Recommandation contextuelle d'utilisateurs pour les plateformes de micro-blogging

Camelia Constantin [1], **Ryadh Dahimene** [2], **Cédric du Mouza** [2], **Quentin Grossetti** [2]

1. *Univ. Pierre et Marie Curie, 2 Place Jussieu, 75005 Paris, France*
   camelia.constantin@lip6.fr
2. *CNAM Paris, 2 rue Conté, 75141 Paris, France*
   dahimene.ryadh@gmail.com,dumouza@cnam.fr,quentin.grossetti@cnam.fr

RÉSUMÉ. *Les services de micro-blogging sont devenus récemment une source d'information importante. Cependant, victimes de leur succès, ils doivent actuellement gérer une quantité sans précédent d'informations générées par les utilisateurs. Il devient par conséquent difficile pour les utilisateurs de trouver dans ces services des contenus proches de leurs intérêts. Afin de recommander des utilisateurs à suivre sur un sujet donné, nous proposons dans cet article des scores basés sur la topologie du graphe social ainsi que sur le contenu textuel des microblogs. Pour permettre le passage à l'échelle, nous présentons une approche qui s'appuie sur l'utilisation de landmarks pour pré-calculer des recommandations pour certains comptes choisis dans le graphe. Nos expériences confirment la pertinence de notre score de recommandation par rapport à des approches existantes ainsi que le passage à l'échelle de notre algorithme basé sur l'utilisation des landmarks.*

ABSTRACT. *Micro-blogging systems have become a prime source of information. However due to their unprecedented success, these systems have to face an exponentially increasing amount of user generated content. As a consequence, finding users who publish quality content that matches precise interests is a real challenge for the average user. We present in this article a recommendation score which takes advantage of the social graph topology and of the existing contextual information to recommend users to follow on a given topic. We also introduce a landmark-based algorithm that precomputes recommendation scores for a given set of graph nodes and that allows to scale. Our experimental results confirm the relevance of this score against existent approaches as well as the scalability of our landmark-based algorithm.*

MOTS-CLÉS : *recommandation, réseaux sociaux, micro-blogging*

KEYWORDS: *recommendation, social networks, micro-blogging*

DOI:10.3166/ISI.21.3.93-118

1. Introduction

Les micro-blogs sont devenus un moyen de communication incontournable du Web 2.0. Twitter, le principal service de micro-blog, a connu une croissance spectaculaire pour atteindre 570 millions d'utilisateurs en avril 2014 en moins de 7 ans d'existence. Aujourd'hui, environ un million de nouveaux comptes sont crées sur Twitter chaque semaine, alors qu'ils n'étaient que 1000 en 2008. 500 millions de tweets sont envoyés chaque jour et, en moyenne, un utilisateur Twitter suit 108 comptes. Facebook est un autre exemple avec 1,26 milliard d'utilisateurs qui publient en moyenne 36 posts par mois. Un utilisateur Facebook suit en moyenne 130 "amis" ce qui donne pour un utilisateur qui se connecte 1500 nouveaux posts affichés en moyenne [1]. D'autres systèmes similaires comme Google+, Instagram, Youtube, Sina Weibo ou Plurk, pour citer les plus grands, présentent également une telle croissance.

Suite à ce succès rapide et sans précédent plusieurs défis pour les fournisseurs de services ainsi que pour leurs utilisateurs sont apparus. Tandis que les premiers doivent faire face à un énorme flux de contenu généré par les utilisateurs, les derniers luttent pour trouver de l'information pertinente qui correspond à leurs intérêts : ils doivent prendre un temps conséquent pour lire tout le contenu reçu tout en essayant de filtrer l'information pertinente. Deux approches complémentaires ont émergé pour aider l'utilisateur à trouver l'information pertinente qui correspond à ses intérêts dans le grand flux de contenu généré par les utilisateurs : le filtrage de posts comme dans (Kapanipathi *et al*., 2011 ; Koroleva, Röhler, 2012 ; Esparza *et al*., 2012) et la recherche et/ou la recommandation de posts/comptes comme dans (Diaz-Aviles *et al*., 2012 ; Chen *et al*., 2012 ; Chin *et al*., 2013). Les systèmes de réseaux sociaux offrent généralement la possibilité de chercher des posts ou des comptes qui correspondent à un ensemble de mots-clés. Cela peut-être une recherche « locale » pour filtrer les posts reçus, ou une recherche « globale » pour interroger l'ensemble complet de posts/comptes existant. Pour cette dernière recherche il existe deux options : des ensembles de posts pré-calculés qui correspondent aux *hot topics*, ou des recherches personnalisées pour lesquelles le résultat est construit d'après les mots-clés spécifiés d'après l'utilisateur. Cependant la sémantique de *matching total* généralement adoptée par les outils de recherche est très limitée. Même un score de classement basé sur le nombre de mots-clés satisfaits n'est pas suffisant pour retrouver tous les posts d'intérêt. Combiné au manque de sémantique et aux nombres de posts par jour, le large nombre de recherches réalisées chaque jour pose aussi des problèmes de passage à l'échelle. Par exemple en 2012, plus de recherches ont été réalisées chaque mois sur Twitter (24 milliards) que sur Yahoo (9,4 milliards) ou sur Bing (4,1 milliards).

Dans cet article, nous considérons le problème de découvrir des publiants de contenu de qualité en fournissant des recommandations topologiques et contextuelles sur un graphe social de micro-blogging. Les systèmes de micro-blogging sont caractérisés par l'existence d'un large graphe social dirigé où chaque utilisateur (compte)

1. http://expandedramblings.com

peut décider librement de se connecter à n'importe quel utilisateur afin de recevoir ses posts. Dans cet article nous faisons l'hypothèse qu'un lien entre un utilisateur u et un utilisateur v exprime un intérêt de u pour un ou plusieurs topics dans le contenu publié par v. Nous choisissons par conséquent de modéliser le graphe social sous-jacent par un graphe social étiqueté (LSG) où les étiquettes correspondent aux topics d'intérêt des utilisateurs. Notre objectif est de proposer un score de recommandation qui capture à la fois la proximité topologique et la connectivité d'un publiant avec son autorité sur un topic donné et l'intérêt des utilisateurs intermédiaires entre celui devant être recommandé et le publiant.

La taille du graphe social sous-jacent pose des problèmes quand nous considérons des opérations qui impliquent l'exploration du graphe. Dans le but d'accélérer le processus de recommandation, nous proposons un calcul approché basé sur des landmarks, *i.e.*, nous sélectionnons un ensemble de nœuds dans le graphe social, appelé *landmarks*, qui joueront le rôle de hubs et qui stockeront des données concernant leurs voisins. Cet ensemble de landmarks est sélectionné parmi les stratégies existantes (nous comparons certaines dans la Section 5).

Contributions

Dans cet article nous proposons un système de recommandation qui procure des recommandations d'utilisateurs personnalisées. Nos principales contributions sont :

1) en considérant l'idée que les mesures basées sur la topologie du graphe sont de bons indicateurs pour la similarité d'utilisateurs, nous proposons un score topologique qui intègre des informations sémantiques sur les utilisateurs et leurs relations ;

2) de plus, nous introduisons une approche basée sur les landmarks pour améliorer le temps d'exécution et atteindre un gain de 2-3 ordres de grandeur par rapport au calcul exact ;

3) une validation expérimentale de notre approche, incluant une étude comparative avec d'autres approches (TwitterRank (Weng *et al.*, 2010) et Katz (Liben-Nowell, Kleinberg, 2003)).

Dans cet article nous présentons d'abord notre score de recommandation qui exploite les données intrinsèques au graphe social étiqueté que nous nommons *LSG*. Nous décrivons ensuite une approche par *landmark* permettant d'obtenir une approximation efficace de nos scores sur le graphe social. Nous présentons les différentes expériences conduites afin de valider la qualité mais aussi l'efficacité de notre approche. Enfin, une conclusion résume les contributions de notre approche ainsi que les pistes d'amélioration envisagées.

2. État de l'art

Les données issues des réseaux sociaux représentent une véritable mine d'information pour un éventuel système de recommandation. Fonctionnant sur l'adage *« Dis moi qui tu fréquentes, je te dirai qui tu es »*, nous pouvons identifier un nouveau type

de systèmes de recommandation basé sur la présence d'une *communauté* d'utilisateurs liés par des liens sociaux (Ricci *et al*., 2011). Sur les plates-formes sociales, ces systèmes de recommandation permettent de recommander par exemple des utilisateurs à suivre, des publications précises, des éléments multimédias, des groupes (sous-communautés) à intégrer, etc.

La principale caractéristique des réseaux sociaux étant l'existence d'un graphe de relations sociales, cette donnée est l'information principale au centre des diverses stratégies de recommandation. (Liben-Nowell, Kleinberg, 2003) présente une étude qui compare diverses méthodes de prédiction de liens. La prédiction de liens consiste à analyser l'état du réseau à un instant t afin d'anticiper la création de nouveaux liens à un moment $t+1$. Ces techniques sont souvent utilisées dans le contexte de la recommandation. Les méthodes présentées dans (Liben-Nowell, Kleinberg, 2003) se basent sur les propriétés topologiques des réseaux pour le calcul des prédictions. Une autre mesure topologique est présentée dans (Budalakoti, Bekkerman, 2012), les auteurs proposent de combiner deux scores de classement de comptes basés sur deux sources différentes issues du même réseau (les invitations sur le réseau social, ainsi que le graphe social proprement dit). Ces données sont ensuite utilisées pour produire la liste triée des comptes les plus influents sur le réseau.

Certains travaux appliquent des méthodes de filtrage collaboratif ainsi que des méthodes basées sur le contenu dans un contexte social. (Chen *et al*., 2012) introduit une méthode pour classer des *tweets* qui exploite les profils de préférences utilisateur, une mesure d'autorité du compte publiant le *tweet* ainsi que la qualité de la publication. (Hannon *et al*., 2010) compare un ensemble de méthodes d'extraction de profils utilisateur sur le réseau *Twitter*, basées sur le contenu publié par l'utilisateur, celui des comptes qu'il suit ainsi que celui de ses *suiveurs*. Un classement basé sur le score de TF-IDF est ensuite employé pour trouver les utilisateurs similaires. Une méthode similaire de recommandation est adoptée par (Pennacchiotti *et al*., 2012). Dans (Diaz-Aviles *et al*., 2012) les auteurs décrivent une approche qui garde trace des interactions passées par les utilisateurs pour le calcul en temps-réel des recommandations. Les techniques présentées par (Pennacchiotti *et al*., 2012 ; Diaz-Aviles *et al*., 2012) et (Chen *et al*., 2012) fournissent des recommandations au niveau de granularité du *tweet*. Le passage à l'échelle est problématique étant donné le nombre important de *tweets*, l'aspect temps-réel de la plate-forme et les fréquences élevées de publication.

Les approches citées précédemment ne prennent pas en considération la topologie du graphe dans le calcul des recommandations. (Weng *et al*., 2010) présente une adaptation de l'algorithme *PageRank* pour le micro-blogging appelée *TwitterRank*. Cette approche prend en compte la structure des liens du graphe ainsi que l'autorité des utilisateurs sur des sujets (*topics*) donnés. Les *topics* utilisés par *TwitterRank* sont obtenus par l'application de la méthode LDA (Allocation de Dirichlet Latente) qui est une technique probabiliste permettant la caractérisation du contenu des utilisateurs. (Liang *et al*., 2012) propose une méthode basée sur le contenu afin de fournir des recommandations de sujets (*topics*) en exploitant les liens implicites issus des conventions adoptées sur les plates-formes de micro-blogging. (Kywe *et al*., 2012) propose

une technique de recommandation de *hashtags* personnalisée. Certains *hashtags* ayant un cycle de vie extrêmement court, les auteurs proposent de combiner le contenu des *tweets* avec une approche de filtrage collaboratif sur une fenêtre temporelle d'un mois afin de recommander des *hashtags* pertinents aux utilisateurs. Une approche présentée par (Chaoji *et al.*, 2012) vise à maximiser la découverte de nouveau contenu lors de la tâche de recommandation. Ce problème s'apparente à un problème d'optimisation multi-objectif *NP-difficile*. Les auteurs proposent une approximation qui atteint un degré de propagation de contenu important. Cependant l'application de cette méthode sur de grands graphes demeure problématique. Dans (Gupta *et al.*, 2013), les auteurs présentent le système de recommandation mis en production par *Twitter* pour sa fonction « *Qui suivre ?* » (*Who to follow ?*). Il se base sur le déploiement de l'algorithme SALSA (Lempel, Moran, 2001) dans un environnement centralisé. SALSA fonctionne en créant un graphe biparti avec d'un côté le cercle de confiance d'un utilisateur (pouvant être calculé par exemple comme l'ensemble les comptes avec qui l'utilisateur interagit le plus ou bien à partir d'un algorithme de marche aléatoire) et de l'autre coté les comptes les plus suivis par ce cercle de confiance, considéré comme des autorités. Cette approche ne prend pas en considération le sujet (*topic*) sur lequel les comptes recommandés peuvent être une autorité.

3. Modèle

Dans cette section, nous définissons les différentes composantes du score de recommandation proposé.

3.1. Graphe social étiqueté

Nous modélisons le réseau social *Twitter* par un graphe dirigé étiqueté *G=(U, E,* $\mathcal{T}$*,* l_u, l_e*)* où $\mathcal{U}$ est l'ensemble des sommets tels que chaque sommet $u \in \mathcal{U}$ est un utilisateur (compte). $E \subseteq \mathcal{U} \times \mathcal{U}$ est l'ensemble des arêtes où une arête $e = (u, v) \in E$ existe si u suit v, $i.e$ u reçoit les publications de v. La fonction d'étiquetage $l_u : \mathcal{U} \rightarrow 2^{\mathcal{T}}$ associe à chaque utilisateur u l'ensemble des sujets (topics) qui caractérise ses publications, choisis dans un vocabulaire de topics défini $\mathcal{T}$. Les topics associés par la fonction d'étiquetage $l_e : E \rightarrow 2^{\mathcal{T}}$ à une arête $e = (u, v)$ décrivent les intérêts de l'utilisateur u pour les posts de v. La détection de topics est un problème largement étudié (Diaz-Aviles *et al.*, 2012) et ne rentre pas dans le cadre de ce travail. Nous supposons ici un ensemble donné de topics qui peuvent être explicitement définis par les utilisateurs ou inférés à partir des posts comme nous avons fait par exemple dans la section 5. Pour un utilisateur u, on définit $\Gamma^{u^-}(t)$ l'ensemble des comptes suivant u sur le topic t et par $\Gamma^{u^+}(t)$ l'ensemble des utilisateurs qui suivent u sur t.

Afin d'illustrer le fonctionnement de notre système de recommandation, nous présentons dans la figure 1 un exemple de graphe étiqueté avec le contenu des publications pour les comptes *B* et *C*. Pour simplifier, chaque arête est étiquetée avec un seul topic.

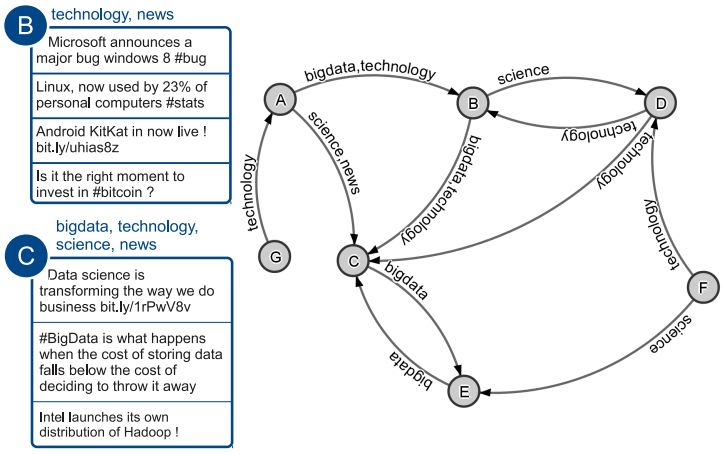

Figure 1. Exemple de graphe étiqueté dans un contexte de recommandation

Notre objectif est de recommander à un utilisateur u du système des comptes pertinents à suivre pour une requête $Q = \{t_1, ..., t_n\}$ composée de plusieurs *topics*. Notre proposition, TR, se base sur les hypothèses suivantes pour calculer un score de recommandation qui prend en considération à la fois la topologie du graphe et la sémantique du contenu :

i) *La proximité topologique* est une donnée cruciale pour le calcul de la recommandation, *c.à.d* un compte u fait confiance à ses voisins directs dans le graphe, aux voisins de ses voisins, etc. Cette confiance décroit avec la distance (Liben-Nowell, Kleinberg, 2003).

ii) *Le nombre de chemins* entre u et v est lui aussi important : un compte v a de grandes chances d'être considéré comme important par un compte u s'il existe un grand nombre d'autres comptes liés à u qui recommandent v.

iii) *La pertinence par rapport à la requête Q* sur les chemins existants entre u et v doit être aussi considérée dans le calcul des recommandations.

3.2. Le score de recommandation

Basé sur ces hypothèses, le score de recommandation $\sigma(u, v, t)$ d'un utilisateur v pour l'utilisateur u sur un *topic* t calculé sur des chemins $p = u \rightsquigarrow v$ est exprimé comme étant la somme des scores de chemin $\overline{\omega}_p(t)$:

DÉFINITION 1 (Score de recommandation). —

$$\sigma(u, v, t) = \sum_{p \in P_{u,v}} \overline{\omega}_p(t) = \sum_{p \in P_{u,v}} \beta^{|p|} \omega_p(t) \quad (1)$$

où $P_{u,v}$ est l'ensemble de tous les chemins existants entre u et v. Le score $\overline{\omega}_p(t)$ sur un chemin de longueur $|p|$ prend en considération à la fois la pertinence $\omega_p(t)$ et le facteur de distance $\beta^{|p|}$ sur le chemin p. Le facteur de décroissance $\beta \in [0, 1]$ permet d'accorder plus d'importance aux chemins courts.

Le score de recommandation final pour la requête Q est calculé comme étant une combinaison linéaire où chaque score d'un topic t_i est pondéré par la pertinence de t_i pour les posts de u qui est calculée pendant l'extraction des topics (voir la section 5).

Un score purement topologique peut être obtenu à partir de l'équation précédente si l'on ignore la pertinence des chemins (*c.à.d* si $\omega_p(t)$ est 1). Ce score est plus élevé s'il existe beaucoup de chemins cours entre u et v et il est exprimé comme suit :

$$topo_\beta(u, v) = \sum_{p \in P_{u,v}} \beta^{|p|} \quad (2)$$

Ce score correspond à la mesure de *Katz* (Liben-Nowell, Kleinberg, 2003) entre un nœud u et un nœud v qui est utilisé dans le contexte de prédiction de liens. Nous comparons notre score TR avec ce score dans la section 5.

La pertinence $\omega_p(t)$ d'un chemin $p = u \rightsquigarrow v$ pour un topic t qui apparait dans l'équation (1) prend en considération à la fois l'*autorité des comptes* et la *pertinence des arrêtes* sur le chemin p pour les topics de la requête Q. Nous définissons ces concepts dans la suite.

3.2.1. L'autorité d'un compte

Nous définissions une fonction d'autorité $auth(u, t)$ pour chaque compte u sur un *topic* t. Cette fonction dépend du nombre d'utilisateurs qui suivent u sur le *topic* t. Le score d'autorité est composé de deux parties : *(i)* un *score d'autorité locale* qui assigne une valeur plus élevée aux utilisateurs u qui sont « spécialisés » sur un *topic* t par rapport aux utilisateurs qui publient sur une grande variété de *topics* et *(ii)* une *popularité globale* qui permet d'attribuer un meilleur score aux utilisateurs les plus suivis sur t dans tout le graphe social. La combinaison d'une composante locale et d'une composante globale est une approche classique traditionnellement adoptée pour l'évaluation de l'importance d'un document (ex. le score de TF/IDF) mais aussi l'importance d'un nœud en termes d'autorité dans un graphe de pages Web (Jeh, Widom, 2003) ou plus récemment dans le monde du micro-blogging (Gupta *et al.*, 2013).

DÉFINITION 2 (Score d'autorité d'un compte). — *Le score d'autorité d'un compte $auth(u, t)$ s'exprime comme suit :*

$$auth(u, t) = \underbrace{\frac{|\Gamma^{u^-}[t]|}{|\Gamma^{u^-}|}}_{local} \times \underbrace{\frac{\log(1 + \Gamma^{u^-}[t])}{\log(1 + max_v(\Gamma^{v^-}[t]))}}_{global}$$

où $|\Gamma^{u^-}[t]|$ *est le nombre de followers de* u *sur le topic* t*, et* $|\Gamma^{u^-}|$ *est le nombre total de followers de* u*.*

La composante d'autorité locale est égale à 1 lorsque u est suivi exclusivement sur le *topic* t tandis que l'autorité globale peut être égale à 1 lorsque u est le compte le plus suivi sur le *topic* t dans tout le réseau. Si aucun utilisateur ne suit u sur t les deux composantes sont égales à zéro. Nous avons décidé d'introduire une fonction logarithmique afin de réduire l'écart entre les comptes populaires avec un nombre important de *followers* et les comptes qui ont très peu de *followers*. Les scores d'autorité pour t sont élevés pour les utilisateurs qui sont principalement suivis sur t et qui ont un nombre important de followers. Le fait de combiner une composante locale à une composante globale dans le calcul de l'autorité d'un utilisateur permet d'assigner une valeur élevée de score d'autorité sur un *topic* t à un compte qui a un grand nombre de *followers* principalement sur ce *topic* t. Cela permet aussi d'assigner la même importance à un compte généraliste et très suivi et à un compte très spécialisé mais avec un petit nombre de *followers*.

À noter que, mis à part pour $max_v(\Gamma^{v^-}[t])$, toutes les composantes du score peuvent êtres calculées "localement", en exploitant seulement les informations du nœud en question dans le graphe. Le calcul en temps réel des recommandations implique l'estimation de la valeur de $max_v(\Gamma^{v^-}[t])$ ce qui peut introduire un sur-coût car cette estimation peut entraîner la consultation de tout le graphe social. Cependant, cette valeur peut être stockée et recalculée périodiquement. En effet, malgré la dynamicité du graphe social (qui touche particulièrement les comptes populaires), l'application de la fonction logarithmique limite l'impact de telles variations ($\log(max_v(\Gamma^{v^-}[t])) \sim \log(max_v(\Gamma^{v^-}[t]) + \Delta)$. Par conséquent, nous pouvons évaluer la fonction d'autorité pendant l'exécution.

EXEMPLE 3 (Autorité globale et locale). — Le graphe exemple illustré en figure 1 contient un aperçu des *tweets* publiés par les utilisateurs B et C ainsi que les *topics* sur lesquels ils publient. L'utilisateur B est plus pertinent sur le *topic* `technology` que l'utilisateur C. En effet, B et C ont la même autorité globale avec deux *followers* sur ce *topic* pour les deux comptes. Cependant, le score d'autorité locale de B sur `technology` est plus élevé que celui de C du fait que sur les 3 arcs entrant de B, 2 sont étiqueté avec `technology` tandis que pour C ce ratio n'est que de 2 sur 6. Pour le *topic* `bigdata`, l'autorité locale de B et de C est similaire (1 sur 3 pour B et 2 sur 6 pour C) mais il apparaît que C est plus suivi sur `bigdata` (avec 2 comptes contre 1 pour B). Dès lors, le score d'autorité totale de C sur le *topic* `bigdata` est plus important. □

3.2.2. La pertinence des arêtes

Un chemin $p = u \rightsquigarrow v$ dans le graphe est considéré pertinent par rapport à un *topic* t lorsque les *topics* présents sur les arêtes constituant p sont sémantiquement proches de t. Cela peut être vu comme une mesure de la "conductivité" du chemin p pour le topic t. En s'appuyant sur notre hypothèse de proximité, nous considérons qu'une arête distante sur le chemin contribue moins au score de recommandation qu'une arête

proche de u. Cet aspect est contrôlé par un facteur de décroissance $\alpha \in [0, 1]$ qui réduit l'influence d'une arête $edge$ lorsque sa distance d à partir de u augmente. Plus précisément, nous définissons la pertinence d'une arête $edge$ à la distance d de u sur un chemin p comme suit :

$$\varepsilon_{edge}(t) = \alpha^d \times max_{t' \in l_e(edge)}(sim(t', t)) \quad (3)$$

où $l_e(edge)$ est une fonction d'étiquetage retourne les *topics* qui étiquettent une arête $edge$ et $sim : \mathcal{T}^2 \rightarrow \mathbb{R}$ évalue la similarité sémantique entre deux *topics* t et t'. Dans notre implantation nous utilisons la mesure de *Wu and Palmer* (Wu, Palmer, 1994) sur l'ontologie WORDNET [2] mais d'autres mesures peuvent tout aussi bien être employées (ex. Resnik, Disco [3] , etc.). Le choix de la meilleure mesure de similarité dépasse le cadre de notre approche de recommandation. Lorsqu'une arête $edge$ est étiquetée avec plusieurs *topics*, nous ne gardons que le *topic* avec le plus grand score de similarité sémantique afin d'éviter de donner un score trop important aux arêtes étiquetées par un grand nombre de *topics*.

3.2.3. Le score d'un chemin pour un topic

Finalement, nous considérons que l'importance d'un chemin p est élevée lorsque la pertinence des nœuds ainsi que celle des arêtes sur ce chemin sont élevées :

$$\omega_p(t) = \sum_{edge \in p} \varepsilon_{edge}(t) \times auth(end(edge), t) \quad (4)$$

où $end(edge)$ retourne le nœud destination d'une arête dirigée $edge$. Le score de recommandation de v pour l'utilisateur u sur un *topic* t est alors obtenu en substituant $\omega_p(t)$ dans l'équation (1) par sa formule donnée par l'équation (4). Le score de recommandation ainsi obtenu tient compte de la topologie du graphe (proximité et connectivité) avec les intérêts des followers (exprimés par des arrêtes étiquetés) et des scores d'autorité sur le topic t des utilisateurs sur le chemin p.

EXEMPLE 4 (Pertinence d'un chemin). — Dans l'exemple illustré en figure 1, nous voulons recommander à l'utilisateur A des comptes à suivre sur le *topic* `technology` en effectuant une exploration (pour cet exemple nous supposons une exploration dans un périmètre $k = 2$ sauts). Les utilisateurs D et E peuvent êtres atteints respectivement par les chemins $p_1 = A \rightarrow B \rightarrow D$ et $p_2 = A \rightarrow C \rightarrow E$, chacun de longueur 2. La pertinence de l'arête $A \xrightarrow{bigdata,technology} B$ est plus importante que celle de $C \xrightarrow{technology} E$ du fait que la première est à une distance 1 de A tandis que la deuxième est à distance 2. De plus, l'autorité sur le *topic* `technology` du nœud B calculé comme $(local) \times (global) = \frac{2}{3} \times \frac{log(1+2)}{log(1+2)}$ est plus importante

2. http://wordnet.princeton.edu/
3. http://www.linguatools.de/disco/disco_en.html

que l'autorité de C sur ce même *topic* qui est égale à $\frac{2}{6} \times \frac{log(1+2)}{log(1+2)}$. Plus globalement, la pertinence sémantique des arêtes sur le chemin p_1 par rapport à `technology` est plus élevée que celle des arêtes sur p_2 par conséquent le nœud D obtient un meilleur score de recommandation que le nœud E. □

3.3. Analyse du calcul des scores

Nous allons montrer dans la suite la formule itérative utilisée pour le calcul des scores ainsi que la propriété de composition de scores qui est utilisée dans la section 4.3.

3.3.1. Calcul itératif

Les scores de recommandation $\sigma(u, v, t)$ (équation (1)) sont calculés en utilisant la méthode de la puissance itérée (voir l'algorithme (1)). Le score $\sigma(u, u, t)$ est initialisé à 1 et pour tout $v \neq u$ le score $\sigma(u, v, t)$ est initialisé à 0. À chaque itération i, un nouveau score $\sigma(u, v, t)^{(i)}$ (qui considère tous les chemins de longueur $\leq i$ de u à v) est calculé en utilisant les scores $\sigma(u, v, t)^{(i-1)}$ des voisins $w \in \Gamma^{v^-}$ calculés à l'itération $(i - 1)$. La calcul est effectué jusqu'à la convergence. La formule itérative de calcul de scores est la suivante :

PROPOSITION 5 (Calcul itératif). —

$$\sigma^{(i)}(u, v, t) = \sum_{w \in \Gamma^{v^-}} (\beta.\sigma^{(i-1)}(u, w, t) + \\ + topo_{\alpha\beta}^{(i-1)}(u, w).\overline{\omega}_{w \to v}(t)) \quad (5)$$

où $topo_{\alpha\beta}^{i-1}(u, w)$ est le score topologique (voir l'équation (2)) avec un factor de décroissance de $\alpha.\beta$. Le score $\overline{\omega}_{w \to v}(t) = \beta.\alpha.max_{t' \in l_e(w \to v)}(sim(t', t)).auth(v, t))$ est celui d'un chemin qui contient seulement l'arrête $w \to v$ avec le topic t.

PREUVE 6. — Supposons un chemin p de longueur $k \leq i$ de u à v. Ce chemin est composé d'un chemin p_1 de longueur $k - 1$ de u au voisin w de v et d'une arrête e de w à v de longueur 1. En utilisant les Équations (3) et (4), le score $\overline{\omega}_p(t)$ est calculé comme étant:$\overline{\omega}_p(t) = \beta.\overline{\omega}_{p1}(t) + \beta^{|p_1|}.\alpha^{|p_1|}.\overline{\omega}_e(t) = \beta.\overline{\omega}_{p1}(t) + \beta^{k-1}.\alpha^{k-1}.(\beta.\alpha.\max_{t' \in l_e(edge)}(sim(t', t)).auth(v, t))$. Le score $\overline{\omega}_{p1}(t)$ correspond à un chemin qui se finit à w.

On peut réorganiser les chemins de l'équation (1) en regroupant ceux qui passent par le même voisin w de v: $\sigma(u, v, t) = \sum_{p \in P_{u,v}} \overline{\omega}_p(t) = \sum_{w \in \Gamma^{v^-}(t)} (\sum_{p \in P_{u,v}, w \in p} \overline{\omega}_p(t))$. En replaçant la formule de $\overline{\omega}_p(t)$ dans cette équation on obtient la formule du calcul itératif. ■

3.3.2. Composition des scores

Nous pouvons déduire le score total $\overline{\omega}_p(t)$ pour le topic t d'un chemin p de u à v à partir des scores d'autres chemins faisant partie de p qui ont été déjà calculés, en utilisant la propriété suivante :

PROPOSITION 7 (Composition des scores de recommandation). — *Supposons un chemin $p = p_1.p_2$ ainsi que les scores $\overline{\omega}_{p_1}(t)$ et $\overline{\omega}_{p_2}(t)$ qui représentent les scores déjà calculés correspondants aux chemins p_1 et p_2 pour un topic t. Le score total correspondant à p est calculé comme suit :*

$$\overline{\omega}_p(t) = \beta^{|p_2|}.\overline{\omega}_{p_1}(t) + \beta^{|p_1|}.\alpha^{|p_1|}.\overline{\omega}_{p_2}(t)$$

PREUVE 8. — En utilisant la formule du calcul récursif on peut prouver par induction la proposition pour des chemins plus longs. ■

3.3.3. Convergence du calcul itératif

Afin de montrer la convergence du calcul itératif des scores de recommandation $\sigma(u, v, t)$ (équation (5)), on peut exprimer ce calcul sous forme matricielle comme suit :

$$R_t^{(k+1)} = (\beta A)R_t^{(k)} + (\beta\alpha)S_t T_{\alpha\beta}^{(k)} \tag{6}$$

où $R_t^{(k)}$ est le vecteur de recommandation pour le topic t calculé à l'itération k ($R_t^{(k)}[v]$ est le score de recommandation $\sigma(u, v, t)$ calculé à l'itération (k)). A est la matrice d'adjacence du graphe ($A[v][u] = 1$ si u suit v). S_t est la matrice de similarité de d'autorité ($S_t[v][u] = sim(max_{t' \in l_e(u \to v)}(t', t)) \times auth(v, t)$). Le vecteur $T_{\alpha\beta}^{(k)}$ est le vecteur topologique calculé à l'itération k ($T_{\alpha\beta}^{(k)}[v]$ est le score topologique $topo_{\alpha\beta}^{(k)}(u, v)$). Il peut être exprimé de la manière suivante :

$$T_{\alpha\beta}^{(k+1)} = \alpha\beta A T_{\beta}^{(k)} + I$$

où $I[u] = 1$ et $I[u] = 0$ pour tous les utilisateurs $u \neq v$.

On peut déduire que la convergence du calcul est obtenue avec la condition suivante :

PROPOSITION 9 (Convergence du calcul de scores). — *Si $\beta < 1/\sigma_{max}(A)$, où $\sigma_{max}(A)$ est la plus grande valeur propre de A, alors le calcul itératif des scores de recommandation converge.*

PREUVE 10. — En se basant sur la formule récursive qui définit le vecteur topologique, la matrice est définie par le développement en série

$$T_{\alpha\beta} = \sum_{i=1}^{\infty} \alpha\beta^i A^i = (I - \alpha\beta A)^{-1} - I$$

converge lorsque $I - \alpha\beta A$ est positive définie, donc $\alpha\beta < 1/\sigma_{max}(A)$. Considérons que $T_{\alpha\beta}$ converge à une itération k'. Pour chaque itération $k > k'$, nous obtenons

le calcul itératif $R^{(k+1)} = (\beta A)R^{(k)} + C$ où $C = (\alpha\beta)ST^{(\infty)}_{\alpha\beta}$ est une constante. La convergence pour $R^{(k+1)}$ est obtenue quand $R^{(\infty)} = (\beta A)R^{(\infty)} + C$ et quand $R^{(\infty)} = (I - \beta A)^{-1} \times C$. Cette condition peut être obtenue si $\beta < 1/\sigma_{max}(A)$. Sachant que $\beta > \alpha\beta$, la deuxième condition est suffisante pour obtenir la convergence. ■

4. Une estimation efficace des recommandations

4.1. Vue d'ensemble de l'approche basée sur les landmarks

Notre score de recommandation suppose l'exploration de *tous* les chemins à partir du nœud pour lequel les recommandations sont générées. Afin de générer des recommandations à la volée pendant l'exécution, nous proposons un algorithme basé sur une approche par *landmarks* en deux étapes :

(i) une étape de pré-traitement qui calcule les recommandations pour un ensemble de nœuds, pour chaque *topic* du vocabulaire des *topics*.

(ii) une approximation en temps-réel, au moment de la requête, du calcul des recommandations pour un nœud sur un *topic* donné en exploitant les données pré-calculées.

Dans notre contexte, la taille du graphe social rend difficile le calcul efficace de recommandations basées sur des propriétés topologiques. Nous pré-calculons donc, pour un échantillon de nœuds choisis dans le graphe (*landmarks*), un *top-k* de recommandations (k étant un paramètre pré-établi par le système) sur chacun des *topics* $t \in \mathcal{T}$. Puis, pendant l'exécution, afin de déterminer les recommandations pour un compte n, nous explorons le graphe autour de n jusqu'à une profondeur $k < K$, où K est la profondeur d'exploration pour un calcul à la convergence. Après avoir récolté toutes les recommandations des différents *landmarks* rencontrés pendant l'exploration, celles-ci sont pondérées en se basant sur notre score de recommandation entre n et les *landmarks*. Une fusion des différents résultats obtenus permet d'obtenir les recommandations finales pour n.

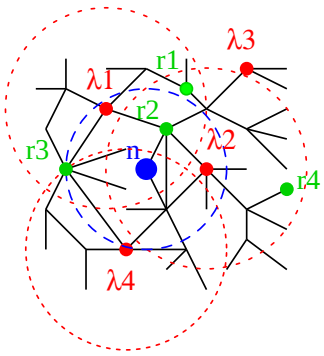

Figure 2. Exemple de recommandation par landmarks

La figure 2 illustre notre approche, n est le nœud pour lequel nous calculons les recommandations, λ_1, λ_2, λ_3 et λ_4 sont des *landmarks*. En explorant le graphe en effectuant un *BFS* (*Breadth-First-Search*[4], représenté par la ligne bleue en pointillés) à partir de n, seulement λ_1, λ_2 et λ_4 ont été rencontrés. Si l'on observe les recommandation générées, nous remarquons que r_1 n'a pas été rencontré durant l'exploration. Son score est le résultat de la combinaison des scores obtenus à travers les chemins $n - \lambda_1 - r_1$ et $n - \lambda_2 - r_1$. Le score de r_2 est quant à lui le résultat de la combinaison des scores $n - \lambda_1 - r_1$ et $n - \lambda_2 - r_1$ avec le score du chemin à distance 2 pour $n - \lambda_2$ car il a été croisé durant le *BFS*. On remarque aussi que la recommandation r_4 a été générée uniquement grâce au *landmark* λ_2.

À noter que cette approche estime une borne inférieure des différents scores de recommandation proposés. Généralement, les approches basées sur les *landmarks* dans un contexte de calcul de plus court chemin (voir section 2) sont utilisées pour l'estimation d'une borne supérieure en exploitant l'inégalité triangulaire. La raison est la suivante : lorsqu'un *landmark* λ recommande un compte n_2 à un compte n_1, le calcul considère l'ensemble P_1 de tous les chemins entre n_1 et λ de longueur inférieure à k_1 ainsi que l'ensemble P_2 de tous les chemins existants entre λ et n_2 de longueur inférieure à k_2. Cependant l'ensemble $P_{1,2} = \{p_i.p_j | p_i \in P_1 \wedge p_j \in P_2\}$ est inclus dans l'ensemble de tous les chemins existants entre n_1 et n_2 de longueur inférieure à $k_1 + k_2$.

4.2. Pré-traitement

Dans l'étape de pré-traitement, nous supposons un sous-ensemble $\mathcal{L} \subset N$ de nœuds, appelés *landmarks*, tel que $|\mathcal{L}| \ll |N|$. Différentes stratégies de sélection existent pour déterminer l'ensemble $\mathcal{L}$. Par exemple, les stratégies utilisant des *landmarks* dans un contexte de calcul de plus court chemin se basent essentiellement sur des propriétés de « centralité » des nœuds (centralité d'intermédiarité ou de proximité) afin de choisir les *landmarks*. Les caractéristiques de notre graphe social dirigé ainsi que le double rôle des utilisateurs (*publisher, follower*) nous permettent d'envisager d'autres stratégies de sélection basées sur la topologie comme par exemple choisir les nœuds ayant le plus fort degré entrant (comptes les plus populaires sur le réseau) ou bien les nœuds qui suivent le plus grand nombre de comptes (les lecteurs les plus actifs). On peut aussi mettre en place des contraintes pour maximiser la couverture comme par exemple obliger les *landmarks* choisis à respecter une distance minimale entre eux. Le choix de la stratégie de sélection de *landmarks* impacte la qualité ainsi que les performances globales du système de recommandation. Nous avons testé et comparé un ensemble de stratégies expérimentalement en section 5.

Pour chaque *landmark* $\lambda \in \mathcal{L}$, nous effectuons une exploration *BFS* sans éviter les cycles. Nous appelons l'ensemble des nœuds atteints à la profondeur k à partir

4. http://fr.wikipedia.org/wiki/Algorithme_de_parcours_en_largeur

de λ le *k-voisinage* de λ noté $\Upsilon_k(\lambda)$. $\Upsilon_\infty(\lambda)$ désigne l'ensemble de tous les nœuds atteignables à partir de λ. Nous calculons pour chaque nœud $u \in \Upsilon_\infty(\lambda)$ un vecteur de recommandation R_t tel que $R_t[u] = \sigma(\lambda, u, t)$ ainsi qu'un score topologique $topo_\beta(\lambda, u)$. Le score $\sigma(\lambda, u, t)$ est le score du *landmark* λ pour le compte u sur le topic t, et $topo_\beta(\lambda, u)$ représente le score de *topologique* présenté dans en section 3.2 avec un facteur de décroissance β, utilisé par la suite afin d'estimer le score de recommandation final.

Notre algorithme effectue une "fusion" des différents scores R_t pour tout $u \in \Upsilon_\infty(\lambda)$ afin d'obtenir une *posting list* pour laquelle chaque entrée qui correspond à un terme t est associé une liste de nœuds recommandés, classés par pertinence en fonction de leur score de recommandation $\sigma(\lambda, u, t)$. Nous gardons dans ces listes uniquement un top-k des recommandations calculées.

L'algorithme 1 effectue le calcul des scores de recommandation (nous avons omis les différentes initialisations à 0 par soucis de simplification). Les paramètres utilisés sont le facteur de décroissance choisi pour la fonction topologique ainsi que la profondeur maximale pour notre exploration *BFS*. Cette valeur est utilisée par l'étape de *BFS* pour le nœud qui pose la requête de recommandation (voir algorithme 2). Pour

Algorithme 1 : LANDMARK_RECOMM(λ, max_k, $\mathcal{T}$, β, n)

Require: landmark (λ), maximum exploration (max_k), set of topics ($\mathcal{T}$), topological decay factor (β), number of results to keep (n)
Ensure: a set of recommendation list R_λ, a topological vector $topo_\beta(\lambda)$

1: $\Upsilon_0 := \lambda, k := 0$
2: **while** $k < max_k$ **and** $converged = false$ **do**
3: $\quad\Upsilon_{k+1} := \emptyset$
4: $\quad$**for all** $u \in \Upsilon_k$ **do**
5: $\quad\quad\Upsilon_{k+1} := \Upsilon_{k+1} \cup \Gamma^u$
6: $\quad\quad$**for all** $v \in \Gamma^u$ **do**
7: $\quad\quad\quad$**for all** $t \in \mathcal{T}$ **do**
8: $\quad\quad\quad\quad\sigma^{(k+1)}(\lambda, v, t) += \beta \times \sigma^{(k)}(\lambda, u, t) + topo^k_{\alpha\beta}(\lambda, u) \times \overline{\omega}_{u \to v}$
9: $\quad\quad\quad$**end for**
10: $\quad\quad\quad topo^{(k+1)}_\beta(\lambda, v) += \beta \times topo^k_\beta(\lambda, u)$
11: $\quad\quad$**end for**
12: $\quad\quad R_t[u] += \sigma^{(k)}(\lambda, u, t)$
13: $\quad\quad topo_\beta(\lambda, u) += topo^k_\beta(\lambda, u)$
14: $\quad$**end for**
15: $\quad$**if** $(\sum_{u \in \Upsilon_k} \sigma^k(\lambda, u, t))/|R_t| < tol,\ \forall t \in \mathcal{T}$ **then**
16: $\quad\quad converged := true$
17: $\quad$**end if**
18: $\quad k := k + 1$
19: **end while**
20: **return** for all $t \in \mathcal{T}$ top-n($(R_t), topo_\beta(\lambda)$)

un *landmark* donné, cet algorithme est appelé avec en paramètre l'ensemble de *topics* $\mathcal{T}$. Les itérations en ligne 2 permettent d'explorer le k-voisinage de λ. À chaque ité-

ration, nous ajoutons au k-voisinage les nœuds atteignables avec un saut additionnel (l. 3-4). Pour tous les nœuds atteints à cette étape (l. 5), nous calculons le score de recommandation pour chacun des termes du vocabulaire des *topics* (si ce nœud à déjà été rencontré, nous mettons à jour cette valeur) (l. 6-7) ainsi que le score topologique (l. 10). Par la suite, nous mettons à jour le score de recommandation pour u (l. 12) et son score topologique (l. 13). Un test de convergence est effectué en lignes 15-16. Lorsque tous les vecteurs finissent par converger, l'algorithme retourne les listes de recommandations pour chaque *topic* ainsi que le score topologique (l. 20).

4.3. Approximation rapide des recommandations

Nous présentons dans cette section l'algorithme réalisant l'estimation des scores de recommandation à l'arrivée d'une requête, pendant l'exécution, en se basant sur les calculs qui ont été effectués pour chaque *landmark* à l'étape de pré-traitement. Nous supposons que nous devons générer les recommandations de comptes à suivre pour un compte n sur un *topic* t.

Le traitement commence par une exploration *BFS* à partir du nœud u similaire à celle effectuée dans le pré-traitement pour une profondeur d'exploration maximale max_k. L'objectif de ce BFS est triple : (i) l'exploration du voisinage de n et la découverte des *landmarks* les plus proches, (ii) le calcul des scores de chemins pour le *topic* t pour les chemins existants entre n et chaque *landmark* découvert et (iii) la mise à jour des scores de recommandation stockés par les *landmarks* en tenant compte des chemins entre n et les *landmarks*.

Afin de définir notre score, nous supposons $\Lambda \subseteq \mathcal{L}$ l'ensemble des *landmarks* trouvés par le *BFS*. Pour chaque λ les top-n comptes recommandés v avec leurs scores de recommandation $\sigma(\lambda, v, t)$ et leurs scores topologiques $topo_\beta(\lambda, v)$ sont déjà calculés pendant l'étape de pré-traitement. Le score de recommandation approché d'un noeud v pour un utilisateur u est une agrégation des scores de v calculés par tous les landmarks $\lambda \in \Lambda$:

DÉFINITION 11 (score de recommandation approché). — *Le score de recommandation approché d'un noeud v pour un noeud u sur un topic t par rapport à un ensemble de landmarks Λ est défini comme étant :*

$$\widetilde{\sigma}(u, \Lambda, v, t) = \sum_{\lambda \in \Lambda} \widetilde{\sigma}(u, \lambda, v, t)$$

où le score $\widetilde{\sigma}(u, \lambda, v, t)$ est le score de v qui prend en considération l'ensemble de chemins $P_{u,\lambda,v}$ de u à v qui passent à travers le landmark λ.

Le score $\widetilde{\sigma}(u, \lambda, v, t)$ est calculé par la composition des scores $\sigma(u, \lambda, t)$ et $topo_{\beta\alpha}(u, \lambda)$ obtenus pendant la phase d'exploration avec les scores $\sigma(\lambda, v, t)$ et $topo_\beta(\lambda, v)$ qui sont stockés dans les listes triées de λ.

PROPOSITION 12 (Calcul des scores approchés). — *Le score de recommandation de v pour u via le landmark λ est calculé comme suit :*

$$\widetilde{\sigma}(u,\lambda,v,t) = \sigma(u,\lambda,t) \times topo_{\beta}(\lambda,v) + topo_{\beta\alpha}(u,\lambda) \times \sigma(\lambda,v,t)$$

PREUVE 13. — Un chemin p dans $P_{u,\lambda,v}$ est composé des chemins p_1 et p_2, où $p_1 \in P_{u,\lambda}$ et $p_2 \in P_{\lambda,v}$. Chaque chemin $p = p_1.p_2$ avec $p_1 \in P_{u,\lambda}$ et $p_2 \in P_{\lambda,v}$ est un chemin dans $P_{u,\lambda,v}$. Par conséquence, basé sur la proposition (7) on obtient :

$$\begin{aligned}
\widetilde{\sigma}(u,\lambda,v,t) &= \sum_{p \in P_{u,\lambda,v}} \overline{\omega_p}(t) \\
&= \sum_{p_1 \in P_{u,\lambda}} \sum_{p_2 \in P_{\lambda,v}} \beta^{|p_2|}.\overline{\omega}_{p_1}(t) + \beta^{|p_1|}.\alpha^{|p_1|}.\overline{\omega}_{p_2}(t) \\
&= \sum_{p_1 \in P_{u,\lambda}} \overline{\omega}_{p_1}(t). \sum_{p_2 \in P_{\lambda,v}} \beta^{|p_2|} + \sum_{p_1 \in P_{u,\lambda}} (\beta.\alpha)^{|p_1|}. \sum_{p_2 \in P_{\lambda,v}} \overline{\omega}_{p_2}(t) \\
&= \sigma(u,\lambda,t).topo_{\beta}(\lambda,v) + topo_{\beta.\alpha}(u,\lambda).\sigma(\lambda,v,t)
\end{aligned}$$

■

À noter que notre approche calcule une borne inférieure des scores de recommandation tandis que les approches existantes basées sur les landmarks qui sont utilisées pour le calcul des plus courts chemins calculent des bornes supérieures en exploitant l'inégalité triangulaire. Dans notre contexte les scores approchés ne prennent pas en considération tous les chemins de u à v, mais seulement un sous-ensemble de chemins $P_{u,\lambda,v}$ qui passent par λ. Nos expériences montrent que cette approximation permet de calculer des scores de recommandations proches des scores obtenus par un calcul exact. Le calcul des scores de recommandation approchés pour un noeud u et un topic t

Algorithme 2 : APPROX_RECOMM(u,k,t,β,α)

Require: a node u, a max. depth k, a topic t, the decay factor for path β and for edge α.
Ensure: an ordered list of recommendations $\widetilde{R_t}$ for n

1: $(R_t, topo_{\beta.\alpha}(u)) \leftarrow$LANDMARK_RECOMM$(u,k,t,\beta.\alpha)$
2: **for all** $v \in R_t$ **do**
3: **if** $v \in \mathcal{L}$ **then**
4: **for all** w recommended by v **do**
5: $\widetilde{R_t}[w] += \sigma(u,v,t).topo_{\beta}(v,w) + topo_{\beta.\alpha}(u,v).\sigma(v,w,t)$
6: **end for**
7: **end if**
8: **end for**
9: **return** $\widetilde{R_t}$

est réalisé par l'algorithme 2. Il appelle d'abord l'algorithme LANDMARK_RECOMM pour calculer des scores de recommandation et les scores topologiques pour tous les

noeuds situés à une distance k de u (l. 1). À la différence de l'étape de pré-traitement, l'algorithme n'est pas exécuté jusqu'à la convergence, l'exploration se fait à une distance k (2 dans nos expériences). Les scores de recommandation sont calculés pour un topic t. Le facteur de décroissance est ici $\beta.\alpha$. Pour chaque landmark λ rencontré pendant l'exploration (l. 2-3) l'algorithme combine ses scores de recommandation pour le topic t avec le score de recommandation calculé par u pour λ d'après la Proposition 12 (l. 5).

5. Expérimentations

Dans cette section, nous décrivons les expérimentations que nous avons conduites afin de valider notre approche de recommandation sur le jeu de données issu du réseau *Twitter*.

À partir du jeu de données fourni par (Ahn *et al*., 2007) nous avons extrait un sous-graphe respectant les tendances topologiques de Twitter composé d'utilisateurs principalement anglophones. Ce sous-graphe contient 2 182 867 utilisateurs et 125 451 980 arcs. Pour chacun de ses utilisateurs nous avons utilisé l'API de Twitter [5] en mars 2015 afin de récupérer leurs tweets. Nous avons obtenu plus de 3 milliards de tweets, soit environ 1500 tweets par utilisateur en moyenne. Afin de générer les topics d'intérêt de chaque utilisateur, nous avons concaténé ses tweets pour former un unique document que nous avons envoyé à OpenCalais afin de récupérer les topics. Cette méthode nous a permis de catégoriser 10 % des utilisateurs (contraintes temporelles, l'utilisation de l'API est assez lente). Avec ces utilisateurs catégorisés nous avons utilisé Mulan [6] afin de générer un modèle SVM (Support Vector Machine) qui nous a per- mis de catégoriser avec une précision de 90 % les topics des autres utilisateurs. Enfin, nous avons construit le graphe social étiqueté en choisissant comme étiquette sur les arcs l'intersection des topics des utilisateurs qui composent l'arc. La distribution des étiquettes attribuées dans le graphe est visible figure 3.

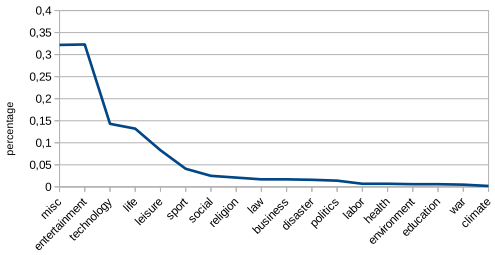

Figure 3. Distribution des topics dans notre jeu Twitter

5. http://dev.twitter.com/rest/public
6. http://mulan.sourceforge.net

5.1. Qualité des recommandations

Afin d'estimer la qualité des recommandations générées par notre système, nous adoptons la méthodologie décrite par (Cremonesi *et al.*, 2010) :

$i)$ nous construisons un ensemble candidat de T comptes en retirant des arêtes du graphe d'après une stratégie de sélection d'arêtes ;

$ii)$ pour chaque compte i de l'ensemble candidat sont choisis aléatoirement 1000 comptes du graphe social ;

$iii)$ le système calcule les scores de recommandations des 1001 comptes (1000 comptes choisis + compte de l'ensemble candidat) et génère une liste triée L ;

$iv)$ lorsque le compte de l'ensemble candidat appartient au top-N de L nous comptabilisons un succès (*hit*), sinon un échec.

$v)$ l'itération sur l'ensemble-candidat se poursuit.

La mesure de précision (resp. rappel) que nous utilisons est donnée par la formule $\#hits/N.T$ (resp. $\#hits/T$) (Cremonesi *et al.*, 2010). Pour nos expérimentations, la taille de l'ensemble candidat a été initialisée à $T = 100$ et les valeurs considérées sont les moyennes calculées sur 10 essais successifs. Nous comparons la qualité des recommandations générées avec les résultats produits par deux autres approches : la mesure standard de *Katz* (Liben-Nowell, Kleinberg, 2003) qui est strictement topologique, et TwitterRank (Weng *et al.*, 2010) qui considère, en plus de la topologie, une similarité entre les comptes sur des *topics* donnés. La valeur choisie pour le facteur de décroissance β pour *Katz* et notre proposition TR est de $0,0005$. Il a été reporté par (Liben-Nowell, Kleinberg, 2003) que cette valeur permet d'obtenir de bons résultats sur les graphes issus du Web. La valeur des coefficients α et β pour TwitterRank a été initialisée à $0,85$ comme indiqué par (Weng *et al.*, 2010). Les résultats présentés ont été obtenus à la convergence.

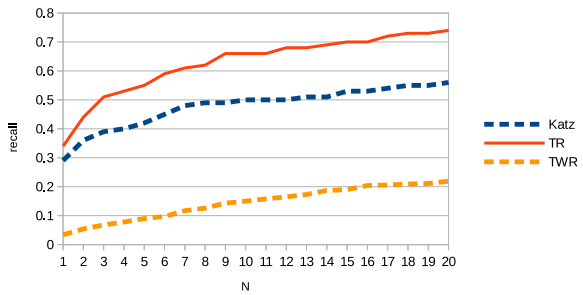

Figure 4. Rappel à N

La figure 4 illustre la performance des trois systèmes de recommandation pour des arêtes choisies aléatoirement lors de la construction des ensembles-candidats. Nous pouvons observer que TwitterRank est distancé par les deux autres algorithmes. En effet, seulement 4 % des recommandations générées avec TwitterRank apparaissent

dans le top-1. Ce taux est de 13 % pour *Katz* et 30 % pour TR. Il apparaît donc que notre approche apporte un gain de respectivement $7,5$ et $2,3$ par rapport à *Katz* et TwitterRank sur le top-1. Sur un top-10, ce gain demeure significatif avec des gains respectifs de $4,1$ et de $1,2$ de TR par rapport à respectivement *Katz* et TwitterRank.

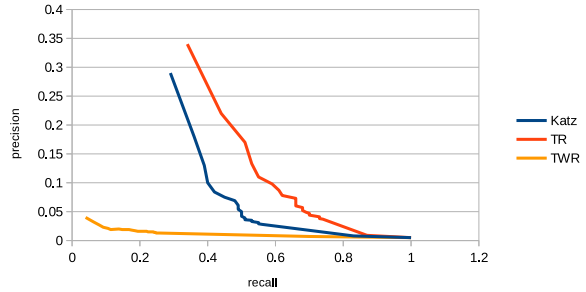

Figure 5. Précision VS rappel

La figure 5 confirme le fait que TR fournit de meilleures recommandations. En effet, pour une même valeur de rappel, nous observons que la précision de notre approche est au minimum le double de celle obtenue par *Katz* et de plus d'un ordre de grandeur plus élevée que celle obtenue par TwitterRank. Par exemple, pour une valeur de rappel de $0,3$, TR offre une précision de $0,3$ lorsque *Katz* est à $0,09$ et TwitterRank à $0,01$.

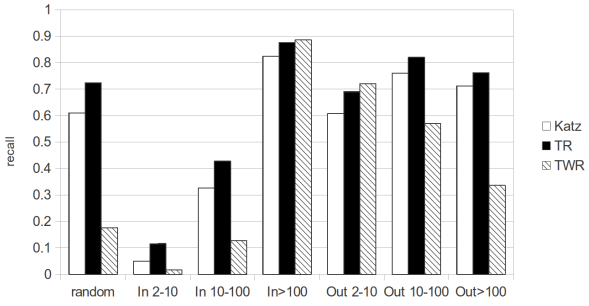

Figure 6. Rappel suivant la stratégie de suppression d'arêtes

Cependant, on observe la présence d'une grande disparité lorsque l'on considère les résultats sur les deux dimensions d'analyse qui sont : la stratégie de sélection des arcs à retirer et la popularité du *topic* sur lequel générer les recommandations. Sur un top-10, la figure 6 montre une faible valeur de rappel pour notre stratégie lorsqu'on génère des recommandations pour un compte qui a moins de 10 followers. Les comptes populaires (ayant une valeur de degré entrant supérieure à 100) sont la plupart du temps présents dans le top-10 des recommandations générées avec une valeur de rappel oscillant entre $0,8$ et $0,9$. Ce phénomène s'explique par l'approche topologique

adoptée qui est basée sur l'existence de chemins, les comptes ayant un grand nombre de chemins entrants voient ainsi leur score augmenter. Il apparaît aussi que TwitterRank obtient le meilleur résultat pour les comptes très populaires. En effet, la plupart des comptes populaires sont étiquetés par plusieurs *topics*. Tandis que TwitterRank se base sur la présence ou l'absence d'un *topic* sans prendre en compte le nombre d'étiquettes sur les arcs entrants, notre score s'appuie sur le nombre d'étiquettes afin de déterminer l'autorité d'un compte. Dans ce cas, la présence de plusieurs arcs entrants avec des *topic* différents entraîne une diminution du score d'autorité du compte pour un *topic* donné. Inversement, un compte avec moins de 10 *followers* a rarement des *topics* différents sur ses arcs entrants. Notre approche qui consiste à considérer les *topics* sur les chemins entre les comptes à recommander et leurs recommandations se montre alors particulièrement efficace. En ce qui concerne le degré sortant des comptes pour lesquels les arcs sont retirés, on observe que TwitterRank obtient un meilleur résultat avec les comptes ayant un faible degré sortant (suivant peu de comptes). La raison est que le fait de retirer un arc sur un compte déjà peu connecté au réseau peut entraîner la déconnexion (complète ou partielle) d'une partie du graphe. Comme Katz et TR se basent sur le nombre de chemins communs, cela peut avoir un impact important sur les scores finaux. Inversement, les comptes suivant plus 1000 autres comptes offrent un grand nombre de chemins alternatifs pour atteindre les comptes recommandés.

Étant donné la distribution biaisée des *topics* dans le graphe social (voir figure 3), nous avons décidé d'étudier l'impact de la popularité d'un *topic* sur les recommandations générées. Les résultats sont illustrés en figure 7 pour les *topics* `social`, `leisure` et `technology`. On observe d'abord que les meilleurs valeurs de rappel sont obtenues pour les *topics* les moins populaires. Par exemple, pour un *topic* peu fréquent comme `social`, nous obtenons une valeur de rappel pour les 10 premières recommandations (top-10) de $0,959$ pour TR, $0,751$ pour Katz et $0,253$ pour TwitterRank. Inversement, pour un *topic* populaire tel que `technology`, nous obtenons respectivement $0,462$, $0,424$ and $0,09$. En effet, pour un *topic* populaire, plusieurs comptes sont découverts dans un voisinage proche du compte pour lequel les recommandations sont générées avec potentiellement de meilleurs scores que les comptes des arcs retirés, ce qui explique le fait que la valeur absolue du rappel baisse. Par ailleurs, nous observons que notre approche, TR, surpasse les approches concurrentes dans les trois cas de figure (*topic* populaire, à popularité moyenne, et peu populaire).

5.2. Validation par les utilisateurs

Afin d'évaluer la pertinence des recommandations générées, nous avons conduit une tâche de validation par 54 utilisateurs (étudiants, doctorants et membres du laboratoire) parmi lesquels 46 % se définissent comme utilisateurs de plates-formes sociales telles que *Twitter*. Nous avons mis en place un sondage en ligne où les utilisateurs doivent évaluer la pertinence d'un ensemble de recommandations pour un *topic* donné sur une échelle de 1 (faible pertinence) à 5 (forte pertinence). Un ensemble de recommandations à évaluer est constitué par le top-3 des recommandations générées respectivement par *Katz*, TwitterRank ainsi que TR. Nous avons donc 9 comptes re-

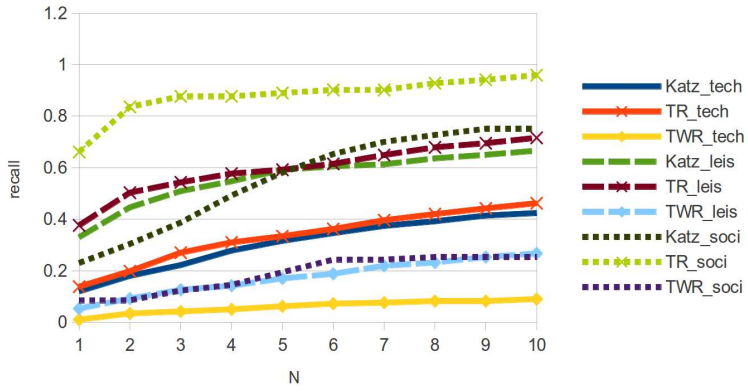

Figure 7. Rappel suivant la popularité des topics

commandés pour les *topic* `Technology`, `Social` et `Leisure`. Sur l'interface du sondage en ligne, la liste des recommandations est triée aléatoirement, et pour chaque recommandation est affichée une liste de 5 publications (*tweets*), choisies aléatoirement parmi les publications du compte recommandé. La figure 8 illustre un aperçu de l'interface de notre système d'évaluation.

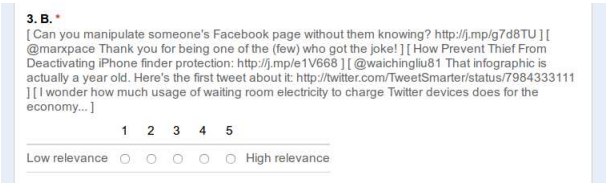

Figure 8. Interface du système d'évaluation pour le topic `Technology`

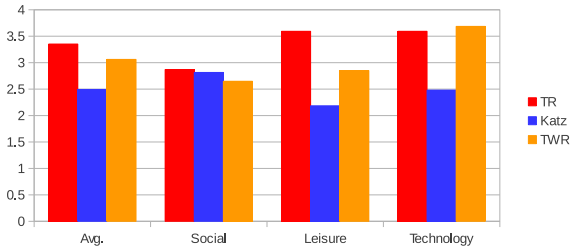

Figure 9. Scores de pertinence obtenus par la validation par les utilisateurs

La figure 9 présente les résultats de notre sondage. Ces résultats montrent que TR et TwitterRank fournissent les recommandations les plus pertinentes. Cependant, selon la popularité du *topic* pour lequel les recommandations sont générées, des différences apparaissent. Par exemple, on observe que le *topic* `social` offre des résultats

homogènes avec des scores oscillant entre $2,7$ pour TwitterRank, $2,8$ pour Katz et $2,9$ pour TR. La raison est que la définition de ce topic est vaste. En effet les comptes catégorisés `social` sont souvent des comptes généralistes qui publient sur plusieurs sujets tandis que les *topics* `leisure` or `technology` sont moins ambigus. Pour ces *topics*, on observe que TR et TwitterRank surpassent *Katz*, ce qui s'explique par l'avantage que ces approches ont en considérant le contenu publié en plus de la topologie seule. Ainsi TR obtient un meilleur score de pertinence sur les *topics* moyennement populaires comme `leisure` grâce à sa capacité à recommander des comptes peu suivis mais spécialisés sur un sujet. Par contre, sur un sujet très populaire comme `technology`, TwitterRank obtient un score légèrement meilleur.

5.3. Approximation du calcul de recommandations

Nous avons conduit une série d'expériences afin d'illustrer les bénéfices de notre approche basée sur les *landmarks* pour le calcul approché des scores de recommandation. Étant donné le fait que les résultats dépendent grandement du choix des *landmarks*, comme le montre (Potamias *et al.*, 2009), nous avons décidé d'implanter et de comparer les recommandations en se basant sur 11 stratégies de sélection de *landmarks* différentes (présentées dans le tableau 1).

Tableau 1. Algorithmes de sélection de Landmarks

Algorithme	Description
RANDOM	Choix des *landmarks* avec une distribution uniforme
FOLLOW	Sélection des *landmarks* grâce à une probabilité dépendant de leur # de followers
PUBLISH	Sélection des *landmarks* grâce à une probabilité dépendant du # de comptes qu'ils suivent
IN-DEG	*Landmarks* choisis parmi les nœuds ayant le plus grand degré entrant
BTW-FOL	Sélection des *landmarks* avec # de *followers* entre [min_follow, max_follow]
OUT-DEG	*Landmarks* choisis parmi les nœuds ayant le plus grand degré sortant
BTW-PUB	Sélection des *landmarks* avec # de comptes suivis entre [min_publish, max_publish]
CENTRAL	Les *landmarks* sont les nœuds atteignables à une certaine distance à partir d'un ensemble de nœuds "*seeds*"
OUT-CEN	Les *landmarks* sont les nœuds qui atteignent le plus de nœuds d'un ensemble de "*seeds*"
COMBINE	Combinaison pondérée de CENTRAL et OUT-CEN
COMBINE2	Combinaison pondérée de BTW-FOL et BTW-PUB

5.3.1. Construction des index de landmarks

Cette première expérience met en avant l'écart important qui peut exister dans la phase de sélection des *landmarks*. Le tableau 2 illustre les temps moyen de sélection et de calcul pour un *landmark* pour chacune des stratégies de sélection. Les stratégies aléatoires telles que RANDOM ou BTW-PUB sont naturellement les plus rapides ($\approx 1ms$ par *landmark*) tandis que les stratégies basées sur la mesure de centralité sont de 4 ordres de grandeur plus lentes avec des temps moyens variant entre 30 et $60s$ (en raison de la complexité de l'ordre de $O(N^2.\log N + NE)$ de la mesure de centralité en appliquant l'algorithme de (Johnson, 1977)). Le tableau 2 illustre aussi les temps nécessaires au calcul des tables de recommandation pour chaque *landmark*. Nous pouvons observer que ce calcul est indépendant de la stratégie de sélection.

Tableau 2. Temps moyen de sélection et de calcul pour un landmark

Stratégie	*Landmarks*	
	sélection. (ms)	calcul. (s)
RANDOM	0,6	27,0
FOLLOW	179,0	30,1
PUBLISH	172,4	29,9
IN-DEG	62,9	31,3
BTW-FOL	7,5	31,8
OUT-DEG	60,4	30,4
BTW-PUB	1,7	29,7
CENTRAL	30 603,0	29,1
OUT-CEN	33 193,1	27,8
COMBINE	65 223,1	27,3
COMBINE2	67 382,0	28,3

5.3.2. Comparaison des stratégies de sélection de landmarks

Nous évaluons la stratégie de sélection de *landmarks* présenté en section 4.3. Nous effectuons une exploration *BFS* à profondeur 2 depuis un nœud donné puis nous combinons les scores obtenus avec les scores des *landmarks* découverts en appliquant l'algorithme 2. Nous comparons ensuite les recommandations obtenues par le calcul approché avec celles calculées à la convergence. Les résultats moyens pour 100 *landmarks* sont reportés sur le tableau 3.

D'abord, nous observons que le nombre de *landmarks* rencontrés pendant l'exploration *BFS* à distance 2 varie d'une stratégie à l'autre allant de $7,3$ en moyenne pour les stratégies de type COMBINE à $25,4$ pour la stratégie OUT-DEG. Les mesures basées sur la centralité affichent le moins de *landmarks* trouvés. Ceci est dû au fait que ces mesures choisissent les *landmarks* parmi les nœuds qui forment des "ponts" entre des composantes connexes du graphe et donc plus difficilement atteignables par une exploration à profondeur 2.

Il apparaît aussi que le temps de traitement est inversement proportionnel au nombre de *landmarks* trouvés, ce qui peut sembler contre-intuitif étant donné que

Tableau 3. Comparaison des stratégies de sélection de landmarks

Stratégie	#lnd	Temps en ms (gain)	L10	L100	L1000
RANDOM	11,0	24 (225)	0,130	0,124	0,125
FOLLOW	17,8	9 (599)	0,377	0,140	0,096
PUBLISH	14,9	12 (449)	0,349	0,136	0,100
IN-DEG	25,0	6 (899)	0,523	0,149	0,066
BTW-FOL	20,8	25 (216)	0,061	0,059	0,058
OUT-DEG	25,4	7 (770)	0,518	0,147	0,064
BTW-PUB	12,7	24 (225)	0,129	0,127	0,123
CENTRAL	13,6	24 (225)	0,134	0,123	0,125
OUT-CEN	8,8	19 (284)	0,172	0,131	0,121
COMBINE	7,3	18 (300)	0,180	0,125	0,118
COMBINE2	10,5	22 (245)	0,129	0,126	0,124

plus de calculs (combinaisons de scores) sont exécutés s'il y a un nombre important de *landmarks*. L'explication réside dans le fait qu'un "*pruning*" est effectué à la rencontre d'un *landmark* pendant le *BFS* c.à.d que les chemins qui passent par un *landmark* ne sont pas considérés pour l'exploration. Le temps de calcul étant largement dominé par l'étape d'exploration, ce choix de *pruning* réduit considérablement le temps global de traitement. Nous observons que le calcul approché permet un gain allant de 2 à 3 ordres de grandeur en comparaison avec l'approche exacte.

Les 3 dernières colonnes du tableau 3 affichent les valeurs de distance de *Kendall Tau* entre les résultats obtenus par la méthode approchée et la méthode exacte (à la convergence) lorsque les *landmarks* stockent respectivement les top-10, top-100 et top-1000 recommandations. La distance de *Kendall Tau* [7] est une mesure utilisée afin de comparer deux listes classées. Plus cette valeur est importante, plus les classements comparés sont dissimilaires. Nous observons par exemple que les stratégies OUT-DEG et IN-DEG présentent des résultats similaires par rapport à la méthode exacte. En effet, il apparaît que les top-100 comptes recommandés par les deux stratégies sont très similaires. Le fait de garder le top-1000 recommandations permet d'atteindre des valeurs de distance de *Kendall Tau* oscillant entre $0,06$ et $0,13$. En effet, un classement de 1000 compte ne va pas être très impacté par une exploration à la convergence. Par contre, un top-10 donne des valeurs de *Kendall Tau* plus importantes car un classement sur 10 comptes seulement est fortement impacté par une exploration plus approfondie (à la convergence). Aussi, l'analyse de notre jeu montre que pour les *landmarks* ayant un fort degré sortant (la distribution de degrés révèle que ce sont aussi ceux ayant un fort degré entrant) les listes à top-10 varient grandement de l'approche exacte. La raison est que ces comptes sont capables d'atteindre très vite (à saut 1 ou 2) un très grand nombre de nœuds .

7. http://en.wikipedia.org/wiki/Kendall_tau_distancehttp://en.wikipedia.org/wiki/Kendall_tau_distance

Enfin, nous observons que, même en gardant les top-1000 recommandations pour chaque *topic*, l'espace occupé par l'index des recommandations d'un *landmark* n'est que de 1,4 Mo. Nous pouvons donc gérer les tables de *landmarks* en mémoire centrale.

6. Conclusion

Nous avons présenté dans cet article TR, notre score de recommandation qui intègre la topologie du graphe et les informations sémantiques concernant les intérêts des utilisateurs disponibles afin de recommander des comptes. Pour permettre le passage à l'échelle dans des larges graphes, nous avons proposé une approche basée sur les *landmarks* qui nécessite une étape de pré-traitement pour un ensemble réduit de nœuds du graphe. Cette approche permet d'atteindre un gain de 2 à 3 ordres de grandeur en comparaison avec l'approche exacte. Les expérimentations conduites ainsi que la validation par les utilisateurs ont permis de démontrer que TR surpasse les approches concurrentes.

Comme pistes d'évolution, nous envisageons d'abord l'étude de stratégies de mises à jour. En effet, les liens dans les réseaux sociaux peuvent avoir un cycle de vie très éphémère. La dynamicité du graphe impactant directement nos scores, nous avons donc besoin de mettre en place des stratégies de rafraîchissement adaptées. Par ailleurs nous avons choisi une approche centralisée, comme *Twitter* avec son système de recommandation de comptes à suivre (Gupta *et al*., 2013). Cependant, avec la croissance continue des tailles de graphe, des stratégies intelligentes de distribution doivent être élaborées. Dans notre approche, nous pouvons envisager la distribution du graphe en prenant en compte la connectivité des différentes composantes, mais aussi en combinant la distribution avec la sélection de *landmarks* ce qui permettrait d'effectuer les calculs de pré-traitement localement, minimisant ainsi les transferts sur le réseau.

Bibliographie

Ahn Y., Han S., Kwak H., Moon S. B., Jeong H. (2007). Analysis of topological characteristics of huge online social networking services. In *Proc. intl. world wide web conference (www)*, p. 835–844.

Budalakoti S., Bekkerman R. (2012). Bimodal Invitation-Navigation Fair Bets Model for Authority Identification in a Social Network. In *Proc. intl. world wide web conference (www)*, p. 709-718.

Chaoji V., Ranu S., Rastogi R., Bhatt R. (2012). Recommendations to Boost Content Spread in Social Networks. In *Proc. intl. world wide web conference (www)*, p. 529-538.

Chen K., Chen T., Zheng G., Jin O., Yao E., Yu Y. (2012). Collaborative personalized tweet recommendation. In *Proc. intl. conf. on research and development in information retrieval (sigir)*, p. 661-670.

Chin A., Xu B., Wang H. (2013). Who Should I Add as a 'Friend'?: a Study of Friend Recommendations Using Proximity and Homophily. In *MSM*, p. 7.

Cremonesi P., Koren Y., Turrin R. (2010). Performance of Recommender Algorithms on Top-n Recommendation Tasks. In *Proc. intl. conf. on recommender systems (recsys)*, p. 39-46.

Diaz-Aviles E., Drumond L., Gantner Z., Schmidt-Thieme L., Nejdl W. (2012). What is happening right now ... that interests me?: online topic discovery and recommendation in twitter. In *Proc. intl. conf. on information and knowledge management (cikm)*, p. 1592-1596.

Esparza S. G., O'Mahony M. P., Smyth B. (2012). Towards the Profiling of Twitter Users for Topic-Based Filtering. In *SGAI*, p. 273-286.

Gupta P., Goel A., Lin J., Sharma A., Wang D., Zadeh R. (2013). WTF: the Who to Follow Service at Twitter. In *Proc. intl. world wide web conference (www)*, p. 505-514.

Hannon J., Bennett M., Smyth B. (2010). Recommending twitter users to follow using content and collaborative filtering approaches. In *Proc. intl. conf. on recommender systems (recsys)*, p. 199-206.

Jeh G., Widom J. (2003). Scaling Personalized Web Search. In *Proc. intl. world wide web conference (www)*, p. 271-279.

Johnson D. B. (1977). Efficient algorithms for shortest paths in sparse networks. *J. ACM*, vol. 24, n° 1, p. 1–13.

Kapanipathi P., Orlandi F., Sheth A. P., Passant A. (2011). Personalized Filtering of the Twitter Stream. In *SPIM*, p. 6-13.

Koroleva K., Röhler A. B. (2012). Reducing Information Overload: Design and Evaluation of Filtering & Ranking Algorithms for Social Networking Sites. In *ECIS*, p. 12.

Kywe S. M., Hoang T.-A., Lim E.-P., Zhu F. (2012). On Recommending Hashtags in Twitter Networks. In *Proc. intl. conf. on social informatics (socinfo)*, p. 337-350.

Lempel R., Moran S. (2001). Salsa: The stochastic approach for link-structure analysis. *ACM Transactions on Information Systems*, vol. 19, n° 2, p. 131-160.

Liang H., Xu Y., Tjondronegoro D., Christen P. (2012). Time-aware topic recommendation based on micro-blogs. In *Proc. intl. conf. on information and knowledge management (cikm)*, p. 1657-1661.

Liben-Nowell D., Kleinberg J. (2003). The Link Prediction Problem for Social Networks. In *Proc. intl. conf. on information and knowledge management (cikm)*, p. 556–559.

Pennacchiotti M., Silvestri F., Vahabi H., Venturini R. (2012). Making your interests follow you on twitter. In *Proc. intl. conf. on information and knowledge management (cikm)*, p. 165-174.

Potamias M., Bonchi F., Castillo C., Gionis A. (2009). Fast shortest path distance estimation in large networks. In *Proc. intl. conf. on information and knowledge management (cikm)*, p. 867-876.

Ricci F., Rokach L., Shapira B., Kantor P. B. (2011). *Recommender systems handbook*. Springer.

Weng J., Lim E.-P., Jiang J., He Q. (2010). TwitterRank: Finding Topic-sensitive Influential Twitterers. In *Proc. intl. conf. on web search and web data mining (wsdm)*, p. 261-270.

Wu Z., Palmer M. (1994). Verbs Semantics and Lexical Selection. In *Acl*, p. 133–138.