Apprentissage de Représentation appliqué à la Recommandation pour la Littérature Scientifique

Robin Brochier [1]

Université de Lyon, Lyon 2, ERIC EA3083
Digital Scientific Research Technology
robin.brochier@univ-lyon2.fr

RÉSUMÉ. *La littérature scientifique forme un large réseau d'information reliant des acteurs variés (laboratoires, entreprises, institutions, etc.). La vaste quantité de données générées par ce réseau constitue un graphe hétérogène attribué dynamique, dans lequel de nouvelles informations sont constamment produites et dont il est de plus en plus difficile d'extraire du contenu d'intérêt. Dans cet article, je présente mes premiers travaux de thèse réalisés en partenariat avec un acteur industriel. Celui-ci propose un outil de veille scientifique abordant différentes problématiques, telles que la recommandation d'articles et la recherche d'experts. Tout d'abord, je détaille les données habituellement associées à la littérature scientifique. Ensuite, j'aborde le problème de la recherche d'experts, son évaluation et propose une nouvelle méthode pour juger de la qualité d'un algorithme. Enfin, j'introduis un algorithme d'apprentissage de représentations abordant le problème du plongement des nœuds d'un graphe dans un espace de faible dimension, étendu pour intégrer l'information textuelle liée à ces nœuds.*

ABSTRACT. *The scientific literature is a large information network linking various actors (laboratories, companies, institutions, etc.). The vast amount of data generated by this network constitutes a dynamic attributed heterogeneous network, in which new information is constantly produced and from which it is increasingly difficult to extract content of interest. In this article, I present my first thesis works in partnership with an industrial company. This later offers a scientific watch tool addressing various issues, such as the recommendation of articles and the search for experts. First, I detail the data usually associated with the scientific literature. Then I present the problem of expert finding, its evaluation and I suggest a new method to juge the quality of an algorithm. Finally, I introduce a representation learning algorithm for the problem of embedding the nodes of a graph in a vector space of small dimension, extended to integrate the textual information related to these nodes.*

MOTS-CLÉS : *littérature scientifique, système de recommandation, recherche d'information*

KEYWORDS: *scientific literature, recommender system, information retrieval*

DOI:10.3166/RIA.28.1-14

1. Introduction

De nombreuses applications, devenues outils du quotidien, proposent de chercher et filtrer les vastes sources de données disponibles sur le Web. En particulier, on compte une multitude de plateformes traitant de la littérature scientifique. Du simple moteur de recherche d'articles scientifiques au réseau social pour chercheurs, toutes utilisent comme données les publications quotidiennement produites à travers le monde. Pour le chercheur, faisant face à ce déluge d'information, il est devenu laborieux, voire impossible, de réaliser une veille régulière et exhaustive de ses domaines d'expertise.

Mes travaux de thèse, réalisés en partenariat avec un acteur industriel, traitent le problème de l'apprentissage de représentations dans des graphes de documents appliqué à la recommandation d'articles et d'experts scientifiques en temps réel. Je présente dans cet article plusieurs problématiques et mes propositions. Premièrement, s'il est possible de collecter massivement des données publiques liées à la publication scientifique, certaines problématiques apparaissent quant à leurs traitements. Je détaille brièvement dans la section 2 l'acquisition de données de la littérature scientifique. Ensuite, dans la section 3, j'explore particulièrement la tâche de recherche d'experts. Je décris les évaluations usuellement pratiquées dans la littérature et leurs limitations, puis introduis une alternative. Finalement, pour faciliter la compréhension et l'exploitation des données de la littérature scientifique, il est nécessaire de résoudre des tâches variées telles que la recommandation de lecture, la recherche d'experts et la prédiction de liens. Une approche usuelle est d'apprendre de manière non supervisée des représentations de ces données, indépendamment de leurs utilisations finales, mais dans le but de faciliter grandement la résolution de chacune de ces tâches. Je présente dans la section 4 mes travaux sur le plongement de nœuds issus d'un graphe de documents. Je conclus en présentant quelques perspectives de recherche.

2. Acquisition des données de la littérature scientifique

La littérature scientifique désigne l'ensemble des publications scientifiques diffusant l'information produite par les chercheurs. Une importante partie de cette littérature prend la forme d'articles scientifiques ayant subit un processus d'évaluation par les pairs et étant publiés dans des revues. À l'ère du tout digital, une importante partie de ces publications est accessible publiquement, soit directement sur les sites internet des maisons d'éditions, soit au travers d'agrégateurs de contenus. Cependant, nombre de publications sont payantes et il n'est possible d'en extraire à moindre coût que des informations partielles, appelées méta-données, telles que leurs titres, leurs résumés ou les noms de leurs auteurs.

Si l'information scientifique d'une publication est majoritairement contenue dans le texte qui la compose, en l'occurrence son titre, son résumé et/ou son contenu intégrale, des renseignements complémentaires riches se nichent dans ses méta-données. Ainsi, les réseaux de co-auteurs et de citations ou les lieux de publication d'un article renferment une information importante pour la réalisation et l'évaluation d'un sys-

tème de recherche d'information scientifique. De plus, plusieurs initiatives ont permis de mettre à disposition des chercheurs des jeux de données spécifiques à l'évaluation de certaines tâches, comme la recherche d'experts.

Dans cette section, je présente tout d'abord les traitements usuels lié à la construction d'une base de données de littérature scientifique. Ensuite je détaille certains jeux de données publiques, largement utilisés dans la littérature scientifique.

2.1. Le traitement des données

Une publication scientifique est constituée d'un contenu textuel et de méta-données. Il convient de s'assurer de la qualité d'extraction de ces données et de les traiter pour obtenir des informations exploitables. On s'intéresse dans cet article aux attributs suivants:

– le contenu, qui peut être l'ensemble du texte de l'article, ou simplement la concaténation de son titre et de son résumé.

– les auteurs, qui serviront à construire un graphe de co-auteurs.

– les citations, qui serviront à construire un graphe de citations.

– le lieu de publication, journal ou conférence, qui permettra d'étiqueter les articles.

Parmi ces données, le contenu, les citations et les auteurs nécessitent une attention particulière. Le contenu, souvent réduit au titre et résumé de l'article, est composé de texte brute. Il s'agit alors de pré-traiter le texte selon les méthodes usuelles de vectorisation comme la représentation par sac de mots. Le traitement du nom des auteurs demande un important travail de désambiguïsation. S'il existe des initiatives d'identification des chercheurs tel que ORCID (Open Researcher and Contributor ID), ceux-ci sont encore trop peu adoptés, et une étape de clustering, souvent semi-supervisée, est nécessaire. Enfin, les citations sont soit directement fournies avec les méta-données par le site de l'éditeur, soit extraites à partir du contenu complet de l'article avec des méthodes d'extraction automatiques. Il est par la suite nécessaire d'appliquer un algorithme de résolution de référence pour lier la citation avec l'article cité en base de données.

2.2. Quelques jeux de données

Il existe une multitude de bases de données de littérature scientifique accessibles publiquement et gratuitement. Cependant, un nombre restreint de jeux de données ont particulièrement intéressé la communauté scientifique car ils sont suffisamment traités pour être fiables et sont parfois agrémentés d'une vérité terrain. Je présente ci-dessous deux petits jeux de données, Cora et CiteSeer, selon les traitements appliqués dans (Sen *et al*., 2008) ainsi qu'un grand jeu, DBLP, largement utilisé par la communauté scientifique :

– **Cora** (McCallum *et al*., 2000) est un réseau de citations d'articles scientifiques dans le domaine de l'apprentissage artificiel, regroupant 7 classes (des sous-domaines scientifiques) avec 2708 documents, 1433 mots distincts dans le vocabulaire et 5429 liens de citations.

– **CiteSeer** (Giles *et al*., 1998) est aussi un réseau de citations d'articles scientifiques regroupant 6 classes sur 3312 documents, 3703 mots distincts dans le vocabulaire et 4732 liens de citations.

– **DBLP** (Ley, 2002) est une base de données de plusieurs millions d'articles scientifiques dans le domaine de l'informatique, démarrée en 1993. Un système de désambiguïsation performant permet d'identifier les auteurs (Ley, 2009). Dans (Deng *et al*., 2008), Google Scholar a été utilisé pour compléter des informations manquantes de DBLP telles que les résumés des articles et des sous-domaines de l'informatique ont été associés aux articles grâce à *eventseer.net*. Ensuite, ces associations ont été confirmées par jugements humains et selon la pertinence vraisemblable de plusieurs moteurs de recherches afin de créer un jeu de données d'experts (associations auteurs-expertise) selon (Zhang *et al*., 2007).

3. Contribution sur l'évaluation de la recherche d'experts

L'expertise est un concept flou, difficile à formaliser. De nombreux travaux ont cherché à construire des algorithmes de recherche d'experts dans de grandes bases de données pour divers domaines d'applications. Un exemple d'application consiste à trouver des chercheurs experts d'un domaine pour qu'ils effectuent une relecture d'un article soumis à un journal (*peer reviewing*). Il est commun de considérer l'expertise comme un savoir implicite, que des personnes portent et partagent de plusieurs façons. La recherche d'experts consiste alors à identifier ce savoir à travers des artefacts explicites, tels que des communications, des actions et des interactions mises en œuvre par ces personnes.

Formellement, un système de recherche d'experts reçoit une requête textuelle et produit un classement de candidats. L'évaluation d'un algorithme de recherche d'expert nécessite d'avoir un ensemble vérité terrain de candidats liés à plusieurs domaines d'expertises puis de confronter cet ensemble aux classements produits par l'algorithme pour différentes requêtes. Ma contribution concerne l'évaluation de cette tâche. Elle consiste à proposer un nouvel ensemble de requêtes qui se veut plus représentatif d'une application de recherche de relecteurs scientifiques (*reviewers*).

La plupart des travaux réalisés dans la recherche d'experts proposent de soumettre, comme requête, l'intitulé du domaine d'expertise. Ainsi, si notre base de données possède certains experts dans le domaine de la *recherche d'information*, on évaluera le classement des candidats produit par l'algorithme selon la requête textuelle «recherche d'information». Cependant, ces requêtes ne sont pas pleinement représentatives d'applications réelles pour deux raisons majeures :

– Il arrive souvent que différents utilisateurs proposent différentes formulations même s'ils cherchent la même chose. Ainsi, l'algorithme doit être capable de proposer des résultats similaires pour les requêtes «apprentissage automatique» et «apprentissage artificiel».

– On cherche rarement des experts dans des domaines aussi larges. Si une entreprise cherche un nouveau collaborateur, il est plus probable que le recruteur fournisse une description détaillée du poste à pourvoir plutôt qu'un court terme générique tel que «développeur informatique».

Je présente dans la section 3.2 deux types d'évaluations pour la recherche d'experts. La première, *topic-query*, est celle usuellement utilisée dans la littérature. La seconde, *document-query*, constitue ma contribution (Brochier *et al*., 2018). Dans la section 3.3, je confronte les résultats obtenus avec ces deux évaluations sur 3 algorithmes issus de la littérature et pour 3 représentations des documents, montrant que le choix du type de requêtes est décisif dans la sélection d'un meilleur algorithme.

3.1. État de l'art

L'automatisation de la recherche d'expert est apparue en même temps que la création des premières grandes base de données lors des digitalisations des bibliothèques et de la démocratisation des outils informatiques en entreprise. *P@noptic Expert* (Craswell *et al*., 2001) est l'un des premiers travaux sur la recherche d'expert. Le modèle proposé transforme la tâche de recherche de candidats en une tâche de similarité textuelle en agrégeant, pour chaque candidat de la base, l'ensemble de ses productions dans un méta-document. Classer les candidats selon une requête revient alors à calculer la similarité entre celle-ci et les méta-documents.

TREC-2005 Enterprise Track, Expert search task (Craswell *et al*., 2005) présenta un jeu de données et une évaluation pour la recherche d'expert. Une formalisation du problème émergea dans (Balog *et al*., 2006), avec le modèle génératif *document-model*. On note q une requête textuelle, d un document et e un candidat. La tâche revient à estimer la probabilité d'un candidat d'être un expert étant donnée une requête $P(e|q) = \frac{P(q|e)P(e)}{P(q)}$. L'approche générale consiste à introduire les documents dans la probabilité $P(q|e)$, qui n'est pas estimable en l'état. Les modèles de votes (Macdonald, Ounis, 2006) relâchent l'aspect probabiliste de cette dernière équation. Un exemple d'un tel modèle calcule le score d'un candidat selon une requête en agrégeant les scores de l'ensemble des documents auxquels le candidat est lié vis-à-vis de la requête.

Dans (Serdyukov *et al*., 2008), les auteurs adaptent l'algorithme PageRank (Page *et al*., 1999) de sorte à propager l'affinité de la requête avec les documents à travers le graphe bipartite documents-candidats. La propagation est un processus itératif où un auteur fortement connecté à des documents ayant reçus un score élevé reçoit à son tour un score élevé. Cette propagation peut être modérée à travers un coefficient

d'amortissement $\eta \in [0, 1]$, qui force les scores des nœuds du graphe bipartite à rester plus ou moins dans l'entourage de leur nœud initial.

3.2. Évaluations

Pour évaluer le classement de candidats produit par un algorithme selon une requête, on utilise différentes métriques usuelles en recherche d'information telles que la précision au rang K (P@K), la précision moyenne (AP) et le rang réciproque (RR) et l'aire (AUC) sous la courbe ROC. Pour chacune de ces métriques, on calcule leurs moyennes et écarts-types, à travers l'ensemble des requêtes ou à travers les domaines d'expertise.

L'évaluation *topic-query*

Cette approche est celle utilisée usuellement dans la littérature. Pour un domaine particulier, son nom ou sa description est directement utilisée comme requête, et les candidats associés à ce domaine, les experts, représentent la vérité terrain. On compare alors grâce aux métriques le classement produit par l'algorithme et la liste d'experts.

L'évaluation *document-query*

Une alternative consiste à échantillonner les documents liés aux experts de chaque domaine et à les utiliser comme requêtes. Au lieu d'utiliser directement la description du domaine, on utilise le contenu textuel des documents associés à la vérité terrain. On crée ainsi un ensemble de plusieurs requêtes par domaine, autant qu'il y a de documents associés aux experts de la vérité terrain.

Pour le jeu de données de recherche d'experts dans DBLP, on dénombre 210 experts, associés à 7 domaines, dont on peut extraire 3410 documents-requêtes. On aura donc 7 requêtes pour l'évaluation *topic-query* et 3410 pour l'évaluation *document-query*.

3.3. Expérimentations

Les résultats présentés dans le tableau 1 montrent les scores obtenus par trois algorithmes précédemment présentés dans la section 3.1: P@noptic, un modèle de vote et un modèle de propagation dont le coefficient d'amortissement $\eta = 0.1$ est faible, impliquant une forte propagation. De plus, chacun de ces modèles ont été testés avec trois représentations différentes des documents: TF (term frequency), TF-IDF (term frequency-inverse document frequency) et LSI (latent semantic indexing). Pour les deux types d'évaluation *topic-query* et *document-query*, la représentation des documents TF-IDF obtient généralement le meilleur score. La courbe ROC indique que LSI est meilleur pour classer les candidats les moins bien classés, mais est moins bon pour les experts les mieux classés.

Pour l'évaluation *topic-query*, le modèle de propagation est le meilleur, sauf quand réalisé avec les représentations LSI. À l'inverse, pour l'évaluation *document-query*,

Tableau 1 – Résultats de la recherche d'experts. Les valeurs en gras sont les meilleurs scores obtenus pour chaque métrique à travers chaque représentation de document.

(a) Scores moyens et écarts-types pour l'évaluation *topic-query*

		TF	TF-IDF	LSI
P@noptic	AUC	0.809±0.114	0.815±0.119	0.853±0.059
	P@10	0.757±0.176	0.814±0.146	**0.771**±0.183
	AP	0.580±0.175	0.613±0.186	0.580±0.157
	RR	**1.000**±0.000	**1.000**±0.000	1.429±0.495
Vote	AUC	0.788±0.131	0.793±0.136	**0.857**±0.048
	P@10	0.786±0.136	0.800±0.141	0.729±0.158
	AP	0.607±0.158	0.636±0.171	**0.599**±0.123
	RR	1.286±0.452	**1.000**±0.000	**1.000**±0.000
Prop ($\eta = 0.1$)	AUC	**0.860**±0.097	**0.866**±0.100	0.834±0.052
	P@10	**0.829**±0.148	**0.843**±0.118	0.686±0.181
	AP	**0.647**±0.142	**0.676**±0.139	0.564±0.140
	RR	**1.000**±0.000	**1.000**±0.000	1.143±0.350

(b) Scores moyens et écarts-types pour l'évaluation *document-query*.

		TF	TF-IDF	LSI
P@noptic	AUC	0.599±0.112	0.626±0.123	0.621±0.121
	P@10	0.324±0.278	0.387±0.285	0.343±0.287
	AP	0.282±0.145	0.318±0.164	0.302±0.158
	RR	4.904±5.731	3.557±4.976	4.810±5.895
Vote	AUC	**0.611**±0.120	**0.637**±0.129	**0.634**±0.132
	P@10	**0.370**±0.257	**0.417**±0.274	**0.370**±0.266
	AP	**0.303**±0.148	**0.338**±0.168	**0.318**±0.161
	RR	**3.211**±4.637	**2.752**±4.211	**3.686**±5.174
Propagation ($\eta = 0.1$)	AUC	0.596±0.113	0.625±0.121	0.612±0.119
	P@10	0.325±0.269	0.381±0.283	0.339±0.278
	AP	0.283±0.147	0.319±0.163	0.298±0.158
	RR	4.512±5.510	3.557±5.171	4.558±6.141

(c) Écarts-types entre domaines.

TF	TF-IDF	LSI
0.061	0.058	0.064
0.194	0.171	0.195
0.095	0.094	0.101
3.265	2.019	3.142
0.059	*0.056*	*0.058*
0.110	*0.112*	*0.115*
0.067	*0.073*	*0.071*
1.107	*0.908*	*1.307*
0.077	0.074	0.079
0.190	0.181	0.195
0.101	0.104	0.108
3.120	2.206	3.320

c'est le modèle de vote qui l'emporte. Une possible explication pour ce comportement est qu'un algorithme de propagation est plus robuste à une requête courte et vague telle que « data mining » puisqu'il utilise fortement le voisinage d'un document pour calculer son score. Inversement, le modèle de vote sera plus précis pour des requêtes complexes, telle qu'un résumé d'article puisqu'il sera capable d'identifier précisément les articles proches de cette requête, sans être bruité par l'aspect réseau du corpus.

Un élément intéressant est la différence entre l'écart-type entre domaines d'expertise des deux approches d'évaluation. Celles-ci sont significativement moindres pour l'approche *document-query*. Ceci signifie que l'utilisation des noms des domaines comme requêtes est biaisée. On peut l'expliquer par le fait que certains termes utilisés sont plus discriminatifs que d'autres. Par exemple, les mots « data » et « mining » ont plus de chances d'être utilisés en apprentissage artificiel, quelque soit le domaine d'expertise, que les termes « planning » et « agents », qui sont assez spécifiques à leurs sous-domaines.

Finalement, il est intéressant d'observer que le modèle de vote, pour l'évaluation *document-query*, engendre l'écart-type entre domaines d'expertise le plus bas. Ceci fournit un élément d'analyse important pour le choix d'un algorithme à utiliser sur un cas pratique. En effet, si la performance d'un algorithme est importante on peut aussi vouloir favoriser sa constance à travers différents domaines, notamment ceux les moins représentés en base de données.

4. Contribution sur le plongement de graphe

La structure des liens au sein d'un réseau renferme d'importantes informations sur ses nœuds. Une approche pour faciliter le traitement de ces informations consiste à apprendre des représentations de ces nœuds en utilisant des techniques originellement appliquées au plongement de mots dans un espace de faible dimension.

Ma contribution (Brochier *et al.*, 2019) consiste à (1) proposer un algorithme de plongement de graphe, nommé *GVNR* (Global Vectors for Node Representation). Inspiré par GloVe (Pennington *et al.*, 2014), cet algorithme factorise la matrice des comptes de co-occurrences des nœuds lors de marches aléatoires. En formulant le problème comme une régression sur les valeurs non nulles de la matrice et sur des valeurs nulles aléatoirement échantillonnées, il est possible d'apprendre des représentations aussi performantes que l'état de l'art. (2) Je montre ensuite comment étendre ce modèle pour intégrer l'information textuelle liée aux nœuds. (3) J'évalue cet algorithme sur plusieurs jeux de données et montre qu'il présente non seulement de meilleurs résultats que GloVe, mais aussi de meilleurs résultats que certains travaux récents de la littérature.

4.1. État de l'art

La qualité de l'information capturée dans des représentations de données influence fortement la performance d'algorithmes de recherche d'information. Pour cette raison, beaucoup d'effort est dévoué à trouver de nouvelles méthodes d'apprentissage de représentations (Bengio *et al.*, 2013). Le plongement de graphe (*network embedding*), c'est à dire l'apprentissage de représentation de nœuds dans un graphe, est fortement lié à l'apprentissage de représentations des mots (*word embedding*).

L'hypothèse distributionnelle constitue la base des algorithmes de plongement de mots (Sahlgren, 2008). Celle-ci suppose que la similarité distributionnelle des mots corrèle fortement avec leur similarité sémantique. En d'autres termes, si l'on apprend une représentation d'un mot qui permette de prédire les autres mots qui occurrent dans son contexte, on aura une représentation de son sens.

Skip-Gram (Mikolov *et al.*, 2013) est un algorithme qui apprend des représentations des mots en maximisant la log-vraisemblance d'un ensemble de paires de mots co-occurrents, C: $\sum_{(w_i,w_j)\in C} \log p(w_j|w_i)$. Ces paires sont extraites d'un corpus en glissant une fenêtre contextuelle. Skip-Gram avec échantillonnage négatif (Skip-gram with Negative Sampling: SGNS) est une variante proposée dans (Mikolov *et al.*,

2013) afin d'approcher efficacement la log-vraisemblance. Ceci est réalisé en réduisant la tâche en une classification qui consiste à distinguer des paires de mots qui co-occurrent avec des fausses paires qui ne co-occurrent pas. Une approche alternative, GloVe (Pennington *et al.*, 2014), apprend des représentations des mots en factorisant une matrice de comptes de co-occurrences d'un corpus. Son objectif est de minimiser l'erreur de reconstruction de la matrice, en ne considérant que les valeurs non-nulles de co-occurrences.

Si l'hypothèse distributionnelle émergea de la linguistique, (Perozzi *et al.*, 2014) ont établi que la fréquence à laquelle les nœuds d'un graphe apparaissent dans de courtes marches aléatoires suit la distribution d'une loi de puissance. Ils proposent alors d'adapter Skip-Gram pour l'apprentissage de représentations de graphes, appliquant la fenêtre de contexte sur des séquences de nœuds, assimilables à des phrases. De nombreux travaux, tels que (Grover, Leskovec, 2016) et (Dong *et al.*, 2017) ont étendu l'utilisation de SGNS, pour le plongement de graphe. En particulier, NetMF (Qiu *et al.*, 2018) unifie certains de ces travaux et les reformule comme des factorisations de matrices. Enfin, certains algorithmes comme TADW (Yang *et al.*, 2015) permettent d'intégrer des attributs des nœuds, particulièrement du texte, pour améliorer les représentations apprises.

4.2. Formulation du modèle

On considère un graphe pondéré et non-orienté $G = (V, E)$ constitué de n nœuds V connectés par des arêtes E. On cherche une représentation vectorielle de faible dimension d qui préserve leurs similarités en terme de co-occurrences C, analogues à celles de mots dans Skip-Gram. Ces co-occurrences sont générées en réalisant γ marches aléatoires de longueurs t à partir de chaque nœud, en fonction des poids des arêtes. La matrice de co-occurrences X est construite en comptant les co-occurrences dans une fenêtre de taille l sur les séquences générées par les marches.

Une étape additionnelle réalise un filtrage de la matrice X. Ses coefficients proches de zéro peuvent être légitimement considérés comme du bruit, car ils sont le fruit de très rares co-occurrences lors des marches aléatoires. Ainsi, tous les éléments inférieurs à un seuil $x_{\min}$ sont mis à zéro. Ce traitement présente deux avantages. Premièrement, il permet de réduire drastiquement le nombre de valeurs non-nulles de X, réduisant ainsi le temps de sa factorisation. Ensuite, comme présenté dans la section 4.4, cela améliore la qualité des représentations apprises dans la pratique.

Notre objectif est alors d'apprendre deux matrices de représentation des nœuds U et V et leurs biais respectifs b^U et b^V, où U correspond aux représentations des nœuds cibles et V aux représentations des nœuds contextes. *GVNR* consiste alors à optimiser l'objectif suivant, avec (u_i, b_i^U) et (v_j, b_j^V) les représentations et biais respectivement du nœud cible i et du nœud contexte j :

$$\underset{U,V,b^U,b^V}{\operatorname{argmin}} \sum_{i=1}^{n} \sum_{j=1}^{n} s(x_{ij}) \big(u_i \cdot v_j + b_i^U + b_j^V - \log(c + x_{ij})\big)^2 \qquad (1)$$

La constante $c \in]0; 1]$ permet le lissage de X tout en maintenant le logarithme négatif lorsque $x_{ij} = 0$. La fonction s sélectionne les coefficients de X que l'on considère pour mesurer l'erreur de reconstruction :

$$s(x_{ij}) = \begin{cases} 1 & \text{si } x_{ij} > 0, \\ m_i & \text{sinon, avec } m_i \sim \text{Bernoulli}\left(\frac{k \times n_i}{n - n_i}\right). \end{cases} \tag{2}$$

Cette fonction prend la valeur 1 pour toutes les valeurs non nulles de X ainsi que pour une partie des valeurs nulles. Pour les autres valeurs, elle prend la valeur zéro, réduisant la complexité de l'algorithme. Notez que, pour un nœud donné i, le nombre de nœuds non-cooccurrents retenus vaut en moyenne k fois le nombre de nœuds co-occurrents n_i. En pratique, les meilleurs résultats sont obtenus avec $k = 1$.

4.3. Extension pour un réseau de documents de petites tailles

Il est possible d'étendre la formulation précédente pour prendre en compte le texte associé aux nœuds. En considérant que l'ordre des mots est négligeable, on peut représenter le document d'un réseau par son sac de mots doc_j. L'extension consiste à représenter le vecteur contexte v_j d'un nœud par le barycentre des représentations de ces mots. On introduit pour cela une matrice $W \in \mathbb{R}^{m \times d}$, représentations des m mots présents dans le vocabulaire du corpus de documents. Le vecteur contexte d'un nœud j s'écrit alors:

$$v_j = \frac{\text{doc}_j \, W}{|\text{doc}_j|_1} \tag{3}$$

De cette façon, le modèle apprend conjointement des représentations des nœuds, des mots et des documents d'un réseau. De plus, il devient possible d'obtenir la représentation d'un nouveau document en appliquant l'équation (3) à sa représentation par sac de mots.

4.4. Expérimentations

Un algorithme de plongement de graphe produit un ensemble de représentations. Pour évaluer celles-ci, il est courant de les utiliser comme espace d'entrée pour un algorithme linéaire afin de résoudre une tâche particulière, telle que la classification multi-classe. Je présente ici les résultats de *GVNR* comparés à d'autres algorithmes de la littérature.

4.4.1. Tâche et mesure d'évaluation

L'efficacité de *GVNR* est évaluée en réalisant une tâche de classification multi-classe. Je reporte ici les résultats obtenus sur deux jeux de données, Cora et CiteSeer, présentés dans la section 2.2. Pour chacun des jeux de données, la performance de *GVNR* et de son extension sont évalués. L'évaluation consiste à apprendre des représentations sur un jeu de données, puis d'utiliser ces représentations comme entrées

d'un classifieur linéaire, en l'occurrence une régression logistique utilisant l'implémentation LIBLINEAR (Fan *et al.*, 2008). Pour chaque ensemble de représentations produites par un algorithme précis, on fait varier la proportion de vecteurs d'apprentissage de 10% à 50% et la précision moyenne des prédictions du classifieur est calculée sur le reste des nœuds. Cette démarche est répétée 10 fois pour chaque proportion, tirant à chaque fois aléatoirement l'ensemble d'apprentissage.

4.4.2. Représentations comparées

GVNR et GloVe sont opérés sur une matrice de co-occurrence produite par $\gamma = 80$ marches aléatoires par nœuds de longueurs $t = 40$ et en appliquant une fenêtre de taille $l = 5$. Pour les algorithmes ci-dessous, la dimension de représentations apprises est de $d = 80$:

– GloVe: les résultats sont obtenus avec un seuillage $x_{\max} = 10$.

– NetMF: les résultats sont obtenus avec la version exacte (pour petit graphe) fournie par les auteurs [1] et $k = 10$ échantillons négatifs.

– NetMF+SVD: représente la concaténation de vecteurs des nœuds issus de NetMF et de représentations des documents obtenus par décomposition en valeurs singulières de leurs représentations TF-IDF.

– TADW: obtenus avec l'implémentation fournie par *OpenNE* [2] avec 20 itérations.

– *GVNR* : les résultats sont obtenus avec $c = 1$, $x_{\min} = 1$ et $k = 1$.

4.4.3. Résultats

Les résultats présentés dans les tableaux 2 et 3 montrent les précisions moyennes obtenues sur les deux jeux de données. Tout d'abord, on observe que *GVNR* , grâce à l'échantillonnage de valeurs non nulles sur X, obtient de meilleurs résultats que Glove. De plus, *GVNR* avec un filtrage $x_{\min} = 1$ produit de meilleurs représentations que sans filtrage, confirmant ainsi l'hypothèse que les valeurs faibles de X peuvent être considérées comme accidentelles. Enfin, *GVNR* ($x_{\min} = 1$) obtient des résultats similaires à NetMF et avec un léger avantage sur CiteSeer.

Lorsque l'on étend *GVNR* avec l'intégration du contenu textuel des documents, on est capable d'améliorer significativement les résultats, obtenant de meilleures performances que TADW qui repose pourtant sur des représentations textuelles plus coûteuses en temps, obtenues par SVD.

5. Conclusion et Perspectives

Mes travaux sur l'apprentissage de représentations pour la recommandation de données issues de la littérature scientifique ont couvert deux points. Le premier concerne

1. https://github.com/xptree/NetMF
2. https://github.com/thunlp/OpenNE/blob/master/src/openne/tadw.py

Tableau 2 – Résultats de la classification multiclasse sur le jeu de données Cora

% of training data	10%	20%	30%	40%	50%
GloVe	57.7	62.4	69.5	72.8	73.8
GVNR ($x_{\min} = 0$)	58.5	62.5	70.7	73.4	75.0
NetMF	65.7	**72.9**	**76.4**	**78.6**	79.4
GVNR ($x_{\min} = 1$)	**69.5**	72.6	75.9	78.1	**80.2**
SVD	54.7	61.0	62.4	63.0	62.8
NetMF+SVD	72.5	77.2	78.1	78.1	77.9
TADW	77.1	78.8	78.2	78.8	78.6
GVNR (avec texte)	**79.3**	**80.7**	**80.8**	**81.4**	**81.1**

Tableau 3 – Résultats de la classification multiclasse sur le jeu de données CiteSeer

% of training data	10%	20%	30%	40%	50%
GloVe	42.8	53.5	55.3	56.2	56.8
GVNR ($x_{\min} = 0$)	38.7	46.8	49.1	50.4	50.9
NetMF	**51.2**	54.8	55.1	55.0	54.8
GVNR ($x_{\min} = 1$)	45.6	**55.6**	**57.3**	**58.7**	**59.0**
SVD	52.0	54.7	54.7	58.4	65.7
NetMF+SVD	57.0	59.6	59.6	59.4	60.7
TADW	60.6	60.1	60.1	66.2	69.3
GVNR (avec texte)	**63.3**	**62.5**	**64.9**	**68.6**	**70.4**

l'étude de l'évaluation d'une tâche précise, la recommandation d'experts, dont l'approche classique ne parait pas adaptée à la sélection d'un algorithme pour l'application au cas industriel de recherche de *reviewer*. J'ai donc proposé une nouvelle évaluation qui souligne cette inadéquation. Pour parfaire cette proposition, l'étiquetage des documents-requêtes serait une contribution intéressante pour la communauté scientifique. Le second est une nouvelle méthode d'apprentissage de représentations des nœuds d'un réseau, permettant de prendre en compte l'information textuelle leurs étant associée.

Dans de futurs travaux, j'explorerai deux nouveaux aspects. Premièrement, ma contribution pour le plongement de graphe ne s'applique pas aux réseaux hétérogènes. Pour la recherche d'experts, il est nécessaire de prendre en compte la variété des types de données en présence (articles, auteurs, lieux de publications) dont on pourrait chercher à trouver un espace de représentation commun, comme introduit dans (Dong *et al.*, 2017). Ensuite, l'exploitation de l'information textuelle peut être améliorée en intégrant dans l'apprentissage de représentation des méthodes mieux adaptées. De récentes techniques, telles que les mécanismes d'attention (Vaswani *et al.*, 2017), ont montrées d'excellentes performances en terme de précision et de complexité algorithmique en traitement automatique du langage.

Bibliographie

Balog K., Azzopardi L., De Rijke M. (2006). Formal models for expert finding in enterprise corpora. In *Proceedings of the 29th annual international acm sigir conference on research and development in information retrieval*, p. 43–50.

Bengio Y., Courville A., Vincent P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, n° 8, p. 1798–1828.

Brochier R., Guille A., Rothan B., Velcin J. (2018). Impact of the query set on the evaluation of expert finding systems. In *Birndl 2018 (sigir 2018).*

Brochier R., Guille A., Velcin J. (2019). Global vectors for node representations. In *Proceedings of the 2019 world wide web conference (www '19).*

Craswell N., Hawking D., Vercoustre A.-M., Wilkins P. (2001). P@ noptic expert: Searching for experts not just for documents. In *Ausweb poster proceedings, queensland, australia*, vol. 15, p. 17.

Craswell N., Vries A. P. de, Soboroff I. (2005). Overview of the trec 2005 enterprise track. In *Trec*, vol. 5, p. 199–205.

Deng H., King I., Lyu M. R. (2008). Formal models for expert finding on dblp bibliography data. In *Data mining, 2008. icdm'08. eighth ieee international conference on*, p. 163–172.

Dong Y., Chawla N. V., Swami A. (2017). metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining*, p. 135–144.

Fan R.-E., Chang K.-W., Hsieh C.-J., Wang X.-R., Lin C.-J. (2008). Liblinear: A library for large linear classification. *Journal of machine learning research*, vol. 9, n° Aug, p. 1871–1874.

Giles C. L., Bollacker K. D., Lawrence S. (1998). Citeseer: An automatic citation indexing system. In *Proceedings of the third acm conference on digital libraries*, p. 89–98.

Grover A., Leskovec J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, p. 855–864.

Ley M. (2002). The dblp computer science bibliography: Evolution, research issues, perspectives. In *International symposium on string processing and information retrieval*, p. 1–10.

Ley M. (2009). Dblp: some lessons learned. *Proceedings of the VLDB Endowment*, vol. 2, n° 2, p. 1493–1500.

Macdonald C., Ounis I. (2006). Voting for candidates: adapting data fusion techniques for an expert search task. In *Proceedings of the 15th acm international conference on information and knowledge management*, p. 387–396.

McCallum A. K., Nigam K., Rennie J., Seymore K. (2000). Automating the construction of internet portals with machine learning. *Information Retrieval*, vol. 3, n° 2, p. 127–163.

Mikolov T., Sutskever I., Chen K., Corrado G. S., Dean J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, p. 3111–3119.

Page L., Brin S., Motwani R., Winograd T. (1999). *The pagerank citation ranking: Bringing order to the web.*. Rapport technique. Stanford InfoLab.

Pennington J., Socher R., Manning C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)*, p. 1532–1543.

Perozzi B., Al-Rfou R., Skiena S. (2014). Deepwalk: Online learning of social representations. In *Proceedings of the 20th acm sigkdd international conference on knowledge discovery and data mining*, p. 701–710.

Qiu J., Dong Y., Ma H., Li J., Wang K., Tang J. (2018). Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *Proceedings of the eleventh acm international conference on web search and data mining*, p. 459–467.

Sahlgren M. (2008). The distributional hypothesis. *Italian journal of linguistics*, p. 23–53.

Sen P., Namata G., Bilgic M., Getoor L., Galligher B., Eliassi-Rad T. (2008). Collective classification in network data. *AI magazine*, vol. 29, n° 3, p. 93.

Serdyukov P., Rode H., Hiemstra D. (2008). Modeling multi-step relevance propagation for expert finding. In *Proceedings of the 17th acm conference on information and knowledge management*, p. 1133–1142.

Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N. *et al*. (2017). Attention is all you need. In *Advances in neural information processing systems*, p. 5998–6008.

Yang C., Liu Z., Zhao D., Sun M., Chang E. Y. (2015). Network representation learning with rich text information. In *Ijcai*, p. 2111–2117.

Zhang J., Tang J., Li J. (2007). Expert finding in a social network. In *International conference on database systems for advanced applications*, p. 1066–1069.