulent l'atténuation des intérêts utilisateurs au cours du temps. Dans *Zheng et al.*, [248], les auteurs développent une fonction exponentielle adaptative qui simule l'atténuation graduelle des intérêts utilisateurs. *Zheng et al.*, combinent deux mesures de poids, la première concerne la ressource par rapport au tag qui la décrit, la deuxième fait état de la valeur temporelle attribuée à la ressource. *Abel et al.*, [2] analysent les activités individuelles sur *Twitter*⁷. Ils comparent différentes stratégies pour la création de profils utilisateurs. Ils étudient comment ces profils changent au fil du temps et comment la dynamique temporelle influence l'exactitude du processus de personnalisation. Les auteurs considèrent deux types de profil utilisateur : les utilisateurs actifs qui interagissent pendant une longue période de temps et ceux qui le sont moins et interagissent de manière plus irrégulière. La plupart des approches exploitent aussi des facteurs temporels après la construction du profil utilisateur afin de décrire son évolution dans le temps. Dans leurs études visant à évaluer la manière dont les profils à court et à long terme interagissent, *Bennett et al.*, [20] définissent dans un premier temps, le profil à court terme comme étant celui de la session en cours, tandis que le profil à long terme est construit en fonction de plusieurs sessions de recherche. Ensuite [20] ils étudient comment chacun de ces profils peut être utilisé isolément ou en combinaison pour contribuer de manière optimale aux gains de pertinence lors d'une recherche personnalisée. Le profil à court terme n'intègre que les interactions extraites de la session en cours, mais ne prend pas en compte les intérêts de celui-ci à long terme. De plus, le profil à long terme met en avant d'anciens intérêts sans pour autant tenir compte des besoins actuels de l'utilisateur. Afin de remédier à ces limites, *Kacem et al.*, [109] proposent de considérer les

5. <https://del.icio.us/>

6. <https://www.flickr.com>

7. <https://twitter.com/>

deux types de profil (court et long terme) dans un même *Framework* et d'ajuster le poids des termes constituant les profils en fonction de sa date d'apparition. Le profil utilisateur est ainsi représenté à l'aide d'un vecteur de termes. Pour les utilisateurs pas très actifs sur les services que proposent les réseaux sociaux, le profil à court terme peut éliminer des résultats pertinents qui sont davantage liés à leurs intérêts personnels. Cela est dû au fait que leurs activités sociales sont peu nombreuses et éparpillées dans le temps. De plus, pour les utilisateurs très actifs, l'agrégation des activités récentes sans pour autant ignorer les anciens intérêts serait très intéressant à mettre en œuvre car ce type de profil évolue généralement dans le temps. De manière générale, l'importance de chaque terme est ajustée en fonction de sa date d'apparition. Cette mesure de "fraîcheur" de terme est calculée grâce à une fonction à noyau gaussien, qui permet le calcul d'une distance temporelle entre la date courante et la date où apparaît le terme (l'interaction sociale).

- **Profils affectifs** : Certains auteurs exploitent la sémantique des tags dans une perspective d'analyse de sentiment. Dans *Cambria et al.*, [35], les auteurs construisent un dictionnaire aux multiples niveaux de concepts, afin de déterminer les rapports affectifs et être en mesure de les catégoriser. Dans *Xie et al.*, [231], les auteurs introduisent une approche visant à incorporer la valeur affective attribuée pour chaque tag dans le but de construire des profils utilisateurs ou de ressources, ils utilisent une fonction qui transforme la valeur de pertinence du tag d'un espace de tags vers un espace de sentiments. Les profils utilisateurs sont d'abord représentés sous la forme de matrices représentant le degré de préférence de chaque utilisateur par rapport à un tag. Ce degré de préférence est dérivé du paradigme de normalisation de la fréquence de termes *nTF* [33]. Cependant, d'autres paradigmes peuvent être utilisés comme (*tf-idf*) [145] et *BM25* [219] pour le calcul du degré de pertinence du tag. Afin de mapper l'espace de tags \mathbf{W} à l'espace de sentiment \mathbf{V} , les auteurs les combinent en employant des opérations de multiplications internes entre \mathbf{W} et le dictionnaire des sentiments associé (extrait à partir de *SenticNet*⁸ par exemple). Le classement de ressources contient à la fois la pertinence du contenu de la ressource et la pertinence de l'intérêt de l'utilisateur. Plus précisément, la pertinence des ressources est mesurée par la similarité entre les requêtes (ensemble de tags) et les profils de ressources. Comme il y a deux types de profils de ressources : les profils de ressources basés sur les tags et des profils de ressources basés sur des sentiments, ils sont combinés par agrégation. À l'issue de cette agrégation, deux scores de pertinences seront calculés et serviront de mesure pour le classement des ressources : celle de la requête par rapport à la ressource et celle de la ressource par rapport à l'utilisateur.

2.3 Approches de Filtrage collaboratif basées sur l'apprentissage automatique

Une tâche de recommandation est souvent associée à un modèle permettant la prédiction de notes attribuées à des items spécifiques. Pour cela, le modèle en question doit être en mesure d'exploiter des représentations d'utilisateurs et d'items. Dans le cadre de notre thèse, ces représentations sont issues d'annotations sociales. Ainsi, contraire-

8. <https://sentic.net/>

ment aux approches de recommandation citées précédemment, les approches basées sur l'apprentissage automatique considèrent différentes caractéristiques pour construire un modèle d'apprentissage. Ces caractéristiques qui sont souvent extraites manuellement se fondent principalement sur l'activité d'annotation des utilisateurs et sont exploitées conjointement à l'historique des attributions antérieurs des notes d'items.

Les interactions utilisateur-item représentées sous forme matricielle constituent une représentation usuelle de l'historique des notes dans ce type de modèle. De cette façon, les modèles de factorisation matricielle [111, 3] représentent les utilisateurs et les items comme des vecteurs de caractéristiques de même taille dans un espace de représentation latent. Les interactions sont modélisées comme des produits internes dans cet espace. Pour chaque utilisateur u et item i un vecteur latent de valeur réels p_u et q_i respectivement tel que le produit scalaire de ces deux vecteurs correspondent à la note prédite soit $r_{u,i} = q_i^T p_u$. Le problème posé s'apparente à l'estimation des scores de notes d'utilisateurs non-observées en minimisant l'erreur carrée entre les produits scalaires des vecteurs latents p_u et q_i et les notes à prédire. Formellement :

$$\min(p, q) = \sum_{(u,i \in k)} (r_{u,i} - q_i^T p_u)^2 \quad (3.7)$$

Le modèle de recommandation décrit doit être capable de généraliser un ensemble d'apprentissage, pour y parvenir un facteur de régularisation ainsi que des biais sont inclus au modèle qui est minimisé à l'aide de l'algorithme de descente de gradient stochastique[26]. Le facteur de régularisation λ est multiplié par la somme carrée des vecteurs d'utilisateurs et d'items, ainsi l'équation 3.7 peut être redéfinie avec le paramètre de régularisation λ de la manière suivante :

$$\min(p, q) = \sum_{(u,i \in k)} (r_{u,i} - q_i^T p_u)^2 + \lambda(\|q_i\|^2 \|p_u\|^2) \quad (3.8)$$

Dans ces circonstances, si une note basse est attribuée par un utilisateur à un item sans autre évaluation observée de cet utilisateur alors l'algorithme de descente de gradient stochastique minimisera l'erreur en attribuant à q_i une valeur élevée. Cela entraîne généralement une prédiction des notes de cet utilisateur à la baisse, ce qui n'est pas cohérent. Le fait d'élever au carré les valeurs des vecteurs latents de l'équation 3.8 permet de minimiser le risque que de telles situations se produisent. De plus, afin de réduire l'écart entre la valeur prédite et la valeur réelle, certaines caractéristiques de l'ensemble des données peuvent y être ajouter. Ces caractéristiques concernent l'ensemble des paires utilisateur/item (u, i) et prennent la forme de biais b_i et b_u représentant respectivement la moyenne des notes attribuée à l'item i , et la moyenne des notes données par l'utilisateur u comme mentionné dans l'équation 3.10. Enfin un dernier paramètre permettant la prise en compte de la note moyenne attribuée à tous les items notée μ qui permet de redéfinir une nouvelle fois l'équation 3.7 comme suivant :

$$\hat{r}_{ui} = q_i^T p_u + \mu + b_u + b_i \quad (3.9)$$

L'équation 3.8 finale à minimiser peut s'écrire alors sous la forme :

$$\min(p, q, b_u, b_i) = \sum_{(u,i \in k)} (r_{u,i} - q_i^T p_u - \mu - b_i - b_u)^2 + \lambda(\|q_i\|^2 \|p_u\|^2 + b_u^2 + b_i^2) \quad (3.10)$$

La propriété principale des modèles d'apprentissage pour la recommandation est que les caractéristiques des données ont un impact direct sur les modèles prédictifs et leurs résultats. En d'autres termes, les données d'apprentissage doivent être décrites de manière suffisamment cohérente pour que le modèle prédictif soit en mesure de généraliser et d'être ainsi performant.

Après avoir explicité le principe de l'apprentissage automatique appliqué à des approches de filtrage collaboratif de base et qui implique la prédiction d'une interaction simple (une note) entre un utilisateur et un item, nous présentons dans ce qui suit certains travaux d'apprentissage automatique ayant intégré des annotations sociales aux données d'apprentissage dans le but de résoudre certaines tâches de recommandation.

2.3.1 *Exploitation des annotations sociales dans des approches de filtrage collaboratif basées sur l'apprentissage automatique*

L'idée de prendre en compte des informations (données) additionnelles telles que l'historique de navigation, texte ou tags en complément des avis d'utilisateurs a conduit indéniablement à l'amélioration des systèmes de recommandation [16]. En effet, intuitivement, ces données additionnelles permettent d'enrichir l'espace des caractéristiques associées aux données d'apprentissage en favorisant l'apprentissage de représentations jointes de certaines caractéristiques.

En intégrant les annotations sociales parmi les données d'apprentissage, certains travaux [148, 76] ont exploité des tags comme caractéristiques jointes à la matrice utilisateur-item. Ces modèles exploitent des matrices de caractéristiques dans le but de saisir les préférences que peuvent avoir certains utilisateurs à annoter avec certains tags spécifiques et de surcroît généraliser les mécanismes latents ayant favorisé la recommandation d'un ensemble spécifique d'items. Les travaux de *Chen et al.*, [51] ont exploré l'exploitation de matrices de caractéristiques groupées en catégories sociales ou textuelles. Les caractéristiques sociales ont été extraites à partir de valeurs qui mettent en évidence l'action commune d'annoter certaines ressources avec un même ensemble de tags. Les caractéristiques textuelles sont quant à elles extraites des schémas de pondération classiques de tags comme la fréquence locale (*tf*) ou globale (*iuf*) où une combinaison des deux.

À première vue, la prise en compte de contenus supplémentaires en plus des notes communément utilisés en recommandation implique de considérer une technique de représentation adéquate. Les modèles d'apprentissage intégrant les annotations sociales [148, 76, 51] exploitent souvent le modèle en sac de mots pour encoder simplement les tags dans l'espace de représentations des utilisateurs et des items. Ces modèles sont issues d'extensions de modèles de factorisation de matrices telles que SVD [226, 206], ils emploient une formule globale permettant la décomposition d'une matrice composée de notes d'utilisateurs globale combiné à des indicateurs comme l'importance d'un tag par rapport à un ensemble d'items. *Koren et al.*, quant à eux [118] introduisent un ensemble supplémentaire de facteurs latents qui permet de mieux saisir l'impact des tags des utilisateurs dans l'estimation des prédictions. *Zhang et al.*, [246] partent du principe que plus les tags appartiennent à des sujets divers et variés plus ils permettent

une plus large considérations des intérêts des utilisateurs à travers de multiples catégories. Les plates-formes de recommandation comme *MovieLens* ou *Del.icio.us* fournissent aux utilisateurs des catégories qui font référence à différentes thématiques reflétant les préférences d'utilisateurs pour différents domaines. L'idée est d'attribuer au sein des matrices de caractéristiques tag-catégorie, un poids plus fort aux tags associées au plus grand nombre de catégories. Ces tags sont associés aux items figurant parmi les catégories n'étant pas décrites avec suffisamment de tags dans la matrice item-tag. Récemment, *TagEmbedSVD* [222] qui est un modèle basé sur la décomposition en valeur singulière a été introduit comme méthode collaborative pour la recommandation afin de répondre à la problématique du manque de données relatives à certaines catégories d'items lors de processus de recommandation, tout comme dans les travaux de [246]. Cependant, dans le modèle *TagEmbedSVD*, les auteurs ont opté pour l'exploitation de représentations vectorielles de tags issues de *word2vec*. Ils ont démontré qu'un espace de représentation dense commun (*common embedding space*) associé aux tags permettait une flexibilité dans la sélection d'une dimension adaptée aux tags décrivant certaines catégories grâce à des matrices de projections. Ainsi, les utilisateurs et les items peuvent être comparés grâce à leurs représentations projetées dans un espace commun dense.

La majorité des approches de filtrage collaboratif basées sur l'apprentissage automatique [76, 51], combinent linéairement les facteurs latents d'utilisateurs et d'items [148, 246]. À l'inverse, les systèmes de recommandation basés sur le *deep learning* [235, 138] parviennent à extraire des caractéristiques abstraites pendant que les systèmes de recommandation classiques, basés sur l'apprentissage statistique [51, 134] identifiera des caractéristiques prédéfinies. Ces modèles aboutissent d'ailleurs à des performances raisonnables [213]. Par comparaison, l'apprentissage profond ou "*deep learning*" a considérablement changé l'architecture des systèmes de recommandation en améliorant leurs performances. L'apprentissage profond est capable de capturer efficacement des relations non-linéaires et non-triviales entre utilisateurs et items. En outre, il permet l'encodage d'abstractions complexes sous la forme de représentations de données au sein des couches supérieures d'un réseau de neurones profonds. Il permet aussi de mettre en évidence des relations existant au sein même de la donnée comme : le contexte, le texte ainsi que les informations visuelles [244]. Les apports des modèles neuronaux exploités dans diverses tâches de recommandation sont plus explicités dans la section suivante.

3 Modèles de réseaux de neurones pour la recommandation

Les réseaux de neurones profonds (DNN) apprennent des représentations dites profondes grâce aux multiples niveaux de représentations et d'abstractions extraits à partir de données bruts. D'une part, la performance d'un modèle d'apprentissage est associée à la qualité des représentations fournies à l'entrée du modèle. D'autre part, le choix d'une stratégie de recommandation va de pair avec le choix de l'architecture neuronale utilisée. Les modèles de DNN pour la recommandation sont souvent des modèles entièrement différentiable et qui s'entraînent de bout en bout (*end-to-end*). De plus, les biais inductifs doivent être adaptés au type de données d'entrée [254]. À ce propos, le développement des systèmes de recommandation est longtemps resté tributaire du domaine relatif aux entités à recommander. À titre d'exemple, la recommandation de

films en ligne exige une fine analyse des commentaires des utilisateurs représentés sous la forme de données textuelles qui exigent très souvent de lourds pré-traitements lexicaux. À l'inverse, les systèmes de recommandations basés sur des réseaux de neurones prennent en entrée du texte brut sous toutes ses formes, en plus des possibilités qu'offrent ces systèmes de représenter toutes sorte de contenus (images et textes) dans un modèle unifié de bout-en bout. Ainsi, les réseaux d'apprentissage profond ont fait des percées dans le traitement des données multimédia et ont montré leur potentiel dans l'apprentissage de représentations à partir de sources de données disparates. Nous abordons dans la sous-section suivante certains modèles de filtrage collaboratif basés sur des modèles neuronaux de bout en bout ou *end-to-end* en anglais en détaillant les architectures ainsi que les approches collaboratives propres à chaque modèle. Ensuite, nous présentons certains modèles neuronaux de recommandation basés sur l'apprentissage de représentations de tags. De surcroît, nous détaillons les modèles neuronaux pour la recommandation basés sur des représentations denses (continues) de tags.

3.1 Filtrage collaboratif basé sur des modèles neuronaux *end-to-end*

Les méthodes d'apprentissage de «bout en bout» ou *end-to-end* sont conçues pour l'apprentissage de caractéristiques utiles des données. C'est ainsi que les approches collaboratives neuronales [95, 94] ont répondu aux faiblesses des modèles de recommandation à variables latentes. En raison de la nature implicite des données, mais aussi face à des méthodes comme la factorisation de matrices qui permet d'associer les utilisateurs et les items à des produits de vecteurs latents et à estimer les interactions des utilisateurs/items linéairement. Comme les modèles neuronaux utilisent des fonctions d'activations non-linéaires, cela permettrait une meilleure approximation des valeurs à prédire. Le filtrage collaboratif neuronal peut être perçu comme une représentation neuronale multi-couches qui modélise les interactions entre utilisateurs et items représentés sous la forme de vecteurs de caractéristiques $f_{(u)}$ et $f_{(i)}$ décrivant respectivement l'utilisateur u et l'item i . Ces vecteurs sont représentés sous la forme binaire en encodage *One-hot* comme dans la figure 3.6. À titre illustratif, nous prenons un réseau de neurones multi-couches (MLP) qui passent par la première couche du réseau (*embedding layer*) : Il s'agit d'une couche complètement connectée qui projette les représentations en vecteurs denses. Ces vecteurs denses d'utilisateurs et d'items peuvent être considérés comme des vecteurs latents. Pendant la phase d'entraînement, ces vecteurs passent à travers des couches neuronales dans l'optique de les "mapper" à des scores de prédiction grâce à une fonction de coût qui minimise l'erreur entre la valeur prédite \hat{r}_{ui} et la valeur réelle r_{ui} . Le modèle prédictif de filtrage collaboratif neuronal peut être formulée de la manière suivante :

$$\hat{r}_{ui} = \mathbf{MLP}(P_u^T f_{(u)}, Q_i^T f_{(i)} | P_u, Q_i, \theta_{\mathbf{MLP}}) \quad (3.11)$$

Où $P_u \in \mathbb{R}^{M \times K}$ et $Q_i \in \mathbb{R}^{N \times K}$ sont les matrices de facteurs latents associées aux utilisateurs et aux items et θ défini les paramètres du modèle de la fonction d'interaction **MLP**. Les paramètres du modèle sont estimés via une regression en minimisant la fonction de perte quadratique exprimée selon l'équation :

$$\zeta = \sum_{(u,i) \in \mathcal{R} \cup \mathcal{R}^-} \mathbf{W}_{u,i} (r_{u,i} - \hat{r}_{u,i})^2 \quad (3.12)$$

Où, \mathcal{R} représente l'ensemble des interactions utilisateur-item observés et \mathcal{R}^- l'ensemble des interactions à prédire et $\mathbf{W}_{u,i}$ est un hyper-paramètre représentant le poids associée à l'instance d'entraînement (u,i) .

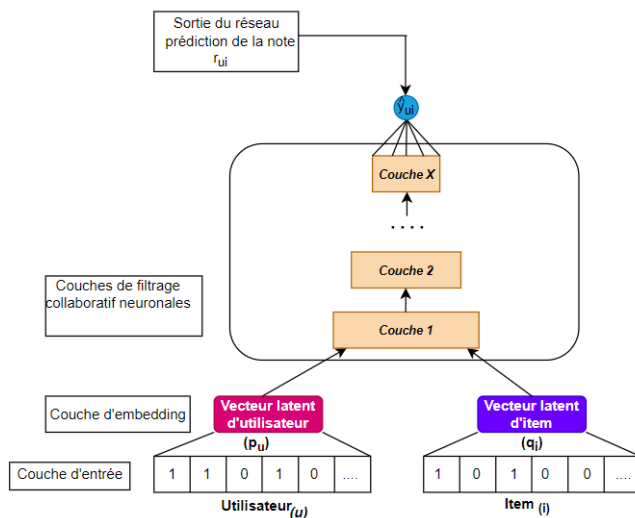
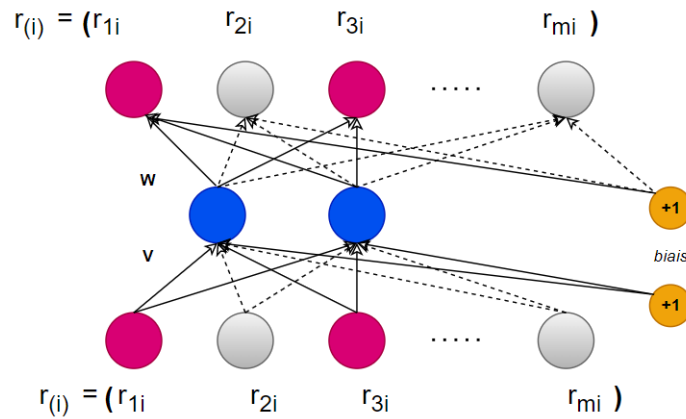


FIGURE 3.6 – Framework de filtrage collaboratif neuronal simple comme explicité dans [95]

He et al., [94] se sont inspirés des travaux de [95] et ont suggéré l'utilisation du produit tensoriel entre les matrices P_u et Q_i et ont proposé l'utilisation de réseaux de neurones à convolution [124] dans le but d'apprendre des corrélations de haut niveau parmi les dimensions d'embeddings.

Toutefois, le filtrage collaboratif neuronal a été abordé de différentes manières, à travers d'autres approches et en exploitant d'autres types d'architectures neuronales. C'est le cas aussi des modèles de recommandation exploitant des architectures en *Autoencoder* [196, 22, 254].

Autorec illustré dans la figure 3.7 [196] est un modèle de recommandation neuronal qui a été conçu pour prendre en entrée des vecteurs $r_{(u)}$ partiels représentant les notes d'utilisateurs pour des items (*I-Autorec*) ou de manière similaire, des vecteurs $r_{(i)}$ partiels représentant des notes d'items attribuées par des utilisateurs (*U-Autorec*). Le but étant d'exploiter les propriétés de l'autoencoder pour prédire les valeurs correspondantes aux notes dans la couche de sortie. L'autoencodeur compte parmi les architectures neuronales les plus performantes. Cette architecture permet l'apprentissage par rétro-propagation de représentation de caractéristiques neuronales, qui sont encodés à travers plusieurs couches neuronales dans un espace de représentation dense [217]. Pour autant, d'autres moyens peuvent être mis en œuvre pour exploiter d'autres caractéristiques non-triviales. Les techniques abordées dans la prochaine sous-section ont la particularité de considérer des représentations en graphes et mettent en avant différentes méthodes pour le filtrage collaboratif neurale.



Le vecteur $r_{(i)}$ des notes d'utilisateurs contenant des notes non-observées (en gris) avec $i = 1 \dots n$

FIGURE 3.7 – Architecture de *Autorec* illustré pour l'ensemble des notes des items représenté sous la forme d'un vecteur $r_{(i)}$ contenant des notes à prédire (nœuds en gris). Ces vecteurs sont projetés dans un espace latent de faible dimension puis sont reconstruits dans l'espace de sortie pour prédire les notes manquantes en vue de processus de recommandation avec $u \in U = \{1 \dots m\}$ l'ensemble des utilisateurs et $i \in I = \{1 \dots n\}$, l'ensemble des items

3.2 Modèles neuronaux end-to-end de recommandation basés sur l'apprentissage de représentation de graphes

Le filtrage collaboratif neuronal s'illustre aussi à travers des modèles neuronaux prenant en entrée un graphe. Ces modèles [91, 130, 22, 9] considèrent des structures bipartites d'utilisateurs et d'items représentées à partir d'une matrice M de taille $N \times M$. Les interactions sont définies à travers un graphe biparti non-orienté $G = (\mathcal{V}, \mathcal{E}, \mathcal{R})$ d'utilisateurs et d'items tel que : les nœuds $u_i \in \mathcal{V}_u$ avec $i \in \{1 \dots N\}$ représentent l'ensemble des utilisateurs et les nœud $v_j \in \mathcal{V}_j$ avec $j \in \{1 \dots M\}$ représentent l'ensemble des items, avec $\mathcal{V}_u \cup \mathcal{V}_j = \mathcal{V}$. Les notes d'utilisateurs $(u_i, r, v_j) \in \mathcal{E}$ représente l'ensemble des notes attribuées par les utilisateurs aux items soit : $r \in \{1 \dots R\} = \mathcal{R}$.

Certains algorithmes d'apprentissage pour la recommandation basés sur les graphes [91, 130] se sont axés sur des méthodes de complétion de la matrice M en rapportant la problématique de recommandation en une chaîne de traitement, i.e un pipeline comprenant un modèle d'apprentissage pour l'extraction de caractéristique des nœuds du graphe et un modèle pour la prédiction de lien. Dans les travaux de *Li et al.*, [130], chaque modèle est entraîné séparément. Cependant, de récents travaux [9, 22] ont montré de meilleurs résultats en considérant des modèles d'apprentissage qui modélisent les structures des graphes bipartite à l'aide d'un modèle neuronal entraîné de bout-en-bout [22]. Le modèle de graphe neuronal en autoencoder décrit dans les travaux de *Berg et al.* [22], visible dans la figure 3.8, se compose d'un encodeur $Z = f(X, A)$ qui prend en entrée une matrice de caractéristiques de taille $N \times D$ ainsi que la matrice d'adjacence A et produit des représentations denses de nœuds (*node embeddings*) sous forme de matrice de taille $N \times E$, avec N définissant le nombre de nœuds, D le nombre

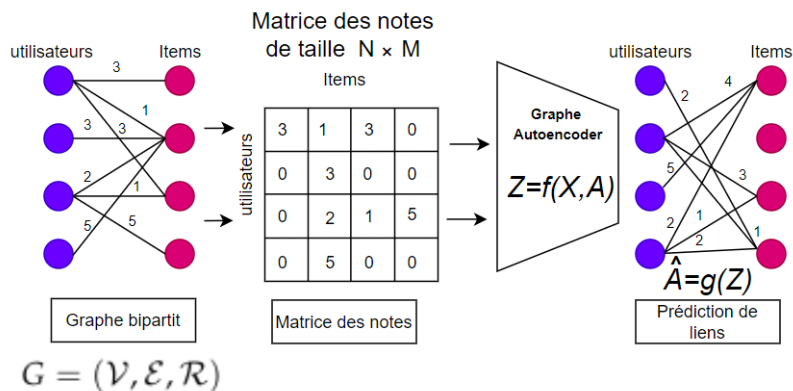


FIGURE 3.8 – Framework de filtrage collaboratif des graphes neuronaux (NGCF) comme explicité dans [22]

de caractéristiques en entrée et E la taille des représentations denses des nœuds. Ces représentations denses de nœuds peuvent être représentées sous la forme suivante à l'aide des propriétés de transposition : $Z = [z_1^T, \dots, z_N^T]^T$. Le décodeur $\hat{A} = g(Z)$ intègre les représentations denses des nœuds par paires (z_i, z_j) et prédit les entrées respectives \hat{A}_{ij} dans la matrice d'adjacence.

Ainsi, le modèle de recommandation en graphe bipartit $G = (\mathcal{V}, \mathcal{E}, \mathcal{R})$ peut être formulé à l'aide des matrices d'embeddings (matrices denses) U et V de taille respective $N \times E$ et $M \times E$ comme suit : L'encodeur est défini selon la fonction $[U, V] = f(X, M_1, \dots, M_R)$ avec $M_r \in \{0, 1\}$ la matrice d'adjacence de taille $N \times M$ associée aux notes des utilisateurs $r \in \mathcal{R}$, de cette manière la représentation dense des nœuds d'utilisateurs V_u et d'items V_j sont des vecteurs d'embeddings à valeurs réelles. Le décodeur $[\hat{M}] = g(U, W)$ reconstruit la matrice \hat{M} de taille $N \times M$. L'entraînement du modèle est réalisé en minimisant l'erreur de reconstruction entre la matrice des notes prédites \hat{M} et la matrice d'interaction M des notes initialement observées à l'aide de la perte quadratique moyenne si les notes à prédire sont à valeurs continues ou en optimisant l'entropie croisée si les notes sont représentées sous la forme de classes.

Par contraste, Hamilton *et al.* [91], ont quant à eux développé un modèle inductif permettant l'apprentissage de fonctions d'agrégation qui tiennent compte du voisinage des nœuds et de leurs caractéristiques, comme par exemple le degré d'un nœud dans un graphe $G = (\mathcal{V}, \mathcal{E}, \mathcal{R})$. Ainsi *GraphSAGE* a été conçu pour intégrer les caractéristiques de nœuds grâce à un algorithme d'apprentissage qui permet d'apprendre la structure topologique de chaque nœud en fonction de son voisinage. Les auteurs ont été en mesure d'entraîner un ensemble de fonctions d'agrégation qui apprennent des agrégats de caractéristiques issues du voisinage proche de nœuds. La génération d'un embedding de nœud issu d'un tel graphe bipartit est illustré à titre d'exemple dans la figure 3.9 en utilisant cette technique. D'abord les auteurs définissent une fonction d'agrégation $Agg^{(k)}$ au voisinage $k \in \{1, \dots, K\}$ ainsi qu'une matrice de poids \mathbf{W}^k qui est utilisé pour propager les informations entre les différentes couches du modèle. Chaque nœud $v \in \mathcal{V}$ agrège les représentations des nœuds de son voisin immédiat $\{h_u^{k-1}, \forall u \in \mathcal{N}(v)\}$ en un vecteur $h_{\mathcal{N}(v)}^{k-1}$. Une fois l'agrégation des nœuds voisins (et de leurs caractéristiques) ter-

minée, le modèle concatène la représentation du nœud courant h_v^{k-1} avec le vecteur du voisinage agrégé $h_{\mathcal{N}(v)}^{k-1}$. Le vecteur résultant est passé par une couche entièrement connectée avec une fonction d'activation non-linéaire σ qui relaie les représentations au prochain k-voisin i.e $h_v^k \forall v \in \mathcal{V}$. La représentation finale d'un nœud est dénotée z_v , elle est considérée comme étant la représentation dense agrégée au k-ième voisin. Une fonction objectif est appliquée à ces embeddings que l'on nomme z_u l'ensemble des embeddings des nœuds utilisateurs $\forall v_u \in \mathcal{V}$. Les matrices de poids à ajuster $\mathbf{W}^k \forall k \in \{1, \dots, k\}$ et les paramètres des fonctions d'agrégations sont ajustés grâce à l'algorithme de Descente de gradient stochastique [26]. La fonction objectif utilisée incite les nœuds voisins à avoir des représentations similaires tout en contraignant les représentations de nœuds dissemblables à être distincts. Cet algorithme d'apprentissage a permis de généraliser l'algorithme GraphSAGE pour des graphes hétérogènes et bipartis. En concaténant les représentations entre paires de nœuds, des représentations d'arcs sont formées et passées à travers des couches entièrement connectées d'un modèle de bout en bout qui minimise une fonction de perte (perte quadratique par exemple) entre les notes prédites et les notes réelles.

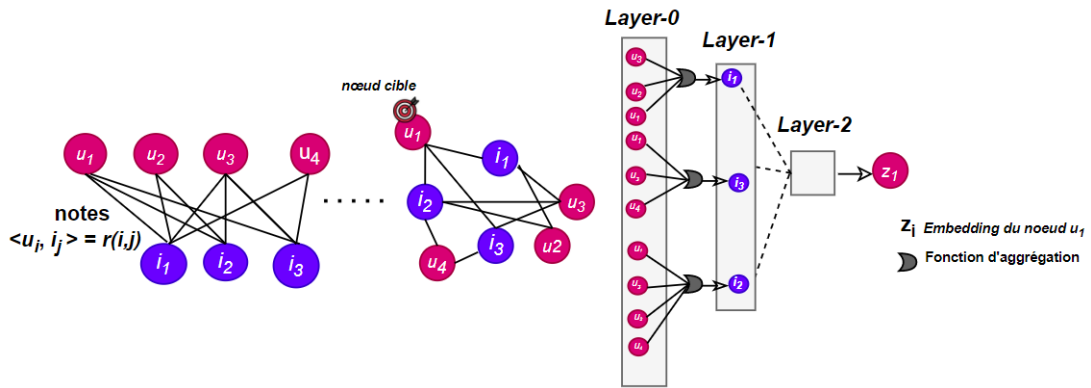


FIGURE 3.9 – Cette illustration explicite la génération de l'embedding du nœud u_1 . La représentation dense de ce nœud tient compte du voisinage auquel ce nœud est associé ($K=2$).

Récemment, les performances des approches de filtrage collaboratif neuronal dans le domaine de la recommandation ont conduit au développement d'approches de recommandation combinant apprentissage de représentations profondes de contenus et techniques de filtrage collaboratif. Ces approches sont hybrides du fait qu'elles intègrent des processus collaboratifs et que l'extraction de vecteurs de caractéristiques associés à ces contenus sont joints aux données d'apprentissage. Ces caractéristiques sont exploitées pour enrichir sémantiquement l'espace des représentations latent. Elles sont soit déterminées à l'issue du processus d'apprentissage faisant intervenir une architecture neuronale comme dans *Zuo et al.*, [254] ou extraites à partir de modèles pré-entraînés tel que : *word2vec* [132] lorsque le contenu est textuel. Nous présentons dans les prochaines sous-sections les modèles neuronaux de recommandation basés sur l'apprentissage de représentations de tags, par la suite, nous aborderons certains modèles neuronaux basés sur des représentations de tags issues de modèles de langue neuronaux pré-entraînés, utilisées comme caractéristiques dans un processus de filtrage collaboratif.

3.3 Modèles neuronaux de recommandation basés sur l'apprentissage de représentations

Les approches d'apprentissage profond ont montré un grand potentiel dans l'apprentissage de représentations. Dans le cas du filtrage collaboratif, il s'agit d'apprendre des représentations d'utilisateurs et d'items. Ces approches tentent de capturer, de manière fine, les interactions complexes entre utilisateurs et items en permettant d'atteindre un niveau d'abstraction élevé dans la dans la représentation des contenus. De plus, la diversité du contenu à exploiter est un moyen efficace d'affiner les représentations d'items et de saisir la nature complexe du comportement de certains utilisateurs [244].

Certaines de ces approches [235, 138] ont exploré l'apprentissage de représentations de profils d'utilisateurs et d'items au moyen de projection dans un espace vectoriel commun de faible dimension, les représentations ainsi produites sont comparées au moyen d'une simple fonction cosinus. Les profils sont initialement construits à partir d'activité d'annotation (représentés dans l'espace des tags). Ces profils sont positionnés à l'entrée de deux réseaux de neurones en perceptron multi-couches (MLP). Les deux réseaux de neurones possèdent de multiples couches cachées et partageant des paramètres communs. La raison d'une telle manœuvre tient au fait que les profils d'utilisateur et d'item sont modélisés à partir de la même folksonomie, ils partagent le même espace de tag. Il est donc raisonnable d'utiliser des paramètres partagés pour réduire les coûts de calcul et de mémoire lors de la phase d'entraînement du modèle. De plus les paramètres partagés font que les deux réseaux neuronaux utilisent le même espace de caractéristiques abstraites pour décrire les utilisateurs et les items, ce qui est utile pour mesurer leur pertinence. Le passage de ces profils à travers les multiples couches cachées permet la construction de représentations abstraites des profils d'utilisateurs et de ressources. Ces profils sont ensuite comparés par une simple fonction de similarité.

Formellement, un profil d'utilisateur est représenté par un vecteur de valeurs p_u contenant le nombre de fois qu'un utilisateur u annote des items avec le tag t_j . De manière similaire le profil d'item p_i est un vecteur qui représente le nombre de fois qu'un item i a été annoté avec le tag t_j . Une matrice de poids \mathbf{W}_1 ainsi qu'un biais b_1 et les sorties intermédiaires h_1 provenant de la première couche cachée sont définies comme suivant :

$$h_1(u) = \tanh(\mathbf{W}_1 p_u + \mathbf{b}_1) \quad (3.13)$$

$$h_1(i) = \tanh(\mathbf{W}_1 p_i + \mathbf{b}_1) \quad (3.14)$$

Où \tanh est utilisée comme fonction d'activation de type tangente hyperbolique. De manière similaire la sortie intermédiaire de la j ème couche h_j , avec $j \in \{2...K\}$ est définie selon les équations suivantes :

$$h_j(u) = \tanh(\mathbf{W}_j h_{j-1}(u) + \mathbf{b}_j) \quad (3.15)$$

$$h_j(i) = \tanh(\mathbf{W}_j h_{j-1}(i) + \mathbf{b}_j) \quad (3.16)$$

Où, \mathbf{W}_j et \mathbf{b}_j sont respectivement les poids de la matrice et le vecteur de biais pour la j -ème couche cachée, K est le nombre total de couches cachées. Les sorties à la K ème

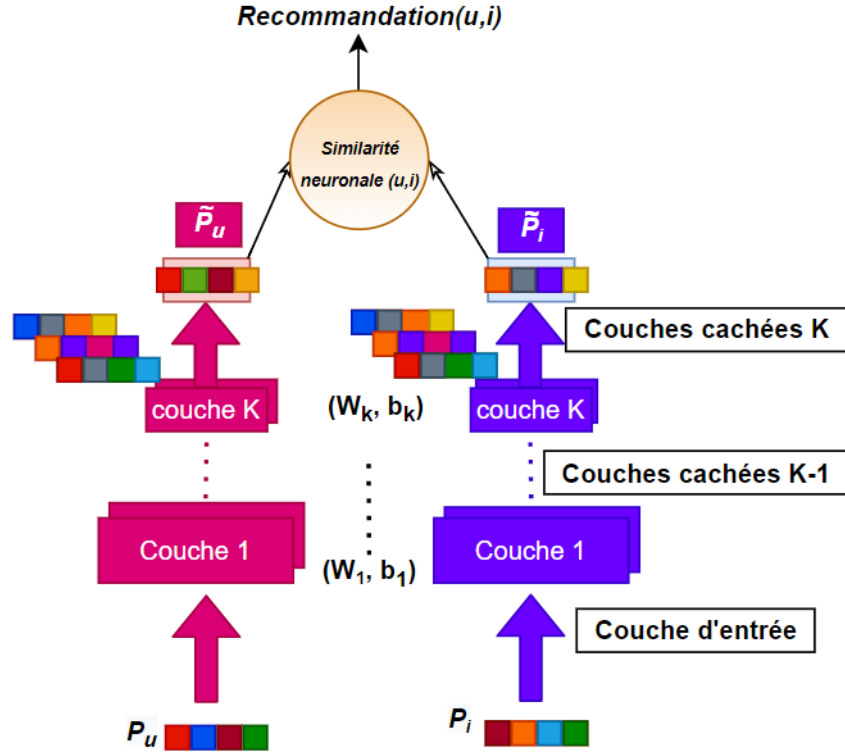


FIGURE 3.10 – Architecture en MLP permettant l’apprentissage de de représentations de profils d’utilisateur et d’items et le calcul de la similarité entre ces profils

couche cachée sont les représentations abstraites des profils d’utilisateurs et d’items notées respectivement \tilde{p}_u et \tilde{p}_i . De manière formelle, $\tilde{p}_u = h_K(u)$ et $\tilde{p}_i = h_K(i)$. Ainsi la similarité neuronale entre profils d’utilisateurs et d’items est calculé selon la mesure du cosinus, soit :

$$Sim_{(u,i)} = \frac{\tilde{p}_u \cdot \tilde{p}_i}{|\tilde{p}_u| |\tilde{p}_i|} \tag{3.17}$$

Afin de générer des recommandations personnalisées, la fonction *softmax* est utilisé pour mesurer la pertinence d’un item i pour un utilisateur u , elle est définie sur l’ensemble des items comme suit :

$$Rec_{(u,i)} = \frac{e^{Sim_{(u,i)}}}{\sum_{i' \in I} e^{Sim_{(u,i')}}} \tag{3.18}$$

Le modèle est entraîné pour maximiser la similarité entre paires (utilisateur,item) pertinents qui peut être aussi abordé en minimisant la similarité entre paires d’utilisateurs/items non-pertinents.

En représentant θ comme les paramètres de \mathbf{W}_j et \mathbf{b}_j dans le réseau de neurones et (u, i^*) comme ensemble d’entraînement qui se caractérise par des paires d’utilisateurs u et d’items i^* générées par l’affectation de tags à des items par des utilisateurs, i.e (u, t, i^*) , la fonction objectif (*loss function*) utilisée pour cela est définie comme suit :

$$L(\theta) = \sum_{(u,i^*)} \log(\text{Rec}_{(u,i^*)}) = - \sum_{(u,i^*)} [\log(e^{\text{Sim}_{(u,i^*)}}) - \log(\sum_{i' \in I} e^{\text{Sim}_{(u,i')}})] \quad (3.19)$$

Par la suite, les mêmes auteurs [138, 235] ont exploité une architecture neuronale en *Autoencoder* dans un modèle hybride pour l'apprentissage de représentations denses de profils pour la recommandation. Ils partent du principe que l'entraînement de réseaux de neurones denses par rétro propagation peut être une tâche difficile. Étant donné que, dans un réseau de neurones profonds il arrive que l'intensité des signaux neuronaux deviennent insignifiant lors de la rétro propagation vers les premières couches du réseau, cela entraîne une difficulté du modèle à converger. Pour remédier à ce problème, certains auteurs [19] pré-entraînent chaque couche du réseau avant la rétro-propagation. Cette technique est plus connue sous le nom de "*fine-tuning*". Grâce à cette technique, les auteurs améliorent les scores de recommandation de leurs précédents travaux [235].

3.4 Modèles neuronaux pour la recommandation basés sur des représentations denses de tags

Une attention particulière a été portée sur certains travaux qui tentent d'établir quels types de représentations denses d'utilisateurs et d'items utilisés à l'entrée du réseau de neurones. Nous nous focalisons sur les travaux (modèles neuronaux) qui ont exploité des vecteurs denses. Les questions qui ont été abordées dans ce cadre concernent, le type de vecteurs à utiliser (*word2vec*, *Bert*, ...), la manière de les agréger ou encore la manière de les combinés avec les notes d'utilisateurs [181, 230, 91].

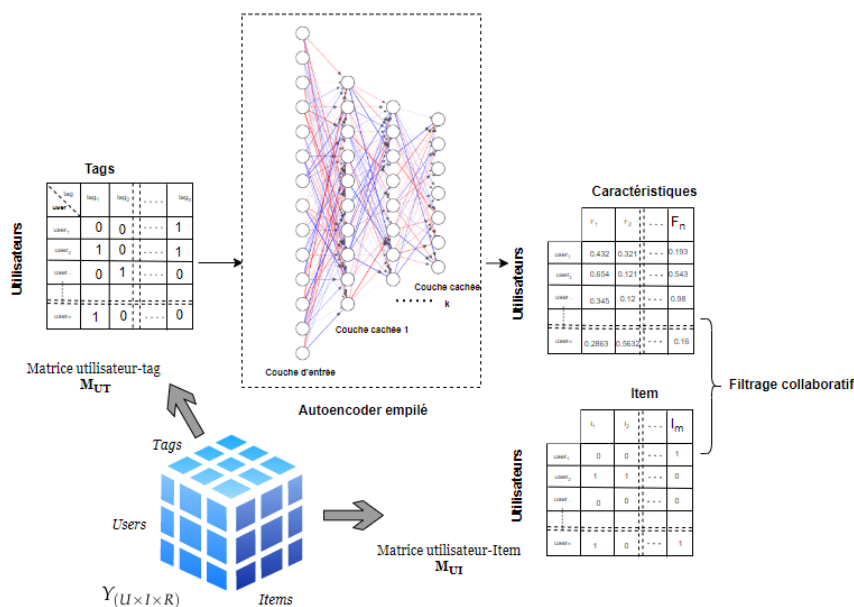


FIGURE 3.11 – Illustration de l'approche de Zuo *et al.*, [254] utilisant le filtrage collaboratif neuronal en exploitant un autoencoder empilés à l'aide de plusieurs couches cachées.

Les auteurs de Zuo *et al.*, [254] ont suggéré la projection de la relation ternaire $Y_{(U \times I \times R)}$ entre utilisateurs, items et tags sous la forme de deux matrices de dimen-

sions ($dim = 2$), la matrice utilisateur-item M_{UI} ainsi que la matrice utilisateur-tag M_{UT} comme illustré dans la figure 3.11. Le but étant d'extraire les profils d'utilisateur à partir de la matrice M_{UT} . D'autre part, cette matrice est passée à travers une structure en autoencoder épars (*sparse autoencoder*) [99] permettant d'obtenir des représentations de plus petite dimension comparée à la dimension de la matrice initiale M_{UT} du fait du grand nombre de tags à traiter. Cependant, les auteurs autorisent la parcimonie sur les couches cachées de l'autoencoder durant la phase d'apprentissage. Cela permet de représenter de façon épars les entrées, ces représentations pouvant être utilisées pour le pré-entraînement de tâches spécifiques. Les auteurs définissent les représentations obtenues comme étant des représentations abstraites de profils d'utilisateurs obtenues à travers les couches cachées de l'encoder. Ils sont définis de la manière suivante :

$$\hat{P}_u = (h_1, h_2, \dots, h_k) \quad (3.20)$$

Avec, h_k qui représente le k -ième facteur latent. Les auteurs assurent que cette technique permet d'extraire des caractéristiques dans l'espace utilisateur-tag permettant l'identification du voisinage $v \in N_u$ de l'utilisateur u d'après une mesure de similarité. Ainsi la note d'un utilisateur $r_{u,i}$ est calculée au voisinage de son profil \hat{P}_u selon l'équation suivante :

$$r_{u,i} = sim_{(u,v)} = \sum_{v \in N_u} sim_{(u,v)} \cdot M_{(v,i)} \quad (3.21)$$

Ainsi, selon les notes que le modèle prédit, les top-n items possédant les scores de similarité les plus élevés sont recommandés. Les auteurs précisent par ailleurs, que la matrice M_{UT} est utilisée afin d'identifier des utilisateurs similaires $sim_{(u,v)}$ et que la matrice M quant à elle est utilisée pour recommander des items selon le degré de similarité entre utilisateurs. De ce fait, les auteurs parviennent à exploiter pleinement la relation ternaire entre utilisateurs, tags et items.

Liang et al., [132], modélisent les utilisateurs et les items dans l'espace des tags représentés par des vecteurs denses. Comme un tag peut être formé de plusieurs mots (tokens), les auteurs proposent de représenter chaque tag en fonction des *embeddings* (*word2vec*) dans le cadre de ces travaux) des mots qui le composent. Pour le reste, on retrouve les mêmes architectures que les modèles précédents, comme l'illustre la figure 3.12, les vecteurs passent par différentes couches cachées. *Liang et al.*, [132] ont utilisé un réseau de neurones récurrents (RNN) [150] pour la construction de profils d'utilisateurs. Pour cela, ils ont considéré les vecteurs de la couche cachée du RNN comme étant des vecteurs de caractéristiques latents. Les auteurs [132] ont relevé que l'architecture d'un RNN s'avère être particulièrement efficace lorsqu'il s'agit d'apprendre des données séquentielles. En d'autres termes, le produit scalaire des deux vecteurs de caractéristiques, le premier issu d'un DNN exploitant la moyenne des embeddings de tags qui sont extraits du modèle *word2vec* et le deuxième, issu d'un RNN sont employés pour approximer la valeur de la note à prédire. Les auteurs affirment que la combinaison des caractéristiques latentes issues de ces deux architectures (DNN et RNN) améliore significativement la performance du modèle pour une tâche de prédiction de notes.

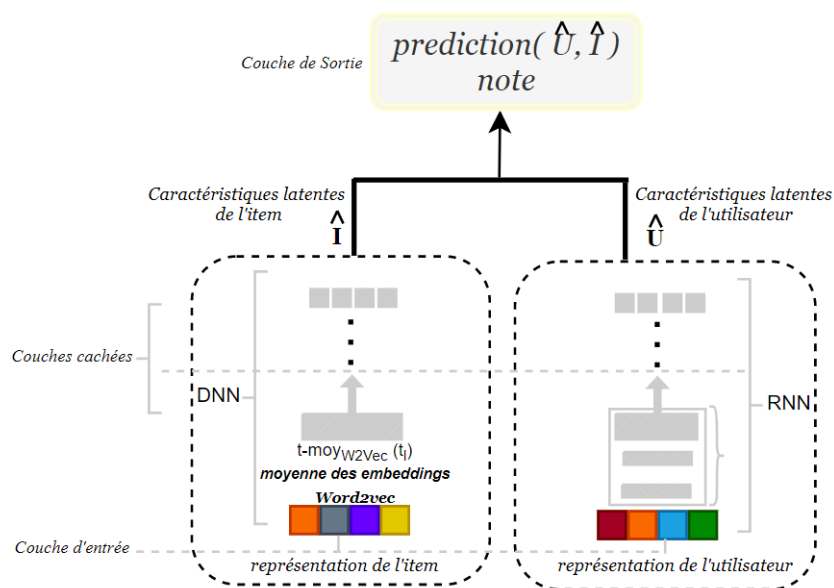


FIGURE 3.12 – Illustration du modèle neuronal TRSDL [132], d’une part la modélisation de l’utilisateur à travers un réseau de neurones récurrents et d’une autre part la modélisation des items à travers un réseau de neurones profonds exploitant la moyenne des embeddings de tags extraits de *word2vec*.

4 Limitations

De manière générale, Il existe des limitations récurrentes pour la mise en œuvre de processus collaboratifs exploitant des tags dans des systèmes de recommandation. Dans ce qui suit, nous avons mis en exergue certaines d’entre elles :

1. Tout d’abord, concernant les approches dites statiques qui se basent sur la similarité vectorielle entre profils d’utilisateurs et profils d’items :
 - La représentation de tags en sac de mots pour la construction de profils folksonomiques peut conduire à des irrégularités. Du point de vue de l’utilisateur, les approches [233, 219, 162] qui supposent que les tags les plus fréquents sont les plus susceptibles de mieux décrire un profil d’utilisateur ou de ressource se heurtent à des problématiques qui peuvent biaiser les résultats de classement personnalisé. En effet, l’utilisation de fréquences absolues de tags, sans tenir compte du fait que certains utilisateurs sont plus actifs que d’autres, augmente anormalement les poids des utilisateurs actifs et diminue les poids des autres utilisateurs moins actifs. Dans ce sens, des travaux comme [214, 207, 33] ont eu pour objectif l’exploitation de l’activité de l’utilisateur pour déduire ses intérêts, ils proposent d’ajuster, précisément, de normaliser le poids des tags en fonction des activités de l’utilisateur.
 - Dans les systèmes d’annotations sociales, les utilisateurs sont libres d’utiliser n’importe quel vocabulaire pour annoter n’importe quelle ressource. Cela conduit l’utilisateur à employer un large éventail de tags qui est en relation avec un large éventail de thèmes, qui peuvent également être redondants et ambigus. Un seul

ensemble (ou vecteur) de tags ne peut pas capturer la diversité des centres d'intérêt. Pour faire face à ce problème, certaines approches [200, 14, 237, 20, 207] ont été proposées pour rassembler les tags similaires en clusters qui reflètent les préférences des utilisateurs pour un sujet d'intérêt particulier. Les techniques de *clustering* sont souvent basées sur la similarité entre les vecteurs de tags [237, 78]. En d'autres termes, elles calculent la similarité entre deux tags représentés comme des vecteurs pondérés d'utilisateurs et de ressources basés sur des modèles de pondération tf ou $tf*idf$. Cependant, bien que ce type d'approches offre une perception plus large du profil de l'utilisateur, de telle sorte que les centres d'intérêts soient plus divers et variés et mieux adaptés pour représenter le large éventail des intérêts des utilisateurs, elles ne considèrent pas l'annotation dans le contexte de l'activité de l'utilisateur, et ne fournissent pas de moyens d'identifier les tendances dont les intérêts et les communautés en regardant à la fois l'activité des utilisateurs à travers différents tags et l'activité dans les tags à travers différents utilisateurs. Ces approches souffrent également de la fréquence non normalisée des termes.

- Même si certains travaux sont parvenus à lever les limitations dues aux activités d'annotations hétérogènes des utilisateurs [34] ainsi qu'à la diversité de leurs centres d'intérêts[237], il n'en demeure pas moins que les préférences de l'utilisateur évoluent avec le temps. Les travaux ayant réussi à prendre en compte la dimension temporelle des annotations à travers des profils [109, 2, 248], ont exploité dans un espace vectoriel, des représentations de tags auxquelles des mesures de poids représentant la popularité d'un ensemble de tag vis à vis d'un utilisateur ou d'une ressource ont été combinées à des fonctions modélisant l'atténuation graduelle de l'intérêt. Cette dynamique est portée par l'utilisateur et/ou la ressource. Du point de vue de l'utilisateur, cela a permis de modéliser des profils d'utilisateurs selon l'évolution de ses intérêts en lui proposant des recommandations actualisées. D'un point de vue des ressources, cela permet aux systèmes de recommandation de mettre à jour l'ordre du classement préférentiel des différents contenus proposés aux usagers. Cependant, si l'on considère que les intérêts des utilisateurs sont variés et multiples, les approches considérées peuvent faire face à certaines incohérences dans la mise à jour des contenus proposés en n'offrant pas la possibilité de bénéficier d'une politique de mise à jour dynamique du contenu recommandable prenant en compte le changement des préférences des utilisateurs dans le temps et à travers différents domaines.

2. Concernant les modèles neuronaux

- Nous avons souligné que le filtrage collaboratif pour la recommandation a aussi été mis en œuvre grâce à des modèles neuronaux. Ces modèles ont la particularité de pouvoir apprendre des représentations de données. De cette manière le choix des caractéristiques à prendre en considération est prise en charge par le modèle. Cependant, les modèles de filtrage collaboratif neuronal abordés intègrent les tags implicitement dans leurs processus de recommandation. En d'autres termes, les vecteurs des matrices représentant les interactions utilisateur-tag, item-tag et utilisateur-item sont représentés sous leurs formes binarisées en utilisant un en-

codage disjonctif (encodage one-hot) [196, 95, 94]. Par conséquent, les matrices d'interactions sont creuses et ne prennent pas en compte l'aspect sémantique et contextuel pouvant être exploité pour saisir toute la complexité de l'annotation sociale. Certains travaux, ont proposé d'extraire les caractéristiques des tags grâce à des architectures neuronales permettant l'encodage dans un espace de représentations associé aux tags. Une fois ces caractéristiques extraites, elles sont exploitées dans des méthodes de filtrage collaboratif, par similarité. Ils utilisent [222, 132] des représentations denses de tags issues de modèles tel que *word2vec* qui malgré la possibilité de représenter ces tags dans un espace vectoriel sémantiquement riche peuvent être assimilés à des représentations statiques. Enfin, d'autres travaux [254, 92] ont exploré l'utilisation de représentations d'utilisateurs et d'items via des représentations explicites de tags en optimisant les dimensions de l'espace de représentation associé aux tags, mais n'exploitent peu ou pas le voisinage de ces représentations.

5 Discussion

Dans cette section, nous synthétisons quelques points (idées) qui certes, ont été discutés précédemment, mais qui apportent des précisions quant aux contextes où se s'inscrivent nos contributions.

Les approches de personnalisation exploitant des profils pour la recommandation ont été illustrées à travers de nombreux travaux de recherche [241, 237, 233, 219, 175, 162, 65, 63, 42, 33]. La plupart utilise des représentations de tags en sac de mots exploitant des schémas de pondération tel que $tf * idf$ ou des variantes de ce modèle [233, 219, 162, 33, 13]. Les profils sont alors considérés comme un unique vecteur de tags pondéré. D'autres travaux [241, 237, 78, 14], considèrent ces profils comme des ensembles multiples de vecteurs de tags, où chaque ensemble représente un centre d'intérêt particulier. De notre point de vue, ces approches peuvent être catégorisées en deux catégories :

1. Profils centrés sur un centre d'intérêt unique : Les profils d'utilisateurs/ressources sont représentés dans un espace vectoriel en combinant différentes caractéristiques relatifs à la popularité des tags, leurs temporalité, leurs valeurs affectives ou communautaires [233, 219, 162, 33, 13]. Ces profils sont des vecteurs de tags pondérés combinant plusieurs de ces caractéristiques grâce à des mesures de pondération spécifiques indiquant la popularité des tags par l'utilisateur, ou de manière similaire d'un point de vue de la ressource la popularité d'un tag utilisé pour annoter une ressource.
2. Profils basés sur de multiples centres d'intérêts : Les profils issus de ces approches sont représentés sous la forme multiples vecteurs d'intérêts et se basent sur des méthodes d'apprentissage non-supervisé tel que le *clustering* pour regrouper les tags similaires sous la forme de clusters qui reflètent les préférences des utilisateurs pour certains domaines d'intérêts, mais aussi qui traduisent leurs préférence à l'égard de certaine ressources connexes [241, 237, 233, 200, 78, 63, 14]. Les travaux dans ce sens représentent les intérêts des utilisateurs sous la forme de clusters d'in-

térêts. Chaque cluster est considéré comme un unique topic d'intérêt composé de tags pondérés selon l'importance de ces derniers au sein de ce cluster.

Cependant, nous considérons que de telles représentations de profils sont statiques. Les profils d'utilisateurs évoluent avec le temps et donnent lieu à de nouveaux centres d'intérêts qui doivent être intégrés aux profils. De nos jours les systèmes de recommandation favorisent la diversité des contenus en proposant à leurs usagers la capacité de s'adapter et d'actualiser selon certaines tendances issues de pratiques sociales. En considérant les tags comme composantes de base, la construction de profils d'utilisateurs évolutifs saisissant des centres d'intérêts multiples représente un enjeu de taille.

Par ailleurs, nous avons montré que certains modèles neuronaux pour le filtrage collaboratif exploitent des tags pour servir de vecteurs de caractéristiques supplémentaires pour enrichir sémantiquement l'espace des représentations associé aux utilisateurs et aux items. Des travaux ont proposé l'intégration de ces tags sous forme de simples mots clé (représentation en *One-hot*) [51, 76, 148] ou comme des embeddings issus de modèles pré-entraînés tel que *word2vec* [222, 132] ou Glove [92]. Cependant, comme nous l'avons explicité dans le chapitre précédent les représentations denses issues du modèle *word2vec* sont statiques et sujets à différentes limitations.

D'autres modèles [91, 232, 130] suggèrent l'apprentissage des caractéristiques de nœuds décrivant des utilisateurs et des items dans une structure en graphe. L'apprentissage de ces modèles repose uniquement sur la matrice d'adjacence souvent engendrée dans l'espace des représentations des notes attribuées par les utilisateurs aux items. D'autres travaux [22, 37] exploitent ces matrices d'adjacences comme caractéristiques encodées à l'aide d'architectures neuronales. Certaines méthodes de représentations de nœuds permettent la découverte de connaissance plus profonde à travers l'exploration de leurs voisinages [91, 9]. Ces modèles apprennent les représentations des nœuds en fonction des voisins.

Dans le but d'étendre ces techniques d'apprentissage dans un espace de représentation intégrant des tags, nous avons pris l'initiative d'adapter certains modèles de filtrage collaboratif neuronal (*end-to-end*) en y intégrant différentes représentations de tags (*embeddings*). Nous avons ensuite formalisé des techniques permettant de prendre en compte des embeddings de tags à l'entrée de ces modèles. Ces *embeddings* sont issus de modèles de langues neuronaux pré-entraînés statiques (*word2vec*) et contextualisés (*BERT*) et sont associés à des représentations d'utilisateurs et d'items. Enfin, nous avons exploré l'utilisation d'embeddings contextualisés à un certain voisinage grâce notamment à une représentation initiale en graphe biparti et d'une architecture neuronale de bout en bout.

6 Conclusion

Dans ce chapitre, nous avons synthétisé une partie de l'état de l'art. D'abord, nous avons introduit les différents concepts relatifs au domaine de la recommandation en abordant les applications les plus populaires du domaine dont la prédiction de notes d'items, souvent associées à des contenus recommandables, en se basant sur l'historique

des attributions passées de l'utilisateur. Nous avons présenté différentes techniques permettant l'intégration de tags dans des tâches de recommandation personnalisées en détaillant certains travaux connexes. Nous nous sommes intéressés particulièrement aux méthodes de filtrage collaboratif. Cela nous a permis de considérer les processus de recommandation intégrant des tags comme étant tributaires de la pertinence de profils folksonomiques. Dans ce sens, nous avons abordé l'étude de ces profils à travers différentes dimensions et avons fait état de leurs apports dans la modélisation d'usagers de folksonomies mais aussi d'items annotés par ces usagers.

Parmi les approches de recommandation citées, nous avons présenté certains modèles basés sur l'apprentissage qui se basent sur des transformations linéaires opérées sur des données d'apprentissage associées à des tags et qui ont été intégrés avec succès comme caractéristiques sociales d'usagers et d'items au sein de ces modèles.

Des travaux plus récents ont été abordés. C'est ainsi que les modèles neuronaux visant à l'apprentissage des interactions utilisateurs/items grâce à des fonctions d'activation non-linéaires ont été abordés. Ces modèles ont adopté des stratégies de recommandations basées sur le principe du filtrage collaboratif neuronales en employant différentes architectures. D'autres travaux ont employé des représentations en graphes bipartis d'utilisateurs-items afin de saisir des caractéristiques topologiques pouvant s'étendre à un voisinage donné. Ces travaux ont exploité ces représentations à l'entrée de modèles neuronaux entraînés de bout en bout. L'intégration des tags dans ces modèles a surtout été illustré à travers l'extraction de caractéristiques à partir de matrice utilisateur-tags et leur mise en œuvre dans des processus collaboratif. D'autres travaux sont parvenus à extraire ces caractéristiques à partir de modèles de langues neuronales pré-entraîné tel que *word2vec*.

Vers la fin de ce chapitre, nous avons fait état des différentes limitations rencontrées par l'utilisation de ces modèles de recommandation et avons discuté des possibles perspectives et axes d'améliorations permettant de lever ces limitations, en nous focalisant particulièrement sur les axes suivants :

- La nécessité de développer l'aspect évolutif des profils folksonomiques en tenant compte des multiples centres d'intérêts des utilisateurs pour améliorer la pertinence de ces profils et de les exploiter dans un contexte de recommandation.
- L'exploration de modèles neuronaux permettant l'apprentissage de caractéristiques aux voisinages des représentations d'utilisateurs et d'items, en intégrant au sein de ces modèles des représentations denses de tags.

STATISTIQUES DE LA COLLECTION

Statistiques détaillées de la collection MovieLens

Chapitre 4

DESCRIPTION DE LA COLLECTION

1 Introduction

GROUPELENS est un laboratoire de recherche américain du département d'informatique et d'ingénierie de l'université du Minnesota, spécialisé dans les systèmes de recommandation, les communautés en ligne, les technologies mobiles, les bibliothèques numériques et les systèmes d'information géographique. Parmi ces nombreux axes d'études, le projet *MovieLens* créée en 1997 est celui qui a suscité le plus d'intérêts.

Il s'agit d'un système de recommandation en ligne de contenus soutenu par une communauté virtuelle active qui recommande à ses utilisateurs des films à regarder, en fonction de leurs préférences cinématographiques et en utilisant différentes techniques issues du filtrage collaboratif, de factorisation de matrices et d'historique de notes et de critiques d'utilisateurs.

Le groupe de recherche ayant développé *MovieLens* s'est intéressé à plusieurs problématiques inhérentes aux systèmes de recommandation dont le problème de démarrage à froid ou encore la prédiction de notes d'utilisateurs à partir d'historique ou d'annotations sociales. Le groupe a mis à disposition des bases de données révélant les diverses activités des utilisateurs. En particulier, dans le cadre de notre thèse, nous nous focaliserons sur les données issues d'annotations sociales d'utilisateurs.

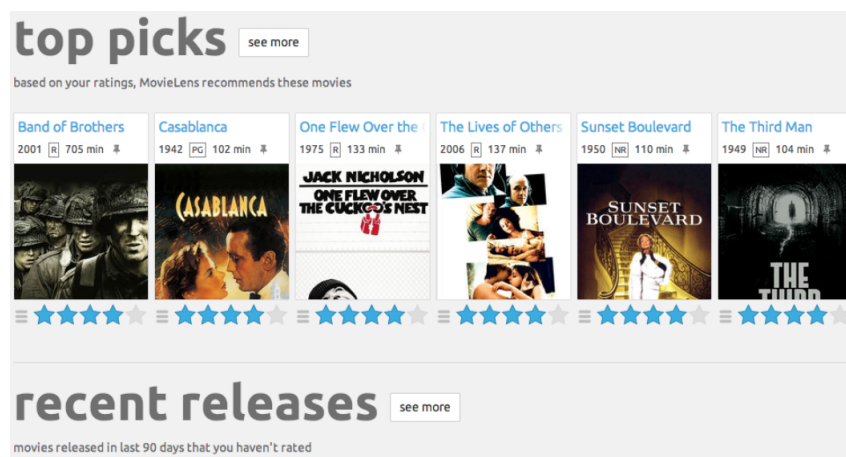


FIGURE 4.1 – Page principale de la plate-forme de recommandation *MovieLens*

2 Analyse de la collection *MovieLens*

Dans le cadre de nos contributions, nous avons exploité les données de datasets provenant de la collection. Elle se compose de 3 sous ensembles distincts, *MovieLen-10M*, *MovieLens-20M* et *MovieLens-25M* dont nous détaillons à titre indicatif, le contenu ex-

plotté dans le cadre des différentes expérimentations rapportées dans cette thèse.

1. Les utilisateur (*userId*) : Sont décrits sous la forme d'identifiants anonymisés.
2. Les ressources (Items *movieId*) : Correspondent à des ressources cinématographiques (des films). Il est à noter qu'il existe dans la collection que des films ayant au moins une note ou une annotation. Les films sont aussi représentés sous la forme d'identifiants.
3. Les notes : Il s'agit de notes d'utilisateurs à l'égard de films, ces notes sont formulées selon l'appréciation des utilisateurs sur la base de scalaire de 0.5 indiquant une faible appréciation à 5 indiquant une forte appréciation.
4. Timestamps : représente la valeur temporelle à laquelle un utilisateur a consommé une ressource, soit en lui attribuant une note soit en l'annotant via un tag. Cette valeur est formalisée comme étant le nombre de secondes écoulées entre l'activité de l'utilisateur et le 1^{er} janvier 1970.
5. Tag : Un tag et généralement inclus au sein du quadruple $\langle userId, movieId, tag, timestamp \rangle$. Les tags sont associés à des méta-données générées par les utilisateurs sur les films. Chaque tag est constitué d'un seul mot ou d'une courte phrase. La signification, la valeur et l'objectif d'un tag particulier sont déterminés par chaque utilisateur.

3 Statistiques descriptives

Les tags sont les constituantes principaux des folkonomies, dans cette section nous fournissons plus de détails sur ces données personnelles à savoir : La distribution des fréquences de tags ainsi que la distribution des mots composant un tag dans les trois dataset.

3.1 *MovieLens-10M*

Ce dataset comporte 4009 utilisateurs uniques, de 7601 instances de tags uniques et 16526 films distincts. Le tableau inclut dans la figure 3.1 résume la fréquence de tags en fonction du nombre de *token* ou de mots les composants. Cette illustration présente aussi la distribution des activités d'annotations ainsi que la distribution des notes des utilisateurs ayant annotés au moins un tag.

3.2 *MovieLens-20M*

Ce dataset se compose de 7801 utilisateurs uniques, de 38643 instances de tags uniques et 19545 films distincts. Le tableau dans la figure 3.2 résume la fréquence de tags en fonction du nombre de *token* ou de mots les composants. Dans la même figure, nous donnons, la distribution des activités d'annotations ainsi que la distribution des notes des utilisateurs ayant annotés au moins un tag dans la collection *MovieLens-20M*.

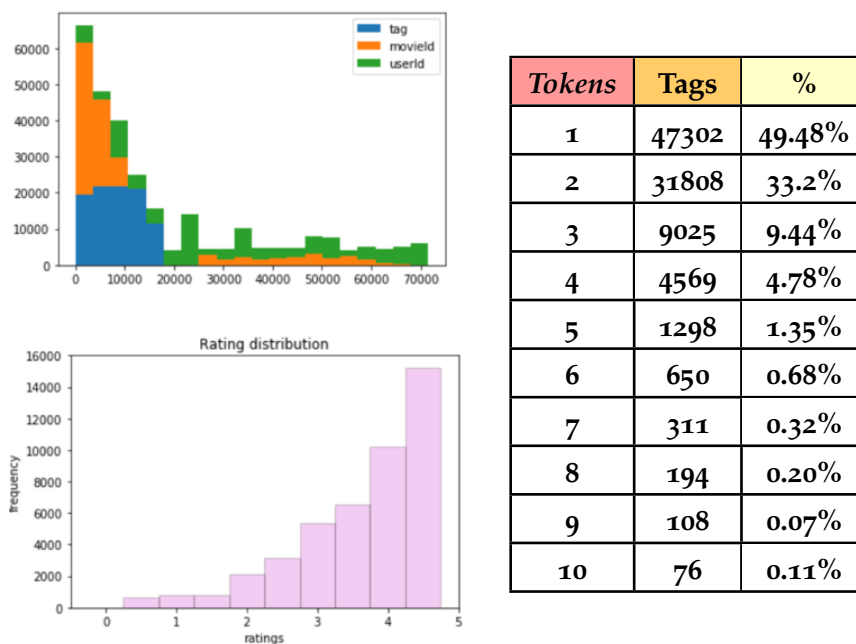


FIGURE 4.2 – Distribution du nombre de *tokens* (mots) composant un tag dans *MovieLens-10M*, de l'activité d'annotation et de l'ensemble des notes des utilisateurs ayant annotés au moins un tag *MovieLens-10M*.

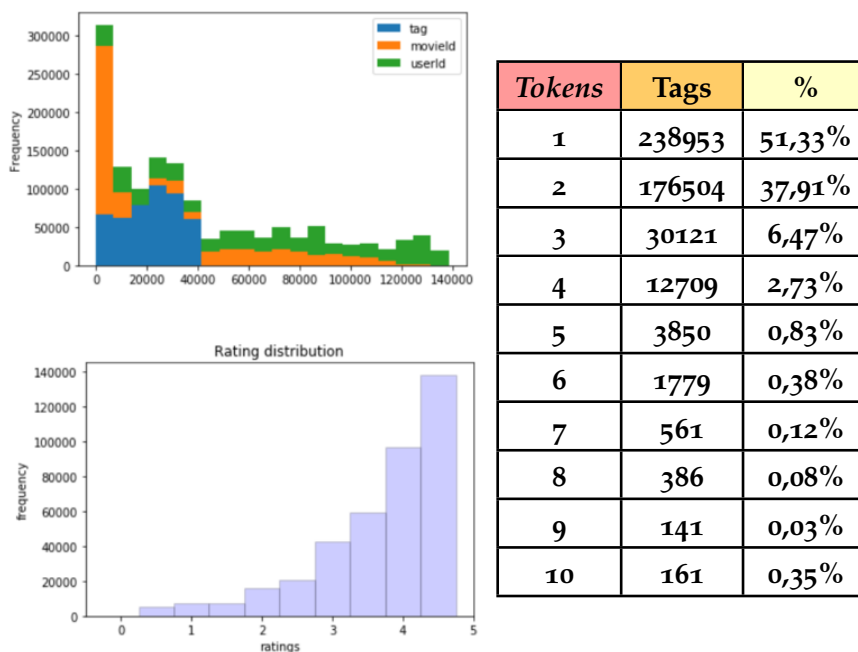


FIGURE 4.3 – Distribution du nombre de *tokens* (mots) composant un tag dans *MovieLens-20M*, de l'activité d'annotation et de l'ensemble des notes des utilisateurs ayant annotés au moins un tag dans *MovieLens-20M*.

3.3 *MovieLens-25M*

Ce dataset se compose de 14592 utilisateurs uniques, de 45251 instances de tags uniques et 73050 films distincts. Le tableau de la figure 3.3 résume la fréquence de tags en fonction du nombre de *token* ou de mots les composants. Enfin, nous présentons aussi dans la figure 3.3, la distribution des activités d’annotations ainsi que la distribution des notes des utilisateurs ayant annotés au moins un tag dans la collection *MovieLens-25M*.

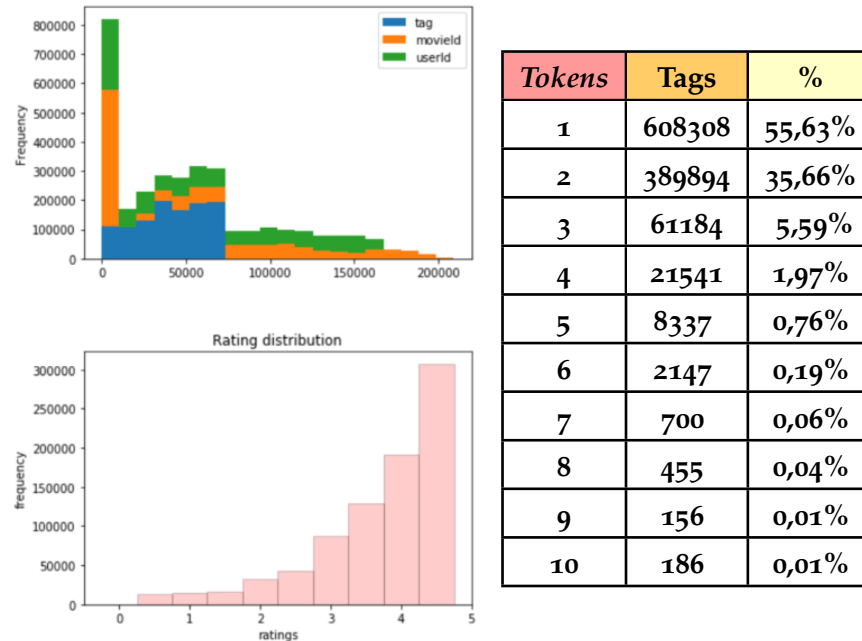


FIGURE 4.4 – Distribution du nombre de *tokens* (mots) composant un tag dans *MovieLens-25M*, de l’activité d’annotation et de l’ensemble des notes des utilisateurs ayant annotés au moins un tag dans *MovieLens-25M*.

3.4 *Plongement lexicaux de tags avec word2vec et visualisation à l’aide de t-SNE*

A l’issue de cette thèse, nous avons étudié les représentations que pouvaient engendrer certains modèles neuronaux. En l’occurrence, le modèle de langue neuronal pré-entraîné *word2vec* dont nous avons déjà évoqué le principe. Nous avons vu que ce modèle permet d’associer une représentation vectorielle de mots dans un espace aux multiples dimensions. À partir de ce modèle, nous avons extrait des représentations de tags et les avons associées aux mots les plus similaires ou sémantiquement proche du corpus à partir duquel le modèle a été entraîné. Par la suite, nous avons exploité un algorithme permettant de réduire les dimensions en vue de visualiser ces données. Cet algorithme, au nom de t-SNE (t-distributed stochastic neighbor embedding), utilise une méthode non linéaire permettant de représenter un ensemble de points d’un espace à grande dimension dans un espace réduit à deux dimensions. Nous pouvons observé

dans la figure 4.5 une instance de cet algorithme, qui projette dans l'espace sémantique engendré par Word2Vec, un corpus de tags extrait de *MovieLens-10M*. Ainsi, nous observons que les tags "Marvel" et "comics" sont sémantiquement proches, de même pour les tags "War" et "Invasion".

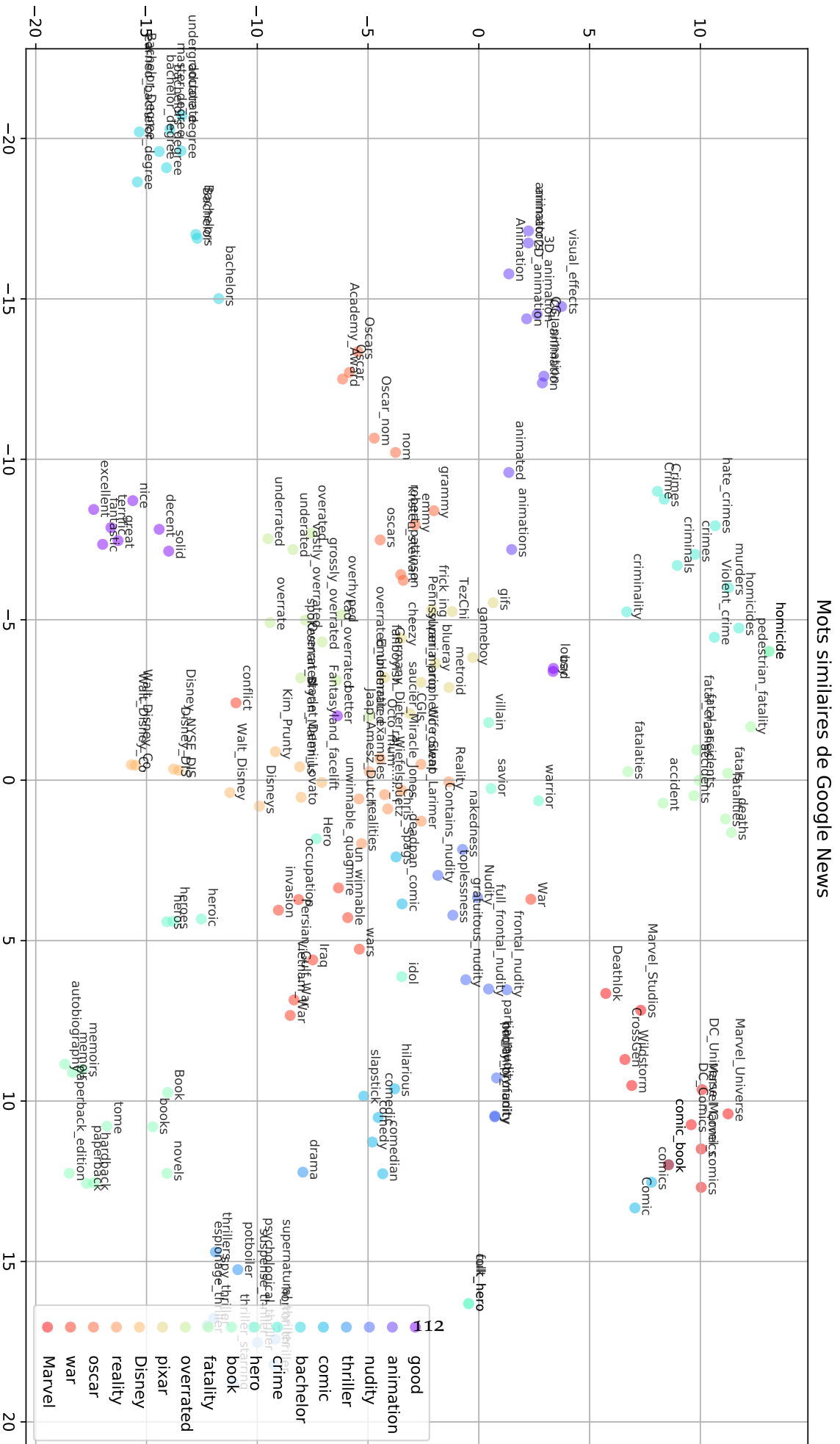


FIGURE 4-5 – MovieLens-10M : Les tags sont représentés à droite {'good', 'animation', 'nudity', 'thriller', 'comic', 'bachelor', 'crime', 'hero', 'book', 'fatality', 'overrated', 'pixar', 'Disney', 'reality', 'oscar', 'war', 'Marvel'} et leurs correspondance avec des mots similaires plongé dans un espace à k-dimension

3.5 Tableau récapitulatif

Nous présentons également le tableau 4.1 qui est un récapitulatif intégrant les différentes proportions d'utilisateurs de tags et d'items par dataset utilisé tout au long de ce manuscrit. Nous y avons aussi intégré des informations sur la période à laquelle ces entités ont été extraites. Le tableau 4.1 présente aussi les datasets dans un contexte de représentation en graphe biparti de nœuds entre l'ensemble des nœuds utilisateurs et l'ensemble des nœuds items. Une arête existe entre les deux ensembles quand un utilisateur note une ressource.

<i>Collection</i>	<i>Movielens 10M</i>	<i>Movielens 20M</i>	<i>Movielens 25M</i>
<i>Utilisateurs</i>	71567	138000	162541
<i>Nombre d'items</i>	10681	27000	62423
<i>TAS(Tag assignment)</i>	95580	465000	1093360
<i>Notes</i>	10000054	2000000	25000095
<i>Nombre de noeuds</i>	7114	20555	35363
<i>Nombre d'arcs</i>	24564	126080	210725
<i>Periode</i>	<i>December 2015</i>	<i>Jan 09,1995 - Oct, 2016</i>	<i>November 21, 2019</i>

TABLE 4.1 – Statistical details of the 10M, 20M and 25M collections from MovieLens

CONTRIBUTION I

*Nouvelles approches pour la génération de profils folksonomiques dans le domaine
de la recommandation*

Chapitre 5

MULTIPLES CENTRES D'INTÉRÊTS D'UTILISATEURS ÉVOLUTIFS

1 Introduction

DANS ce chapitre, nous abordons notre première contribution qui développe le concept de multiple centre d'intérêt évolutif. Dans l'optique de prendre en compte l'activité des utilisateurs dans la construction de profils pertinents, nous présentons une approche originale pour la recommandation de ressources. Les ressources décrites à l'issue de cette approche proviennent de collections de ressources cinématographiques extraites de la plate-forme *MovieLens*. Cette approche peut intégrer d'autres types de contenus et a la particularité de saisir les multiples centres d'intérêt des utilisateurs. Le but étant de construire des profils à même de décrire efficacement les intérêts des utilisateurs à partir de tags. Nous détaillons par la suite la variante temporelle de cette approche qui nous a permis d'intégrer notamment, la dimension temporelle des tags dans une approche de recommandation.

Ce chapitre se présente comme suit, dans la section suivante nous donnons une description générique de l'approche en rappelant certains concepts propres aux représentations sous la forme de graphes bipartis de personnalités d'utilisateurs. Ensuite, dans la section 3.1 nous détaillons l'approche et donnons le cadre expérimental des évaluations en section 4.1. La section 5 est dédiée à la variante temporelle de cette approche. La section 6 conclut ce chapitre.

2 Motivations

Dans ce qui suit, nous nous appuyeront sur les notations du tableau 5.1 où un document d_k peut être associé à une ressource r_k ou encore à un item i_k . Cela nous permettra par la suite d'illustrer certaines limitations rencontrées à la construction de profils pour la recommandation.

Les principales limitations rencontrées dans la plupart des approches, dites *classiques*, qui développent des profils d'utilisateurs basés sur des intérêts thématiques uniques, concernent les schémas de pondération. De manière générale, le poids de chaque tag dans un profil d'utilisateur ou de ressource est présenté sous la forme d'un vecteur pondéré par des tags selon les fréquences tf ou tf^*iuf comme suit :

$$w(t_i, u_m) = tf_{u_m}(t_i) \cdot iuf(t_i) \quad (5.1)$$

Dans le cas du modèle *Bm25*, les poids des tags dans un profil d'utilisateur sont calculés comme suit :

$$w(t_1, u_m) = Bm25_{u_m}(t_1) = iuf(t_1) \cdot \frac{tf_{u_m} \cdot (k_1 + 1)}{tf_{u_m} + k_1(1 - b + b \cdot |u_m| / avg(|u_m|))} \quad (5.2)$$

Notation	Référence	Formule	Mention
$tf_{u_m}(t_l)$	fréquence de tag/utilisateur	$tf_{u_m}(t_l) = count(t_l, u_m)$	Le nombre de fois que le tag t_l est utilisé par l'utilisateur u_m
$tf_{d_k}(t_l)$	fréquence de tag/document	$tf_{d_k}(t_l) = count(t_l, d_k)$	Le nombre de fois que le tag t_l a été utilisé pour annoter le document d_k
$idf(t_l)$	fréquence inverse tag/documents	$idf(t_l) = \log \frac{N}{n_d(t_l)}$; N : Nombre de documents. $n_d(t_l)$: Nombre de documents annotés avec le tag t_l .	Le nombre de fois que le tag t_l est utilisé à travers toute la collection D
$iuf(t_l)$	fréquence inverse tag/utilisateurs	$iuf(t_l) = \log \frac{M}{n_u(t_l)}$; M : Le nombre total d'utilisateurs. $n_u(t_l)$ est le nombre d'utilisateurs ayant annoté avec t_l	Le nombre de fois que le tag t_l est utilisé parmi tous les utilisateurs U

TABLE 5.1 – Rappel des différents schémas de pondération associés aux tags, dans ce tableau un document peut aussi bien faire référence à une ressource qu'à un item.

Comme l'ont déjà souligné les travaux de *Cai et al.* [34], les schémas de pondération tf ou tf^*iuf ou encore $BM25$ utilisés pour pondérer un tag dans un profil révèlent être insuffisant pour indiquer dans quelle mesure un utilisateur présente de l'intérêt pour les tags décrivant son profil. Le poids de chaque tag dans un profil d'utilisateur est censé représenter le degré de préférence de l'utilisateur u_i par rapport au tag t_l . Dans un profil de ressource, il reflète la pertinence d'un tag par rapport à une ressource ; il mesure avec quelle précision le tag décrit la ressource. Cependant, si certains utilisateurs actifs annotent les ressources plus fréquemment que certains utilisateurs moins actifs, les poids des tags des utilisateurs plus actifs seront plus élevés que ceux des utilisateurs moins actifs. Soit $u_1 = (t_1 : 124, t_2 : 177, \dots, t_3 : 154)$, le profil d'un utilisateur ayant annoté 500 ressources (t_1, t_2, t_3 sont des tags) et soit $u_2 = (t_1 : 10, t_2 : 20, \dots, t_3 : 24)$ le profil d'un utilisateur moins actif qui a annoté seulement 30 ressources. u_2 devrait avoir autant d'intérêt pour les trois ressources que u_1 , du fait que u_2 a utilisé ces 3 tags de manière plus proportionnelle que u_1 et par conséquent, la probabilité que u_2 soit intéressé par ces ressources devrait être aussi élevée que celle de u_1 . En ce qui concerne les ressources, les utilisateurs actifs influencent la fréquence à laquelle les ressources ont été annotées, les rendant plus populaires en raison de la tendance disproportionnée des utilisateurs actifs à annoter ces ressources. Cela affecte donc le classement total de ces ressources, subissant une distorsion systématique causée par des utilisateurs plus actifs que d'autres. Nous en concluons que, tf n'est pas suffisamment représentatif des préférences de l'utilisateur. De même, dans un profil de ressource, tf n'indique pas dans quelle mesure le tag est pertinent pour la ressource. D'autre part, l'utilisation de $tf-iuf$ et idf comme schéma de pondération dans la construction du profil utilisateur n'est pas souhaitable pour indiquer la préférence d'un utilisateur pour un tag. iuf peut aider à distinguer un utilisateur d'un autre mais ne reflète pas l'intérêt d'un utilisateur pour un tag particulier. Par conséquent, cette limitation s'applique également à idf .

Le tableau 5.2 illustre ce propos, pour un utilisateur u_i d'un système qui compte 1000 utilisateurs, si nous adoptons tf^*iuf pour mesurer les poids des tags dans le profil de cet utilisateur, alors la valeur iuf pour le tag t_3 sera plus élevée que celle des autres tags malgré la fréquence de tags inférieure. Ce résultat n'est pas raisonnable. Comme tf et $iuf(idf)$ sont des variables de la plupart des modèles de pondération en RI (VSM, BM25), les limitations mentionnées ci-dessus affectent tous ces modèles de pondération. *Cai et*

fréquences	tags	t_1	t_2	t_3
tf		38	41	6
n_t		500	820	1
iuf		0.3	0.08	3
tf^*iuf		11.4	3.28	18

TABLE 5.2 – Limitations des schémas de pondération tf^*iuf et iuf pour un utilisateur u_i

al., [33] arrivent à lever ces limitations en proposant une approche qui prend en compte la proportion d'utilisateurs ayant annoté un tag spécifique. Ils pondèrent les profils des utilisateurs et des ressources en fonction de la fréquence normalisée des termes (nTF). Un profil d'utilisateur est représenté selon $\vec{U}_i = (v_{i,1}, v_{i,2}, \dots, v_{i,n})$, où un tag est pondéré selon l'équation suivante :

$$v_{i,l} = nTF(u_i, t_l) = \frac{\text{count}(t_l, u_i)}{n_r(u_i)}$$

Où $v_{i,l}$ est le degré de préférence de l'utilisateur u_i par rapport à t_l , n est le nombre total de tags utilisé par u_i , $\text{count}(t_l, u_i)$ est le nombre de fois que l'utilisateur u_i utilise le tag t_l pour annoter des ressources, $n_r(u_i)$ est le nombre de ressources annotées par l'utilisateur u_i .

De manière similaire, pour une ressource r_j , nous considérons un profil de ressource formalisé de la manière suivante : $\vec{R}_j = (w_{j,1}, w_{j,2}, \dots, w_{j,m})$, en associant au tag la pondération suivante :

$$w_{j,l} = nTF(r_j, t_l) = \frac{\text{count}(t_l, r_j)}{n_u(r_j)}$$

Où $w_{j,l}$ indique le degré de popularité associé à l'utilisation du tag t_l pour décrire r_j , N est le nombre de tags utilisés pour décrire la ressource r_j , $\text{count}(t_l, r_j)$ est le nombre d'utilisateurs utilisant le tag t_l pour annoter la ressource r_j , et $n_u(r_j)$ est le nombre total d'utilisateurs qui ont utilisé des tags pour annoter la ressource r_j .

2.1 Profilage des centres d'intérêts multiples des utilisateurs

Afin de tirer profit des solutions apportées aux limitations mentionnées, nous les mettons en pratique dans une représentation étendue des multiples centres d'intérêts des utilisateurs. En particulier, dans [237], Yeung et al., révèlent une approche pour rassembler les tags similaires lors de l'étude de la génération de profils d'utilisateurs à partir de folksonomies. La principale limitation rencontrée dans l'approche de [237] est liée à la question de la pondération des tags. Précisément, cette approche est censée regrouper les ressources partageant des tags communs au moyen d'une méthode de *clustering*. Les tags associés à chaque cluster sont censés représenter un sujet d'intérêt pour un utilisateur. Cette façon simpliste de regrouper les ressources, afin de différencier les intérêts des utilisateurs, peut conduire à des regroupements inappropriés. Nous supposons que la prise en compte de l'importance d'un tag par rapport aux activités de l'utilisateur (comme discuté précédemment) peut conduire à une distribution différente

des poids des arêtes qui relient les ressources entre elles, ce qui pourrait aboutir à un meilleur partitionnement des ressources et donc à une meilleure inférence des intérêts des utilisateurs.

Le tableau 5.3 illustre une limitation concernant cette approche. Considérons un utilisateur donné qui a annoté 3 ressources r_1, r_2 et r_3 avec les tags $\{t_1\}, \{t_2\}, \{t_3\}$. Par exemple, pour ce même utilisateur, t_1 est un tag commun aux ressources r_1 et r_2 , donc de l'arête $\{r_1, r_2\}$. De plus, nous supposons que les préférences de cet utilisateur concernant les tags sont : $u_i = (t_1 = 0.5; t_2 = 0.25; t_3 = 0.75)$.

Poids des arêtes	$\{r_1, r_2\}$	$\{r_2, r_3\}$	$\{r_1, r_3\}$
Tags en communs	$\{t_1\}$	$\{t_2\}$	$\{t_3\}$
Nbre de tags en commun	1	1	1
$\sum v_{i,l}$	0.25	0.75	0.5

TABLE 5.3 – Illustration de la limitation associée aux poids des arêtes dans l'approche de Yeung et al., [237]

Tandis que les poids des arêtes (c'est-à-dire $\{r_i, r_j\}$) restent les mêmes en utilisant un simple comptage des tags en communs comme poids des arêtes comme dans [237], la somme du degré de préférence de l'utilisateur pour chaque tag en commun affecte considérablement la valeur de distribution des poids des arêtes. Notre intuition repose sur le fait qu'une telle différence dans la distribution du schéma de pondération des arêtes peut produire un partitionnement différent des *clusters*. Par conséquent, nous considérons que les poids des arêtes sont des informations supplémentaires efficaces qui devraient être intégrées dans le processus de *clustering*; nous pensons qu'un tel facteur peut générer des profils d'utilisateurs plus précis puisque le degré de préférence de l'utilisateur $v_{i,l}$ sur ses tags communs est une mesure personnalisée qui révèle à quel point un utilisateur est intéressé par un tag particulier.

3 Description générale de l'approche

Le point de départ de notre approche consiste en une folksonomie F définie comme un 4-uplet $F = (U, T, R, A)$, où U est l'ensemble des utilisateurs annotant les ressources de l'ensemble R avec $U = \{u_1, u_2, \dots, u_m\}$ où chaque u_i est un utilisateur; T est l'ensemble des tags que comprend le vocabulaire exprimé par la folksonomie $T = \{t_1, t_2, \dots, t_l\}$; R est l'ensemble des ressources annotées par les utilisateurs $R = \{r_1, r_2, \dots, r_n\}$; $A = \{u_m, t_l, r_n\} \in U \times T \times R$ est l'ensemble des annotations de chaque tag t_l à une ressource r_n par un utilisateur u_m .

Nous définissons le profil d'un utilisateur u_i comme étant le vecteur $\vec{u}_i = (v_{i,1}, \dots, v_{i,l})$, où $v_{i,l}$ désigne le degré de préférence de l'utilisateur u_i sur le tag t_l . Il est défini comme le rapport du nombre de fois que le tag t_l a été utilisé par u_i pour annoter une ressource par le nombre de ressources annotées par cet utilisateur, c'est à dire :

$$v_{i,l} = nTF(u_i, t_l) = \frac{\text{count}(t_l, u_i)}{n_r(u_i)}. \quad (5.3)$$

Où, $n_r(u_i)$ est le nombre de ressources annotées par l'utilisateur u_i .

De même, pour un profil de ressource nous définissons le vecteur $\vec{r}_k = (w_{k,1}, w_{k,2} \dots w_{k,l})$. Où $w_{k,l}$ est le nombre de fois qu'une ressource r_k a été annotée avec le tag t_l , formellement :

$$w_{j,l} = nTF(r_j, t_l) = \frac{\text{count}(t_l, r_j)}{n_u(r_j)} \quad (5.4)$$

L'approche que nous proposons exploite une personomie d'un utilisateur donné $u \in U$ que nous définissons comme P_u , la restriction de la folksonomie à cet utilisateur, c'est-à-dire $P_u \leftarrow (T_u, R_u, A_u)$ avec A_u comme étant l'ensemble des annotations de l'utilisateur u : $A_u = \{(t, r) | (u, t, r) \in A\}$, T_u est l'ensemble des tags utilisateur : $T_u = \{t | (t, r) \in A_u\}$, et R_u est l'ensemble des ressources de l'utilisateur : $R_u = \{r | (t, r) \in A_u\}$. La figure 5.1 illustre la personomie P_u .

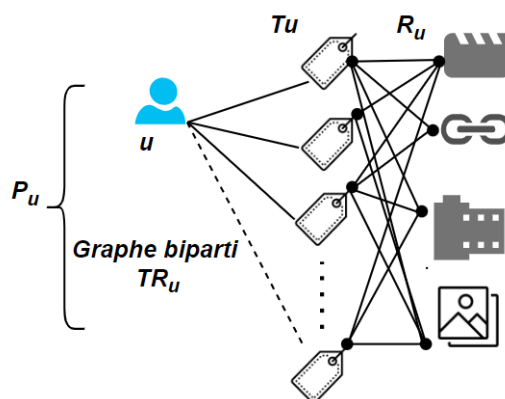


FIGURE 5.1 – Illustration de la personomie P_u

Formellement, le graphe biparti TR_u associé à la personomie P_u peut être défini comme suit : $TR_u = (T_u \cup R_u, E_{tr})$, $E_{tr} = \{(t, r) | (t, r) \in A_u\}$. Ce graphe (visible dans la figure 5.1) est construit à partir de sommets (nœuds) représentant les ressources R_u et les tags T_u associés à l'utilisateur u_i . Une arête existe entre un sommet de la classe de ressources et un sommet de la classe des utilisateurs si et seulement si un tag est assigné à une ressource par cet utilisateur. Nous proposons de pondérer ce graphe en fonction de l'activité de l'utilisateur. Une projection 1-mode est un processus qui consiste à sélectionner une classe de nœuds, et à relier deux nœuds s'ils sont connectés au même nœud de l'autre classe. Par exemple, la projection du graphe 1-mode sur l'ensemble des ressources reliera deux nœuds de ressources s'ils partagent un même tag. Le lien peut être pondéré par le nombre de tags qu'ils partagent. Au lieu de simplement compter le nombre de tags parmi les ressources communes, nous proposons de regarder au-delà de ce simple comptage afin de propager le degré de préférence de l'utilisateur sur un tag en l'exploitant dans une approche de *clustering*. Nous détaillons notre approche

dans la section suivante.

3.1 Approche détaillée

L'objectif de notre approche est de construire des profils d'utilisateurs à "granularité fine", correspondant aux larges éventails de centres d'intérêts des utilisateurs. Le but est de fournir ainsi un accès personnalisé aux ressources pour l'utilisateur à partir des informations extraites de ses tags. Le principal problème abordé dans notre méthode est centré sur la relation entre les ressources qui sont décrites en utilisant les mêmes tags. Une caractéristique majeure de notre travail réside dans le fait que cette approche peut être facilement étendue pour construire des profils de ressources.

En particulier, pour identifier les différents centres d'intérêts des utilisateurs, nous avons étendu l'approche de *Yeung et al.*, [237]. Ces derniers [237] exploitent un algorithme agglomératif hiérarchique gourmand dit "greedy" [80, 159, 158, 157] sur le graphe 1-mode représentant les ressources de l'utilisateur. Chaque *cluster* généré représente alors un centre d'intérêts utilisateur, il est donc représenté par un ensemble de tags pondérés. Nous définissons par la suite une signature de *cluster* comme étant les tags les plus pertinents décrivant ce *cluster*, nous supposons que le sous-ensemble des tags les plus pertinents liés aux ressources fournissent une meilleure représentation des *clusters*. À la différence de ce que proposent les travaux de *Yeung et al.*, [237], afin de parfaire la création des profils d'utilisateurs, nous exploitons des fréquences normalisées de tags (nTF) pour pondérer les tags, mais aussi les arêtes (liens) entre les ressources.

Ainsi, le point de départ pour la construction de profils d'utilisateurs est le graphe de ressources R_u . Nous proposons de propager les préférences de l'utilisateur à travers le graphe représentant les arêtes du graphe 1-mode R_u .

Cela conduit à la pondération des arêtes du graphe de ressources R_u , comme suit :

$$w_u(t_l, r_k) = v_{u,l} = \frac{\text{count}(t_l, u)}{n_r(u)} \quad (5.5)$$

Où $\text{count}(t_l, u)$, représente le nombre de fois que le tag t_l est utilisé par l'utilisateur u . $n_r(u)$ est le nombre total de ressources annotés par utilisateur u . Nous effectuons ensuite une projection 1-mode du graphe biparti TR_u sur la classe de sommets que nous voulons étudier (dans notre cas, la projection sur les ressources de l'utilisateur R_u). Le résultat de cette projection est un graphe $\{R_u \times R_u\}$ des ressources utilisateur, où les arêtes sont pondérées en additionnant les poids des tags utilisateur qu'ils partagent, il est donné par la formule suivante :

$$w(r_j, r_k) = \sum_{t_l \in \{t_{r_j} \cap t_{r_k}\}} [w_u(t_l, r_j) + w_u(t_l, r_k)] \quad (5.6)$$

$$= \sum_{l: t_l \in \{t_{r_j} \cap t_{r_k}\}} v_{u,l} \quad (5.7)$$

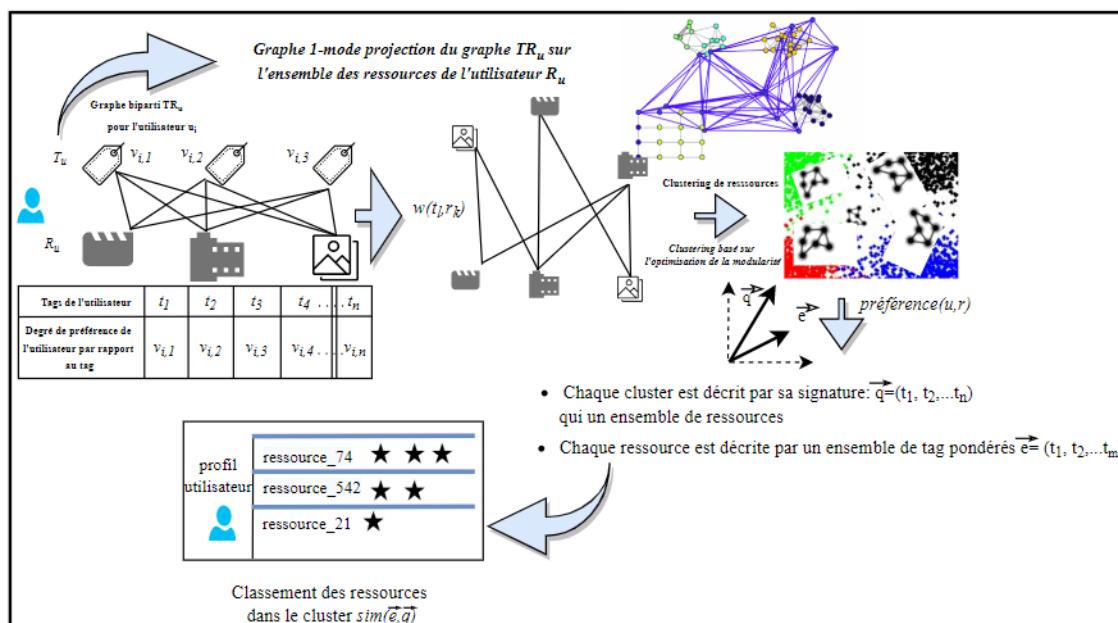


FIGURE 5.2 – Étapes générales de notre approche de personnalisation pour la génération de profils d'utilisateurs. D'abord les tags sont pondérées selon le degré de préférence de l'utilisateur par rapport au tag noté $v_{i,t}$, ensuite le graphe 1-mode est généré par projection du graphe TR_u sur l'ensemble des ressources de l'utilisateur R_u . Après avoir procédé à un clustering sur l'ensemble des noeuds de ce graphe, des signatures propres à chaque cluster sont extraites et comparées à l'ensemble des tags qui apparaît dans chaque ressource. À l'issue de cette comparaison les ressources sont classées selon les préférences de l'utilisateur. Au final, le profil de l'utilisateur est un cluster unique ou un ensemble de clusters contenant une liste de ressources classées par ordre de préférence.

Où $\{t_{r_j} \cap t_{r_k}\}$ représentent les tags en commun entre r_j et r_k annotés par l'utilisateur u . Nous effectuons ensuite une optimisation de la *modularité* sur le graphe des ressources $\{R_u \times R_u\}$, en utilisant l'algorithme décrit dans [157]. L'utilisation de cet algorithme figure parmi les méthodes d'apprentissage non-supervisé utilisées dans les graphes pondérés et proposés par Newman et al., [157]. L'efficacité et la performance de cet algorithme ont déjà été prouvées, en particulier, pour les grands graphes de terrains. Pour chaque communauté (*cluster*) obtenu via la division finale du graphe, on obtient un ensemble de ressources propres à chaque *cluster*. Chaque cluster est caractérisé par une signature qui correspond à un ensemble des tags pondérées associées aux ressources.

3.2 Signature de clusters

La signature d'un cluster est l'ensemble des tags qui permet de décrire de manière fidèle les ressources associées à ce *cluster*. Une signature naïve consiste à sélectionner tous les tags associés aux ressources du *cluster*. L'une des problématiques sous-jacente que nous abordons est la possibilité de sélectionner le plus petit ensemble de tags décrivant au mieux l'ensemble des ressources du cluster. Pour ce faire, différentes solutions peuvent être alors proposées. On peut sélectionner ces tags en fonction du nombre ou

de la proportion de ressources que ces tags retournent, mais aussi en fonction de leur poids. Nous estimons que cet ensemble de tags est sélectionné d'après la mesure de poids attribuée à chaque tag. Cet ensemble représente les tops tags ρ . Nous verrons par la suite que ce paramètre affectera la précision du profil dans la modélisation des intérêts de l'utilisateur. En général, pour une petite valeur de ρ , seuls les tags les plus populaires (pondérées) seront inclus dans la signature, alors qu'une grande valeur de ρ , inclura plus de tags. Nous avons détaillé les algorithmes de *clustering* basés sur l'optimisation de la *modularité* par sélection de critères dans l'annexe I que nous avons intégrée à la fin de ce manuscrit.

Comme nous l'avons décrit précédemment, afin de s'assurer de la bonne qualité des *clusters* générés par l'algorithme, nous utilisons une mesure quantitative : la *modularité* [159], qui est définie comme le nombre d'arêtes intraclusters, comparé au nombre moyen d'arêtes intraclusters obtenu après un brassage aléatoire du graphe. Soit $e_{i,j}$ la fraction d'arêtes reliant les nœuds d'un groupe i à un autre groupe j . Soit $a_i = \sum_j e_{ij}$ la sommes de ces arêtes. La *modularité* Q , est définie par l'équation [159, 158] :

$$Q = \sum (e_{ii} - a_i^2) \quad (5.8)$$

En commençant par un état initial dans lequel chaque sommet (dans notre cas chaque sommet appartenant au graphe de ressources) est le seul membre de l'une des communautés n et après un certain nombre d'itérations, les communautés sont formés par propagation du degré de préférence de l'utilisateur. Nous appliquons cet algorithme pour rassembler les ressources en fonction des centres d'intérêts des utilisateurs d'une collection de données extraite depuis *MovieLens*.

Dans la section suivante, nous examinons à travers une série d'expérimentation notre approche basée sur le Degré de Préférence de l'Utilisateur Normalisé et propagé par les tags des utilisateurs que l'on nomme DPN_{Tu} et qui nous permettra une meilleure compréhension du processus permettant de rassembler puis de classer des ressources similaires en fonction du centre d'intérêt de l'utilisateur. Cela nous permettra aussi une meilleure approximation du paramètre ρ .

4 Expérimentations

Nous avons conduit une série d'expériences sur un échantillon de *MovieLens*, à savoir l'ensemble de données *MovieLens-20M*. Ce dataset est détaillé dans le chapitre 4. Nous avons extrait à partir de cet échantillon (*MovieLens-20M*) 100 personnalités d'utilisateurs ayant au moins 100 tags dans leurs activités d'annotation afin de s'assurer qu'il y ait suffisamment de tags de manière à ce que le *clustering* soit significatif.

Nous évaluons par la suite la performance de l'approche personnalisée en fonction de la pertinence graduée des ressources par utilisateur. Le profil est représenté par de multiples ensembles de tags pondérées, des clusters. Dans ce qui suit nous évaluons :

- L'impact du nombre de tags (i.e. ρ) à inclure dans la signature d'un *cluster*.

- L'impact de la prise en compte des préférences des utilisateurs nommées dans la construction de profils.
- La précision d'une représentation à multiples centre d'intérêts par rapport à une représentation vectorielle classique basée sur des intérêts thématiques uniques.

Afin d'évaluer ces différents points, nous avons comparé notre approche avec celle de *Yeung et al.*, [237] et *Cai et Li.*, [33]. Nous considérons les signatures des *clusters* comme étant des requêtes. Notre but est de retourner les ressources de la collection associées à la requête, idéalement celles qui appartiennent seulement au *cluster* correspondant à l'un des centres d'intérêts effectifs de l'utilisateur et d'éviter d'obtenir des ressources d'autres *clusters* considérés comme non pertinents. Par conséquent, nous supposons que chaque ressource est représentée initialement comme un vecteur de tags pondérés, c'est-à-dire :

$$\vec{r} = (t_1 : w_1, t_2 : w_2, t_2 : w_2, \dots, t_N : w_N). \quad (5.9)$$

Chaque tag est pondéré en utilisant le degré de préférence d'un utilisateur sur un tag, qui est calculé de la manière suivante :

$$t_1 : w_1 = t_1 : nTF(u_i, t_1) = \frac{\text{count}(t_1, u_i)}{n_r(u_i)} \quad (5.10)$$

Où N est le nombre total de tags de l'espace de tags, $\text{count}(t_1, u_i)$ est le nombre de fois que l'utilisateur u_i consomme le tag t_1 pour annoter les ressources, $n_r(u_i)$ est le nombre de ressources annotées par l'utilisateur u_i .

De même, nous rappelons que chaque signature de cluster considérée comme une requête, est représentée dans l'espace des tags. Les ressources retournées sont classées par ordre décroissant de leur score de similarité (cosinus) avec les clusters. Pour évaluer les performances, nous utilisons une précision commune. Précisément, pour un cluster donné c_u^j , (cluster c^j de l'utilisateur u , $j = \{1..#C_u\}$), soit D_{c^j} l'ensemble des ressources dans le cluster, et $D_{x(c^j)}$ l'ensemble de ressources retournées par la signature du cluster.

La précision et le rappel du système sur C sont définis comme suit :

$$\text{Précision}(c^j) = \frac{|D_{x(c^j)} \cap D_{c^j}|}{|D_{x(c^j)}|} \quad (5.11)$$

$$\text{Rappel}(c^j) = \frac{|D_{x(c^j)} \cap D_{c^j}|}{|D_{c^j}|} \quad (5.12)$$

La précision à @10 correspond à la précision obtenue si l'on ne prend en compte que les 10 premières ressources de l'ensemble des ressources du cluster. De même, @20 pour les 20 premières ressources. La précision et le rappel sont ensuite moyennés sur tous les clusters pour un utilisateur donné, puis sur tous les utilisateurs pour obtenir une précision moyenne. En plus de ces mesures basées sur des ensembles, à savoir le rappel et la précision, nous avons également évalué la performance du classement personnalisé des ressources en utilisant la mesure du NDCG : (*Normalized Discount Cumulative Gain*).

4.1 Résultats et discussions

Nous avons comparé notre approche avec deux approches de base, *Yeung et al.*, [237] et *Cai et Li* [33]. L'approche de *Yeung et al.*, [237], exploite les multiples centre d'intérêts des utilisateurs, elle permet de regrouper les ressources partageant des tags en communs, mais n'intègre pas l'activité de l'utilisateur. Les tags associés à chaque *cluster* sont juste censés représenter un sujet d'intérêt pour un utilisateur. Cette façon simpliste de regrouper les ressources pour différencier les intérêts des utilisateurs, peut conduire à un *clustering* incohérent. Par ailleurs, l'approche de *Cai and Li* [33] représentent les profils d'utilisateurs sous la forme de vecteurs uniques de tags pondérés, exploitant une fréquence normalisée de tag (NTF) mais ne prenant pas en compte les multiples centre d'intérêts des utilisateurs. Dans ce qui suit, nous présentons les impacts de chaque approche.

4.1.1 Impacte du degré de préférence de l'utilisateur

Afin d'évaluer l'impact de la prise en compte de l'activité des utilisateurs dans la construction de profils utilisateurs, nous avons comparé notre approche avec celle de *Yeung et al.*, [237], nous avons fait varier dans un premier temps la valeur de ρ , correspondant à la proportion de tags à considérer parmi les valeurs de ρ à 20%, 40% et 60%. Le but d'une telle démarche est de déterminer à partir de différentes signatures la proportion de tags nécessaire pour décrire avec plus de précision les *clusters*. À l'issue de cette expérimentation, nous pourrions synthétiser l'impact du degré de préférence de l'utilisateur, en supposant que dans notre approche un nombre minimum de tags peut constituer une bonne signature pour un *cluster* et peut également récupérer un plus grand nombre de ressources en fonction du degré de préférence de l'utilisateur par rapport au multiples centre d'intérêts de l'utilisateur. Le tableau 5.4 compare la précision des deux approches.

Valeur de ρ	Précision			
	Yeung et al. [237]		DPN_u	
	$p@10$	$p@20$	$p@10$	$p@20$
$\rho = 20\%$	0,117	0,133	0,147	0,156
$\rho = 40\%$	0,138	0,145	0,192	0,224
$\rho = 60\%$	0,198	0,236	0,265	0,269

TABLE 5.4 – Comparaison de notre approche avec la baseline *Yeung et al.*, [237]

Après avoir calculé les similarités entre les signatures des clusters \vec{q} et les vecteurs de tags \vec{e} correspondant à chaque ressources, dénoté $s(m_k)$. Nous sélectionnons ensuite les ressources qui atteignent un certain seuil δ de similarité ($0 \leq \delta < 1$). Une interprétation possible des résultats du tableau 5.4 implique que la prise en compte de l'activité de l'utilisateur à travers un schéma de pondération normalisé des tags permet d'identifier les intérêts de l'utilisateur, améliore la classification des ressources et génère des clusters plus précis et plus représentatifs des divers intérêts des utilisateurs.

4.1.2 Impact du clustering

Le but de cette expérimentation est de prendre connaissance de la nature des intérêts de l'utilisateur lors de la construction des profils. Sont-ils uniques? sont ils mieux décrits à travers des centres d'intérêts multiples?

A cet égard nous comparons notre approche avec celle de *Cai et Li* [33]. Plus précisément, si nous considérons un profil comme un ensemble de tags à intérêts uniques, serait-il plus pertinent que ce profil soit représenté par plusieurs sous-ensembles de tags traduisant ses multiples centres d'intérêts.

Nous répétons les mêmes séries d'expérimentations et les comparons avec l'approche de *Cai et Li* [33]. Par conséquent, la proportion ρ représentera 20% des tags qui apparaissent le plus parmi tous les tags de tous les utilisateurs annotant des ressources, alors que dans notre approche cette proportion est toujours égale à un ensemble restreint de tags dans chaque cluster et représentant ce que nous considérons comme une signature unique propre à chaque centre d'intérêts. En d'autres termes, nous comparons la représentation par profils d'utilisateurs uniques avec une représentation intégrant des centres d'intérêts multiples. On pourrait également évaluer la pertinence de la signature du cluster lorsqu'on l'utilise comme requête pour retourner les ressources utilisateurs dans le but d'identifier lesquelles correspondent le plus aux centres d'intérêts identifiés. Dans ce cas, on se demande si la signature du cluster constituée par cet ensemble de tags pondérés est suffisamment significatif pour retourner l'ensemble des ressources de cet utilisateur? toutes les ressources de la collection?

Nous présentons dans le tableau 5.5, un résumé de la précision globale par profil utilisateur. Nous avons considéré à cette fin, un score de précision global calculé d'abord sur l'ensemble des ressources utilisateur puis sur l'ensemble des ressources de la collection; nous identifions la proportion de tags minimale dans une signature, c'est-à-dire l'ensemble de tags pondérés qui renvoie les ressources de l'utilisateur à son centre d'intérêt parmi les ressources de ce même utilisateur ou parmi les ressources de toute la collection.

Recouvrement	Précision					
	$\rho = 20\%$		$\rho = 40\%$		$\rho = 60\%$	
	$p@10$		$p@10$		$p@10$	
Value of ρ	Cai et Li [33]	DPN _{Tu}	Cai et Li [33]	DPN _{Tu}	Cai et Li [33]	DPN _{Tu}
$\forall r \in R_u$	0,112	0,147	0,182	0,192	0,231	0,265
$\forall r \in R_{collection}$	0,078	0,088	0,093	0,0941	0,174	0,168

TABLE 5.5 – Ce tableau compare les précisions pour différentes valeurs des signatures ρ parmi toutes les ressources d'un utilisateur, parmi l'ensemble des ressources de la collection. Comparaison avec *Cai et al.*, [33].

Les résultats suggèrent des scores de précision @10 favorables à notre approche, dans le sens où la signature que nous générons et qui se situe autour des 40% (pour une

couverture optimale de toutes les ressources des clusters) permet de retourner plus de ressources comparées aux signatures extrait de l'approche de Cai et Li [33]. À ce propos, nous avons estimé la proportion de tags dans ces signatures à 100%, soit la totalité des tags formés par l'exploitation de vecteurs de tags à intérêts uniques. Cela prouve qu'un sous-ensemble pondéré de tags utilisé comme signature décrit de manière plus fine le large éventail d'intérêts que peut décrire un utilisateur.

4.1.3 *Évaluation du classement personnalisé des ressources par utilisateur*

Les ressources contenues dans chaque profil d'utilisateur sont classées en fonction des préférences de l'utilisateur. Afin d'évaluer le classement de ces ressources, un moyen couramment utilisé est de traiter l'information extraite des notes (ratings) des utilisateurs.

4.1.4 *Évaluation du classement de pertinence*

Nous définissons une fonction de notation, $f_{u_i}(m_i)$, qui retourne la valeur de notation d'une ressource m_i donnée par un utilisateur u_i . La valeur maximale d'une note est de 5, elle indique une très bonne appréciation de l'utilisateur concernant le film, tandis que la valeur minimale est 1. Les notes varient de 1 à 5. Toutes les ressources contenues dans les clusters et représentant les profils utilisateurs sont décrites par un ensemble de tags pondérés dans un espace vectoriel. Nous calculons la similarité entre ces ressources et les signatures des clusters afin de classer les ressources par similarité. Nous exploitons les notes des utilisateurs que nous considérons comme jugements de pertinence aidant à évaluer le classement des ressources par similarité par rapport à la pertinence graduelle des ressources notées par l'utilisateur. Dans notre cas, pour chaque signature de cluster représentant l'intérêt de l'utilisateur envers un topic obtenu par notre approche, une liste classée de ressources est générée en fonction de la similarité des ressources contenu dans le cluster et la similarité avec la signature du cluster.

Le principe du DCG (Discount Cumulative gain [110]) stipule que les ressources pertinentes apparaissant au bas d'un résultat de recherche doivent être pénalisés, car la valeur de pertinence graduée est réduite de manière logarithmique et proportionnelle à la position du résultat. Plus la position d'une ressource pertinente est basse, moins elle sera pertinente pour l'utilisateur. Le score de pertinence de chaque ressource est utilisé comme mesure de la valeur acquise pour sa position classée dans le résultat et le gain est additionné à la position classée de 1 à n . Le DCG idéal (IDCG) étant les ressources qui sont classés d'après les notes de l'utilisateur.

Ainsi, nous avons calculé les valeurs de NDCG pour les ressources extraites de la signature des clusters en appliquant respectivement $\rho = 20\%$, $\rho = 40\%$, $\rho = 60\%$. Nous comparons les résultats avec *Yeung et al.*, [237]. Le tableau 5.6 résume les résultats du calcul du NDCG@10 et NDCG@20 pour les deux approches pour différentes valeurs de ρ .

Nous observons lorsque $\rho = 40\%$ que la valeur du NDCG@10 de notre approche qui exploite DPN_{Tu} est à 0.168 quand la valeur du NDCG@10 de l'approche de *Yeung et al.*, atteint 0.154. De même, NDCG@20 associée à notre approche atteint 0.25, alors que

Value of ρ	NDCG			
	Yeung [237]		DPN _{Tu}	
	NDCG@10	NDCG@20	NDCG@10	NDCG@20
$\rho = 20\%$	0,112	0,192	0,117	0,205
$\rho = 40\%$	0,154	0,245	0,168	0,256
$\rho = 60\%$	0,184	0,389	0,198	0,397

TABLE 5.6 – NDCG@10 et NDCG@20 pour différentes valeurs des signatures ρ . Comparaison avec [237]

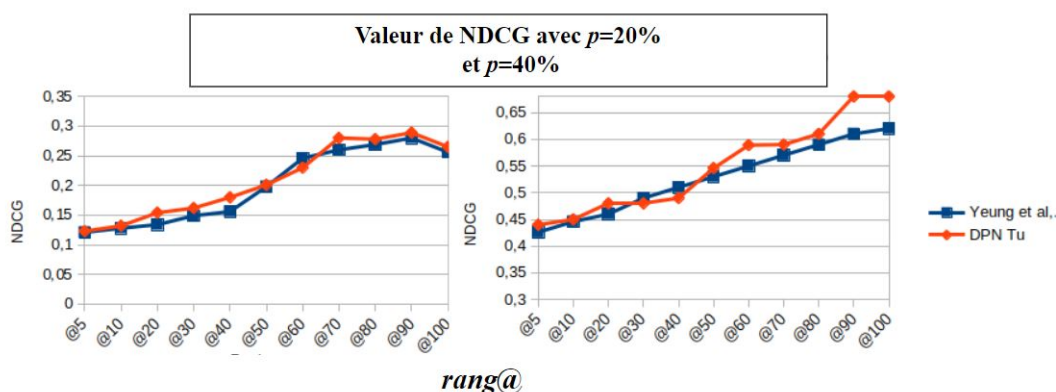


FIGURE 5.3 – NDCG@10 à NDCG@100 pour les approches illustrées par l'exploitation de DPN_{Tu} et Yeung et al., [237] $\rho = 20\%$ $\rho = 40\%$

L'approche de Yeung et al., [237] obtient une valeur NDCG@20 égale à 0.24. Les deux approches sont aussi mises en comparaison dans la figure 5.3 qui détaille la distribution de la mesure du NDCG de @10 à @100 et met en évidence l'efficacité des signatures de *clusters* (ensemble de tags pondérés) exploitées pour classer les ressources au sein de topics d'intérêts. Dans la section suivante, nous détaillons la manière d'identifier pour un utilisateur donné, les centres d'intérêts des utilisateurs indépendamment des *clusters*, en exploitant simplement les signatures.

4.1.5 Évaluation globale

Afin d'identifier les centres d'intérêts des utilisateurs indépendamment des *clusters* pour un utilisateur donné, nous calculons la similarité entre les signatures du cluster et toutes les ressources utilisateur, c'est-à-dire $\forall r \in R_u$ puis sur toutes les ressources de la collection, ie. $\forall r \in R_{Collection}$.

Le but d'une telle évaluation est d'identifier à partir d'une grande quantité de ressources celles qui font partie des ressources les plus importantes correspondant aux préférences des utilisateurs.

Le **tableau 5.7** résume les valeurs des $nDCG$ à $@10$ pour les deux approches (la notre DPN_{Tu}) comparée à celle de *Yeung et al.*, [237]), lorsque $\rho = 20\%$, $\rho = 40\%$, et $\rho = 60\%$.

Recouvrement	NDCG					
	20%		40%		60%	
Value of ρ	NDCG@10		NDCG@10		NDCG@10	
	<i>Yeung</i> [237]	DPN_{Tu}	<i>Yeung</i> [237]	DPN_{Tu}	<i>Yeung</i> [237]	DPN_{Tu}
$\forall r \in R_u$	0.112	0.117	0.154	0,168	0.184	0.19
$\forall r \in R_{collection}$	0.028	0.033	0.068	0.071	0.088	0.091

TABLE 5.7 – Synthèse des valeurs $NDC@10$, comparaison avec [237]. Les valeurs sont calculées parmi toutes les ressources d'un utilisateur, parmi l'ensemble des ressources de la collection pour différentes valeurs de ρ

Le résultat de cette évaluation nous montre que notre approche construit des profils de meilleure précision que ceux générés par l'approche de [237]. En effet, du fait d'avoir propagé le degré d'intérêt d'un utilisateur pour un tag en amont du clustering impacte le classement des ressources au sein de ces profils (*clusters*). De plus, le fait d'avoir pondéré les tags associés aux ressources a permis d'affiner la précision des signatures des *clusters*. Ce qui se traduit par une amélioration des scores de ranking $ndcg@10$ et $ndcg@20$.

4.2 Conclusion partielle

En prenant comme point de départ les travaux décrits dans [237, 33], nous avons établi une nouvelle méthode pour construire des profils d'utilisateurs basés sur des folksonomies. L'originalité de cette méthode tient au fait que l'on prend en compte le degré de préférence de l'utilisateur u_i à l'égard du tag t_l et donc à la précision des tags d'utilisateur concernant une ressource. Ce degré de préférence est obtenu grâce à la mesure normalisée proposée dans [33] et qui a été utilisée pour pondérer les sommets de la classe T_u représentant les tags d'un utilisateur spécifique. Dans ce qui suit nous abordons une extension de cette approche, qui prends en compte, en plus de l'activité de l'utilisateur, son évolution dans le temps. Cette approche répond à la problématique suivante : Pour un utilisateur donné, en sachant les activités de cet utilisateur, ces intérêts restent-ils figés ou évoluent-ils à travers des périodes de temps. C'est ce que tente de mettre en relief la section suivante.

5 Variante temporelle de profils aux multi-centres d'intérêts évolutifs

Cette section décrit la variante temporelle de notre approche nommée DPN_{Temp_u} . Après avoir étudié la propagation des multiples centres d'intérêts lors de la construction du profil de l'utilisateur. Nous intégrons à cette approche une fonction *gaussienne*

simulant l'atténuation graduelle de ses intérêts. L'approche que nous décrivons dans ce qui suit a fait l'objet de deux publications (se référer aux articles [28, 210]). Il est important de prendre connaissance du fait que nous gardons

la représentation initiale du profil utilisateur. De même pour le degré de préférence normalisé que nous dénommons $v_{i,k}$. \vec{U}_i est un vecteur de tags : valeurs, i.e

$$\vec{U}_i = (t_{i,1} : v_{i,1}, t_{i,2} : v_{i,2} \dots t_{i,n} : v_{i,n})$$

Où $t_{i,k}$ est un tag annoté par u_i , n est le nombre total de tags annotés par u_i . $v_{i,k}$ est le degré de préférence de l'utilisateur u_i au tag $t_{i,k}$, qui peut être intuitivement obtenu comme suit :

$$v_{i,k} = \frac{N_{i,k}}{N_i} \quad (5.13)$$

Où $N_{i,k}$ est le nombre de fois qu'un utilisateur i annoté avec le tag k , et N_i est le nombre de ressources annotées par l'utilisateur i .

Dans le but d'intégrer l'évolution de l'intérêt de l'utilisateur lors de la construction de profils, nous proposons de biaiser le degré de préférence par une fonction temporelle. Pour cela, nous exploitons les caractéristiques d'une fonction à noyaux Gaussien [109] pour estimer la distance entre la date courante et la date à laquelle la ressource a été annotée, avec comme paramètre σ qui représente le coefficient d'interpolation avec S^c le moment courant et S_j le moment où l'utilisateur annoté le tag.

$$K(S^c, S_j) = \frac{1}{\sigma\sqrt{2\pi}} \cdot \exp\left[\frac{-(S^c - S_j)^2}{2\sigma^2}\right] \quad (5.14)$$

Le profil utilisateur est représenté cette fois d'après les mesures de pondérations suivantes associées au vecteur de tags :

$$\vec{U} = (t_1^{S_j} : W_1^{S_j}, t_2^{S_j} : W_2^{S_j}, \dots, t_m^{S_j} : W_m^{S_j})$$

Où le poids $W(t)^{S_j}$ d'un tag t dans un profil utilisateur est la somme de ses fréquences normalisées et biaisées par la fonction temporelle, et est définie comme suit :

$$W(t_k)^{S^c} = \sum v_{i,k}(t_k)^{S_j} K(S^c, S_j) \quad (5.15)$$

Nous gardons le cadre expérimental détaillé en section 4, précisément, à partir d'une représentation en graphe biparti nous exploitons un algorithme de clustering qui propage le degré de préférence de l'utilisateur, mais contrairement à ce qui a été explicité précédemment, nous biaisons à l'aide d'une fonction temporelle ce degré de préférence. Le but d'une telle manœuvre est de prendre en compte en plus de l'activité d'annotation de l'utilisateur, l'évolution de ses centres d'intérêts.

Ainsi, comme dans la sous-section 4.1.1, nous évaluons la performance de l'approche personnalisée en fonction de la pertinence graduée des ressources pour chaque utilisateur et la comparons à celles d'approches, notamment, celle de *Yeung et al.*, et DPN_{Tu} qui ne tiennent pas compte de la temporalité associée aux tags [28].

Aussi, nous proposons de mettre en comparaison cette méthode de génération de profils utilisateurs, à savoir notre approche qui exploite le degrés de préférence de

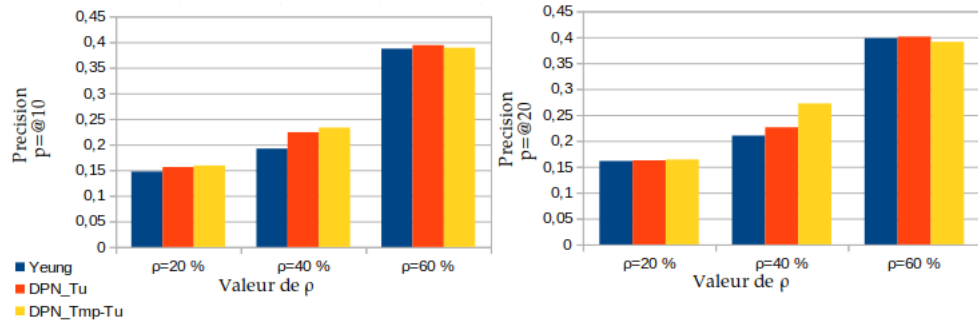


FIGURE 5.4 – Précisions à $p@10$ et $p@20$ des approches *Yeung et al.*, [237], DPN_{Tu} et DPN_{Tmp-u} lorsque $\rho = 40\%$

Valeur de ρ	Precision					
	<i>Yeung et al.</i> , [237]		DPN_{Tu}		DPN_{Tmp-u}	
	$p@10$	$p@20$	$p@10$	$p@20$	$p@10$	$p@20$
$\rho = 20\%$	0,147	0,161	0,156	0,1623	0,159	0,164
$\rho = 40\%$	0,192	0,21	0,224	0,226	0,233	0,272
$\rho = 60\%$	0,387	0,398	0,394	0,401	0,389	0,391

TABLE 5.8 – Précisions pour différentes valeurs des signatures ρ . Comparaison *Yeung et al.*, [237] et DPN_{Tu} .

l'utilisateur normalisé et biaisé par une fonction temporelle DPN_{Tmp-u} avec DPN_{Tu} et l'approche de *Yeung et al.*, [237]. Le **tableau 5.8** synthétise les valeurs des précisions à $p = @10$ et $p = @20$. La Figure 5.4, représente les résultats des précisions calculés à différentes valeurs de ρ . On y observe une meilleure précision quand $\rho = 40\%$. Nous en concluons qu'à $\rho = 40\%$, la proportion de tags décrivant les ressources dans les clusters est exploitable. Nous conserverons donc ce paramètre lors du calcul du nDCG. Nous comparons ce score de précision avec les baselines, *Yeung et al.*, [237] et DPN_{Tu} .

Recouvrement	Précision globale								
	20%			40%			60%		
	<i>Yeung et al.</i> , [237]	DPN_{Tu}	DPN_{Tmp-u}	<i>Yeung et al.</i> , [237]	DPN_{Tu}	DPN_{Tmp-u}	<i>Yeung et al.</i> , [237]	DPN_{Tu}	DPN_{Tmp-u}
$\forall r \in R_u$	0,112	0,121	0,128	0,117	0,151	0,161	0,231	0,247	0,258
$\forall r \in R_{collection}$	0,078	0,088	0,089	0,093	0,094	0,096	0,174	0,168	0,175

TABLE 5.9 – Précisions globales pour différentes valeurs des signatures ρ . Les valeurs sont calculées parmi toutes les ressources d'un utilisateur, parmi l'ensemble des ressources de la collection. Comparaison *Yeung et al.*, [237].

À l'issue de la comparaison rapportée dans le **tableau 5.9**, nous pouvons affirmer que la prise en compte du paramètre temporel augmente la valeur du score de pertinence globale. En effet, il représente mieux les divers centres d'intérêts de l'utilisateur à travers le temps, parmi l'ensemble des ressources de l'utilisateur ($\forall r \in R_u$), de toutes les ressources de la collection ($\forall r \in R_{collection}$).

En ce qui concerne les résultats d'évaluation de la pertinence du classement gradué sur l'ensemble des ressources de l'utilisateur, on observe (**Figure 5.5**) une nette amélio-

ration de la position de classement en faveur de l'approche $DPN_{T_{mp-u}}$ à $r@ = 5$ à $r@ = 10$, visible dans le tableau 5.10. La valeur $nDCG$ de cette dernière augmente pour atteindre 0.69 à $r@ = 90$ et $r@ = 100$, tandis que l'approche de *Yeung et al.*, [237] atteint la valeur maximale de 0.62 lorsque $r@ = 90$ et $r@ = 100$. La valeur $nDCG$ de l'approche DPN_{T_u} indique quant à elle une amélioration qui surpasse celle de *Yeung et al.*, [237] et atteint une valeur maximale de 0.64 lorsque $r@ = 90$ et $r@ = 100$. Nous présentons dans le tableau 5.11 un comparatif des valeurs moyennes, déviation standard ainsi que p -value associées à chaque approche que nous comparons statistiquement aux baselines.

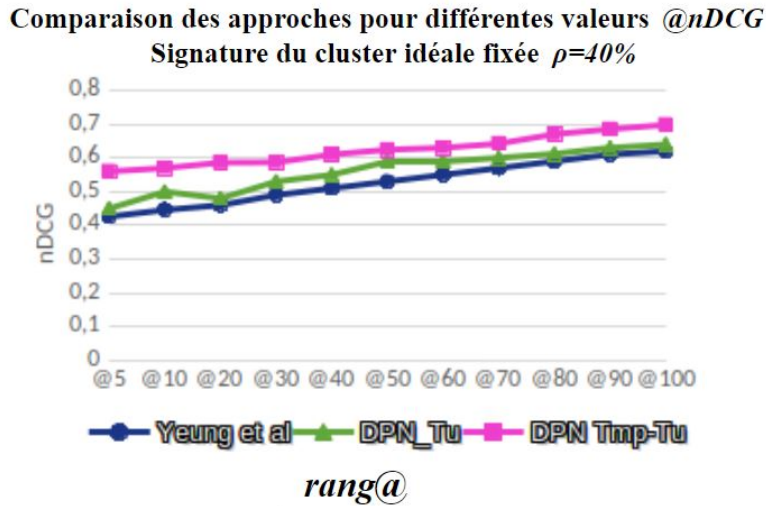


FIGURE 5.5 – Comparaison des approches parmi différentes valeurs de la mesure $nDCG$ pour une signature de cluster idéale fixée à $\rho = 40\%$

Rang @	NDCG Yeung et al.,	NDCG_DPN _{T_u}	NDCG_DPN _{T_{mp-u}}
r@5	0.426	0.45	0.56
r@10	0.446	0.5	0.57
r@20	0.465	0.48	0.586
r@30	0.49	0.53	0.587
r@40	0.51	0.55	0.61
r@50	0.53	0.591	0.624
r@60	0.55	0.59	0.63
r@70	0.57	0.611	0.6423
r@80	0.59	0.612	0.67
r@90	0.61	0.63	0.685
r@100	0.62	0.64	0.698

TABLE 5.10 – Le tableau présente les scores de $NDCG@10$ à $NDCG@100$ en comparaison avec les approches *Yeung et al.*, [237], DPN_{T_u} et $DPN_{T_{mp-T_u}}$ [29].

Variable	Mean	Déviati on standard	<i>p-value</i>
<i>Yeung et al</i>	0.52	0.06	1.29 e – 05(★)
DPN_{Tmp-u}	0.62	0.04	
DPN_{Tu}	0.48	0.064	1.43 e – 06(★)
DPN_{Tmp-u}	0.52	0.06	
<i>Yeung et al</i>	0.52	0.06	2.00 e – 05(★)
DPN_{Tu}	0.48	0.064	

TABLE 5.11 – Dans ce tableau, les résultats avec (★) sont statistiquement significatifs comparés aux baselines.

De notre point de vue, cela met en évidence d'une part, l'impact de l'attribution de poids qualitatifs représentant les préférences de l'utilisateur, mais aussi l'importance de prendre en compte l'évolution des intérêts de celui-ci. Par conséquent, la combinaison d'un schéma de pondération normalisée de termes avec une mesure permettant l'atténuation progressive de l'importance de ces termes au cours du temps, améliore le classement personnalisé des ressources retournées à l'utilisateur lors du processus de recherche.

6 Conclusion

Dans ce chapitre nous avons mis en évidence deux méthodes permettant la construction de profils aux multiple centres d'intérêts. L'une **DPN_{Tu}** intègre le degré de préférence normalisé et propagé de tags d'utilisateurs et l'autre **DPN_{Tmp-u}** inclus le degré de préférence normalisé et propagé de tags d'utilisateurs biaisé paune fonction temporelle.

Après avoir évalué ces deux approches dans un cadre expérimental commun incluant le classement de pertinence personnalisé par utilisateur, nous somme parvenu aux conclusions suivantes que nous avons synthétisé sous forme de points :

- Les intérêts des utilisateurs sont multiples et variés, dans la mesures ou nous avons prouvé qu'ils peuvent prendre la forme de clusters d'intérêts. Nous avons aussi démontré que ces clusters pouvaient être identifier grâce à la plus petite proportion de tags au sein de ces clusters d'intérêts et qui correspondent au paramètre ρ .
- Les profils générés grâce à l'approche **DPN_{Tmp-u}** a démontré de meilleurs résultats à l'évaluation, ce qui indique qu'en plus d'être multiples les intérêts des utilisateurs évoluent avec le temps.

- L'évaluation de la pertinence de ces profils permettent leurs exploitation dans le domaine de la recommandation personnalisée en s'ajustant aux pratique et habitudes des usagers de folksonomies de manière dynamique et évolutive.

CONTRIBUTION II

Approches neuronales basées sur les folksonomies pour la recommandation

Chapitre 6

APPROCHES NEURONALES BASÉES SUR LES FOLKSONOMIES POUR LA RECOMMANDATION

1 Introduction

DANS ce chapitre, nous abordons notre deuxième contribution qui porte sur l'exploitation de modèles neuronaux de recommandation intégrant explicitement le vocabulaire engendré par les tags.

Dans notre synthèse de l'état de l'art, nous avons mis en exergue certains modèles neuronaux de filtrage collaboratif de la littérature qui exploitent des entrées sous la forme de vecteurs U ou I , représentés le plus souvent dans l'espace des tags (représentation en *One-hot*, autrement dit si un utilisateur annote avec un tag ou pas) ou sous la forme de vecteurs d'*embeddings* de tags. Ces modèles exploitent en entrée des représentations d'utilisateurs et d'items qui sont :

- Des vecteurs *One-hot* dans l'espace des items [196, 95, 94].
- Des vecteurs *One-hot* dans l'espace des tags [155, 235, 254].
- Des vecteurs d'*embeddings* statiques dans l'espace des tags [132, 222].

Dans le cadre des modèles neuronaux de recommandation exploitant des représentations d'utilisateurs et d'items dans l'espace des tags, le problème que nous traitons se situe au niveau de la façon de construire les représentations à l'entrée de ces modèles. Comme nous l'avons explicité dans la synthèse de l'état de l'art, ces représentations sont construites de manière séparées à l'entrée du réseau de neurones [155, 235] ou bien agrégées (concaténation, multiplication par élément, en anglais *pointwise product* [138, 254]) ou encore sous la forme de graphe en exploitant la matrice d'adjacence associée à ce graphe [22, 9]. Cependant, les modèles présentés n'explorent pas la possibilité d'intégrer le voisinage des représentations de nœuds.

Plus précisément, dans nos travaux, nous tenterons de répondre aux questions de recherches que nous formulons comme suit :

- Comment intégrer explicitement le vocabulaire des tags à travers des vecteurs de caractéristiques issus des modèles de langue neuronaux pré-entraînés à savoir *word2vec* et *BERT* dans un modèle de filtrage collaboratif neuronal pour la recommandation ?
- Par quel moyen exploiter des *embeddings* de tags issus de représentations d'utilisateurs et d'items qui tiennent compte des relations de voisinage dans un modèle neuronal pour la recommandation ?

Pour le reste, ce chapitre est organisé comme suit, dans la section suivante nous explicitons les représentations des entrées des modèles neuronaux de filtrage collaboratif que

nous avons développés. Plus précisément nous explorons l’exploitation d’autres type de vecteurs d’embeddings (*BERT*), la concaténation des vecteurs d’embeddings de notes avec les vecteurs d’*embeddings* d’utilisateurs/items mais aussi l’exploitation du voisinage de représentations d’utilisateurs/items sous la forme de noeuds d’un graphe et représentées dans l’espace des tags. Par la suite, dans la section 3, nous formalisons différents modèles neuronaux de filtrage collaboratif incluant ces vecteurs d’embeddings. La section 4 est consacrée aux expérimentations ainsi qu’à l’évaluation des modèles. Nous avons comparé nos modèles avec d’autres approches en section 5 et discuté des résultats de nos évaluations dans le tableau 6.2. Enfin, nous dressons notre conclusion dans la section 7 avant d’aborder le dernier chapitre du manuscrit.

2 Représentations des entrées des modèles neuronaux de filtrage collaboratif

Nos travaux dans le cadre de cette thèse exploitent des utilisateurs et des items représentés dans l’espace des tags dense (dans l’espace des *embeddings* de tags). En se référant aux techniques commune de la littérature exploitant des modèles de langues pré-entraînés [62, 155, 132], nous étudions différents types d’entrées que nous passons à travers des modèles neuronaux *end-to-end* pour la recommandation. Nous nous basons sur les notations exprimées dans le tableau 6.1.

2.1 Représentation en sac de mots dans l’espace des tags

Soit un utilisateur u ou un item i représenté implicitement dans l’espace des tags au moyen d’une représentation simple, *One-hot*. La représentation de cet utilisateur peut être formalisée comme suit :

$$u = (w_{t_1}, w_{t_1} \dots w_{t_m}) \quad (6.1)$$

Où, $w_{(t_m)} = 1$ si le tag t_i a été utilisé par u et 0 sinon. De même pour la représentation d’un item que nous pouvons aussi décrire de la manière suivante :

$$i = (w_{t_1}, w_{t_1} \dots w_{t_n}) \quad (6.2)$$

Où, $w_{(t_n)} = 1$ si le tag t_i a été utilisé pour annoté i et 0 sinon.

2.2 Représentation dans un espace dense statique de tags

Supposons maintenant que nous changions d’espace de représentation et que nous voudrions cette fois représenter l’ utilisateur u dans l’espace des représentations induit par les *embeddings* de tags. Dans ce cas nous écrirons :

Notations	Détails
V	Vocabulaire (\mathcal{V} ensemble des nœuds)
\mathbb{F}	Une folksonomie
U	L'ensemble des utilisateurs
I	L'ensemble des Items
T	L'ensemble des tags
A	Matrice d'adjacence
D	Matrice d'adjacence des degrés d'un graphe
N	Le nombre d'items
M	Le nombre d'utilisateurs
T_i	Nombre de tags de l'item i
T_u	Nombre de tags de l'utilisateur u
u_m	Un utilisateur m
i_k	Un item k
t_l	Un tag particulier composé de n -mots
\mathcal{R}	Ensemble des notes d'utilisateurs
W	Matrice de poids
$r_{u,i}$	La note d'un utilisateur u attribuée un item i
\mathbb{E}	matrice de projection
H	Neural hidden states (couches neuronales cachées)
\tanh	Fonction de la tangente hyperbolique
\mathbf{b}	Biais optionnel

TABLE 6.1 – Notations détaillées

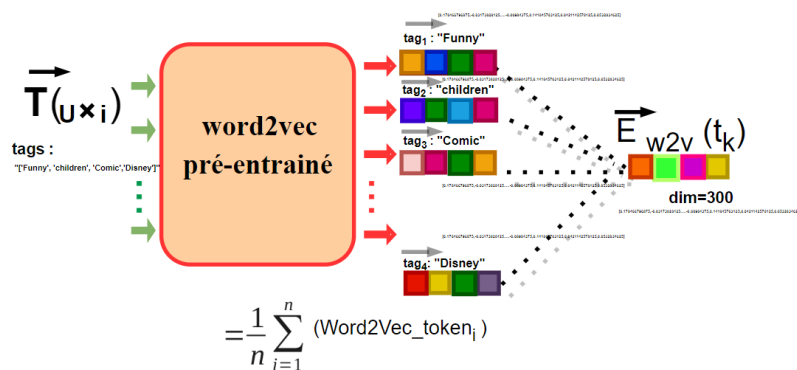


FIGURE 6.1 – Construction des *embeddings* de tags $\vec{T}_{(u \times i)}$ à partir du modèle de langue pré-entraîné Word2vec

$$u = [\mathbb{E}(t_1), \mathbb{E}(t_2) \dots \mathbb{E}(t_T)] \quad (6.3)$$

Où $\mathbb{E}(t_k)$ représente l'*embedding* du tag t_k .

De même pour un item i , que nous pouvons aussi formaliser dans l'espace des représentations induit par les *embeddings* de tags où $\mathbb{E}(t_k)$ représente l'*embedding* du tag t_k .

$$i = [\mathbb{E}(t_1), \mathbb{E}(t_2) \dots \mathbb{E}(t_T)] \quad (6.4)$$

Comme nous l'avons déjà explicité, il existent différents moyens de combiner ces vecteurs à l'entrée d'une architecture neuronale. Ces vecteurs sont souvent associés à des vecteurs de caractéristiques, respectivement $v_{(u)}^f \in \mathbb{R}^d$ pour le vecteur de caractéristiques d'utilisateur et $v_{(i)}^f \in \mathbb{R}^d$ pour le vecteur de caractéristiques de l'item avec d la dimension de l'espace des *embeddings*. Il est aussi possible d'agréger un vecteur unique de tags par utilisateur dans l'espace des *embeddings* des tags. Dans ce cas précis, $v_{(u)}^f = \mathbb{R}^{1 \times d}$.

À l'issue de nos travaux, nous considérons deux espaces de représentation denses. Le premier, $\mathbb{E}_{(w2v)}$ est issu du plongement des tags dans l'espace de représentation dense *statique* du modèle neuronal *word2vec*. Dans ce cas, comme l'illustre la figure 6.1 un utilisateur/item est représenté selon la moyenne des *embeddings* de tags extraits de ce modèle. Comme un tag peut être parfois composé de plusieurs mots, nous calculons en amont la moyenne des *embeddings* des *tokens* formant ce tag. Formellement, si $T_{(u \times i)}$ est l'ensemble de tags utilisé par u pour décrire i alors :

$$v_{(u)}^f = \frac{1}{|T_{(u \times i)}|} \sum_{i'=1}^{|T_{(u \times i)}|} \mathbb{E}_{(w2v)}(t_k) \quad (6.5)$$

Où, $\mathbb{E}_{w2v}(t_k)$ est le vecteur d'*embeddings* du tag t_k . Un item est représenté selon la moyenne de tous les *embeddings* des tags décrivant cet item. Nous obtenons ainsi :

$$v_{(i)}^f = \frac{1}{|T_{(i)}|} \sum_{i'=1}^{|T_{(i)}|} \mathbb{E}_{(w2v)}(t_k) \quad (6.6)$$

Où $\dim(\mathbb{E}_{(w2v)}) = 300$ est la dimension de l'espace $\mathbb{E}_{(w2v)} \in \mathbb{R}^{1 \times d=300}$ et $\mathbf{dim} = 300$ la taille des vecteurs en sortie.

2.3 Représentation dans un espace dense contextualisé

Le second espace que nous nommons \mathbb{E}_{Bert} , formalise l'espace de représentation dense *contextualisé* engendré par la projection des tags en exploitant le modèle de langue pré-entraîné *BERT*. L'extraction d'*embeddings* contextualisés à partir du modèle pré-entraîné *BERT* peut être exécuté de différentes manières. Tout dépend de la façon dont les données en entrée du modèle sont disposées, mais aussi, de la manière d'extraire les vecteurs d'*embeddings* contextualisés en sortie. D'un point de vue pratique, le modèle

dispose de 512 entrées qui correspondent chacune à un token. La première entrée correspond à un token spécial le [CLS] pour "classification" et qui permet d'exploiter *BERT* pour une tâche de classification. Il a également 512 sorties de dimension 768 chacune (pour la version *BERT-base*¹). *BERT* peut ainsi prendre en entrée une ou deux phrases et utilise le token spécial [SEP] pour les différencier. La sortie de chacun des 12 encodeurs peut être considérée comme une représentation vectorielle de la séquence passée en entrée. Ainsi, pour une configuration des tokens [CLS] et [SEP] donnée, Il est souvent commun de concaténer ou d'additionner les quatre dernières couches cachées, ce qui donne en sortie un seul vecteur par *token* de dimension 768. Ainsi pour une phrase composée de 4 tokens chaque vecteur sera de dimension $4 \text{ tokens} \times 768 = 3.072$. Selon la littérature [176], il est possible d'extraire à partir du modèle neuronal pré-entraîné *BERT* :

1. Des vecteurs d'*embeddings* correspondant aux représentations à la sortie du token spécial [CLS].
2. Des vecteurs d'*embeddings* individuels pour chacun des tokens en sortie.
3. Un vecteur d'*embeddings* contextualisés unique de la séquence de tokens en agrégeant (*pooling* [176]) les vecteurs en sortie.

Pour notre part, comme l'illustre la figure 6.2, dans le but d'obtenir un vecteur unique pour l'ensemble des tags qu'un utilisateur emploie lorsqu'il décrit un item, nous développons la troisième option, à savoir, considérer cet ensemble de tags comme une séquence ou plus simplement comme une phrase à l'entrée du modèle pré-entraîné. De même pour l'ensemble des tags décrivant un item, à la différence que dans ce cas précis, cet ensemble est plus conséquent car nous calculons cette même moyenne pour l'ensemble de tous les tags ayant annoté l'item indépendamment de l'utilisateur. A partir du token [CLS], nous obtenons un vecteur unique de dimension 768 par utilisateur/item. Dans notre cas, les entrées d'une seule phrase ne requièrent qu'une série de 1, nous complétons donc le vecteur de valeurs à 1 pour chaque token de la séquence de tags en entrée.

Ainsi, en gardant $\mathbb{E}(t_k)$ la représentation vectorielle (*embedding*) du tag t_k , nous formalisons les représentations des vecteurs de caractéristiques d'utilisateurs dans \mathbb{E}_{Bert} selon l'équation :

$$v_{(u)}^f = \frac{1}{|T_{(u \times i)}|} \sum_{i'=1}^{512} \mathbb{E}_{Bert}(t_k) \quad (6.7)$$

Où $\mathbb{E}(t_k) \in \mathbb{R}^d$ est le vecteur dense associé au tag k et c dénote la dimension de l'espace engendré. \mathbb{E} formalise l'espace dense \mathbb{E}_{Bert} avec $\dim(\mathbb{E}_{Bert}) = 768$, $\mathbb{E} \in \mathbb{R}^{1 \times d=768}$ la taille des vecteurs en sortie de l'espace. La valeur 512 correspond au nombre total de vecteurs en sortie.

De la même manière, nous formalisons les représentations des vecteurs de caractéristiques d'items selon l'équation :

$$v_{(i)}^f = \frac{1}{|T_i|} \sum_{i'=1}^{512} \mathbb{E}_{Bert}(t_k) \quad (6.8)$$

1. 12-couches cachées, 768-dimension couches cachées, 12-têtes d'attention, 110M paramètres

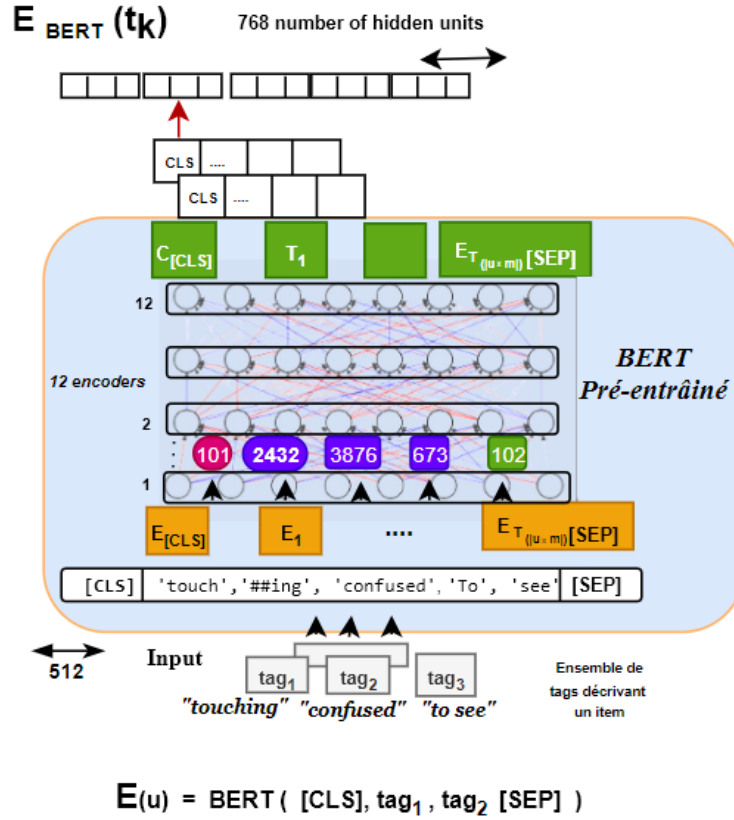


FIGURE 6.2 – Construction des *embeddings* contextualisés de tags $\vec{T}_{(u \times i)}$ à partir de $BERT_{base}$ (12 couches cachées avec la dimension de chaque couche cachée à 768 et 12 têtes d’attentions). La sortie de chacun des 12 encodeurs peut être considérée comme une représentation vectorielle de la séquence passée en entrée. La pertinence de cette représentation est assurée par le mécanisme d’attention mis en œuvre par les encodeurs.

Où $\mathbb{E}(t_k) \in \mathbb{R}^d$ est le vecteur dense associé au tag k et d dénote la dimension de l’espace engendré. \mathbb{E} formalise l’espace dense \mathbb{E}_{BERT} avec $\dim(\mathbb{E}_{BERT}) = 768$, $\mathbb{E} \in \mathbb{R}^{1 \times d=768}$ la taille des vecteurs en sortie de l’espace. La valeur 512 correspond au nombre total de vecteurs en sortie.

2.4 Représentation du voisinage dans un espace dense

Nous formalisons dans ce qui suit un autre type d’entrée exploité particulièrement dans une approche basée sur l’apprentissage de graphe [91]. Cette approche consiste en un modèle à couches, où chaque couche k est la représentation d’un nœud en y intégrant les k voisins. Nous nous appuyons sur ce modèle afin d’inclure des représentations denses de tags étendues à un certain voisinage et de construire à partir des nœuds d’un graphe d’utilisateurs-items des *embeddings* de nœuds. Le point de départ est un graphe biparti d’utilisateurs et d’items, où l’ensemble des arêtes entre les deux classes de nœuds (d’utilisateurs et d’items) représentent l’ensemble des notes attribuées

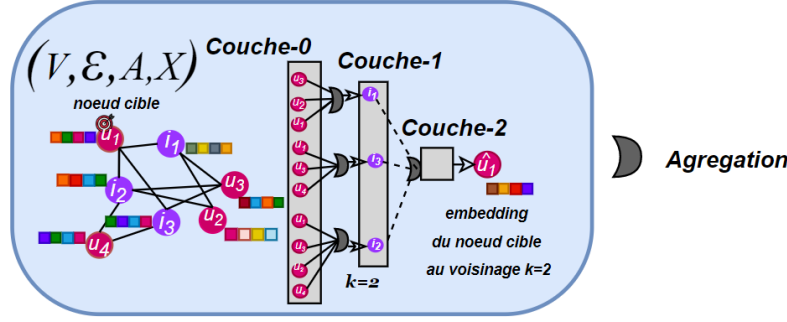


FIGURE 6.3 – Agrégation des représentations vectorielles de noeuds au voisinage $k = 2$ dans l'espace dense \mathbb{E}_{w2v} et \mathbb{E}_{Bert} .

par les utilisateurs aux items. Pour un graphe biparti pondéré et non orienté utilisateur-item $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A, \mathcal{X})$, avec $\mathcal{V} = \{\mathcal{V}_u \cup \mathcal{V}_i\}$, comme étant l'ensemble disjoint de nœuds d'utilisateurs \mathcal{V}_u et d'items \mathcal{V}_i .

Comme l'illustre la figure 6.3, \mathcal{E} est l'ensemble des arêtes non-orientées représentant les notes des utilisateurs pour des items, $r_{(u,i)} \in \{0, 5\}$. A est la matrice d'adjacence et $\mathcal{X} \in \mathbb{R}^{m \times n}$ définie comme la matrice de représentations denses des nœuds du graphe biparti. Notre démarche consiste à construire des *embeddings* de nœuds d'utilisateurs et d'items basés sur un certain voisinage. Pour y parvenir, nous exploitons des agrégats de vecteurs d'embeddings statiques (word2vec) ou contextuels (BERT) de tags. De plus, une fonction de voisinage $N(v)$ agrège les représentations vectorielles de tags (vecteurs d'embeddings) aux voisinages des nœuds. En exploitant différentes fonctions d'agrégation, nous pouvons formaliser le processus d'apprentissage comme suit :

Soit $h_{v(u)}^0 = v_{(u)}^f$ avec $v_u \in \mathcal{V}_u$ qui exprime les caractéristiques agrégées du nœud v_u de la couche $K = 0$. Dans cas, à l'état initial, les caractéristique de ce nœud sont représentées sous la forme vectorielle (un agrégat d'embeddings), c'est à dire que les caractéristiques propre au nœud même y sont inclus dans l'espace dense de représentation sélectionné (\mathbb{E}_{w2v} ou \mathbb{E}_{Bert}). Pour un nœud $v_u \in \mathcal{V}_u$, nous associons le vecteur de caractéristique $v_{(u)}^f$ à ce nœud quand $K = 0$, tel que :

$$v_{(u)}^f = \frac{1}{|T(u)|} \sum_{k \in T(u)} \mathbb{E}_{(w2v/BERT)}(t_k) \quad (6.9)$$

Où $\mathbb{E}(t_k) \in \mathbb{R}^d$ est le vecteur dense associé au tag t_k et d dénote la dimension de l'espace engendré. \mathbb{E} formalise l'espace dense $\mathbb{E}_{w2v/BERT}$.

De la même manière, nous formalisons pour un nœud $v_i \in \mathcal{V}_i$ le vecteur de caractéristique $v_{(i)}^f$, quand $k = 0$:

$$v_{(i)}^f = \frac{1}{|T(i)|} \sum_{k \in T(i)} \mathbb{E}_{(w2v/BERT)}(t_k) \quad (6.10)$$

Où $\mathbb{E}(t_k) \in \mathbb{R}^d$ est le vecteur dense associé au tag t_k et d dénote la dimension de l'espace engendré. Cela se traduit par des agrégats d'embeddings de tags représentant

les nœuds $\in \mathcal{V}_u$ et des agrégats d'*embeddings* de tags représentant les nœuds $\in \mathcal{V}_i$. De la sorte, les caractéristiques à la k -ème couche (voisinage) peuvent être formalisées selon l'équation :

$$h_v^k = \sigma(\mathbf{W}_k \sum_{u \in N(v)} \frac{h_u^{k-1}}{|N(v)|} + \mathbf{b}_k h_v^{k-1}) \quad (6.11)$$

Dans l'équation 6.11, chaque niveau de voisinage K est associé à une couche dense. Ainsi, h_v^{k-1} est le vecteur dense associé au nœud $v \in \mathcal{V}_u$ qui a été agrégé à la $k - 1$ -ème couche, c'est à dire, au voisinage des représentations denses des couches précédentes. où $|N(v)|$ représente le nombre de voisins du nœud v . σ est la fonction d'activation qui permet la non-linéarité. En considérant en plus les paramètres \mathbf{W}_k et \mathbf{b}_k , la représentation dense finale à la K -ième couche du nœud v_u est représentée comme étant : $\hat{v}_u = h_v^k$. Cela peut être aussi exprimer à l'aide d'une multiplication matricielle sur l'ensemble du graphe de la manière suivante :

$$H^{l+1} = \sigma(H^l \mathbf{W}_0^l + \tilde{A} H^l \mathbf{W}_1^l) \quad (6.12)$$

$\tilde{A} = D^{-1/2} A D^{-1/2}$ avec A qui représente la matrice d'adjacence et D la matrice des degrés du graphe.

Dans la section suivante, nous détaillons dans quel contexte ont pu être intégré différentes représentations associées aux vecteurs d'*embeddings* dans des approches de filtrage collaboratif neuronal.

3 Modèles de filtrage collaboratif neuronal basés sur les tags

Dans cette section nous mettons en œuvre notre contribution en formalisant à l'aide de vecteurs denses de caractéristiques issus des tags d'utilisateurs et d'items, différentes architectures neuronales. Plus précisément, nous proposons d'étendre trois modèles de filtrage collaboratif en y intégrant des représentations d'utilisateurs et d'items à base de tags. Dans ce qui suit, chaque modèle est décrit d'après l'architecture qu'il étend mais aussi selon la manière dont les vecteurs de caractéristiques sont considérés à l'entrée de ces modèles. Nous détaillons ces modèles dans les sous-sections suivantes :

3.1 Filtrage collaboratif neuronal à base de tags exploitant une architecture en MLP

Nous présentons l'architecture du réseau de neurones en MLP permettant la prédiction de notes d'utilisateurs que nous formalisons selon l'équation suivante :

$$\hat{r}_{u,i} = \mathbf{MLP}(P_u^T \cdot v_{(u)}^f, Q_i^T \cdot v_{(i)}^f | P_u, Q_i, \theta_{\mathbf{MLP}}) \quad (6.13)$$

Où $P_u \in \mathbb{R}^{M \times k}$ et $Q_i \in \mathbb{R}^{N \times k}$ sont respectivement les vecteurs latents d'utilisateurs et d'items, et $\theta_{\mathbf{MLP}}$ dénote les paramètres de la fonction d'interaction définie comme étant un MLP. $\hat{r}(u, i)$ représente la note de l'utilisateur u attribuée à l'item i . Le modèle peut être généralisé pour un modèle prédictif exploitant un MLP. La figure 6.4 détaille une instance du modèle qui exploite des représentations denses d'utilisateurs

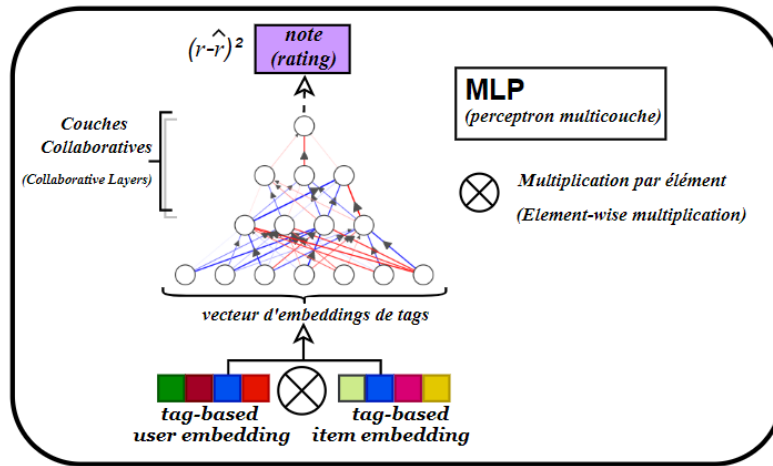


FIGURE 6.4 – Filtrage collaboratif neuronal basé sur une architecture en MLP extrait de l'article Boudiba & Dkaki [27]

et d'items extraits de l'espace de représentations des modèles de langues pré-entraînés. Des couches cachées sont rajoutées pour apprendre les interactions entre vecteurs d'utilisateurs et vecteurs d'items. Le produit de ces vecteurs denses est passé par des couches cachées permettant l'apprentissage des interactions entre utilisateurs et items, la dernière couche cachée comprends une régression pour estimé la note finale.

3.2 Filtrage collaboratif neuronal à base de tags exploitant une architecture en autoencoder

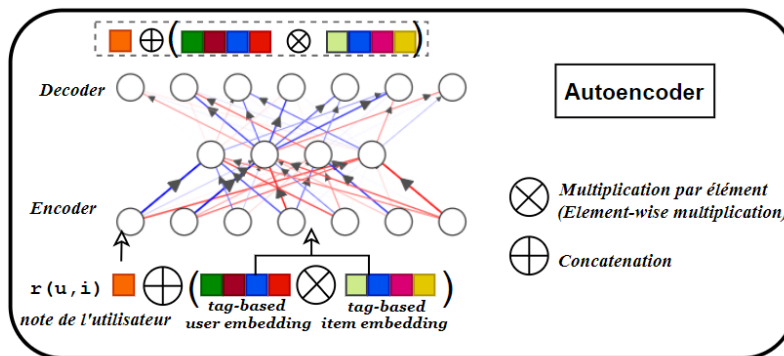


FIGURE 6.5 – Approche de filtrage collaboratif basée sur un modèle neuronal avec une architecture en autoencoder extrait de l'article Boudiba & Dkaki [27]

Ce modèle exploite des représentations de tags denses dans le cadre d'une architecture neuronale en autoencoder. L'objectif d'une telle démarche est de définir un modèle neuronal prédictif capable d'intégrer le produit des vecteurs denses de tags et de les concaténer aux représentations de notes d'utilisateur dans un espace de caractéristiques dense. Les notes des utilisateurs $r(u_m, i_l)$ sont reconstruites à l'aide d'une fonction objective θ qui minimise :

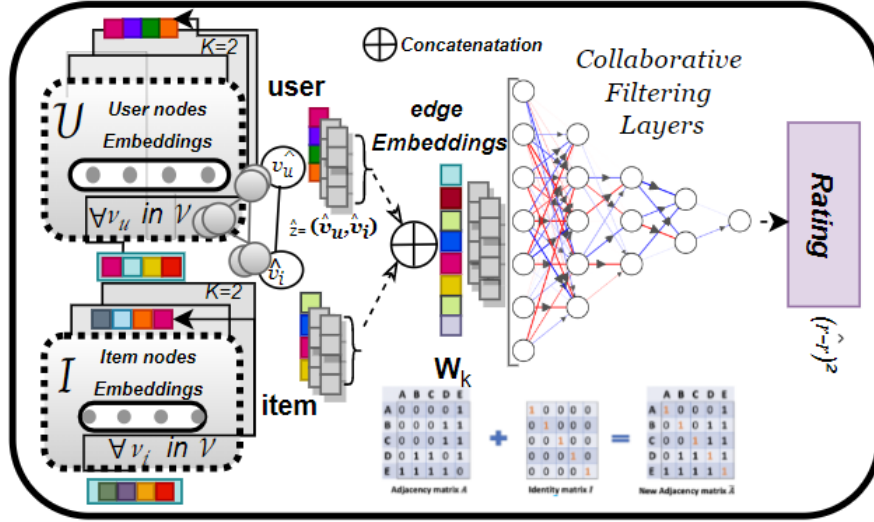


FIGURE 6.6 – Approche basée sur le filtrage collaboratif de graphes neuronaux ayant comme caractéristiques des embeddings de tags étendus à un certain voisinage.

$$\sum ||r(u_m, i_l) \oplus (v_{(u)}^f \otimes v_{(i)}^f) - h(r(u_m, i_l) \oplus (v_{(\tilde{u})}^f \otimes v_{(i)}^f); \theta)||^2 \quad (6.14)$$

Où, $(r(u_m, i_l), \theta)$ est la reconstruction de l'entrée $r(u_m, i_l) \in \mathbb{R}^d$. L'opérateur \otimes dénote le produit interne entre les vecteurs de caractéristiques d'utilisateurs et d'items. L'opérateur \oplus formalise un opérateur de concaténation. \tanh est la fonction d'activation. Le modèle prédictif en autoencodeur exploite le produit des représentations denses d'utilisateurs et d'items extraits de l'espace de représentations des modèles de langues neuronaux pré-entraînés 6.5. Ces représentations sont concaténées à l'aide d'un opérateur de concaténation et passées à travers des couches neuronales cachées 6.5 permettant l'apprentissage des interactions entre utilisateurs et items dont les vecteurs de caractéristiques sont compressés ainsi que les vecteurs des notes d'utilisateurs dans un l'espace dense. En sortie, la reconstruction de cet espace dense produit les notes finales des utilisateurs.

3.3 Filtrage collaboratif neuronal en graphe exploitant le voisinage de représentations continues de tags

Nous avons porté un intérêt au moyen d'apprendre des représentations d'utilisateurs et d'items qui permettent de tenir compte du voisinage auquel est associé chaque utilisateur ou item. Nous estimons qu'une représentation d'utilisateur ou d'item à un voisinage donné constitue un bon moyen d'identifier certaines connaissances de proximité et de les utiliser comme caractéristiques dans un modèle de recommandation neuronal

basé sur le filtrage collaboratif.

Le point de départ de notre modèle est le graphe biparti \mathcal{G} (défini dans la sous-section 2.4) qui est échantillonné pour permettre au modèle l'apprentissage des représentations sous forme vectorielle de chaque nœud. Les vecteurs d'*embeddings* de nœuds, \hat{v}_u , représentant les utilisateurs, $v_u \in \mathcal{V}_u$ et les vecteurs d'*embeddings* de nœuds, \hat{v}_i , représentant les items, $v \in \mathcal{V}_i$ sont concaténés au moyen d'un opérateur de concaténation \oplus . Le but d'une telle démarche est de construire des représentations denses d'arcs $\hat{z}(\hat{v}_u \oplus \hat{v}_i) = \hat{v}_u \oplus \hat{v}_i$ reliant les paires de nœuds $e(v_u, v_i)$. Une fois que les embeddings d'arcs ont construits, les représentations denses d'arcs, formalisées sous la forme d'une matrice, est passée à travers différentes couches collaboratives incluant une couche de régression pour prédire les notes d'utilisateurs-items comme l'illustre la figure 6.6. Le modèle est entraîné de bout en bout en minimisant une fonction de perte (RMSE) en utilisant la méthode de descente en gradient sur les données d'apprentissages issues de l'échantillonnage du graphe biparti initialement considéré. Parmi les différentes fonctions d'agrégation que nous avons testées dans le cadre de cette approche, la plus simple exploite la moyenne des agrégats au voisinage $N(v)$ par couche dense soit :

$$h_{N(v)}^k = \frac{1}{|N(v)|} \cdot D_p[h_v^{k-1}] \quad (6.15)$$

Nous donnons la passe avant du modèle à la couche k comme suit :

$$h_v^k = \sigma(\text{concat}[\mathbf{W}^k D_p[h_v^{k-1}], \mathbf{W}_{\text{voisinage}}^k h_{N(v)}^k] + \mathbf{k}^k) \quad (6.16)$$

Où, \mathbf{W}^k et $\mathbf{W}_{\text{voisinage}}^k$ sont des paramètres (sous la forme de matrices) entraînaibles. \mathbf{b} est un biais optionnel, σ est la fonction d'activation non-linéaire et D_p est le *random dropout*². $N(v)$ représente le voisinage du nœud $v \in \{\mathcal{V}_u \cup \mathcal{V}_i\}$. Le nombre total de paramètre entraînable à la couche k pour cette fonction d'agrégation est estimé à $d_k \cdot d_{k-1} + d_k$, avec d^k la caractéristique associée à la dimension du nœud au voisinage considéré.

Nous avons aussi envisagé un autre type de fonction d'agrégation, qui mémorise les informations issues des agrégats incluant les représentations antérieurs des couches denses h_v^{k-1} à l'aide d'une couche de convolution. Ainsi, les caractéristiques des nœuds sont mis à jour selon l'équation :

$$h_{N(v)}^k = \frac{1}{|N(v)| + 1} (h_v^{k-1} + \sum_{u \in N(v)} h_u^{k-1}) \quad (6.17)$$

Où, la passe avant au voisinage k est définie de la manière suivante :

$$h_v^k = \sigma(\mathbf{W}^k \cdot h_{N(v)}^k + \mathbf{k}^k) \quad (6.18)$$

4 Expérimentation

L'entraînement de nos modèles *end-to-end* a été réalisé pour résoudre une tâche de prédiction de score(rating) d'un ensemble d'items donné. Pour y parvenir, nous avons

² Une probabilité D_p est appliquée aux arguments des vecteurs utilisés pour réduire le sur-apprentissage

considéré les données d'apprentissage issues des *datasets* respectifs *MovieLens 10-M*, *MovieLens 20-M* et *MovieLens 25-M* dont seules les instances d'utilisateurs ayant annotés plus de 20 tags ont été sélectionnées. Les *datasets* ont fait l'objet d'une étude détaillée dans le chapitre 4 ayant révélée les éléments suivants :

- Une forte distribution des notes des utilisateurs d'une certaine classe, (les utilisateurs ont majoritairement attribués la note de 5 aux items qu'ils ont annotés). Cette limitation peut faire perdre au modèle d'apprentissage sa capacité à généraliser. Cette limitation a été levé grâce à l'utilisation d'une méthode de sur-échantillonnage [106] qui ajuste la distribution de classe d'un ensemble de données en multipliant aléatoirement les données issues de la classe minoritaire jusqu'à ajustement du nombre d'instance de chaque classe.
- De par la nature personnel du tags, certains tags sont en dehors du vocabulaire (*out of vocabulary*) engendré par l'espace de représentation de *word2vec*. Nous avons réalisé une étude visant à déterminer la proportion de ces tags pour chaque dataset. Les résultats font part des proportions suivantes : 8%, 5%, 5% respectivement pour les datasets *MovieLens-10M*, *MovieLens-20M* et *MovieLens-25M*. Ainsi, nous avons assigné à ces tags des vecteurs à valeurs réelles aléatoires compris uniformément sur l'intervalle fixé par le modèle pré-entraîné. Cette limitation nous a aussi encouragée à envisager l'utilisation des modèles de langues neuronaux contextuels tel que : BERT (bidirectional encoder representations from transformers) pour construire des vecteurs d'embeddings contextuels de tags et de les exploiter sous la forme de représentations jointes d'utilisateurs et d'items à l'entrée de modèles neuronaux entraînés de bout en bout.

Avant d'explicitier le processus aboutissant à l'extraction d'embeddings de tags statiques et contextuels, nous donnons un aperçu des métriques utilisées pour l'évaluation ainsi qu'une description détaillée des paramètres d'apprentissage des modèles dans les sous-sections suivantes.

4.1 Métriques d'évaluation

Nous avons évalué nos modèles prédictifs en utilisant deux métriques largement répandues dans le domaine de la recommandation, L'erreur absolue moyenne (MAE *Mean Absolute Error en anglais*) et L'erreur quadratique moyenne (RMSE pour *Root-Mean-Square Error*). Pour une prédiction de la note d'un utilisateur donné $\hat{r}_{u,i}$ et la valeur représentant la note réelle $r_{u,i}$, RMSE est calculé comme suit :

$$RMSE = \sqrt{\frac{1}{N} \sum_{u,i} (r_{u,i} - \hat{r}_{u,i})^2} \quad (6.19)$$

Où N indique le nombre totale de notes.

MAE est calculé selon l'équation :

$$MAE = \frac{1}{N} \sum_{u,i} |r_{u,i} - \hat{r}_{u,i}| \quad (6.20)$$

De plus, nous avons aussi évalué le classement des listes d'items recommandées à l'aide de la mesure $NDCG @10$. Explicitement, l'évaluation du classement des listes d'items recommandées a été réalisé sur la base de la moyenne des notes (*i.e.*, 2.5) sur l'ensemble des datasets test de la collection. Ainsi, une note en dessous de cette valeur suggère que l'item n'est pas pertinent pour l'utilisateur, tandis qu'un item atteignant au plus cette note est considérée comme pertinent pour l'utilisateur.

4.2 Paramètres d'apprentissage

Nous avons divisé les données issues des datasets de façon aléatoire en ensemble d'entraînement 90% et avons gardé 10% de ces ensembles pour constituer les ensembles de tests. De surcroît, nous avons conduit une série d'expérimentations visant à évaluer la capacité des modèles à prédire des notes avec une marge d'erreur moindre. Pour cela, nous avons mené pour chaque modèle une série de 10-validation croisée en moyennant à chaque validation la mesure du MAE et du RMSE. Les expérimentations ont été menées sur la plateforme *Google colab*³ qui nous a permis de disposer d'unités de traitement tensorielles (*Tensor Processing Unit*) ou plus simplement TPU.

4.2.1 Hyper-paramètres

Lors de la phase d'entraînement, les modèles de prédiction que nous avons développés convergent vers une solution optimale, *i.e.* une erreur moyenne acceptable après plus de 1.5×10^5 itérations, ce qui correspond à 20 *epochs*, la taille des *batches* quant à eux sont de 128. De plus, nous avons optimisé les modèles grâce une méthode d'optimisation (*hyper-parameter optimization*) permettant de tester une série de paramètres et de comparer les performances pour en déduire le meilleur paramétrage. À l'issue de ce procédé dénommé, *grid-search*, nous avons fixé le taux d'apprentissage parmi les valeurs $\in \{0.0001, 0.0005, 0.001, 0.005\}$ et le nombre d'unités dans les couches cachées $\in \{100, 200, 300, 400, 500, 1000\}$ pour les modèles employant une architecture en MLP et en Autoencoder. À ce sujet, concernant l'architecture en autoencoder, une note par défaut de 2.5 a été attribuée aux items des ensembles de test. Enfin, les modèles de filtrage collaboratif neuronal basés sur des graphes bipartis prennent en compte un voisinage égal à $k = 2$ lors de la construction d'embeddings de noeuds. Concernant le *dropout ratio* D_p visible dans la sous-section 3.3, il est compris entre les valeurs $\in \{0.0, 0.1, \dots, 0.8\}$. Les modèles ont été optimisés à l'aide de l'algorithme d'optimisation *Adam*.

4.3 Extraction d'embeddings de tags à partir de modèles pré-entraînés

Dans cette sous-section, nous explicitons le processus permettant l'extraction d'embeddings de tags statiques en exploitant *word2vec* et contextuels en exploitant le modèle *BERT*. Nous rapportons dans ce qui suit les principales caractéristiques issues de l'exploitation de ces modèles prè-entraînés et la construction de représentations denses de tags.

3. <https://colab.research.google.com/>

- Représentations de tags issues de *word2vec* : Les tags issus de ces représentations ont été exploités sous la forme de vecteurs denses de dimension 300. Pour un utilisateur décrivant un item à l'aide d'un ensemble de tags, la moyenne de ces vecteurs denses a été utilisée comme vecteur de caractéristiques pour représenter l'utilisateur, de même pour un item donné. Cette méthode a permis de projeter les utilisateurs et les items dans un espace dense de caractéristiques régis par le vocabulaire des tags employés. Cependant, du fait que certains mots soient en dehors du vocabulaire en raison de mots d'argots, de l'orthographe mal mené, ou de la nature personnelle des tags, ces derniers ont été sujet à un pré-traitement linguistique diminuant (de peu) l'impact de ces tags sujets à ce type de limitation. Le reste de tags a été formalisé à l'aide d'un vecteur de valeurs réelles aléatoires. Par ailleurs, le modèle pré-entraîné *word2vec* a été entraîné sur un corpus de 3 milliards de mots à partir de GoogleNews⁴. Les représentations denses de tags générées gardent de ce fait le contexte général sur lequel le modèle a été initialement entraîné. Cela nous permet de qualifier ces représentations denses de *statiques*.
- Représentations de tags issues du modèle BERT : d'abord, il est utile de préciser que le modèle BERT apprend de façon non supervisée, qu'il est issu d'une architecture de type "*Transformer*" [221] et qu'il exploite des réseaux de neurones récurrents ainsi que des mécanismes d'attention. Afin d'exploiter pleinement ce modèle, nous avons considéré l'ensemble des tags qu'un utilisateur a attribué à un item comme étant une séquence de tags qui peut être composée de plusieurs mots ou phrases. Le *padding* est utilisé pour rendre toutes les séquences de même longueur. Ainsi, si une séquence est plus courte que la longueur maximale qui est de 512, alors le *padding* est utilisé pour allonger cette séquence. Inversement, si l'une des séquences dépasse la longueur maximale, cette dernière est tronquée (en supprimant les jetons supplémentaires) pour réduire la séquence. Nous avons mis en œuvre le modèle BERT-BASE pré-entraîné⁵ (12 blocs de dimension cachée 768, 12 têtes pour l'attention) en définissant le [CLS] qui indique le début d'une séquence et le [SEP] comme étant la fin de cette séquence. Dans ce cas précis, une séquence peut être constituée d'un ou plusieurs tags.

Dans la section suivante, nous comparons les modèles que nous avons développés à différentes approches de la littérature. Certaines de ces approches se basent exclusivement sur l'apprentissage des interactions entre utilisateurs et items sans pour le moins exploiter de contenus additionnels [196, 95]. Ces approches présentent dans leurs modèles des architectures qui incluent souvent des couches supplémentaires. Ces couches permettent l'apprentissage de ces interactions et figurent dans la plupart des approches de filtrage collaboratif, en autoencoder [196] ou sous la forme de *perceptron multicouches* [95]. Nous comparons aussi nos modèles avec certains modèles de la littérature tel que *Liang et al.*, [132] dont les auteurs intègrent des représentations de tags extraites de *word2vec* dans leurs modèles de recommandation neuronales. Enfin, nos modèles font aussi l'objet d'une

4. <https://news.google.com/>

5. BERT a été pré-entraîné sur les corpus *BookCorpus* qui est un corpus composé de 11038 livres non publiés appartenant à 16 domaines différents et de 2500 millions de mots provenant de passages de texte de Wikipédia en anglais.

comparaison avec certaines approches neuronales permettant l'apprentissage des représentations des nœuds d'un graphe, c'est le cas de *HinSAGE* [91].

5 Évaluation et comparaison avec d'autres approches

Dans le but de démontrer l'efficacité de nos stratégies de recommandation, nous les avons comparé avec certains modèles de base. Il s'agit pour la plupart, de modèles de recommandation issus de la littérature exploitant le filtrage collaboratif neuronal et qui exploitent différentes architectures de réseaux neuronaux. Nous avons détaillé ces modèles comme suit :

- **GMF CF-MLP** décrit dans [95] est une instance du modèle de filtrage collaboratif exploitant un perceptron multi-couches. Cette approche inspirée de la factorisation matricielle vise à l'apprentissage de la fonction d'interaction utilisateur-item. L'entrée du réseau consiste en deux vecteurs binarisés (encodage one-hot) représentant l'utilisateur et l'item. Un tel modèle n'utilise que l'identité d'un utilisateur et d'un item comme caractéristique d'entrée.
- **Neural CF-MLP⁺⁺** décrit en section 3, ce modèle intègre les tags sous la forme de deux vecteurs de caractéristiques formés par l'ensemble des représentations denses de tags par utilisateurs et par item. Ces vecteurs de caractéristiques sont extraits à partir de représentations vectorielles denses issues du modèle de langue neuronal *word2vec*, avec une dimensionnalité de l'ordre de 300, ce modèle est nommé CF-MLP⁺⁺_{*word2vec*}. Nous distinguons aussi la variante de ce modèle qui emploie des représentations vectorielles denses issues du modèle *BERT* avec une dimensionnalité de l'ordre de 768 dénommé CF-MLP⁺⁺_{*BERT*}.
- **U-Autorec** est un framework de filtrage collaboratif neuronal décrit dans [196] pour la prédiction des notes d'utilisateurs qui exploite une architecture en Autoencoder. Le framework prend les vecteurs d'utilisateurs incluant les notes attribuées aux items en entrée du réseau et les reconstruit dans la couche de sortie. Les valeurs des vecteurs reconstruits sont les valeurs prédites aux positions correspondantes.
- **CF-Autoencoder⁺⁺** Il s'agit de notre approche collaborative exploitant une architecture en autoencoder. Ce modèle intègre en entrée du réseau des représentations denses de tags sous la forme de produit de vecteurs de caractéristiques associés aux utilisateurs et aux items. Le produit de ces vecteurs denses est concaténé au vecteur représentant les notes d'utilisateurs puis et reconstruit pour prédire les notes. Nous dissocions le modèle exploitant des représentations denses (statiques) de tags issues de *word2vec* CF-Autoencoder⁺⁺_{*word2vec*} du modèle exploitant des représentations denses de tags contextuelles nommé CF-Autoencoder⁺⁺_{*BERT*}.

- **CF-GNN⁺⁺** ce modèle prédictif génère des représentations denses de nœuds à partir de graphes bipartis d'utilisateurs et d'items en exploitant une fonction d'agrégation (équation 6.15) qui agrège les caractéristiques des nœuds, c'est à dire l'ensemble des tags associés au nœuds à un certain voisinage exprimée sous forme de couches denses. Nous distinguons entre l'approche exploitant des représentations denses de tags statiques CF-GNN⁺⁺_{word2vec} et contextuelles CF-GNN⁺⁺_{BERT}.
- **CF-GCN⁺⁺** tout comme **CF-GNN⁺⁺**, ce modèle prédictif génère des représentations denses de nœuds à partir de graphes bipartis d'utilisateurs et d'items en exploitant une fonction d'agrégation (équation 6.17) qui agrège les caractéristiques des nœuds, c'est à dire l'ensemble des tags associés au nœuds à un certain voisinage. Cependant, dans le cas de ce modèle, ce voisinage est exprimée sous forme de couches à convolution. Nous distinguons entre l'approche exploitant des représentations denses de tags statiques CF-GCN⁺⁺_{word2vec} et contextuelles CF-GCN⁺⁺_{BERT}.
- **GraphSAGE (HinSAGE)** est un framework de la littérature [91] permettant l'apprentissage des représentation de nœuds d'un graphe générant des représentations denses de nœuds en se basant sur les informations de voisinage qui consiste au degré de chaque nœud (quantité d'arêtes qui lui sont incidentes) qui est combiné à des informations démographiques concernant les utilisateurs.
- **TRSDL_(word2vec)** [132] est un modèle neuronal hybride intégrant 2 réseaux de neurones. Le premier est un réseau de neurones profonds DNN prenant la moyenne des embeddings des tags par item. Les embeddings sont extraits à partir de *word2vec*. Le second se base sur un réseau de neurones récurrents RNN permettant l'exploitation de représentations temporelles d'utilisateurs. Les représentations latentes d'utilisateurs et d'items résultantes sont exploitées pour prédire les notes d'utilisateurs.

6 Résultats et discussion

Les résultats de nos expérimentations sont rapportés dans le tableau 6.2. Il décrit les scores des modèles prédictifs évalués selon les mesures du MAE et du RMSE.

6.1 Impacts des architectures neuronales

Les scores les plus prometteurs sont en faveur des approches neuronales en graphe exploitant le voisinage des représentations denses des nœuds, CF-GCN⁺⁺ avec un score de RMSE à 0.78 (*MovieLens-20M*) et CF-GNN⁺⁺ avec un score de RMSE à 0.799 (*MovieLens-10M*). Il est cependant à noter que ces scores sont associés aux modèles exploitant des représentations denses contextuelles de tags. Le tableau 6.2 synthétise les scores donne un aperçu de la qualité de classement des recom-

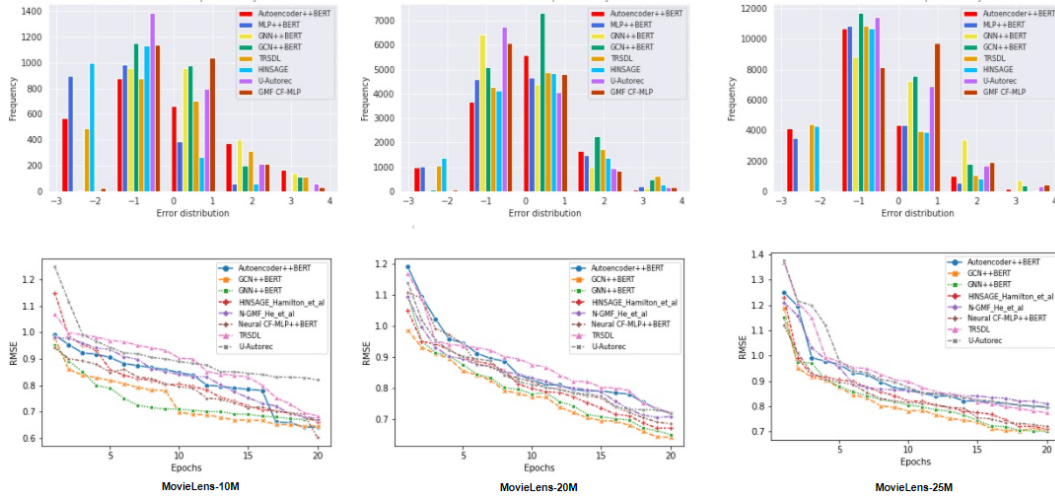


FIGURE 6.7 – En haut de cette figure, nous avons présenté la distribution des erreurs de chaque modèle neuronal. En bas, nous avons donné les scores de validation des modèles après 20 epochs.

mandations en exploitant la mesure du $NDCG@10$. Il est à noter que les approches collaboratives neuronales en graphe obtiennent de meilleures scores lorsque le dataset devient plus conséquent. Concernant les approches de la littérature, nous observons un score du $NDC@10$ à 0.47 (*MovieLens-20M*) et de 0.475 (*MovieLens-25M*) en faveur de l’approche HINSAGE [91] qui décrit des performances respectables.

6.2 Impacts de l’exploitation des embeddings contextels de tags

Nous avons détaillé la fréquence des erreurs de distribution parmi les ensembles tests issus des datasets *MovieLens-10M*, *MovieLens-20M* et *MovieLens-25M*. Pour montrer que les modèles qui exploitent des représentations contextuelles de tags peuvent rivaliser en terme de performance avec les modèles neuronaux de la littérature, voire les surpasser, nous avons comparé comme illustré dans la figure 6.7, la distribution des fréquences d’erreurs de nos modèles avec TRSDL[132], U-Autorec [196], HINSAGE[91] et GMF CF-MLP [95].

En comparant les modèles neuronaux selon le type de représentations denses auxquelles ils sont associés, nous avons mis en évidence l’impact des représentations contextuelles de tags sur la performance des modèles prédictifs. Comme le montre le tableau 6.2, concernant le dataset *MovieLens-10M*, les meilleurs scores RMSE et MAE ont été observé pour le modèle $CF-GCN^{++}_{conv} Agg(k=2) Bert$ avec $MAE = 0.715$ et $RMSE = 0.791$. Notre modèle qui exploite des embeddings de tags contextuels à aussi atteint le meilleur score de classement avec $NDCG@10 = 0.48$. Nous avons remarqué que l’extension de ce modèle qui exploite des embeddings statiques de tags $CF-GCN^{++}_{conv} Agg(k=2) W2V$ a aussi achevé de bonnes performances en surpassant la plupart des modèles de la littérature mise à part le modèle TRSDL [132] qui atteint $MAE = 0.73$, $RMSE = 0.810$ et $NDCG@10 = 0.45$.

Models	Measures								
	MoviLens 10M			MovieLens 20M			MovieLens 25M		
	MAE	RMSE	ndcg@10	MAE	RMSE	ndcg@10	MAE	RMSE	ndcg@10
Neural CF-MLP ⁺⁺ _{word2vec}	0.77	0.98	0.43	0.88	0.96	0.381	0.84	1.01	0.42
Neural CF-MLP ⁺⁺ _{Bert}	0.72	0.93	0.46	0.791	0.86	0.42	0.791	0.83	0.46
CF-Autoencoder ⁺⁺ _{word2vec}	0.83	1.1	0.411	0.85	0.97	0.39	0.80	1.02	0.42
CF-Autoencoder ⁺⁺ _{Bert}	0.76	0.96	0.42	0.811	0.89	0.44	0.798	0.865	0.445
CF-GNN ⁺⁺ _{Mean Agg(k=2)word2vec}	0.88	1.10	0.47	0.80	1.02	0.49	0.82	1.04	0.44
CF-GNN ⁺⁺ _{Mean Agg(k=2)Bert}	0.774	0.89	0.451	0.78	0.85	0.441	0.772	0.799	0.471
CF-GCN ⁺⁺ _{conv Agg(k=2)word2vec}	0.798	0.821	0.47	0.74	0.838	0.464	0.79	0.801	0.465
CF-GCN ⁺⁺ _{conv Agg(k=2)Bert}	0.715	0.791	0.48	0.723	0.782	0.47	0.712	0.787	0.48
U-Autorec Sedhain et al., [196]	0.82	1.09	0.38	0.84	1.07	0.37	0.81	1.01	0.40
GMF CF-MLP He et al., [95]	0.73	0.98	0.44	0.89	1.025	0.39	0.87	0.92	0.43
HINSAGE [91]	0.75	0.85	0.48	0.771	0.801	0.448	0.74	0.791	0.475
TRSDL [132]	0.73	0.810	0.45	0.74	0.820	0.461	0.75	0.87	0.44

TABLE 6.2 – Tableau récapitulant les scores RMSE, MAE et ndcg@10 des approches présentées

En analysant les résultats d'évaluation du modèle *Hinsage* [91] qui a atteint un score $MAE = 0.75$, $RMSE = 0.85$ et un score de classement (ranking) estimé à $NDCG@10 = 0.48$ ainsi que du modèle CF-GNN⁺⁺_{Mean Agg(k=2)Bert} qui atteint $MAE = 0.75$, $RMSE = 0.85$ avec un score de classement de $NDCG@10 = 0.451$. Nous sommes tentés d'affirmer que les modèles intégrant des approches de graphes neuronales décrivent de meilleures performances comparées à d'autres approches collaboratives neuronales, peu importe la nature des embeddings de tags intégrés initialement aux modèles. Cependant, en considérant la performance significative des modèles neuronaux intégrant des embeddings de tags contextuels tel que CF-MLP⁺⁺_{Bert} qui atteint $RMSE = 0.72$ et $MAE = 0.93$ ou encore CF-Autoencoder⁺⁺_{Bert} qui atteint $RMSE = 0.96$ et $MAE = 0.76$, notre réflexion s'est portée sur le moyen de déterminer quelle représentation (*embeddings* de tags) est la plus efficace en terme de performance parmi les architectures neuronales intégrant explicitement sous forme d'embeddings de tags statiques/contextuels ou sous la forme d'agrégats d'embeddings qui se basent sur le voisinage d'utilisateurs ou d'items. Les résultats ont montrés une amélioration significative de la qualité des recommandations pour les modèles exploitant des représentations de tags contextuelles agrégées à un voisinage jointes à une architecture intégrant des couches collaboratives au sein du modèle neuronal. Les modèles exploitant des représentations denses de tags statiques sont tout aussi efficace, dans la mesure où ces modèles fournissent des résultats comparable aux modèles qui ont été proposés dans la littérature.

6.3 Impact de la fonction d'agrégation sur les modèles d'apprentissages

Comme l'illustre la figure 6.7 (en bas), nous donnons pour chaque modèle les scores de validation après 20 epochs. Cela nous permet d'estimer la capacité des modèles qui ont été préalablement entraîné à généraliser. Dans la figure 6.7, en

haut, nous avons analysé les modèles à taux de convergence optimal. Il apparaît que parmi les collections $ML-10M$, $ML-20M$, $ML-25M$, le taux de convergence des modèles exploitant des approches neuronales en graphe est significativement plus optimal. En particulier, $CF-GNN^{++}_{Mean\ Agg(k=2)\ Bert}$ and $CF-GCN^{++}_{conv\ Agg(k=2)\ Bert}$ qui exploitent des embeddings contextualisés de tags agrégés. Cela nous permet d'affirmer que l'exploitation d'embeddings de tags contextualisés et agrégés à travers une fonction d'agrégation spécifique qui inclut le voisinage d'utilisateurs et d'items est une piste prometteuse en voie d'améliorer les modèles de recommandation neuronaux. Nous soutenons que ce type d'approche tient compte de la sémantique complexe que véhicule les tags et qui inclut une formalisation abstraite du voisinage des tags et qui a conduit à des représentations plus explicites. L'apprentissage de la fonction d'agrégation permet de saisir des connaissances exploitées sous la forme d'*embeddings* de tags à même de distinguer entre utilisateurs au comportement d'annotation voisin ou entre items ayant été décrits en utilisant une sémantique relativement proche.

7 Conclusion

Les approches neuronales proposées exploitent des méthodes collaboratives pour la recommandation. Elles ont comme particularité l'exploitation explicite du vocabulaire engendré par les tags des utilisateurs. Les résultats d'expérimentations ont mis en lumière l'utilité de représentations denses de tags à travers différentes architectures neuronales. Dans cette recherche, nous nous sommes attelés à montrer que ces représentations denses de tags impactaient la qualité des modèles de recommandation, mais influaient également sur le classement de ces recommandations.

Les résultats des évaluations ont aussi permis de mettre en évidence le choix de la méthode collaborative à privilégier. Nous avons prouvé que l'exploitation de représentations denses grâce à des fonctions d'agrégation au voisinage d'utilisateurs et d'items permettent d'améliorer la qualité de ces représentations denses. En effet, au lieu de simplement plonger les vecteurs d'utilisateurs et d'items dans un espace de représentation dense, nous avons exploré la prise en compte dans ce même espace, du voisinage associé à ces utilisateurs et à ces items grâce notamment à une modélisation en graphe biparti. Cela nous a permis d'exploiter ces représentations dans un modèle collaboratif de graphe neuronal de bout en bout, et a révélé de meilleures performances dans la tâche consistant à prédire des notes d'utilisateur à des items.

Parmi les architectures neuronales que nous avons testées, nous estimons que les modèles exploitant le filtrage collaboratif sous la forme de graphes neuronaux permettent une meilleure approximation des notes d'utilisateurs, en particulier si l'on associe à ces modèles des représentations continues (denses) contextuelles de tags qui prennent compte d'un voisinage. Ce voisinage découle de la combinaison de représentations denses d'utilisateurs et d'items que l'on a exploitées grâce à une fonction d'agrégation. L'apprentissage de cette fonction que l'on a formalisé

sous différentes manières nous a permis de construire des modèles neuronaux de recommandation performants.

CONCLUSION GÉNÉRALE

Synthèse des contributions

Chapitre 7

CONCLUSION GÉNÉRALE

1 Synthèse des contributions

LES travaux détaillés dans ce manuscrit s’inscrivent dans le contexte général de la recommandation personnalisée. Plus particulièrement, dans l’exploitation des données personnelles d’utilisateurs véhiculées à travers les folksonomies dans des modèles de recommandation. À ce sujet le chapitre 2 s’est focalisé sur les multiples représentations qui pouvaient être engendrées à partir de ces méta-données. Nous avons dans un premier temps, exploré des méthodes de recommandation incluant des représentations de tags sous la forme de profils. Pour cela, nous avons focalisé notre état de l’art sur les approches de filtrage collaboratives permettant de combiner certaines caractéristiques comportementales d’utilisateurs aux représentations classiques de tags dans le but d’améliorer la pertinence de ces profils pour des tâches de recommandation. Le chapitre 3 décrit dans le détail ce type d’approches. Plus spécifiquement, nous avons abordé deux questions principales :

- Lors de la modélisation de profils, les intérêts des utilisateurs décrits à travers leurs tags sont-ils uniques ou multiples ?

Pour cela, nous nous sommes intéressés aux activités d’annotations d’utilisateurs et avons remarqué que les approches décrivant les profils de l’utilisateur sous la forme de vecteurs uniques de tags ne pouvaient représenter efficacement ces divers centres d’intérêts. Dans le chapitre 5, nous avons exploré la possibilité de représenter les multiples intérêts de l’utilisateur sous la forme de clusters d’intérêts grâce notamment à une mesure de pondération permettant la propagation du degré de préférence normalisé propagé par les tags de l’utilisateur qui a été mise en œuvre en amont du processus de clustering. Les résultats ont prouvé que les profils issus de cette méthode décrivaient plus précisément les intérêts des utilisateurs.

- En considérant que les profils d’utilisateurs modélisent les multiples centres d’intérêts de l’utilisateur, sont-ils figés ou évoluent-ils dans le temps ?

Pour répondre à cette question, nous avons étendu le concept de propagation du degré de préférence normalisé et propagé par les tags de l’utilisateur à travers une fonction temporelle que nous avons exploitée pour décrire l’évolution des profils dans le temps. Nous avons montré à travers notre contribution que l’exploitation de ces profils dans des tâches de recommandation permettait un classement plus pertinent des listes de ressources à recommander.

Dans un deuxième temps, nous avons exploré des modèles neuronaux de recommandation incluant des représentations denses d'utilisateurs et d'items. Ces modèles ont fait l'objet d'une étude détaillée dans l'état de l'art. Aussi, nous nous sommes intéressés à la possibilité d'intégrer les tags au sein de ces modèles neuronaux. L'idée principale à la base de cette recherche a été d'étudier des techniques de représentations exploitant des vecteurs d'embeddings de tags dans un espace de représentation joint d'utilisateurs et d'items. L'apprentissage de ces représentations a pour principal but la découverte de certaines caractéristiques non-triviales que nous exploitons dans un modèle neuronal pour la prédiction de notes d'utilisateurs.

- Dès lors, nous nous sommes posés la question suivante : comment plonger le triplet { utilisateur, item, tags } dans un espace de représentation dense ?

Pour répondre à cette question, nous avons exploité le potentiel des modèles de langues neuronaux pré-entraînés pour générer des représentations denses d'utilisateurs et d'items associées aux tags. Cela a été une piste prometteuse que nous avons développée dans le chapitre 6. En effet, vu que dans le processus « d'embedding » en langage naturel, la représentation extraite garde des propriétés pouvant être décrite comme des relations sémantiques. Cependant, nous avons tenu compte de la complexité du vocabulaire engendré par les folksonomies, notamment en associant les tags à des représentations vectorielles contextuelles et bidirectionnelles. Une fois avoir extrait ces représentations, nous les avons employé comme vecteurs de caractéristiques (vecteurs d'*embeddings*) en exploitant différentes architectures neuronales. Ces architectures sont issues de méthodes de filtrage collaboratif neuronales. Plus spécifiquement, nous avons testés des modèles neuronaux en MLP et en autoencoder. En outre, nous avons pris l'initiative d'enrichir les représentations d'utilisateurs et d'items associées aux tags.

Dans cette optique, nous avons tiré profit de méthodes d'apprentissage neuronales de fonctions d'agrégations aux voisinages d'utilisateurs et d'items représentés initialement sous la forme de graphes bipartis. Cela nous a permis d'agréger des représentations du voisinage d'un nœud et d'obtenir ainsi des représentations avec un plus haut niveau d'abstraction. Les représentations denses d'utilisateurs et d'items issues de ces méthodes d'apprentissage ont été exploitées sous la forme de vecteurs de caractéristiques intégrant notamment des représentations denses extraites de modèles de langues neuronaux pré-entraînés.

De surcroît, ces représentations ont été associées à des modèles de graphes neuronaux collaboratifs pour la recommandation. Ces modèles ont permis l'association de niveaux de voisinage sous la forme de couches denses dans un modèle neuronal en graphe GNN exploitant une fonction d'agrégation spécifique ou sous la forme de couches à convolution GCN qui utilise à son tour une fonction d'agrégation dédiée. En associant à chacun des modèles une couche de régression linéaire ainsi qu'une phase d'apprentissage de bout en bout, les notes des utilisateurs ont pu mieux être approximées.

Nous avons par ailleurs, évalué ces différents modèles neuronaux pour la recommandation et avons prouvé que les modèles implémentant le filtrage collaboratif dans

les graphes neuronaux ont amélioré les résultats de modèles neuronaux traditionnels rencontrés dans la littérature. Les performances de ces modèles se traduisent par la qualité des représentations qui ont été étendues aux voisinages aux moyens de fonctions d'agrégation a permis la découverte de connaissances plus profondes que celles issues des modèles de langues pré-entraînés.

2 Perspectives

Nos contributions présentées au cours de ce manuscrit peuvent s'étendre sur plusieurs axes :

- Lors de notre première contribution, nous avons mis en évidence des représentations prenant la forme de profils d'utilisateurs à multiples centres intérêts évolutifs. Cela a été rendu possible grâce à l'exploitation d'une méthode non supervisée, i.e le clustering [80]. Il existe un certain nombre d'architectures de réseaux de neurones spécialement conçu pour le clustering neuronal [129]. Une possible extension de notre première contribution consisterait en l'intégration d'une architecture neuronale dédiée qui impliquerait cependant un certain paramétrage, notamment concernant le nombre de clusters à retourner par profil d'utilisateur.
- À court terme, nous envisageons d'explorer les moyens d'améliorer la précision des profils d'utilisateurs en explorant la possibilité d'inférer à partir de plongement lexicaux (*embeddings*) de tags des mesures de similarités neuronales ou étendre celles déjà existantes. Ces mesures seraient exploitées dans le cadre de systèmes de recommandation proposant tout comme dans les travaux de *Xu et al.*, [235] un certain niveau d'abstraction dans l'apprentissage des représentations d'utilisateurs et d'items. Cette perspective est envisageable dans le cadre d'une tâche de recommandation personnalisée.
- Dans le contexte actuel de nos recherches, nous avons considéré le triplet {utilisateur,item, tags } dans un espace de représentation dense. Or, ce triplet peut intégrer d'autres informations comme la temporalité ou la polarité. À long terme, la prise en compte de ces informations supplémentaires dans un espace de représentation serait une piste intéressante. Tout comme l'est la perspective d'un modèle d'apprentissage de bout-en-bout d'un espace exploitant conjointement les informations issues d'annotations sociales (tags) et de contenus textuel comme par exemple les avis en tout genre d'utilisateurs (critiques, avis, commentaires).
- Pour y parvenir, l'exploitation d'autres méthodes d'apprentissage automatique comme le modèle *seq2seq* [205] qui est utilisé pour le traitement du langage serait une perspective fondée. À plus long terme, une étude détaillée du sujet révélerait l'impact de l'exploitation de réseaux de neurones récurrents (RNN) ou des variantes de cette architecture comme LSTM dans un processus de filtrage collaboratif neuronal pour la prédiction de notes d'utilisateurs. Cependant, contrairement aux travaux décrits dans *Liang et al.*, [132], nous pensons que l'exploitation d'embeddings contextualisés de tags pourrait aboutir à de meilleures performances. Enfin, on pourrait aussi consi-

dérer l'apprentissage automatique en ligne ou *online machine learning* en mettant à jour l'ensemble des données d'entraînement lors de la phase d'apprentissage.

Annexe I : Apprentissage automatique & réseaux de neurones

ANNEXE I : APPRENTISSAGE AUTOMATIQUE

1 Introduction

L'APPRENTISSAGE automatique fournit l'ensemble des moyens, méthodes statistiques et algorithmes nécessaires dans le but d'optimiser les performances d'une tâche donnée. La démocratisation des processeurs graphiques (Graphic Processing Units ou GPU) [160] a permis un regain d'intérêts dans ce domaine. En cause, leur structure hautement parallèle leur a conféré une plus grande efficacité que les processeurs standards (Central Processing Unit ou CPU)¹, nécessaires pour traiter de grandes quantités de données en parallèle tels que les réseaux de neurones. Récemment, de nouvelles méthodes non-linéaires s'appuyant sur des architectures neuronales composées de plusieurs couches de traitement ont permis l'apprentissage de représentations de données avec plusieurs niveaux d'abstractions. Ces nouvelles méthodes sont associées au concept d'apprentissage profond qui consiste en un ensemble de techniques permettant à un réseau de neurones d'apprendre grâce à un grand nombre de couches permettant d'identifier des caractéristiques.

1.1 Réseaux de neurones formels

Un réseau de neurones est composé de cellules connectées entre elles par des liens pondérés qui permettent le calcul. Les poids des liens représente l'intensité des connexions synaptiques représentant l'activité biologique du cerveau humain. Un réseau de neurones artificiels est un système permettant à des noeuds de calcul connecté, de recevoir, de traiter et de combiner l'information d'autres noeuds grâce à des fonctions de combinaison et d'activation. Dans un tel système, un ensemble de paramètres libres et entraînable, incluant les poids des connexions synaptiques w_i et un biais b sont nécessaire à mettre en oeuvre pour faire parvenir les signaux numérique entre les noeuds. Un ensemble de paramètre nommés *paramètres libres* est associé à chaque neurones pour qu'il soit entraîné pour une tâche précise. C'est ensemble θ est défini par :

$$\theta = (b, \mathbf{w}) \in \mathbb{R} \times \mathbb{R}^n \quad (\text{I.1})$$

Dans l'équation I.1, $\mathbf{w} = (w_1, \dots, w_n)$ représente le vecteur des poids synaptiques associés au vecteur des entrées x de taille n avec b qui représente le biais. Une fonction de combinaison est généralement utilisé pour calculer l'état du neurone, il agrège les éléments d'entrée X avec le vecteur des poids synaptiques \mathbf{w} . Cette fonction Σ est exprimée de la manière suivante :

$$\Sigma(x; \theta) = \sum_{i=0}^n \theta_i x_i = \sum_{i=1}^n w_i x_i + b \quad (\text{I.2})$$

Lors de la réception des signaux d'informations, une somme pondérée des entrées est calculée pour constituer la fonction d'activation du neurone. C'est une fonction ma-

¹. Plus un réseau de neurones comprend de couches, plus le nombre de calculs nécessaire pour l'entraîner sur un CPU augmente

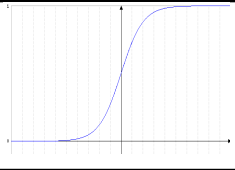
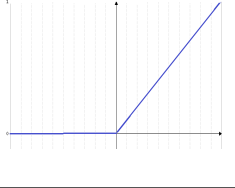
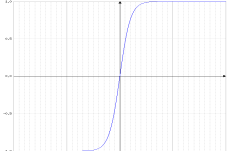
Fonction	Equation	Courbe
Sigmoïde	$\sigma(x) = \frac{1}{1+e^{(-x)}}$	
ReLU	$\sigma(x) = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{sinon} \end{cases}$	
Tanh	$\sigma(x) = \frac{2}{1+e^{(-2x)}} - 1$	

TABLE I.1 – Tableau récapitulatif de fonctions d’activation populaires

thématique qui convertit la somme pondérée des signaux afin de produire la sortie y du neurone. Cette fonction peut aussi apportée une non-linéarité dans le fonctionnement du neurone. Elle est noté σ tel que :

$$\sigma(\Sigma(x, \theta)) = \sigma\left(\sum_{i=1}^n w_i x_i + b\right) \quad (\text{I.3})$$

La fonction d’activation est utilisée pour contrôler la propagation de l’information dans un réseau de neurones multicouche, en fonction d’un seuil ρ . Sur la base de la valeur de ce seuil et de la sortie y , le neurone comporte trois états. Si le neurone est inactif alors $y < \rho$, il est actif si $y > \rho$ et en transition si $y \simeq \rho$. Selon l’objectif, différentes fonctions d’activation peuvent être mise en oeuvre, les plus populaires sont la fonction sigmoïde, la fonction unité de Rectificateur Linéaire et la fonction tangente hyperbolique. D’autres fonctions d’activation comme Unité de rectification linéaire paramétrique (PReLU) ou Unité exponentielle linéaire (ELU) sont utilisées selon leurs caractéristiques respectives. Un aperçu de ces fonctions ainsi que de leurs courbes sont présentées dans le Tableau I.1.

1.2 Modèles de réseaux de neurones artificiels

Les Perceptrons multicouches introduits par Rumelhart et McClelland en 1986 [182], constituent le modèle de neurones artificiels le plus fréquemment utilisé. Chaque neurone calcule une somme pondérée de ses entrées qu’il va transmettre à la fonction σ afin de produire ses sorties. Pour chaque couche du réseau de neurones dans un réseau à perceptron multicouches, un terme de biais est associée à cette dernière. Comme déjà abordé, un biais est un neurone dans lequel la fonction d’activation est en permanence égale à 1. Les neurones et les biais sont organisés dans une structure en couches (feed-forward en anglais ou à propagation avant). Le MLP peut donc s’interpréter comme un

modèle entrée-sortie dans lequel les poids et les seuils étant les paramètres libres et ajustables du modèle. Ils sont théoriquement capable d'estimer n'importe quelle fonction. Le nombre de couches et le nombre d'unités dans chaque couche détermine la complexité de la fonction. Cependant, lors de la conception des Perceptrons Multicouches, il est important de bien spécifier le nombre de couches cachées ainsi que le nombre de neurones dans ces couches [24]. Il est également important de bien choisir les fonctions d'activation et les méthodes d'apprentissage. Le MLP présenté dans la figure I.1 se com-

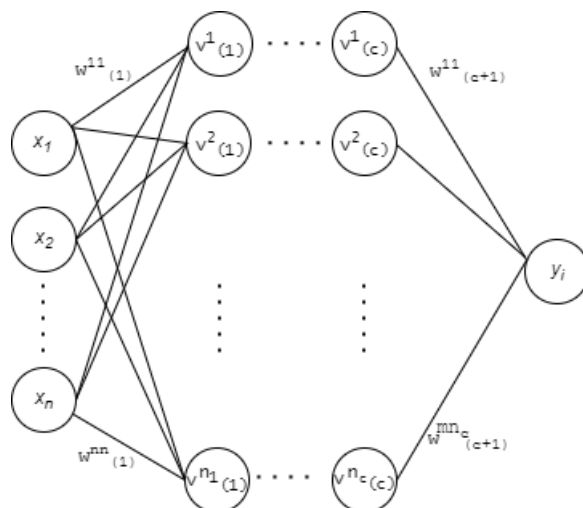


FIGURE I.1 – Architecture MLP pour de la régression

pose d'un ensemble de couches cachées $(l_{(1)}, \dots, l_{(c)})$ ainsi qu'une couche de sortie $l_{(c+1)}$. La couche d'entrée $l_{(0)}$ qui se compose de n entrées doit être connectée qu'avec les neurones de la couche suivante pour garantir une propagation des signaux d'information vers l'avant. Les couches cachées représentent l'état intermédiaires des neurones auquel cas : Un neurone $v_{(i)}^j$ dans une couche $l_{(i)}$ reçoit $n_{(i-1)}$ signaux d'entrée provenant des neurones de la couche précédente $l_{(i-1)}$. De ce fait, la connexion de ce neurone avec le neurone $v_{(i-1)}^k$ de la couche précédente $l_{(i-1)}$ est pondéré par le poids $w_{(i)}^{jk}$.

Pour le neurone situés dans la dernière couche cachée $l_{(c)}$, la valeur de sortie y_i de ce neurone peut être calculer selon l'équation suivante :

$$y_i = v_{(c+1)}^j = \sigma_{(c+1)}\left(\sum_{k=1}^{n_c} w_{(c+1)}^{jk} v_{(c)}^k\right) \quad (\text{I.4})$$

Dans l'équation I.4, il est à noter que le nombre de neurones inclus dans la couche de sortie dépend de la tâche pour laquelle le MLP est préalablement entraîné. Dans le cas du MLP de la figure I.1, il s'agit d'une architecture adaptée à une tâche de régression. Le processus d'entraînement corrige l'ensemble des paramètres libres pour une meilleure approximation de la prédiction via plusieurs itérations d'apprentissage. En effet, les réseaux de neurones doivent être entraînés suffisamment pour être performant. Pour cela, il est nécessaire de définir une fonction de coût permettant d'estimer de manière optimale les paramètres du réseau. L'apprentissage peut être supervisé, si l'on dispose des labels des données d'apprentissage ou non supervisé dans le cas contraire.

1.3 Architecture de réseaux de neurones en Autoencoder

Introduit dès les années 1990 par *Hinton et Geoffrey* [98], l'architecture en autoencoder décrit un réseau de neurones à trois couches : une couche d'entrée une couche de sortie ainsi qu'une ou plusieurs couches cachées les reliant, toutefois la couche de sortie doit posséder le même nombre de nœuds que la couche d'entrée. L'objectif étant de prédire une valeur cible \hat{X} en une valeur la plus proche possible de l'entrée X . Ce type de réseau de neurones est semblable au MLP, il se compose de deux parties, l'encodeur et le décodeur qui sont définies comme étant des transitions ϕ et ψ tel que :

$$\begin{aligned}\phi &: X \rightarrow F \\ \psi &: F \rightarrow X\end{aligned}\tag{I.5}$$

Avec $\phi, \psi = \operatorname{argmin} \|X - (\psi \circ \phi)X\|^2$. Pour un autoencoder avec une seule couche cachée, la phase d'encodage prend l'entrée $\mathbf{x} \in \mathbb{R}^d = X$ en l'associant à $\mathbf{z} \in \mathbb{R}^p = F$ tel que : $\mathbf{z} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b})$. La variable \mathbf{z} représente l'espace des variables latentes ou communément appelé représentation latente et σ la fonction d'activation, \mathbf{W} est la matrice de poids et \mathbf{b} le vecteur de biais. La phase de décodage quant à elle associe \mathbf{z} à la reconstruction de \mathbf{x}' de forme identique à \mathbf{x} tel que : $\mathbf{x}' = \sigma'(\mathbf{W}'\mathbf{z} + \mathbf{b}')$. Un autoencoder est entraîné pour minimiser l'erreur de reconstruction exprimé via la mesure de l'erreur quadratique :

$$\zeta(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|^2 = \|\mathbf{x} - \sigma'(\mathbf{W}'(\sigma(\mathbf{W}\mathbf{x} + \mathbf{b}))) + \mathbf{b}'\|\tag{I.6}$$

Où \mathbf{x} représente généralement la moyenne des données de l'ensemble d'apprentissage.

2 Algorithme d'apprentissage

Une distinction majeure dans le domaine de l'apprentissage automatique consiste à différencier entre apprentissage supervisé, lorsque les données d'apprentissage sont étiquetées (labellisées) et non supervisé lorsque les données d'apprentissage ne le sont pas. De manière générale les algorithmes d'apprentissage peuvent se catégoriser selon les deux principaux modes d'apprentissage qu'ils emploient, le tableau I.2 dresse quelques caractéristiques dont les objectifs et les données que ces modes d'apprentissage manipulent en entrées.

2.1 Apprentissage supervisé de réseaux de neurones

Selon la tâche pour lequel le réseau de neurones est entraîné et une fois l'architecture définie, le choix d'une fonction de coût ζ est nécessaire afin de répondre efficacement à la problématique à traiter. Cette fonction de coût permet d'estimer les paramètres optimaux du réseau θ . Elle est associée à un algorithme d'apprentissage qui se base sur le principe de rétropropagation du gradient [125, 84]. Il s'agit d'une méthode pour calculer le gradient de l'erreur pour chaque neurone [23]. L'algorithme corrige les erreurs en calculant le gradient (dérivé) de la fonction de coût. En d'autres termes, il minimise la fonction de coût en corrigeant les paramètres libres θ du réseau, à partir de données

	Apprentissage supervisé	Apprentissage non supervisé
Observations / Données	- Observations brutes issues de variables aléatoires	- Données issues d'exemples labellisés
Objectif	- Découverte de relations - Variables structurelles latentes	- Prédiction de scores - Nouvelles observations
Données en entrée	- Algorithmes entraînés sur des données labellisées	- Algorithmes/fonction d'optimisation sur des données non labellisées
Utilisation	Réseaux de neurones, régression linéaire/logistique	- Clustering/K-means clustering hiérarchique

TABLE I.2 – Quelques caractéristiques des modes d'apprentissage supervisé/non supervisé

d'apprentissage. La rétropropagation permet par la suite de propager itérativement le gradient à travers tous les noeud du réseau de neurones de la dernière couche à l'entrée.

2.2 Fonction de coût

Formellement, une fonction de coût ζ permet de pénaliser les erreurs observées lors de la phase d'apprentissage. Dans le contexte d'un apprentissage supervisé, pour un vecteur d'entrée x ayant comme valeur observé $y \in \mathbb{R}^+$ et comme valeur prédite \hat{y} à la sortie du réseau de neurones $f(x, \theta)$, la fonction de coût $\zeta(y, \hat{y}, \theta)$ est définie pour minimiser l'erreur moyenne absolue (*MAE* pour *mean average error*) des données issues de l'ensemble d'entraînement. Dans le cas d'une régression d'autres fonctions de pertes sont fréquemment utilisées, c'est le cas de l'erreur quadratique (*square error*) et l'erreur quadratique moyenne (*RMSE* *mean square error*). Les équations de ces fonctions sont détaillés dans le tableau I.3 avec n le nombre d'observations de la collection d'entraînement.

Fonction de coût	Equation
Erreur moyenne absolue (<i>MAE</i>)	$\frac{\sum_{i=1}^n \hat{y}_i - y_i }{n}$
Erreur quadratique	$(y - \hat{y})^2$
Erreur quadratique moyenne (<i>RMSE</i>)	$\sqrt{\frac{\sum_{i=1}^n \hat{y}_i - y_i ^2}{n}}$

TABLE I.3 – Fonctions de coût utilisées pour la régression

2.2.1 Application en TALN : Représentations distribuées des mots

L'une des applications les plus populaires des réseaux de neurones dans le domaine du TALN sont les représentations distribuées des mots. Ces représentations tirent leurs origines du modèle de langue neuronal introduit par *Bengio et al.*, [18]. Il s'agit d'un modèle de langue probabiliste dans lesquels les probabilités des mots sont calculées grâce

à une architecture neuronale qui apprend simultanément les représentations distribuées (plongements lexicaux) et la fonction de probabilité correspondant aux fenêtres contextuelles des mots. Les modèles de langues neuronaux calculent la probabilité $P(w_t|c)$ qu'un mot $w_t \in V^t$ apparaisse dans un contexte $c = \{w_1 \dots w_{|c|}\}$ et soit perçu comme une séquence de mots qui précède le mot w_t . Cette probabilité est estimée avec un réseau de neurones. Le modèle de langue neuronal optimise une fonction ζ de coût durant un processus d'entraînement impliquant tous les mots $w_t \in V^t$ et leurs fenêtres contextuelles. Cette fonction est définie de manière suivante :

$$\zeta(\theta) = \sum_{\langle w_t, c \rangle} \log P(w_t|c; \theta) \quad (\text{I.7})$$

Où θ définit les paramètres du réseau de neurones utilisés pour estimer la probabilité P . L'approche de Mikolov [152] se base sur ce principe et propose deux variantes d'un algorithme : il s'agit de *Word2vec*. Le modèle continu bag-of-words (sac de mots continu CBOW) a pour objectif de prédire un mot à partir de ses mots voisins, le modèle *skip-gram* a pour objectif de prédire les mots du contexte à partir d'un mot central. Cependant, les travaux de Mikolov et al., [152] se sont basés sur plusieurs techniques pour améliorer d'une part l'efficacité de l'apprentissage et d'autre part la qualité des représentations. Pour ce faire, la notion de fenêtre contextuelle d'un mot w_t qui a été adaptée de sorte à ce que ce mot fasse référence aux n mots précédents et aux n mots suivants : $c = \{w_{t-n+1} \dots w_t \dots w_{t+n-1}\}$. Ces deux architectures se présentent sous la forme de réseaux de neurones artificiels simples.

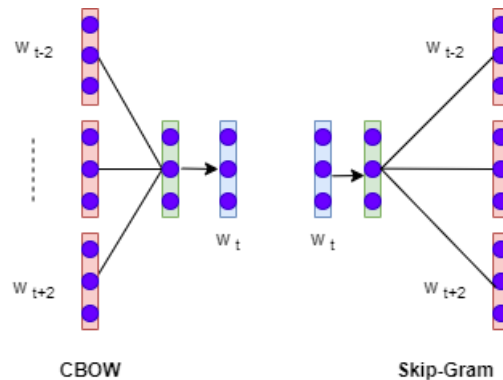


FIGURE I.2 – Architectures CBOW et Skip-gram de *Word2vec*.

Elles sont constituées de trois couches : une couche d'entrée, une couche cachée et une couche de sortie. La couche d'entrée contient un "sac-de-mots" (CBOW) ou un seul mot (Skip-gram) (Figure I.2). La couche cachée correspond à la projection des mots d'entrée dans la matrice des poids. Cette matrice est partagée par tous les mots. Enfin, la couche de sortie est composée de neurones de type "softmax". Pour des raisons de complexité algorithmique due à la couche "softmax", Mikolov et al., [152], ont introduit deux fonctions appelées "échantillonnage négatifs" et "softmax hiérarchique". Plus précisément, dans l'architecture CBOW chaque mot dans le contexte est projeté dans la matrice des poids du modèle. La somme (ou la moyenne) de ces représentations passe ensuite par la couche de sortie. Enfin, le modèle compare sa sortie avec le mot seul et corrige sa représentation grâce à une rétro-propagation du gradient. L'architecture Skip-Gram tente, quant à elle de prédire, pour un mot donné, le contexte dont il est issu. La

couche d'entrée de ce réseau est alors un vecteur ne contenant qu'un seul mot. Le mot est projeté dans la couche cachée puis dans la couche de sortie. Le contexte est ensuite réduit de façon aléatoire à chaque itération. Le vecteur de sortie est ensuite comparé à chacun des mots du contexte réduit et le réseau se corrige par rétro-propagation du gradient. De cette manière, la représentation du mot d'entrée va se rapprocher de chacun des mots présents dans le contexte. Dans ces deux configurations, pour chaque paire mot/contexte, dénoté (w_t, c) , un score neuronal $score_\theta(w_t, c)$ est calculé par le produit entre la représentation d'entrée du contexte c et la représentation de sortie du mot à prédire w_t :

$$Score_\theta(w_t, c) = e_{sortie}(w_t) \times e_{entree}(c) = (\vec{w}_t \times e_{sortie})(\vec{c} \times e_{entree}) \quad (I.8)$$

Où e_{entree} et $e_{sortie} \in \mathbb{R}^{V^t \times |dim|}$ sont les poids synaptiques de la couche de taille $|dim|$ et définissent l'espace de représentation des plongements lexicaux. Les vecteurs \vec{c} et \vec{w}_t sont respectivement les vecteurs de représentation de contexte et du mot.

D'autres méthodes de représentations de mots en fonction de leurs contextes ont fait leurs apparition. C'est le cas de *GloVe* (*Glove global vectors for word representation*) [167], qui est un modèle de représentation qui se base sur des matrices de cooccurrences mot-contexte. *Pennington et al.*, [167] ont proposé de combiner contexte local et global à travers une matrice de co-occurrence globale pour l'apprentissage de représentations de mots. Ce procédé est décrit par l'équation :

$$\zeta = \sum_{i,j}^{V^t} f(x_{i,j})(e(w_i) \cdot e(w_j) + b_i + b_j - \log x_{ij})^2 \quad (I.9)$$

Avec, $x_{i,j}$ qui représente le nombre de co-occurrences des mots w_i et w_j , $e(w_i)$ et $e(w_j)$ les représentations distribuées des mots w_i et w_j . V^t le vocabulaire et la fonction $f(x)$ une fonction de poids qui est définie de la manière suivante :

$$f(x) = \begin{cases} (x/x_{max})^\alpha & \text{si } x \leq x_{max} \\ 1 & \text{sinon} \end{cases} \quad (I.10)$$

Parmi les méthodes de plongements qui ont été proposées dans la littérature, certaines ont exploitées des représentations de mots sous la forme de graphes pondérés. C'est le cas de l'algorithme *LINE* [211] (*Large-scale Information Network Embedding*). L'algorithme s'appuie sur la notion de proximité entre les *embeddings* générées à partir des nœuds d'un graphe de co-occurrence. Le contexte général d'un mot est encodé en fonction du voisinage d'un nœud du graphe. *LINE* associe à chaque nœud du graphe, une représentation vectorielle des proximités au voisinage de ce nœud. Cette notion de proximité est d'abord abordée sous la forme d'une proximité du premier ordre qui est définie localement entre paire de nœuds, autrement dit, c'est le poids d'une arête reliant deux nœuds. Dans un graphe $G = (V, E)$, elle est défini comme étant le poids $w_{(i,j)}$ d'une arête entre les nœuds v_i et v_j . Si il n'existe pas d'arête entre ces nœuds, cette proximité est nulle. Ainsi, le voisinage du premier ordre d'un nœud v_i contient tous les nœuds v_j qui sont directement connectés à v_i par une arête. Pour atteindre cet objectif, *LINE* [211] minimise la fonction objectif δ_1 suivante :

$$\delta_1 = \sum_{(v_i, v_j) \in E} w_{(i,j)} \log \left(\frac{1}{1 + e^{-(\hat{v}_i, \hat{v}_j)}} \right) \quad (\text{I.11})$$

\hat{v}_i et \hat{v}_j représentent les *embeddings* (plongements lexicaux) des nœuds v_i et v_j dans le graphe G . Cependant, un autre type de proximité a été définie entre paires de nœuds v_i et v_j , il s'agit de la similarité entre les voisinages de premier ordre de ces nœuds. En effet, si $p1_{v_i} = (w_{(v_i,1)}, \dots, w_{(v_i,|V|)})$ représente la proximité de premier ordre du nœud v_i sur l'ensemble des nœuds du graphe, alors la proximité de second ordre entre v_i et v_j est déterminé par le calcul de la similarité entre p_{v_i} et p_{v_j} . De même, s'il n'existe pas de lien entre v_i et v_j alors la proximité de second ordre entre v_i et v_j est nulle. De manière similaire, cette proximité est minimisée via la fonction objectif δ_2 exprimée dans ce cas de la manière suivante :

$$\delta_2 = - \sum_{(v_i, v_j) \in E} w_{(i,j)} \log \left(\frac{e^{-(\hat{v}_i, \hat{v}_j)}}{\sum_{k=1}^{|V|} e^{(\hat{v}_i, \hat{v}_k)}} \right) \quad (\text{I.12})$$

De cette façon, les *embeddings* de nœuds sont construits séparément lors d'un processus d'apprentissage intégrant les fonctions objectifs associées aux proximités de premier et de second ordre puis sont concaténées. Ainsi, les deux variantes de proximité peuvent influencer la similarité des nœuds. L'algorithme LINE a été exploité avec succès en utilisant toute sorte d'entités représentant les nœuds.

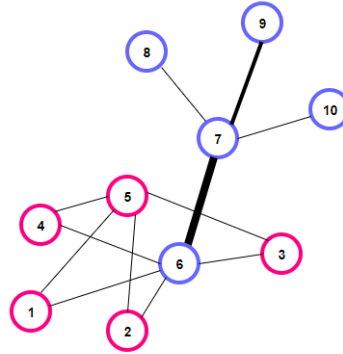


FIGURE I.3 – Illustration des voisinages de nœuds dans un graphe tel que défini dans [211]

Dans l'exemple de la figure I.3, les nœuds en bleu font parti du voisinage direct (du premier ordre) du nœud v_7 . L'épaisseur des arêtes indique le poids du lien reliant les nœuds. L'arête entre les nœuds v_6 et v_7 a un poids $w_{(6,7)}$. Les nœuds en rouge définissent le voisinage indirect (de second ordre) des nœuds v_5 et v_6 .

2.3 Apprentissage non supervisé : Clustering

Le *clustering* est une méthode d'apprentissage non-supervisée qui consiste à diviser les données en groupes d'entités similaires entre elles et non-similaires entre entités d'autres groupes. Contrairement aux approches supervisées, il n'y a pas de va-

riable cible à prédire, formellement, les clusters sont représentés sous la forme de sous-ensembles $C = C_1, \dots, C_k \in S$ de sorte que $S = \bigcup_{i=1}^k C_i$ avec $C_i \cap C_j = \emptyset$ pour $i \neq j$. Les trois principaux types de mesures utilisées pour rassembler des entités similaires au sein d'un cluster sont les mesures de distance comme la distance euclidienne [93, 141], les mesures de similarité telles que celles basées sur le cosinus [180, 89] ainsi que les mesures basées sur l'optimisation de critères qui se divisent en deux catégories, les critères intra-cluster comme l'homogénéité intra-groupe, la séparabilité inter-groupes ou une combinaison de ces deux éléments telle que la somme des carrés des erreurs [146] et les critères extra-cluster, utiles pour examiner si la structure des clusters correspond à une classification prédéfinie de l'instance.

1. Clustering hiérarchique : Consiste à générer des clusters qui ont un ordre prédéterminé de haut en bas. Par exemple, tous les fichiers et dossiers du disque dur sont organisés selon une hiérarchie. Nous comptons deux de types hiérarchique, Diviseur et Agglomératif.
 - a) La méthode de la division : Cette méthode assigne à toutes les observations un seul groupe. Ce groupe est ensuite divisé en deux groupes les moins semblables. Ce procédé est répété de façon récursive sur chaque groupe jusqu'à ce qu'il y ait un groupe pour chaque observation.
 - b) Méthode agglomérative : Dans la méthode agglomérative ou ascendante, nous attribuons chaque observation à son propre cluster. Ensuite, le calcul de la similarité (p. ex., la distance) entre chacun des clusters détermine les deux clusters les plus semblables.
2. K-Means clustering : Partitionne n objets en k clusters dans lesquels chaque objet appartient au cluster avec la moyenne la plus proche. Cette méthode produit exactement k clusters différents de la plus grande distinction possible. Le meilleur nombre de clusters k conduisant à la plus grande séparation (distance) n'est pas connu a priori et doit être calculé à partir des données. L'objectif du clustering K-Means est de minimiser la variance intra-cluster totale, ou la fonction d'erreur quadratique.
3. Clustering de graphe : Ces méthodes de clustering reposent sur l'analyse de caractéristiques représentées sous la forme de graphes. Les arêtes du graphe connectent les instances représentées sous la forme de nœuds. Cette méthode de clustering est particulièrement adaptée à la détection de communautés [69, 157].

2.4 Application : Structures communautaires dans les systèmes de tagging

Dans ce qui suit, nous passons en revue les méthodes existantes de détection de la structure communautaire et discutons de la façon dont ces approches peuvent être mis en œuvre. La méthode traditionnelle de détection de la structure communautaire dans des réseaux tels que celui illustré à la figure 1.4 se base sur un clustering hiérarchique.

On calcule d'abord un poids W_{ij} pour chaque paire i, j de sommets dans le réseau (graphe), ce qui représente dans un certain sens à quel point les sommets sont étroitement liés. (Nous donnons quelques exemples de ces poids possibles (voir ci-dessous). Puis on prend les n sommets en le réseau, sans arêtes les liant entre eux, et ajoutons des

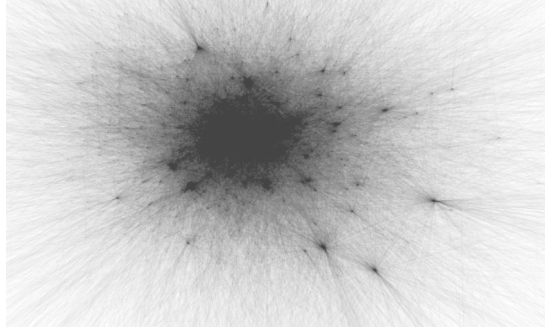


FIGURE I.4 – Graphe d'utilisateurs de tags *Movielens-10M* Number of nodes : 11610 Number of edges : 55484 Average degree : 9.5580

arêtes, entre les paires nœud par nœud dans l'ordre des poids reliées par des arêtes, en commençant par la paire avec le poids le plus fort et en continuant jusqu'au poids des arêtes les plus faibles. Au fur et à mesure que les arêtes sont ajoutés, le graphe qui en résulte montre un ensemble de clusters de plus en plus grands se former (sous-ensemble de sommets), qui seront considérés par la suite comme des communautés. Différentes pondérations ont été proposées pour l'utilisation d'algorithmes de classification hiérarchique. Une définition possible du poids est le nombre de chemins indépendants des nœuds entre les sommets. Deux chemins qui relient la même paire de sommets sont dits être indépendant s'ils ne partagent aucun des mêmes sommets autres que leurs sommets initiaux et finaux.

On peut également compter les chemins indépendants des arêtes. On sait [147] que le nombre de chemin indépendants des nœuds (indépendants des arêtes) entre deux sommets i et j dans un graphe est égal au nombre minimum de sommets (arêtes) qui doivent être supprimés du graphe afin de délier i et j l'un de l'autre. Ainsi, ces nombres sont en quelque sorte une mesure de la robustesse du réseau à la suppression de nœuds (arêtes) [228]. Les nombres de chemins indépendants peuvent être calculés rapidement en utilisant des algorithmes tel que ceux utilisés pour répondre au problèmes du "flow maximum" [4].

2.4.1 Clustering pour la détection de communautés

Dans [80], Newman *et al.*, suggèrent que les poids des arêtes entre nœuds se calcule en comptant le nombre total de chemins entre eux (tous, pas seulement ceux qui sont indépendant du point de vue des nœuds ou des arêtes), mais comme ce nombre est infini vu que les chemins forment des cycles, ils pondèrent les chemins de longueur l avec un facteur α^l avec α petit, de sorte que nombre le poids total des arêtes (chemins) converge. Ainsi, les longs chemins contribuent de façon exponentielle à maintenir un poids exponentiellement inférieur à celui des chemins courts.

Si A est la matrice d'adjacence du graphe, de sorte que A_{ij} est égal à 1 s'il y a un bord entre les sommets i et j et 0 sinon, alors les poids dans cette définition sont donnés par les éléments de la matrice :



FIGURE I.5 – Algorithme de Détection de communautés (*Newman*) via clustering sur le graphe d'utilisateurs d'après le nombre de leurs tags en communs lors d'activités d'annotation sur *Movielens10M* Number of nodes : 11610 Number of edges : 55484 Average degree : 9.5580 Nombre de communautés détectées : 80

$$W = \sum_{l=0}^{\infty} (\alpha.A)^l = [I - \alpha.A]^{-1} \quad (\text{I.13})$$

Dans leurs travaux relatifs à l'étude des structure dans les réseaux sociaux pondérés [80, 159, 158, 157], Au lieu d'essayer de construire une mesure qui révèlent les arêtes les plus centrales pour extraire des communautés, [80] se concentrent plutôt sur les arêtes qui sont les moins centrales. En effet, plutôt que de construire des communautés en reliant les arêtes les plus denses à un ensemble de sommets initialement vides, les auteurs les construisent en supprimant progressivement les arêtes du graphe original. La centralité des nœuds a été étudié comme étant une mesure de centralité et d'influence des nœuds dans un graphe. D'abord défini par Freeman [71], la centralité d'un nœud i est défini comme étant le nombre de plus court chemins entre 2 paires de nœuds. Il s'agit d'une mesure de l'influence d'un nœud sur le flux d'informations entre d'autres nœuds, en particulier dans le cas où de l'information est véhiculer dans un graphe elle suit principalement le chemin le plus court possible. Afin d'identifier les arêtes du graphe qui sont centraux entre paires de nœuds, ils généralisent le concept de centralité de [71] en l'étendant aux arêtes, et définissent la mesure de centralité d'une arête comme étant le nombre de plus petits chemins entre paires deux nœuds connectés. Enfin dans leur analyse des graphes pondérés servant à l'identification de structures communautaires [157], les auteurs introduisent une mesure : la modularité [158], celle ci est garante d'un bon partitionnement des communautés entre elle. En effet, soit c_i la communauté dans lequel le nœud i est assigné. La fraction des arêtes du graphe qui connecte les nœuds intra-cluster est donnée par :

$$\frac{\sum_{ij} A_{ij} \delta(c_i, c_j)}{\sum_{ij} A_{ij}} = \frac{1}{2m} \sum_{ij} A_{ij} \delta(c_i, c_j) \quad (\text{I.14})$$

où la fonction $\delta(u, v)$ est égal à 1 si $u = v$ est à 0 sinon (u et v sont deux nœuds d'un graphe) et $m = \frac{1}{2m} \sum_{ij} A_{ij}$ est le nombre d'arêtes dans le graphe. En gardant le degrés

des sommes et en connectant d'autres nœuds de manière aléatoire, alors la probabilité qu'une arête existe entre deux nœuds i et j est de $k_i k_j / 2m$, où k_i est le degré du nœud i . Dans ce cas la modularité Q , est définie comme suit :

$$Q = \frac{1}{2m} \sum_{ij} (A_{ij} - \frac{k_i k_j}{2m}) \delta(c_i, c_j) \quad (\text{I.15})$$

En pratique cette mesure est un excellent moyen de déterminer si une division particulière du graphe en communautés est bonne ou mauvaise. Les valeurs non nulles indiquent les écarts par rapport à une distribution aléatoire des arêtes, tandis que les valeurs autour de 0,3 ou plus indiquent généralement de bonnes divisions. La valeur maximale possible de Q est 1. La même idée est utilisée pour parfaire les divisions communautaires dans les graphes pondérés comme l'illustre la figure I.5. En effet, si A_{ij} représente le poids de l'arête reliant i et j , le degré k_i est défini selon l'équation : $k_i = \sum A_{ij}$ avec, $m = \frac{1}{2m} \sum_{ij} A_{ij}$ définit avant.

BIBLIOGRAPHIE

- [1] ABBASI, R., AND STAAB, S. Richvsm : enriched vector space models for folksonomies. In *Proceedings of the 20th ACM conference on Hypertext and hypermedia* (2009), pp. 219–228.
- [2] ABEL, F., GAO, Q., HOUBEN, G.-J., AND TAO, K. Analyzing temporal dynamics in twitter profiles for personalized recommendations in the social web. In *Proceedings of the 3rd International Web Science Conference* (2011), ACM, p. 2.
- [3] AGARWAL, D., AND CHEN, B.-C. Regression-based latent factor models. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* (2009), pp. 19–28.
- [4] AHUJA, R. K., ORLIN, J. B., SECHI, G. M., AND ZUDDAS, P. Algorithms for the simple equal flow problem. *Management Science* 45, 10 (1999), 1440–1455.
- [5] ARORA, S., LIANG, Y., AND MA, T. A simple but tough-to-beat baseline for sentence embeddings.
- [6] BAEZA-YATES, R., AND RIBEIRO-NETO, B. Modern information retrieval addison-wesley longman. *Reading MA* (1999).
- [7] BAEZA-YATES, R., AND RIBEIRO-NETO, B. Modern information retrieval : The concepts and technology behind search (acm press books), 2011.
- [8] BAEZA-YATES, R.-N. *Ricardo Baeza-Yates, Berthier Ribeiro-Neto : Modern Information Retrieval. Chapter 3*. Addison Wesley, 1999.
- [9] BAI, T., WEN, J.-R., ZHANG, J., AND ZHAO, W. X. A neural collaborative filtering model with interaction-based neighborhood. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (2017), pp. 1979–1982.
- [10] BALPE, J.-P., LELU, A., AND SALEH, I. Hypertextes et hypermédias : réalisations, outils et méthodes. *Hermès* (1995).
- [11] BANGALORE, S., AND RAMBOW, O. Using tags, a tree model, and a language model for generation. In *Proceedings of the Fifth International Workshop on Tree Adjoining Grammar and Related Frameworks (TAG+ 5)* (2000), pp. 33–40.
- [12] BAO, S., XUE, G., WU, X., YU, Y., FEI, B., AND SU, Z. Optimizing web search using social annotations. In *Proceedings of the 16th international conference on World Wide Web* (2007), pp. 501–510.
- [13] BAO, S., XUE, G., WU, X., YU, Y., FEI, B., AND SU, Z. Optimizing web search using social annotations. In *Proceedings of the 16th international conference on World Wide Web* (2007), ACM, pp. 501–510.

- [14] BEGELMAN, G., KELLER, P., SMADJA, F., ET AL. Automated tag clustering : Improving search and exploration in the tag space. In *Collaborative Web Tagging Workshop at WWW2006, Edinburgh, Scotland* (2006), pp. 15–33.
- [15] BELKIN, N. INGWERSEN, P. A. M. S. Proceedings of the 15th annual international acm sigir conference on research and development in information retrieval.
- [16] BELL, R., KOREN, Y., AND VOLINSKY, C. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining* (2007), pp. 95–104.
- [17] BENDER, M., CRECELIUS, T., KACIMI, M., MICHEL, S., NEUMANN, T., PARREIRA, J. X., SCHENKEL, R., AND WEIKUM, G. Exploiting social relations for query expansion and result ranking. In *2008 IEEE 24th International Conference on Data Engineering Workshop* (2008), IEEE, pp. 501–506.
- [18] BENGIO, Y., DUCHARME, R., VINCENT, P., AND JANVIN, C. A neural probabilistic language model. *The journal of machine learning research* 3 (2003), 1137–1155.
- [19] BENGIO, Y., LAMBLIN, P., POPOVICI, D., AND LAROCHELLE, H. Greedy layer-wise training of deep networks. In *Advances in neural information processing systems* (2007), pp. 153–160.
- [20] BENNETT, P. N., WHITE, R. W., CHU, W., DUMAIS, S. T., BAILEY, P., BORISYUK, F., AND CUI, X. Modeling the impact of short-and long-term behavior on search personalization. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval* (2012), ACM, pp. 185–194.
- [21] BENZ, D., HOTHO, A., STÜTZER, S., AND STUMME, G. Semantics made by you and me : Self-emerging ontologies can capture the diversity of shared knowledge.
- [22] BERG, R. V. D., KIPF, T. N., AND WELLING, M. Graph convolutional matrix completion. *arXiv preprint arXiv :1706.02263* (2017).
- [23] BERNER, J., GROHS, P., KUTYNIOK, G., AND PETERSEN, P. The modern mathematics of deep learning. *arXiv preprint arXiv :2105.04026* (2021).
- [24] BISHOP, C. M., ET AL. *Neural networks for pattern recognition*. Oxford university press, 1995.
- [25] BLEI, D. M., NG, A. Y., AND JORDAN, M. I. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.
- [26] BOTTOU, L. Stochastic gradient descent tricks. In *Neural networks : Tricks of the trade*. Springer, 2012, pp. 421–436.
- [27] BOUDIBA, T., AND DKAKI, T. Exploring contextualized tag-based embeddings for neural collaborative filtering. In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence - Volume 3 : ICAART*, (2022), INSTICC, SciTePress, pp. 158–166.

- [28] BOUDIBA, T.-R., AND AHMED-OUAMER, R. Approche temporelle pour la génération personnalisée de profils folksonomiques.
- [29] BOUDIBA, T.-R., AND AHMED-OUAMER, R. Approche temporelle pour la génération personnalisée de profils folksonomiques. *Document numérique* 21, 1 (2018), 91–114.
- [30] BOUDIBA, T.-R., AND DKAKI, T. Exploring contextualized tag-based embeddings for neural collaborative filtering.
- [31] BOUGHANEM, M., CHRISMENT, C., AND SOULÉ-DUPUY, C. Query modification based on relevance back-propagation in an ad hoc environment. *Information processing & management* 35, 2 (1999), 121–139.
- [32] BRAZDIL, P. B., AND SOARES, C. A comparison of ranking methods for classification algorithm selection. In *European conference on machine learning* (2000), Springer, pp. 63–75.
- [33] CAI, Y., AND LI, Q. Personalized search by tag-based user profile and resource profile in collaborative tagging systems. In *Proceedings of the 19th ACM international conference on Information and knowledge management* (2010), ACM, pp. 969–978.
- [34] CAI, Y., LI, Q., XIE, H., AND YU, L. Personalized resource search by tag-based user profile and resource profile. In *International Conference on Web Information Systems Engineering* (2010), Springer, pp. 510–523.
- [35] CAMBRIA, E., OLSHER, D., AND RAJAGOPAL, D. Senticnet 3 : a common and common-sense knowledge base for cognition-driven sentiment analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2014), vol. 28.
- [36] CANTADOR, I., BELLOGÍN, A., AND VALLET, D. Content-based recommendation in social tagging systems. In *Proceedings of the fourth ACM conference on Recommender systems* (2010), pp. 237–240.
- [37] CAO, S., LU, W., AND XU, Q. Grarep : Learning graph representations with global structural information. In *Proceedings of the 24th ACM international on conference on information and knowledge management* (2015), pp. 891–900.
- [38] CARMAN, M. J., BAILLIE, M., AND CRESTANI, F. Tag data and personalized information retrieval. In *Proceedings of the 2008 ACM workshop on Search in social media* (2008), ACM, pp. 27–34.
- [39] CARMAN, M. J., BAILLIE, M., GWADERA, R., AND CRESTANI, F. A statistical comparison of tag and query logs. In *Proceedings of the 32nd international ACM SIGIR Conference on Research and Development in information Retrieval* (2009), pp. 123–130.
- [40] CARMEL, D., ROITMAN, H., AND YOM-TOV, E. Who tags the tags ? : a framework for bookmark weighting. In *Proceedings of the 18th ACM conference on Information and knowledge management* (2009), ACM, pp. 1577–1580.
- [41] CARMEL, D., ROITMAN, H., AND YOM-TOV, E. Social bookmark weighting for search and recommendation. *The VLDB Journal* 19, 6 (2010), 761–775.

- [42] CARMEL, D., ZWERDLING, N., GUY, I., OFEK-KOIFMAN, S., HAR'EL, N., RONEN, I., UZIEL, E., YOGEV, S., AND CHERNOV, S. Personalized social search based on the user's social network. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (New York, NY, USA, 2009)*, CIKM '09, ACM, pp. 1227–1236.
- [43] CATTUTO, C., BALDASSARRI, A., SERVEDIO, V. D., AND LORETO, V. Vocabulary growth in collaborative tagging systems. *arXiv preprint arXiv :0704.3316* (2007).
- [44] CATTUTO, C., BENZ, D., HOTHO, A., AND STUMME, G. Semantic grounding of tag relatedness in social bookmarking systems. In *International semantic web conference* (2008), Springer, pp. 615–631.
- [45] CATTUTO, C., LORETO, V., AND PIETRONERO, L. Collaborative tagging and semiotic dynamics. *arXiv preprint cs/0605015* (2006).
- [46] CATTUTO, C., LORETO, V., AND PIETRONERO, L. Semiotic dynamics and collaborative tagging. *Proceedings of the National Academy of Sciences* 104, 5 (2007), 1461–1464.
- [47] CATTUTO, C., LORETO, V., AND SERVEDIO, V. A yule-simon process with memory. *EPL (Europhysics Letters)* 76, 2 (2006), 208.
- [48] CHEN, C. Generalised similarity analysis and pathfinder network scaling. *Interacting with computers* 10, 2 (1998), 107–128.
- [49] CHEN, S.-Y., AND ZHANG, Y. Improve web search ranking with social tagging. In *1st International Workshop on Mining Social Media* (2009).
- [50] CHEN, W., ZHENG, H.-T., AND MAO, X.-X. Extracting deep semantic information for intelligent recommendation. In *International Conference on Neural Information Processing* (2017), Springer, pp. 134–144.
- [51] CHEN, X., AND SHIN, H. Tag recommendation by machine learning with textual and social features. *Journal of Intelligent Information Systems* 40, 2 (2013), 261–282.
- [52] CHENG, Y., QIU, G., BU, J., LIU, K., HAN, Y., WANG, C., AND CHEN, C. Model bloggers' interests based on forgetting mechanism. In *Proceedings of the 17th international conference on World Wide Web* (2008), ACM, pp. 1129–1130.
- [53] CHO, Y. H., KIM, J. K., AND KIM, S. H. A personalized recommender system based on web usage mining and decision tree induction. *Expert systems with Applications* 23, 3 (2002), 329–342.
- [54] CLARK, K., LUONG, M.-T., LE, Q. V., AND MANNING, C. D. Electra : Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv :2003.10555* (2020).
- [55] CRÉPEL, M. Les folksonomies comme support émergent de navigation sociale et de structuration de l'information sur le web. *Réseaux*, 6 (2008), 169–204.
- [56] CROFT, W. B., METZLER, D., AND STROHMANN, T. *Search engines*. Pearson Education, 2010.

- [57] DAI, H., DAI, B., AND SONG, L. Discriminative embeddings of latent variable models for structured data. In *International conference on machine learning* (2016), pp. 2702–2711.
- [58] DAI, Z., AND CALLAN, J. Deeper text understanding for ir with contextual neural language modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (2019), pp. 985–988.
- [59] DE MEO, P., QUATTRONE, G., AND URSINO, D. A query expansion and user profile enrichment approach to improve the performance of recommender systems operating on a folksonomy. *User Modeling and User-Adapted Interaction* 20, 1 (2010), 41–86.
- [60] DELLSCHAFT, K., AND STAAB, S. An epistemic dynamic model for tagging systems. In *Proceedings of the nineteenth ACM conference on Hypertext and hypermedia* (2008), ACM, pp. 71–80.
- [61] DEVLIN, J., CHANG, M.-W., LEE, K., AND TOUTANOVA, K. Bert : Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv :1810.04805* (2018).
- [62] DHINGRA, B., LIU, H., SALAKHUTDINOV, R., AND COHEN, W. W. A comparative study of word embeddings for reading comprehension. *arXiv preprint arXiv :1703.00993* (2017).
- [63] DIEDERICH, J., AND IOFCIU, T. Finding communities of practice from user profiles based on folksonomies. In *Innovative approaches for learning and knowledge sharing, ec-tel workshop proc* (2006), pp. 288–297.
- [64] DOU, Z., SONG, R., AND WEN, J.-R. A large-scale evaluation and analysis of personalized search strategies. In *Proceedings of the 16th international conference on World Wide Web* (2007), pp. 581–590.
- [65] DU, Q., XIE, H., CAI, Y., LEUNG, H.-F., LI, Q., MIN, H., AND WANG, F. L. Folksonomy-based personalized search by hybrid user profiles in multiple levels. *Neurocomputing* (2016).
- [66] DUPRET, G., AND PIWOWARSKI, B. Sigir’08 : Proceedings of the 31st annual international acm sigir conference on research and development in information retrieval. *New York, NY, USA : ACM* (2008), 331–338.
- [67] DURAO, F., AND DOLOG, P. A personalized tag-based recommendation in social web systems. *Adaptation and Personalization for Web 2* (2009), 40.
- [68] DZIUGAITE, G. K., AND ROY, D. M. Neural network matrix factorization. *arXiv preprint arXiv :1511.06443* (2015).
- [69] FORTUNATO, S. Community detection in graphs. *Physics reports* 486, 3-5 (2010), 75–174.
- [70] FOX, E. A. *Extending the Boolean and vector space models of information retrieval with p-norm queries and multiple concept types*. PhD thesis, Cornell University, 1983.

- [71] FREEMAN, L. C. A set of measures of centrality based on betweenness. *Sociometry* (1977), 35–41.
- [72] FURNAS, G. W., FAKE, C., VON AHN, L., SCHACHTER, J., GOLDBER, S., FOX, K., DAVIS, M., MARLOW, C., AND NAAMAN, M. Why do tagging systems work? In *CHI'06 extended abstracts on Human factors in computing systems* (2006), ACM, pp. 36–39.
- [73] GAO, J., TOUTANOVA, K., AND YIH, W.-T. Clickthrough-based latent semantic models for web search. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval* (2011), pp. 675–684.
- [74] GAUSSIER, É., GREFFENSTETTE, G., HULL, D., AND ROUX, C. Recherche d'information en français et traitement automatique des langues. *TAL. Traitement automatique des langues* 41, 2 (2000), 473–493.
- [75] GAUSSIER, E., GREFFENSTETTE, G., AND SCHULZE, M. Traitement du langage naturel et recherche d'informations : quelques expériences sur le français. *Premières Journées Scientifiques et Techniques du Réseau Francophone de l'Ingénierie de la Langue de l'AUPELF-UREF* 15 (1997).
- [76] GEDIKLI, F., AND JANNACH, D. Improving recommendation accuracy based on item-specific tag preferences. *ACM Transactions on Intelligent Systems and Technology (TIST)* 4, 1 (2013), 1–19.
- [77] GEMMELL, J., SCHIMOLER, T., RAMEZANI, M., AND MOBASHER, B. Adapting k-nearest neighbor for tag recommendation in folksonomies. In *ITWP* (2009).
- [78] GEMMELL, J., SHEPITSEN, A., MOBASHER, B., AND BURKE, R. Personalizing navigation in folksonomies using hierarchical tag clustering. In *International Conference on Data Warehousing and Knowledge Discovery* (2008), Springer, pp. 196–205.
- [79] GIROLAMI, M., AND KABÁN, A. On an equivalence between plsi and lda. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval* (2003), pp. 433–434.
- [80] GIRVAN, M., AND NEWMAN, M. E. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences* 99, 12 (2002), 7821–7826.
- [81] GODOY, D., AND AMANDI, A. Exploiting the social capital of folksonomies for web page classification. In *Conference on e-Business, e-Services and e-Society* (2010), Springer, pp. 151–160.
- [82] GOEL, S., AND KUMAR, R. Folksonomy-based user profile enrichment using clustering and community recommended tags in multiple levels. *Neurocomputing* 315 (2018), 425–438.
- [83] GOLDBER, S. A., AND HUBERMAN, B. A. The structure of collaborative tagging system. *Journal of information science* 32, 2 (2006), 198–208.
- [84] GOODFELLOW, I., BENGIO, Y., COURVILLE, A., AND BENGIO, Y. *Deep learning*, vol. 1. MIT press Cambridge, 2016.

- [85] GROVER, A., AND LESKOVEC, J. node2vec : Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining* (2016), pp. 855–864.
- [86] GUDIVADA, V. N., RAGHAVAN, V. V., GROSKY, W. I., AND KASANAGOTTU, R. Information retrieval on the world wide web. *IEEE Internet Computing* 1, 5 (1997), 58–68.
- [87] GUO, L., WEN, Y.-F., AND WANG, X.-H. Exploiting pre-trained network embeddings for recommendations in social networks. *Journal of Computer Science and Technology* 33, 4 (2018), 682–696.
- [88] GUPTA, M., LI, R., YIN, Z., AND HAN, J. Survey on social tagging techniques. *ACM Sigkdd Explorations Newsletter* 12, 1 (2010), 58–72.
- [89] HAHN, S., SETHY, A., KUO, H.-K. J., AND RAMABHADRAN, B. A study of unsupervised clustering techniques for language modeling. In *Ninth Annual Conference of the International Speech Communication Association* (2008).
- [90] HALPIN, H., ROBU, V., AND SHEPHERD, H. The complex dynamics of collaborative tagging. In *Proceedings of the 16th international conference on World Wide Web* (2007), ACM, pp. 211–220.
- [91] HAMILTON, W., YING, Z., AND LESKOVEC, J. Inductive representation learning on large graphs. In *Advances in neural information processing systems* (2017), pp. 1024–1034.
- [92] HASSAN, H. A. M., SANSONETTI, G., GASPARETTI, F., AND MICARELLI, A. Semantic-based tag recommendation in scientific bookmarking systems. In *Proceedings of the 12th ACM Conference on Recommender Systems* (2018), pp. 465–469.
- [93] HATHAWAY, R. J., BEZDEK, J. C., AND HU, Y. Generalized fuzzy c-means clustering strategies using l_1 / l_p norm distances. *IEEE transactions on Fuzzy Systems* 8, 5 (2000), 576–582.
- [94] HE, X., DU, X., WANG, X., TIAN, F., TANG, J., AND CHUA, T.-S. Outer product-based neural collaborative filtering. *arXiv preprint arXiv :1808.03912* (2018).
- [95] HE, X., LIAO, L., ZHANG, H., NIE, L., HU, X., AND CHUA, T.-S. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web* (2017), pp. 173–182.
- [96] HE, X., ZHANG, H., KAN, M.-Y., AND CHUA, T.-S. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval* (2016), pp. 549–558.
- [97] HEYMANN, P., AND GARCIA-MOLINA, H. Collaborative creation of communal hierarchical taxonomies in social tagging systems. Tech. rep., Stanford, 2006.
- [98] HINTON, G. E. Connectionist learning procedures. In *Machine learning*. Elsevier, 1990, pp. 555–610.

- [99] HINTON, G. E., AND SALAKHUTDINOV, R. R. Reducing the dimensionality of data with neural networks. *science* 313, 5786 (2006), 504–507.
- [100] HLAOUA, L. *Reformulation de Requêtes par Réinjection de pertinence dans les Documents Semi-Structurés*. PhD thesis, Université de Toulouse, Université Toulouse III-Paul Sabatier, 2007.
- [101] HOFMANN, T. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* (1999), pp. 50–57.
- [102] HOTH0, A., JÄSCHKE, R., SCHMITZ, C., AND STUMME, G. Information retrieval in folksonomies : Search and ranking. In *European semantic web conference* (2006), Springer, pp. 411–426.
- [103] HOTH0, A., JÄSCHKE, R., SCHMITZ, C., AND STUMME, G. *Information retrieval in folksonomies : Search and ranking*. Springer, 2006.
- [104] HUANG, H. Tag distribution analysis using the power law to evaluate social tagging systems : A case study in the flickr database. In *17th ASIS&T SIG/CR Classification Research Workshop. Abstracts of Posters (S. 14-15)* (2006).
- [105] HUANG, R., WANG, N., HAN, C., YU, F., AND CUI, L. Tnam : A tag-aware neural attention model for top-n recommendation. *Neurocomputing* 385 (2020), 1–12.
- [106] IOFFE, S., AND SZEGEDY, C. Batch normalization : Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv :1502.03167* (2015).
- [107] JACQUEMIN, C., DAILLE, B., ROYAUTÉ, J., AND POLANCO, X. In vitro evaluation of a program for machine-aided indexing. *Information processing & management* 38, 6 (2002), 765–792.
- [108] JÄRVELIN, K., AND KEKÄLÄINEN, J. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.
- [109] KACEM, A., BOUGHANEM, M., AND FAIZ, R. Time-sensitive user profile for optimizing search personalization. In *International conference on user modeling, adaptation, and personalization* (2014), Springer, pp. 111–121.
- [110] KALERVO, J., AND JAANA, K. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.
- [111] KALMAN, D. A singularly valuable decomposition : the svd of a matrix. *The college mathematics journal* 27, 1 (1996), 2–23.
- [112] KAMEL BOULOS, M. N., AND WHEELER, S. The emerging web 2.0 social software : an enabling suite of sociable technologies in health and health care education. *Health Information & Libraries Journal* 24, 1 (2007), 2–23.
- [113] KANAWATI, R. Licod : Leaders identification for community detection in complex networks. In *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third International Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on* (2011), IEEE, pp. 577–582.

- [114] KATARIA, S., AND AGARWAL, A. Distributed representations for content-based and personalized tag recommendation. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)* (2015), IEEE, pp. 1388–1395.
- [115] KENTER, T., BORISOV, A., AND DE RIJKE, M. Siamese cbow : Optimizing word embeddings for sentence representations. *arXiv preprint arXiv :1606.04640* (2016).
- [116] KIM, H.-N., HA, I., JUNG, J.-G., AND JO, G.-S. User preference modeling from positive contents for personalized recommendation. In *Discovery Science* (2007), Springer, pp. 116–126.
- [117] KNUTH, D. E. *The art of computer programming : sorting and searching*, vol. 3. Pearson Education, 1997.
- [118] KOREN, Y. Factorization meets the neighborhood : a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* (2008), pp. 426–434.
- [119] KRESTEL, R., AND FANKHAUSER, P. Tag recommendation using probabilistic topic models. In *DC@ PKDD/ECML* (2009), Citeseer.
- [120] KRESTEL, R., FANKHAUSER, P., AND NEJDL, W. Latent dirichlet allocation for tag recommendation. In *Proceedings of the third ACM conference on Recommender systems* (2009), pp. 61–68.
- [121] KRITIKOPOULOS, A., AND SIDERI, M. The compass filter : Search engine result personalization using web communities. In *IJCAI Workshop on Intelligent Techniques for Web Personalization* (2003), Springer, pp. 229–240.
- [122] KWOK, K. A neural network for probabilistic information retrieval. In *Proceedings of the 12th annual international ACM SIGIR conference on Research and development in information retrieval* (1989), pp. 21–30.
- [123] LE, Q., AND MIKOLOV, T. Distributed representations of sentences and documents. In *International conference on machine learning* (2014), pp. 1188–1196.
- [124] LECUN, Y., BENGIO, Y., ET AL. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks* 3361, 10 (1995), 1995.
- [125] LECUN, Y., BENGIO, Y., AND HINTON, G. Deep learning. *nature* 521, 7553 (2015), 436–444.
- [126] LEE, C.-S., KAO, Y.-F., KUO, Y.-H., AND WANG, M.-H. Automated ontology construction for unstructured text documents. *Data & Knowledge Engineering* 60, 3 (2007), 547–566.
- [127] LI, C., XU, L., YAN, M., AND LEI, Y. Tagdc : A tag recommendation method for software information sites with a combination of deep learning and collaborative filtering. *Journal of Systems and Software* 170 (2020), 110783.
- [128] LI, P., WANG, Z., REN, Z., BING, L., AND LAM, W. Neural rating regression with abstractive tips generation for recommendation. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval* (2017), pp. 345–354.

- [129] LI, Q., KREYSSIG, F. L., ZHANG, C., AND WOODLAND, P. C. Discriminative neural clustering for speaker diarisation. In *2021 IEEE Spoken Language Technology Workshop (SLT)* (2021), IEEE, pp. 574–581.
- [130] LI, X., AND CHEN, H. Recommendation as link prediction in bipartite graphs : A graph kernel-based machine learning approach. *Decision Support Systems* 54, 2 (2013), 880–890.
- [131] LIANG, D., ZHAN, M., AND ELLIS, D. P. Content-aware collaborative music recommendation using pre-trained neural networks. In *ISMIR* (2015), pp. 295–301.
- [132] LIANG, N., ZHENG, H.-T., CHEN, J.-Y., SANGAIAH, A. K., AND ZHAO, C.-Z. Trsd1 : Tag-aware recommender system based on deep learning–intelligent computing systems. *Applied Sciences* 8, 5 (2018), 799.
- [133] LIANG, S., AND DE RIJKE, M. Formal language models for finding groups of experts. *Information Processing & Management* 52, 4 (2016), 529–549.
- [134] LIN, Y., LIN, H., JIN, S., AND YE, Z. Social annotation in query expansion : a machine learning approach. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval* (2011), ACM, pp. 405–414.
- [135] LIU, Y., OTT, M., GOYAL, N., DU, J., JOSHI, M., CHEN, D., LEVY, O., LEWIS, M., ZETTEMAYER, L., AND STOYANOV, V. Roberta : A robustly optimized bert pretraining approach. *arXiv preprint arXiv :1907.11692* (2019).
- [136] LU, Y.-T., YU, S.-I., CHANG, T.-C., AND HSU, J. Y.-J. A content-based method to enhance tag recommendation. In *Twenty-first international joint conference on artificial intelligence* (2009).
- [137] LUHN, H. P. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of research and development* 1, 4 (1957), 309–317.
- [138] LUKASIEWICZ, T., MIAO, Y., AND XU, Z. Tag-aware personalized recommendation using a hybrid deep model.
- [139] LUO, L., XIE, H., RAO, Y., AND WANG, F. L. Personalized recommendation by matrix co-factorization with tags and time information. *expert systems with applications* 119 (2019), 311–321.
- [140] MACAVANEY, S., YATES, A., COHAN, A., AND GOHARIAN, N. Cedr : Contextualized embeddings for document ranking. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (2019), pp. 1101–1104.
- [141] MACQUEEN, J., ET AL. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (1967), vol. 1, Oakland, CA, USA, pp. 281–297.
- [142] MANIEZ, J. A decade of research in classification. *International Classification* 18, 2 (1991), 73–77.

- [143] MARLOW, C., NAAMAN, M., BOYD, D., AND DAVIS, M. Tagging papers, taxonomy, frlickr, academic article, toread. In *Proceedings of Hypertext* (2006).
- [144] MARON, M., AND KUHN, J. On relevance, probabilistic indexing and information retrieval. In *Readings in information retrieval* (1997), Morgan Kaufmann Publishers Inc., pp. 39–46.
- [145] MAZHARI, S., FAKHRAHMAD, S. M., AND SADEGHBEYGI, H. A user-profile-based friendship recommendation solution in social networks. *Journal of Information Science* 41, 3 (2015), 284–295.
- [146] MCLACHLAN, G., AND PEEL, D. Finite mixture models. (john wiley & sons : New york.).
- [147] MENGER, K. Zur allgemeinen kurventheorie. *Fundamenta Mathematicae* 10, 1 (1927), 96–115.
- [148] MICHELMAYR, E., AND CAYZER, S. Learning user profiles from tagging data and leveraging them for personal (ized) information access.
- [149] MIKOLOV, T., CHEN, K., CORRADO, G., AND DEAN, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv :1301.3781* (2013).
- [150] MIKOLOV, T., KARAFIÁT, M., BURGET, L., CERNOCKÝ, J., AND KHUDANPUR, S. Recurrent neural network based language model. In *Interspeech* (2010), vol. 2, Makuhari, pp. 1045–1048.
- [151] MIKOLOV, T., KOMBRINK, S., BURGET, L., ČERNOCKÝ, J., AND KHUDANPUR, S. Extensions of recurrent neural network language model. In *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (2011), IEEE, pp. 5528–5531.
- [152] MIKOLOV, T., SUTSKEVER, I., CHEN, K., CORRADO, G., AND DEAN, J. Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv :1310.4546* (2013).
- [153] MITRA, B., AND CRASWELL, N. Neural models for information retrieval. *arXiv preprint arXiv :1705.01509* (2017).
- [154] MOBASHER, B., AND ANAND, S. S. *Intelligent Techniques for Web Personalization : IJCAI 2003 Workshop, ITWP 2003, Acapulco, Mexico, August 11, 2003, Revised Selected Papers*, vol. 3169. Springer, 2005.
- [155] MUSTO, C., SEMERARO, G., DE GEMMIS, M., AND LOPS, P. Word embedding techniques for content-based recommender systems : An empirical evaluation. In *Recsys posters* (2015).
- [156] NAKAMOTO, R., NAKAJIMA, S., MIYAZAKI, J., UEMURA, S., AND KATO, H. Investigation of the effectiveness of tag-based contextual collaborative filtering in website recommendation. In *Advances in Communication Systems and Electrical Engineering*. Springer, 2008, pp. 309–318.

- [157] NEWMAN, M. E. Analysis of weighted networks. *Physical Review E* 70, 5 (2004), 056131.
- [158] NEWMAN, M. E. Fast algorithm for detecting community structure in networks. *Physical review E* 69, 6 (2004), 066133.
- [159] NEWMAN, M. E. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences* 103, 23 (2006), 8577–8582.
- [160] NICKOLLS, J., BUCK, I., GARLAND, M., AND SKADRON, K. Scalable parallel programming with cuda : Is cuda the parallel programming model that application developers have been waiting for? *Queue* 6, 2 (2008), 40–53.
- [161] NIEBLER, T., HAHN, L., AND HOTHO, A. Learning word embeddings from tagging data : A methodological comparison. In *LWDA (2017)*, p. 229.
- [162] NOLL, M. G., AND MEINEL, C. Web search personalization via social bookmarking and tagging. In *The Semantic Web*. Springer, 2007, pp. 367–380.
- [163] NOLL, M. G., AND MEINEL, C. The metadata triumvirate : Social annotations, anchor texts and search queries. In *Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT'08. IEEE/WIC/ACM International Conference on (2008)*, vol. 1, IEEE, pp. 640–647.
- [164] NUGALIYADDE, A., WONG, K. W., SOHEL, F., AND XIE, H. Enhancing semantic word representations by embedding deep word relationships. In *Proceedings of the 2019 11th International Conference on Computer and Automation Engineering (2019)*, pp. 82–87.
- [165] PAGE, L., BRIN, S., MOTWANI, R., AND WINOGRAD, T. The pagerank citation ranking : Bringing order to the web. Tech. rep., Stanford InfoLab, 1999.
- [166] PAPADOPOULOS, S., KOMPATSIARIS, Y., AND VAKALI, A. A graph-based clustering scheme for identifying related tags in folksonomies. In *International conference on data warehousing and knowledge discovery (2010)*, Springer, pp. 65–76.
- [167] PENNINGTON, J., SOCHER, R., AND MANNING, C. D. Glove : Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (2014)*, pp. 1532–1543.
- [168] PETERS, M. E., NEUMANN, M., IYER, M., GARDNER, M., CLARK, C., LEE, K., AND ZETTLEMOYER, L. Deep contextualized word representations. *arXiv preprint arXiv :1802.05365* (2018).
- [169] PETERS, M. E., NEUMANN, M., ZETTLEMOYER, L., AND YIH, W.-T. Dissecting contextual word embeddings : Architecture and representation. *arXiv preprint arXiv :1808.08949* (2018).
- [170] PINO, J. A. Modern information retrieval. ricardo baeza-yates y berthier ribeiro-neto addison wesley hariow, england, 1999.
- [171] PIRKOLA, A. Morphological typology of languages for ir. *Journal of Documentation* 57, 3 (2001), 330–348.

- [172] PONTE, J. M., AND CROFT, W. B. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval* (1998), pp. 275–281.
- [173] PORTER, M. F. Readings in information retrieval.
- [174] POUSSEVIN, M., GUARDIA-SEBAOUN, E., GUIGUE, V., AND GALLINARI, P. Recommendation par combinaison de filtrage collaboratif et d’analyse de sentiments. In *CORIA 2014-Conférence en Recherche d’Information et Applications* (2014), pp. 27–42.
- [175] QIU, F., AND CHO, J. Automatic identification of user interest for personalized search. In *Proceedings of the 15th international conference on World Wide Web* (2006), ACM, pp. 727–736.
- [176] REIMERS, N., AND GUREVYCH, I. Sentence-bert : Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv :1908.10084* (2019).
- [177] ROBERTSON, S., AND ZARAGOZA, H. *The probabilistic relevance framework : BM25 and beyond*. Now Publishers Inc, 2009.
- [178] ROBERTSON, S. E., KANOULAS, E., AND YILMAZ, E. Extending average precision to graded relevance judgments. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval* (2010), pp. 603–610.
- [179] ROBERTSON, S. E., AND WALKER, S. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval* (1994), Springer-Verlag New York, Inc., pp. 232–241.
- [180] ROKACH, L., AND MAIMON, O. Clustering methods. In *Data mining and knowledge discovery handbook*. Springer, 2005, pp. 321–352.
- [181] RÜCKLÉ, A., EGER, S., PEYRARD, M., AND GUREVYCH, I. Concatenated power mean word embeddings as universal cross-lingual sentence representations. *arXiv preprint arXiv :1803.01400* (2018).
- [182] RUMELHART, D. E., HINTON, G. E., AND WILLIAMS, R. J. Learning internal representations by error propagation. Tech. rep., California Univ San Diego La Jolla Inst for Cognitive Science, 1985.
- [183] SALTON, G. Automatic information organization and retrieval.
- [184] SALTON, G. A comparison between manual and automatic indexing methods. *American Documentation* 20, 1 (1969), 61–71.
- [185] SALTON, G., FOX, E. A., AND WU, H. Extended boolean information retrieval. *Communications of the ACM* 26, 11 (1983), 1022–1036.
- [186] SALTON, G., AND MICHAEL, J. McGill. *Introduction to modern information retrieval* (1983), 24–51.
- [187] SALTON, G., WONG, A., AND YANG, C.-S. A vector space model for automatic indexing. *Communications of the ACM* 18, 11 (1975), 613–620.

- [188] SARWAR, B., KARYPIS, G., KONSTAN, J., AND RIEDL, J. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web* (2001), pp. 285–295.
- [189] SARWAR, B. M., KARYPIS, G., KONSTAN, J., AND RIEDL, J. Recommender systems for large-scale e-commerce : Scalable neighborhood formation using clustering. In *Proceedings of the fifth international conference on computer and information technology* (2002), vol. 1, Citeseer, pp. 291–324.
- [190] SAVOY, J. Indexation manuelle et automatique : une évaluation comparative basée sur un corpus en langue française. In *Actes de la 2ème Conférence en Recherche d'Information et Applications CORIA05* (2005), Institut d'Informatique et Mathématique Appliquée de Grenoble, Laboratoire CLIPS (Communication Langagière et Interaction Personne-Système), pp. 9–23.
- [191] SCHAFER, J. B., FRANKOWSKI, D., HERLOCKER, J., AND SEN, S. Collaborative filtering recommender systems. In *The adaptive web*. Springer, 2007, pp. 291–324.
- [192] SCHENKEL, R., CRECELIUS, T., KACIMI, M., MICHEL, S., NEUMANN, T., PARREIRA, J. X., AND WEIKUM, G. Efficient top-k querying over social-tagging networks. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval* (2008), pp. 523–530.
- [193] SCHIFANELLA, R., BARRAT, A., CATTUTO, C., MARKINES, B., AND MENCZER, F. Folks in folksonomies : social link prediction from shared metadata. In *Proceedings of the third ACM international conference on Web search and data mining* (2010), pp. 271–280.
- [194] SCHMITZ, C., HOTHO, A., JÄSCHKE, R., AND STUMME, G. Mining association rules in folksonomies. In *Data Science and Classification*. Springer, 2006, pp. 261–270.
- [195] SCHMITZ, P. Inducing ontology from flickr tags. In *Collaborative Web Tagging Workshop at WWW2006, Edinburgh, Scotland* (2006), vol. 50, p. 39.
- [196] SEDHAIN, S., MENON, A. K., SANNER, S., AND XIE, L. Autorec : Autoencoders meet collaborative filtering. In *Proceedings of the 24th international conference on World Wide Web* (2015), pp. 111–112.
- [197] SEN, S., LAM, S. K., RASHID, A. M., COSLEY, D., FRANKOWSKI, D., OSTERHOUSE, J., HARPER, F. M., AND RIEDL, J. Tagging, communities, vocabulary, evolution. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work* (2006), ACM, pp. 181–190.
- [198] SEN, S., VIG, J., AND RIEDL, J. Tagommenders : connecting users to items through tags. In *Proceedings of the 18th international conference on World wide web* (2009), pp. 671–680.
- [199] SHEPITSEN, A., GEMMELL, J., MOBASHER, B., AND BUKE, R. Personalization in folksonomies based on tag clustering. AAAI.
- [200] SHEPITSEN, A., GEMMELL, J., MOBASHER, B., AND BURKE, R. Personalized recommendation in social tagging systems using hierarchical clustering. In *Proceedings of the 2008 ACM conference on Recommender systems* (2008), ACM, pp. 259–266.

- [201] SHEUGH, L., AND ALIZADEH, S. H. A note on pearson correlation coefficient as a metric of similarity in recommender system. In *2015 AI & Robotics (IRANOPEN) (2015)*, IEEE, pp. 1–6.
- [202] SIGMAN, M., AND CECCHI, G. A. Global organization of the wordnet lexicon. *Proceedings of the National Academy of Sciences* 99, 3 (2002), 1742–1747.
- [203] SPARCK JONES, K. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation* 28, 1 (1972), 11–21.
- [204] SPITERI, L. F. The structure and form of folksonomy tags : The road to the public library catalog. *Information technology and libraries* 26, 3 (2007), 13–25.
- [205] SRIRAM, A., JUN, H., SATHEESH, S., AND COATES, A. Cold fusion : Training seq2seq models together with language models. *arXiv preprint arXiv :1708.06426* (2017).
- [206] SU, J.-H., CHANG, W.-Y., AND TSENG, V. S. Personalized music recommendation by mining social media tags. *Procedia Computer Science* 22 (2013), 303–312.
- [207] SUGIYAMA, K., HATANO, K., AND YOSHIKAWA, M. Adaptive web search based on user profile constructed without any effort from users. In *Proceedings of the 13th international conference on World Wide Web* (2004), pp. 675–684.
- [208] SUN, K., WANG, X., SUN, C., AND LIN, L. A language model approach for tag recommendation. *Expert Systems with Applications* 38, 3 (2011), 1575–1582.
- [209] SZOMSZOR, M., CATTUTO, C., ALANI, H., O’HARA, K., BALDASSARRI, A., LORETO, V., AND SERVEDIO, V. D. Folksonomies, the semantic web, and movie recommendation.
- [210] TAHAR-RAFIK, B., AND RACHID, A.-O. Towards a new approach for generating user profile from folksonomies. In *ISKO-Maghreb : Concepts and Tools for knowledge Management (ISKO-Maghreb), 2014 4th International Symposium* (2014), IEEE, pp. 1–6.
- [211] TANG, J., QU, M., WANG, M., ZHANG, M., YAN, J., AND MEI, Q. Line : Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web* (2015), pp. 1067–1077.
- [212] TANNIER, X. Extraction et recherche d’information en langage naturel dans les documents semi-structurés. *These de doctorat, Université Jean Monnet, Saint-Etienne, France* (2006).
- [213] TAY, Y., ANH TUAN, L., AND HUI, S. C. Latent relational metric learning via memory-based attention for collaborative ranking. In *Proceedings of the 2018 World Wide Web Conference* (2018), pp. 729–739.
- [214] TEEVAN, J., DUMAIS, S. T., AND HORVITZ, E. Personalizing search via automated analysis of interests and activities. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval* (2005), pp. 449–456.

- [215] TEEVAN, J., MORRIS, M. R., AND BUSH, S. Discovering and using groups to improve personalized search. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining* (2009), pp. 15–24.
- [216] THEOBALD, M., SCHENKEL, R., AND WEIKUM, G. Efficient and self-tuning incremental query expansion for top-k query processing. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval* (2005), ACM, pp. 242–249.
- [217] TSCHANNEN, M., BACHEM, O., AND LUCIC, M. Recent advances in autoencoder-based representation learning. *arXiv preprint arXiv :1812.05069* (2018).
- [218] TSO-SUTTER, K. H., MARINHO, L. B., AND SCHMIDT-THIEME, L. Tag-aware recommender systems by fusion of collaborative filtering algorithms. In *Proceedings of the 2008 ACM symposium on Applied computing* (2008), pp. 1995–1999.
- [219] VALLET, D., CANTADOR, I., AND JOSE, J. M. Personalizing web search with folksonomy-based user and document profiles. In *Advances in Information Retrieval*. Springer, 2010, pp. 420–431.
- [220] VAN DAMME, C., HEPP, M., AND SIORPAES, K. Folksontology : An integrated approach for turning folksonomies into ontologies. *Bridging the Gap between Semantic Web and Web 2, 2* (2007), 57–70.
- [221] VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, Ł., AND POLOSUKHIN, I. Attention is all you need. In *Advances in neural information processing systems* (2017), pp. 5998–6008.
- [222] VIJAIKUMAR, M., SHEVADE, S., AND MURTY, M. N. Tagembedsvd : Leveraging tag embeddings for cross-domain collaborative filtering. In *International Conference on Pattern Recognition and Machine Intelligence* (2019), Springer, pp. 240–248.
- [223] VOORHEES, E. M., ET AL. The trec-8 question answering track report. In *Trec* (1999), vol. 99, Citeseer, pp. 77–82.
- [224] VULIĆ, I., AND MOENS, M.-F. Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval* (2015), pp. 363–372.
- [225] WANG, J., LU, Z., SONG, G., FAN, Y., DU, L., AND LIN, W. Tag2vec : Learning tag representations in tag networks. In *The World Wide Web Conference* (2019), pp. 3314–3320.
- [226] WEI, S., ZHENG, X., CHEN, D., AND CHEN, C. A hybrid approach for movie recommendation via tags and ratings. *Electronic Commerce Research and Applications 18* (2016), 83–94.
- [227] WEI, X., AND CROFT, W. B. Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* (2006), pp. 178–185.

- [228] WHITE, D. R., AND HARARY, F. The cohesiveness of blocks in social networks : Node connectivity and conditional density. *Sociological Methodology* 31, 1 (2001), 305–359.
- [229] WHITE, R. W., BAILEY, P., AND CHEN, L. Predicting user interests from contextual information. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval* (2009), ACM, pp. 363–370.
- [230] WU, L., QUAN, C., LI, C., WANG, Q., ZHENG, B., AND LUO, X. A context-aware user-item representation learning for item recommendation. *ACM Transactions on Information Systems (TOIS)* 37, 2 (2019), 1–29.
- [231] XIE, H., LI, X., WANG, T., LAU, R. Y., WONG, T.-L., CHEN, L., WANG, F. L., AND LI, Q. Incorporating sentiment into tag-based user profiles and resource profiles for personalized search in folksonomy. *Information Processing & Management* 52, 1 (2016), 61–72.
- [232] XIE, M., YIN, H., WANG, H., XU, F., CHEN, W., AND WANG, S. Learning graph-based poi embedding for location-based recommendation. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (2016), pp. 15–24.
- [233] XU, S., BAO, S., FEI, B., SU, Z., AND YU, Y. Exploring folksonomy for personalized search. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval* (2008), ACM, pp. 155–162.
- [234] XU, S., BAO, S., FEI, B., SU, Z., AND YU, Y. Exploring folksonomy for personalized search. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval* (2008), pp. 155–162.
- [235] XU, Z., CHEN, C., LUKASIEWICZ, T., MIAO, Y., AND MENG, X. Tag-aware personalized recommendation using a deep-semantic similarity model with negative sampling. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (2016), pp. 1921–1924.
- [236] XU, Z., FU, Y., MAO, J., AND SU, D. Towards the semantic web : Collaborative tag suggestions. In *Collaborative web tagging workshop at WWW2006, Edinburgh, Scotland* (2006).
- [237] YEUNG, A., MAN, C., GIBBINS, N., AND SHADBOLT, N. A study of user profile generation from folksonomies.
- [238] YIN, D., XUE, Z., HONG, L., AND DAVISON, B. D. A probabilistic model for personalized tag prediction. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining* (2010), pp. 959–968.
- [239] ZAMANI, H., AND CROFT, W. B. Estimating embedding vectors for queries. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval* (2016), pp. 123–132.

- [240] ZANARDI, V., AND CAPRA, L. Social ranking : uncovering relevant content using tag-based recommender systems. In *Proceedings of the 2008 ACM conference on Recommender systems* (2008), ACM, pp. 51–58.
- [241] ZHANG, H., KORAYEM, M., YOU, E., AND CRANDALL, D. J. Beyond co-occurrence : discovering and visualizing tag relationships from geo-spatial and temporal similarities. In *Proceedings of the fifth ACM international conference on Web search and data mining* (2012), ACM, pp. 33–42.
- [242] ZHANG, H., SHEN, F., LIU, W., HE, X., LUAN, H., AND CHUA, T.-S. Discrete collaborative filtering. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval* (2016), pp. 325–334.
- [243] ZHANG, N., ZHANG, Y., AND TANG, J. A tag recommendation system for folksonomy. In *Proceedings of the 2nd ACM workshop on Social web search and mining* (2009), pp. 9–16.
- [244] ZHANG, S., YAO, L., SUN, A., AND TAY, Y. Deep learning based recommender system : A survey and new perspectives. *ACM Computing Surveys (CSUR)* 52, 1 (2019), 1–38.
- [245] ZHANG, S., YAO, L., AND XU, X. Autosvd++ an efficient hybrid collaborative filtering model via contractive auto-encoders. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval* (2017), pp. 957–960.
- [246] ZHANG, Z.-K., LIU, C., ZHANG, Y.-C., AND ZHOU, T. Solving the cold-start problem in recommender systems with social tags. *EPL (Europhysics Letters)* 92, 2 (2010), 28002.
- [247] ZHENG, G., AND CALLAN, J. Learning to reweight terms with distributed representations. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval* (2015), pp. 575–584.
- [248] ZHENG, N., AND LI, Q. A recommender system based on tag and time information for social tagging systems. *Expert Systems with Applications* 38, 4 (2011), 4575–4587.
- [249] ZIPE, G. K. human behaviour and the principle of least effort addison. wesley pub, 1949.
- [250] ZOBEL, J., MOFFAT, A., AND RAMAMOHANARAO, K. Inverted files versus signature files for text indexing. *ACM Transactions on Database Systems (TODS)* 23, 4 (1998), 453–490.
- [251] ZUBIAGA, A., FRESNO, V., MARTINEZ, R., AND GARCIA-PLAZA, A. P. Harnessing folksonomies to produce a social classification of resources. *IEEE transactions on knowledge and data engineering* 25, 8 (2012), 1801–1813.
- [252] ZUBIAGA, A., MARTÍNEZ, R., AND FRESNO, V. Analyzing tag distributions in folksonomies for resource classification. In *International Conference on Knowledge Science, Engineering and Management* (2011), Springer, pp. 91–102.

- [253] ZUCCON, G., KOOPMAN, B., BRUZA, P., AND AZZOPARDI, L. Integrating and evaluating neural word embeddings in information retrieval. In *Proceedings of the 20th Australasian document computing symposium* (2015), pp. 1–8.
- [254] ZUO, Y., ZENG, J., GONG, M., AND JIAO, L. Tag-aware recommender systems based on deep neural networks. *Neurocomputing* 204 (2016), 51–60.