Finding Users of Interest in Micro-blogging Systems

Camelia Constantin
Univ. Pierre et Marie Curie
2 Place Jussieu
F75005 Paris, France
camelia.constantin@lip6.fr

Ryadh Dahimene
CNAM
2 rue Conté
F75141 Paris, France
ryadh.dahimene@cnam.fr

Quentin Grossetti
CNAM
2 rue Conté
F75141 Paris, France
quentin.grossetti@cnam.fr

Cédric du Mouza
CNAM
2 rue Conté
F75141 Paris, France
dumouza@cnam.fr

ABSTRACT

Micro-blogging systems have become a prime source of information. However, due to their unprecedented success, they have to face an exponentially increasing amount of user-generated content. As a consequence finding users who publish quality content that matches precise interest is a real challenge for the average user. This paper presents a new recommendation score which takes advantage both of the social graph topology and of the existing contextual information to recommend users to follow according to user interest. Then we introduce a landmark-based algorithm which allows to scale. The experimental results and the user studies that we conducted confirm the relevance of this recommendation score against concurrent approaches as well as the scalability of the landmark-based algorithm.

1. INTRODUCTION

Micro-blogs have become a major trend over the Web 2.0 as well as an important communication vector. Twitter, the main micro-blogging service, has grown in a spectacular manner to reach more than 570 million users in April, 2014 in less than seven years of existence. Currently around 1 million new accounts are added to Twitter every week, while there were only 1,000 in 2008. 500 million tweets are sent every day and on average a Twitter user follows 108 accounts[1]. Facebook is another example with 1.26 billion users who publish on average 36 posts a month. A Facebook user follows on average 130 "friends" which results in 1,500 pieces of information a user is exposed on average when he logs in[1]. Other similar systems like Google+, Instagram, Youtube, Sina Weibo, Identi.ca or Plurk, to quote the largest, also exhibit dramatic growth.

[1]http://expandedramblings.com

This fast and unprecedented success has introduced several challenges for service providers as well as for their users. While the former have to face a tremendous flow of user generated content, the latter struggle to find relevant data that matches their interests: they usually have to spend a long time to read all the content received, trying to filter out relevant information. Two (complementary) strategies have emerged to help the user to find relevant data that matches his interest in the huge flow of user generated content: posts filtering like in [13, 14, 8] and posts/account searches and/or recommendation like in [7, 4, 5]. Social network systems usually offer the ability to search for posts or accounts that match a set of keywords. This could be a "local" search to filter out the posts received, or a "global" search to query the whole set of existing posts/accounts. For the latter search, there exist two options: some precomputed posts sets that correspond to the hot topics at query time, or customized searches where the query result is built according to the keywords specified by the user. However, the broad match semantics generally adopted by the searching tools is very limited. Even a ranking score based on the number of keywords is not sufficient to retrieve all posts of interest. Combined with the lack of semantics and the number of posts a day, the large number of searches performed every day also raises scalability issue. For instance in 2012, more searches were performed each month (24 billion) on Twitter than on Yahoo (9.4 billion) or on Bing (4.1 billion).

In this paper we consider the problem of discovering quality content publishers by providing efficient, topological and contextual user recommendations on the top of a micro-blog social graph. Micro-blogging systems are characterized by the existence of a large directed social graph where each user (accounts) can freely decide to connect to any other user for receiving all his posts. In this paper we make the assumption that a link between a user u and a user v expresses an interest of u for one or several topics from the content published by v. We consequently choose to model the underlying social network graph as a *labeled social graph*, where labels correspond to the topics of interest of the users. Our objective is to propose a recommendation score that captures both the topological proximity and connectivity of a publisher along with his authority regarding a given topic and the interest of the intermediary users between the one to be recommended and the publisher.

The size of the underlying social graph raises challenging issues especially when we consider operations that involve a graph exploration. In order to speed up the recommendation process we propose a fast approximate computation based on landmarks, *i.e.*, we select a set of nodes in the social graph, called *landmarks*, which will play the role of hubs and store data about their neighborhood. This set of landmarks is selected using different strategies we compare experimentally in Section 5.

Contributions. In this paper we propose a recommendation system that produces personalized user recommendations. Our main contributions are:

1) considering the idea that measures based on the graph topology are good indicators for the user similarity, we propose a topological score which integrates semantic information on users and their relationships;

2) furthermore, we introduce a landmark-based approach to improve recommendation computation time and to achieve a 2-3 order of magnitude gain compare to the exact computation;

3) an experimental validation of our approach, including a comparative study with other approaches (TwitterRank [26] and Katz [16]).

Observe that we illustrate our proposal in the context of micro-blogging systems, but our model is general and may be used for any social networks where users publish content and receive posts from the accounts they follow.

The paper is organized as follows: after the introduction in Section 1, we present the related work in Section 2. Section 3 describes our model and our recommendation scores along with their composition property. Then we propose our fast recommendation computation based on a landmark strategy in Section 4. Our experimental validation is presented in Section 5 while Section 6 concludes the paper and introduces future work.

2. RELATED WORK

Recommendation systems for social networks were recently proposed like [16, 4, 26, 11, 7, 21, 10, 3, 28, 24]. The work in [16] presents a comparison of different topological-based recommendation methods adapted in the context of link-prediction. In [2], authors propose to combine two ranking scores estimated with a fair bets approach on the user invitation graph and on the profile browsing graph. The work in [24] presents a user recommendation system which exploits node similarity scores estimated as a combination of local and global scores. Local score is based on the number of neighbors of the query node and of the recommendation node while global score involved the shortest path between them. Thus the recommendation scores for these approaches are only based on the topology and unlike our proposal do not consider neither content nor authority of the users. On the other side, the work in [20] finds users with high topical authority scores in micro-blogging systems. Unlike our method, those scores are not personalized, the system computes global authority scores, which in our case are used as parameters of the recommendation scores computed for some user.

Other approaches, like [4, 11], consider collaborative filtering. The work in [4] introduces a collaborative tweet ranking based on preference information of the users, authority of the publisher and the quality of the content. Recommendations are produced at a tweet level. Hannon et al. [11] evaluate a set of of profiling strategies to provide user recommendations based on content, *e.g.* the tweets of the user or the tweets of his followers, or collaborative filtering. The methods proposed by [21] and [7] also provide tweet recommendations. Pennacchiotti et al. [21] analyze the tweets content as well as the content of the user's direct neighbors while Diaz-Aviles et al. [7] use the user past interaction to compute rankings in real-time. All these works consider content but unlike our proposal they do not consider paths longer than direct follower/followee links.

Few papers combine content and topology of the social graph. Wend et al. [26] present an extension of the PageRank algorithm named TwitterRank which captures both the link structure and the topical similarity between users. However this similarity is based on topics provided by LDA and their distance-based similarity computation between users does not capture the semantic similarity between topics. We also propose an authority score for an account which estimates the local and global influence of this account for a given topic. Our scoring function also provides higher weight for short paths to favor "local" recommendations. In [10], authors describe the production recommender system implemented in *Twitter*. It relies on an adaptation of the SALSA algorithm [15] which provides user recommendation in a centralized environment based on a bipartite graph: the user's circle of trust, computed with random walks from the user considering content properties, and the accounts followed by the users from the circle of trust (the authorities). Our approach is different since it captures the users interest through the labeled social graph, which allows to compute scores based on semantics, on authority and on topology.

To scale and to accelerate our recommendations computations, we pre-compute scores for a subset of nodes named landmarks. Landmark-based approach is a well-known *divide-and-conquer* strategy for the shortest paths problem that have been shown to scale up to very large graphs [25, 9, 22, 23]. The idea is to select a set L of nodes as landmarks which store the distance to other nodes. The distance $d(u, v)$ between two nodes u and v is then estimated by computing the minimum $d(u, l) + d(l, v)$, where $l \in L$. Das Sarma et al. [23] chose to pre-compute the time-consuming shortest-path operations for each node in structures called "sketches" and to use them to provide shortest-path estimations at query-time which enables scaling for large web graphs. Gubichev et al. [9] extend the sketch-based algorithm proposed by [23] to retrieve shortest paths and improve the overall accuracy. Tretyakov et al. [25] use shortest-path trees to achieve an efficient and accurate estimation which support updates. They also introduce a landmark selection strategy that attempts to maximize the shortest-paths coverage. In [22] authors also investigated the impact of landmark selection on the accuracy of distance estimations. They proved that optimizing the landmark coverage is a NP-hard problem and showed experimentally that clever landmark selection strategies yield better results. Similar to these approaches, we employ landmarks for computation scaling and use some of the existing landmark selection strategies in the context of user recommendation.

3. MODEL

We introduce in this section the underlying social graph model and our recommendation scores. Table 1 lists the different notations used throughout the paper.

N, E	resp. set of nodes and edges
Γ^u, $\Gamma^u(t)$	followers for u (resp. total or on topic t)
$\mathcal{T}$	topics vocabulary
$label_n$, $label_e$	labeling function for nodes and for edges, resp.
$topo_\beta(u,v)$	topological score of v for u with decay factor β
$\sigma(u,v,t)$	recommendation score of v for u on topic t
$\widetilde{\sigma_S}(u,v,t)$	approximate recommendation score considering paths going through a node $n \in S$
$\omega_p(t)$	topical component of the path score for path p
$\overline{\omega}_p(t)$	path score for path p
$auth(u,t)$	node authority score for u on topic t
$\varepsilon_e(t)$	edge relevance of edge e on topic t
α, β	decay factor for an edge and path resp.
λ, $\mathcal{L}$	a landmark, the set of landmarks
$P_{u,v}$	set of all paths between u and v
$P_{u,\lambda,v}$	set of all paths between u and v through λ
$\Upsilon_k(\lambda)$	the *k-vicinity* of λ
$R_{u,v}$	recommendation vector of v for u for all topics

Table 1: List of notations

3.1 Labeled social graph

We model the *Twitter* social network as a directed labeled graph G=*(N, E,* $\mathcal{T}$*,* $label_N, label_E$*)* where N is set of vertices such that each vertex $u \in N$ is a user (account). $E \subseteq N \times N$ is a set of edges where an edge $e = (u,v) \in E$ exists if u follows v, *i.e*, u receives the publications of v. The labeling function $label_N : N \to 2^{\mathcal{T}}$ maps each user to the set of topics that characterize his posts, chosen in a topic vocabulary $\mathcal{T}$. The topics associated by the labeling function $label_E : E \to 2^{\mathcal{T}}$ to an edge $e = (u,v)$ describe the interest of the user u for the posts of v. In this paper topics are extracted from tweets by using OpenCalais[1] combined with a trained Support Vector Multi-Label Model using Mulan[2] (see Section 5). For a user u, we define $\Gamma^u(t)$ the set of nodes following u on topic t and by Γ^u the set of all his followers. An example of such graph is depicted in Figure 1. For users B and C we display their topics of interest along with an excerpt of their tweets.

3.2 Recommendation

For a user u and a query composed of several topics $Q = \{t1, \ldots, t_n\}$, our model recommends users v based on the following criteria which consider both graph topology and content semantics:

(*i*) *user proximity*: u trusts his friends, the friends of his friends, etc., but this confidence decreases with distance ;

(*ii*) *the number of paths* from u to v: user v is likely to be more important for u if there are many other relevant users (*i.e.*, linked to u) who recommend v;

(*iii*) *topical path relevance* of the connections between u and v with respect to Q.

[1]http://www.opencalais.com/
[2]http://mulan.sourceforge.net/

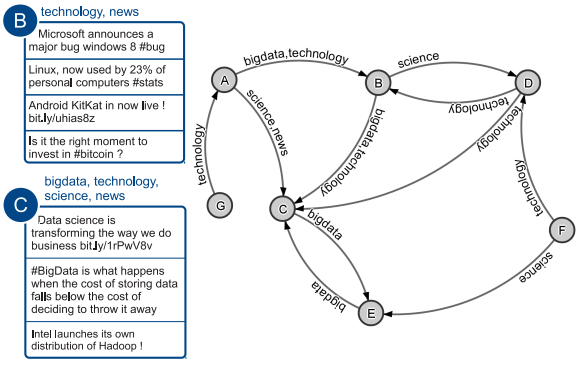

Figure 1: A labeled social graph

Our recommendation score combines the topical relevance of paths with a topological measure which considers *all existing paths* between two nodes u and v of the graph. More precisely, the recommendation score $\sigma(u,v,t)$ of the user v for user u on topic t on paths $p = u \rightsquigarrow v$ is the sum of all path scores $\overline{\omega}_p(t)$ and is expressed as follows:

Definition 1 (Recommendation score).

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

where $P_{u,v}$ denotes the set of all paths between u and v, $\overline{\omega}_p(t)$ is the total path score of a path of length $|p|$ and $\omega_p(t)$ the topical relevance $\omega_p(t)$. The decay factor $\beta \in [0,1]$ gives more importance to shorter paths.

The final recommendation score for the query Q is computed as a weighted linear combination (some are proposed in [1]) where user scores for each individual topic $t_i \in Q$ are weighted by the relevance of t_i for the posts of u which is computed by the topic extraction method (see Section 5).

Note that we can deduce from Equation 1 a score which considers only the topology by ignoring the topical relevance of paths (*i.e.* setting $\omega_p(t)$ to 1). This score is higher if there exist many short paths between u and v and is denoted as :

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

It corresponds to the Katz score [16] that has been successfully employed for link prediction. We will use it as a baseline for comparison with our method in Section 5.

The topical relevance $\omega_p(t)$ of a path $p = u \rightsquigarrow v$ for a topic t in Equation 1 considers both the relevance of nodes (*user authority*) and the topical relevance of edges (*edge relevance*) on the path p *w.r.t* the topics of the query Q. We define in the following these concepts.

Edge relevance:.

Each edge on a path p contributes to the score of p with a semantical score which depends on its topics. Distant edges contribute less to the recommendation score than edges close to u. More precisely, the relevance of an edge e at distance d from u on path p for a topic t is defined as :

$$\varepsilon_e(t) = \alpha^d \times max_{t' \in label_E(e)}(sim(t', t)) \qquad (3)$$

where $label_E(e)$ is the labeling function that returns the topics associated to the edge e. The decay factor $\alpha \in [0,1]$ decreases the influence of an *edge* according to its distance from u. The function $sim : \mathcal{T}^2 \to \mathbb{R}$ computes the semantic similarity between two topics t and t'. We use in the present paper the Wu and Palmer similarity measure [27] on top of the WORDNET [3] database (we have a small number of topics for labeling our dataset without synonymy issues), but other semantic distance measures, such as RESNIK or DISCO [4] could also be used. The choice of the best similarity function is beyond the scope of the current paper. When an *edge* is labeled with several topics, we only keep the maximum similarity to t among all its topics to avoid high scores for edges labeled with many topics that have small similarity to t.

User authority:.

We define a per node topical authority function $auth(u,t)$ of u on a topic t which depends on the number of users who follow u on t. The authority score is decomposed into two scores: (i) the *local authority score* that gives a higher score to users that are specialized on topic t than to users u who publish on a broad range of topics and (ii) the *global popularity score* that gives higher scores to users that are more followed on t. Combination of local and global scores has also been used to compute authorities for Web pages [12] or micro-blogging [10]. The authority score $auth(u,t)$ of a user u on a topic t is consequently defined as follows :

$$auth(u,t) = \underbrace{\frac{|\Gamma^u(t)|}{|\Gamma^u|}}_{local} \times \underbrace{\frac{\log(1+\Gamma^u(t))}{\log(1+max_{v\in N}(\Gamma^v(t)))}}_{global}$$

where $|\Gamma^u(t)|$ is the number of followers of u on t, and $|\Gamma^u|$ is its total number of followers. We used the logarithm function to smooth the difference between popular accounts and accounts with very few followers. The local authority is 1 when u is followed exclusively on t and the global popularity is 1 when u is the most followed user on t. If no other user follows u on t both scores are 0. The authority scores for a given topic t are high for users that are mainly followed on topic t and that have a significant number of followers. The combination of both local and global scores leads to similar authority scores for very specialized accounts with few followers and for very popular but generalist accounts. Observe for scores update that $|\Gamma^u|$ and $|\Gamma^u(t)|$ can be computed on local information of each user, without graph exploration. Oppositely the computation of $max_{v\in N}(\Gamma^v(t))$ may be costly since it requires to query the complete graph. However, the log strongly limits the impact of a variation in the popularity of an account with millions of followers, and we can assume this value is stored (and re-computed periodically).

EXAMPLE 1 (LOCAL AND GLOBAL AUTHORITY). *Consider the example graph in Figure 1, with a sample of tweets for the users B and C along with their topics. User B is more relevant for technology than C. Indeed B and C have the same global popularity with two followers on this topic for both accounts. However the local authority of B on technology is higher than the one of C since 2 out of the three topics on which B is followed are technology, whereas for C only 2 out of the 6 topics on which it is followed are technology. For the topic bigdata, the local authority of B and C is the same (1 out of 3 for B and 2 out of 6 for C) but C is more followed on bigdata (2 users who follow him) than B (1 user). Therefore, the total authority of C on bigdata is higher.*

[3] http://wordnet.princeton.edu/

[4] http://www.linguatools.de/disco/disco_en.html

Topical path relevance:.

Finally, we consider that the path relevance of p is high when both the relevance of the nodes and the one of the edges of p are high:

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

where $end(e)$ returns the end node of the edge e. The recommendation score of v for the user u on topic t is then obtained by replacing $\omega_p(t)$ in equation (1) by its formula given by equation (4). The resulting user recommendation score thus captures the topology (proximity and connectivity) of the graph along with the followers interests (expressed as labeled edges) and the authority score regarding the topic of interest for each user on the path.

EXAMPLE 2 (TOPICAL PATH RELEVANCE). *In Fig. 1, we want to recommend to A users on the topic technology (we suppose a search within a range $k = 2$). Users D and E can be reached with respectively the paths $p_1 = A \to B \to D$ and path $p_2 = A \to C \to E$, each of length 2. The relevance of the edge $A \xrightarrow{bigdata,technology} B$ is higher than the one of $C \xrightarrow{bigdata} E$, since the first one is at distance 1 from A, whereas the second is at distance 2. Moreover, the authority of node B on technology (computed as $(local) \times (global) = \frac{2}{3} \times \frac{log(1+2)}{log(1+2)}$) is higher than the authority of C on technology ($\frac{2}{6} \times \frac{log(1+2)}{log(1+2)}$). Overall, the semantic relevance of the edges on p_1 for technology is higher than the one of edges on p_2 and D obtains a higher recommendation score than E.*

3.3 Score analysis

We will show in the following the iterative formula for score computation and the score composition property that is used in Section 4 for landmark-based computation.

Iterative score computation.

Recommendation scores $\sigma(u,v,t)$ (Equation 1) are computed by using the Power Iteration algorithm [19] (see Algorithm 1 in Section 4). It starts by initializing $\sigma(u,u,t) = 1$ and $\sigma(u,v,t) = 0$ $(\forall u \neq v)$. At each step i, a new score $\sigma(u,v,t)^{(i)}$ (that considers all paths from u to v with length $\leq i$) is computed by using the scores $\sigma(u,v,t)^{(i-1)}$ of the neighbors $w \in \Gamma^v$ computed at step $(i-1)$. The computation is performed until convergence. The iterative formula for score computation is the following :

PROPOSITION 1 (ITERATIVE COMPUTATION).

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

where $topo_{\alpha\beta}^{i-1}(u,w)$ is the topological score (see Equation 2) with a decaying factor of $\alpha.\beta$. The score $\overline{\omega}_{w\to v}(t) = \beta.\alpha.$

$max_{t' \in label_E(w \to v)}(sim(t',t)).auth(v,t))$ *is the score of a path that contains only the edge* $w \to v$ *with topic* t.

PROOF. Suppose a path p of length $k \leq i$ from u to v. This path can be decomposed into a path p_1 of length $k-1$ from u to the neighbor w of v and an edge e from w to v of length 1. By using Equations (3) and (4), the score $\overline{\omega}_p(t)$ is computed as: $\overline{\omega}_p(t) = \beta.\overline{\omega}_{p1}(t) + \beta^{|p_1|}.\alpha^{|p_1|}.\overline{\omega}_e(t) = \beta.\overline{\omega}_{p1}(t) + \beta^{k-1}.\alpha^{k-1}.(\beta.\alpha.\max_{t' \in l_e(edge)}(sim(t',t)).auth(v,t))$. The score $\overline{\omega}_{p1}(t)$ corresponds to a path that finishes at w.

We can re-organize the paths in Equation 1 by grouping those that pass through the same neighbor w of v: $\sigma(u,v,t) = \sum_{p \in P_{u,v}} \overline{\omega}_p(t) = \sum_{w \in \Gamma^{v-}(t)}(\sum_{p \in P_{u,v}, w \in p} \overline{\omega}_p(t))$. By replacing $\overline{\omega}_p(t)$ into this equation we obtain the iterative score formula. □

Score composition.

From the iterative score computation we can deduce for each path p from a node u to a node v its total path score $\overline{\omega}_p(t)$ on topic t from the score of of its sub-paths already computed using the following property :

PROPOSITION 2 (RECOMMENDATION SCORE COMPOSITION). *Assume a path* $p = p_1.p_2$, *with* $\overline{\omega}_{p_1}(t)$ *and* $\overline{\omega}_{p_2}(t)$ *the total path scores of respectively* p_1 *and* p_2 *for a topic* t. *The total path score of* p *can be computed as:*

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

PROOF. By induction using the recursive formula, we prove the proposition for paths with length $k \geq 1$. □

Iterative score computation convergence.

In order to show the convergence of the iterative computation of recommendation scores $\sigma(u,v,t)$ of users v for user u on topic t (Equation (5)), we express this computation in matrix form as :

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

where $R_t^{(k)}$ is the recommendation vector for topic t computed at step k ($R_t^{(k)}[v]$ is the recommendation score $\sigma(u,v,t)$ computed at step (k)). Matrix A is the adjacency matrix of the graph ($A[v][u] = 1$ if u follows v). Matrix S_t is the similarity-authority matrix ($S_t[v][u] = sim(max_{t' \in l_e(u \to v)}(t',t)) \times auth(v,t)$). Vector $T_{\alpha\beta}^{(k)}$ is the topological vector at step k ($T_{\alpha\beta}^{(k)}[v]$ is the topological score $topo_{\alpha\beta}^{(k)}(u,v)$). It can be expressed as follows :

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

where $I[u] = 1$ and $I[v] = 0$ for all $u \neq v$. We deduce that the computation convergence is achieved under the following condition :

PROPOSITION 3 (SCORES COMPUTATION CONVERGENCE). *If* $\beta < 1/\sigma_{max}(A)$, *where* $\sigma_{max}(A)$ *is the highest eigen value for* A, *then the iterative scores computation of our recommendation scores converges.*

PROOF. Based on the recursive formula which defines the topical vector for a given node n, the topical scores matrix defined by the series expansion

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

converges when $I - \alpha\beta A$ is positive definite, so $\alpha\beta < 1/\sigma_{max}(A)$. Consider a step k' when convergence is reached for $T_{\alpha\beta}$. Then for any $k > k'$, we have the recursive computation $R^{(k+1)} = (\beta A)R^{(k)} + C$ with $C = (\alpha\beta)ST_{\alpha\beta}^{(\infty)}$ constant. The convergence for $R^{(k+1)}$ is reached when $R^{(\infty)} = (\beta A)R^{(\infty)} + C$ thus when $R^{(\infty)} = (I - \beta A)^{-1} \times C$. This can be achieve if $\beta < 1/\sigma_{max}(A)$. Since $\beta > \alpha\beta$, this later condition is sufficient to ensure convergence. □

4. LANDMARK-BASED COMPUTATION

The recommendation score computation presented in the previous section assumes to explore *all* paths from a user u to the nodes to be recommended. Computing recommendation scores by graph exploration at k hops for a graph with n nodes supposes to consider out_{avg}^k paths for the average case (out_{avg} denotes the average out degree) and of N^k paths in the worst case for a complete graph. This might be prohibitive in the context of social graphs with millions of nodes and edges. We rely here on a landmark-based approach to propose fast approximate recommendations.

The computation is performed in two steps: (i) in the preprocessing step we precompute for a sample of nodes in the graph, named landmarks, top-n recommendation scores (n being a parameter of the system) for every topic $t \in \mathcal{T}$ and (ii) at query time we compute approximate recommendations by exploring the graph until a given depth (also a parameter of the system) and collect precomputed recommendations from landmarks encountered during this exploration.

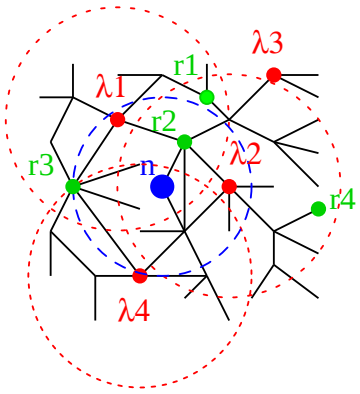

Figure 2: Landmark-based recommendation

EXAMPLE 3. *Figure 2 illustrates this approach, where* n *is the query node and* λ_1, λ_2, λ_3 *and* λ_4 *are landmarks. When performing the graph exploration represented by the blue dashed-line from the node* n, *the landmarks* λ_1, λ_2 *and* λ_4 *are encountered. Node* r_2 *is encountered during exploration and its score for* n *is computed at the same time as the scores for* λ_1, λ_2 *and* λ_4. r_1 *is not encountered during the exploration from* n, *but it is encountered by the explorations starting from landmarks* λ_1 *and* λ_2. *Its recommendation score for* n *is estimated by aggregating the scores of* λ_1 *and* λ_2 *for* n *computed at query time with the scores of* r_1 *for* λ_1 *and* λ_2 *which were precomputed.*

4.1 Preprocessing

For the preprocessing step we consider a subset $\mathcal{L} \subset N$ of nodes, so-called the landmarks, with $|\mathcal{L}| \ll |N|$. Instead of a random sampling, several strategies may be considered to determine $\mathcal{L}$. For instance, landmark-based approaches for computing shortest paths within a large graph rely mainly on centrality properties (betweenness or closeness centrality) to determine the sampling. The publisher-follower characteristics of our graph also allow other topology-based sampling, like a selection of the nodes with the most important number of followers (most popular accounts) or the ones that follow the highest number of accounts (most active readers). While the choice of the landmarks may impact the global performances of our approach we do not investigate further the sampling strategies in the current paper. Nonetheless some of these sampling techniques are experimentally compared in Section 5.

Algorithm 1: LANDMARK_RECOMM(λ, max_k, T, β, n)

Require: landmark (λ), maximum exploration (max_k), set of topics (T), topological decay factor (β), number of results to return (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** $t \in T$ **do**
8: $\quad\quad\quad\quad\sigma^{(k+1)}(\lambda, v, t) +=$
$\quad\quad\quad\quad\beta \times \sigma^{(k)}(\lambda, u, t) + topo^k_{\alpha\beta}(\lambda, u) \times \overline{\omega}_{u \to v}$
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** $(\sum_{u \in \Upsilon_k} \sigma^k(\lambda, u, t))/|R_t| < tol,\ \forall t \in T$ **then**
16: $\quad\quad converged := true$
17: $\quad$**end if**
18: $\quad k := k + 1$
19: **end while**
20: **return** for all $t \in T$ top-n$((R_t), topo_\beta(\lambda))$

Algorithm 1 performs the recommendation computation (we remove the initialization of recommendation scores to simplify the presentation) and is used both in the preprocessing and in the approximate score computation step. It takes as parameters the starting node λ of the graph exploration, the maximum exploration depth max_k, the set of topics on which the recommendations are computed T, the path decay factor β and the number n of final results to be kept.

The set of reached nodes at depth k from λ is called the *k-vicinity* of λ, denoted $\Upsilon_k(\lambda)$. $\Upsilon_\infty(\lambda)$ denotes the set of reachable nodes from λ. For each topic $t \in T$ the algorithm computes a recommendation vector R_t with $R_t[u] = \sigma(\lambda, u, t)$ (see Equation 6), where $u \in \Upsilon_\infty(\lambda)$ is a reachable node from λ. The algorithm also computes the topological scores $topo_\beta(\lambda, u)$ with decay factor β for all $u \in \Upsilon_\infty$ (Equation 2), used to estimate the final recommendation scores at query time (see below). Iteration in line 4 allows to explore the k-vicinity of λ. For each iteration we add in the k-vicinity nodes that could be reached with an additional hop (l. 5). For each node v reached at this step (l. 6), we compute (or update if the node has been already encountered)) the recommendation score for each term of the topic vocabulary (l. 7-8), and the node's topological score (l. 10). The score for u on paths of length k is added to the sorted topical (l.12) and topological (l.13) lists of λ. Finally, only the top-n recommendations for each vector R_t and only the top-n topological scores $topo_\beta(\lambda)$ are stored.

In the preprocessing step, for each landmark $\lambda \in \mathcal{L}$ we compute the recommendation scores on all topics for all nodes encountered during the iterative computation. So Algorithm 1 runs until the convergence is reached (max_k is set to a large value) with the parameter T set to $\mathcal{T}$.

4.2 Fast approximate recommendation

We now present the algorithm for fast approximate recommendations based on the pre-computations performed for each landmark in the preprocessing step. We assume that we want to recommend to an account u other accounts for a topic t.

We first perform a graph exploration starting from u, similar to the one described by the Algorithm 1, for a given maximal depth k, max_k, set to a small value (*e.g.* 2 or 3). The graph exploration finds landmarks in the *k-vicinity* of u and computes path scores on topic t for paths from u to each encountered landmark. These scores are further combined with the scores stored by the landmarks in order to compute the approximate recommendation scores.

More precisely, we denote $\Lambda \subseteq \mathcal{L}$ the set of landmarks encountered by the graph exploration. For each $\lambda \in \Lambda$ the top-n recommended accounts v along with their recommendation scores $\sigma(\lambda, v, t)$ and their topological score $topo_\beta(\lambda, v)$ are already computed in the preprocessing step. The approximate recommendation of a node v for a user u is an aggregation of the scores of v computed by all the landmarks $\lambda \in \Lambda$:

DEFINITION 2 (APPROXIMATE RECOMMENDATION SCORE). *The approximate recommendation score of a node v for a node u on the topic t with respect to the set of landmarks Λ is defined as :*

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

where the score $\widetilde{\sigma}_\lambda(u, v, t)$ denotes the score of v that takes into consideration the set of paths $P_{u,\lambda,v}$ from u to v that pass through the landmark λ.

The score $\widetilde{\sigma}_\lambda(u, v, t)$ is computed by the composition of the scores $\sigma(u, \lambda, t)$ and $topo_{\beta\alpha}(u, \lambda)$ obtained during the exploration phase with the scores $\sigma(\lambda, v, t)$ and $topo_\beta(\lambda, v)$ that are stored in the sorted lists of λ.

PROPOSITION 4 (APPROXIMATE SCORE COMPUTATION). *The recommendation score of v for u with respect to the landmark λ can be computed as follows :*

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

PROOF. Any path p from $P_{u,\lambda,v}$ could be decomposed into p_1 and p_2, with $p_1 \in P_{u,\lambda}$ and $p_2 \in P_{\lambda,v}$. Obviously any path $p = p_1.p_2$ with $p_1 \in P_{u,\lambda}$ and $p_2 \in P_{\lambda,v}$ is a path from $P_{u,\lambda,v}$.

Consequently, based on Proposition 2) we have:

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

□

Note that our approach estimates a lower-bound of the recommendation scores while landmark-based approaches traditionally proposed for shortest paths computation provide score upper-bounds, based on the triangular inequality. Indeed in our setting the approximate scores do not consider all the paths from u to v, but only the subset $P_{u,\lambda,v}$ that pass through λ. However experiments show this approximation allows to retrieve a set of recommendations close to the one retrieved by an exact computation.

Algorithm 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 u

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** $\widetilde{R_t}$

We perform our approximate recommendation for a node u and a topic t by using Algorithm 2. It first calls the LANDMARK_RECOMM algorithm to compute recommendation scores from u to all nodes within a distance k along with their topological score (l. 1). Observe that unlike the preprocessing step, the exploration depth has a small value k (2 in our experiments) so that the algorithm will not be run until the convergence. The recommendations are computed only for a single topic t. Note also that the decay factor is here $\beta.\alpha$. For each encountered landmark (l. 2-3) we combine its recommendation for the topic t with the recommendation score computed from u to the landmark according to Proposition 4 (l. 5).

5. EXPERIMENTS

In this section we present the experiments that we have conducted on a real *Twitter* and *DBLP* datasets to validate our structures and algorithms.

5.1 The datasets

The Twitter dataset we use in our experiments contains approximately 2.2 million users (with their 2.3 billion associated tweets acquired in 2015 from February to April) linked by more than 125 million edges (*i.e* following relationships). Table 2 describes the main topological properties on the generated dataset. For our *Twitter* dataset these properties are very close to the ones of the real *Twitter* network observed in [18].

Property	Twitter	DBLP
Total number of nodes	2,182,867	525,567
Total number of edges	125,451,980	20,526,843
Avg. out-degree	57.8	47.3
Avg. in-degree	69.4	53.6
max in-degree	348,595	9,897
max out-degree	185,401	5,052

Table 2: Datasets topological properties

Topic extraction:.

As already mentioned, in order to generate the topics of the edges we first used the OpenCalais document categorization to tag a subset of the users (nodes) in our graph with topics extracted from their published tweets. This strategy allowed to tag 10 percents of our nodes using a list of 18 standard topics for Web sites/documents proposed by OpenCalais. The user categorization was completed by using a trained Support Vector Multi-Label Model using Mulan, with a precision of 0.90, that associated to each user in the graph his publisher profile (topics on which he publishes). Each follower is characterized by a follower profile containing topics with high frequency among the topics of their followed publishers. Finally the labels of each edge are the topics in the intersection between the corresponding follower and publisher profiles. The resulting graph is a fully labeled social graph with 2.2M nodes and 125M edges. We refer to this dataset as *Twitter*. The edge labels obtained with our generation method show a biased distribution similar to the one observed for Web sites in Yahoo! Directory [17] (see Figure 3).

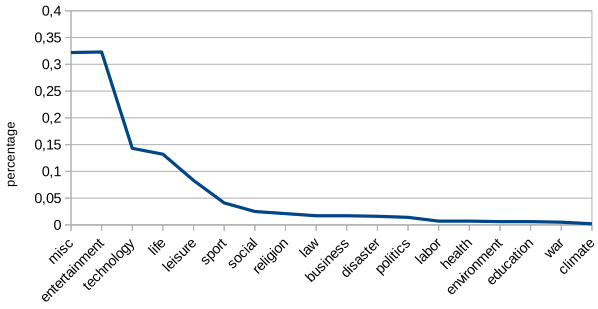

Figure 3: Distribution of edges per topic

For the DBLP dataset, we merged different versions of the ArnetMiner DBLP datasets[5]. The resulting dataset contains 2,291,100 papers, 1,274,860 authors and 4,191,643 citations. From this dataset we build a graph of author citations by creating a directed edge between author u and author v if some paper of u cites a paper of v. This results in a final dataset with 525,567 authors and 20,526,843 citations

[5]http://aminer.org/billboard/citation

between them. Observe that we only kept cited authors. Then we used the Singapore Classification[6] to manually label some of the major conferences. The other conferences are labeled based on the number of authors they have in common with already labeled coferences (topics of two conferences are close if there are many authors that publish in both of them). Paper topics are deduced from the conference topics by assuming that a paper published in a conference is about the main topic of this conference. Author profiles are built from the topics of their published papers. The resulting dataset is summarized in Table 2.

5.2 Implementation

We implemented our solution in Java (JVM version 1.7). The experiments were run on a 64-bit server with a 10 cores Intel Xeon processor at 2.20GHz and 128GB of memory. The server is running Linux with a 3.11.10 kernel.

The topic similarities given by the Wu and Palmer similarity scores are pre-computed and stored in memory as a triangular similarity matrix. We considered here only the 18 common topics for Web documents, which results in a 2.5 KB file, but observe that for 10,000 topics the similarity matrix will require around 750MB so can still easily fit in memory. A similar approach was chosen for the similarity matrix of the DBLP dataset. We stored the landmark recommendations as inverted lists: for each landmark, we have a set of accounts recommended along with their recommendation score for each topic from $\mathcal{T}$. Landmarks were chosen according to one of the selection strategies presented in Table 4. We compare the quality of our recommendations with two related algorithms chosen as baseline: the standard Katz score [16], which considers only the topology (all paths between two accounts along with their length, given by the topological score in Equation 2), and TwitterRank [26] which captures both the link structure and the topical similarity between users. In the following we denote our score as Tr.

The values of parameter β and α are set to 0.0005 and respectively to 0.85, similarily to the values used for the Katz and the TwitterRank alogorithms in [16] and [26].

5.3 Quality of the recommendation

We consider a test set of T edges of the graph together with their corresponding topics representing the ground truth. As observed in [16], to maintain the topological properties of the graph during the evaluation process, the target node of an edge of the test set must have at least k_{in} in-degree and the source node at least k_{out} out-degree ($k_{in} = 3$ and $k_{out} = 3$ in our experiments). All edges from T are then removed from the graph. For each edge $e = u \rightarrow v$ in T we randomly select 1000 accounts in the graph. We compute recommendation scores for the 1001 accounts (the 1000 accounts and v) with respect to u on the topics of e and we form a ranked list (similar to [6]) for each topic. For each list, if v belongs to the $top-n$ accounts of the ranked list we have a hit, otherwise a miss. The overall recall and precision are defined similarly to [6] with $\#hits/T$ and $\#hits/N.T$ respectively. For our experiments we set the test size $T = 100$ and we average values over 100 trials.

Figure 4 illustrates the accuracy of the different recommendation strategies for the Twitter dataset. We see that

[6]http://www.ntu.edu.sg/home/assourav/crank.htm

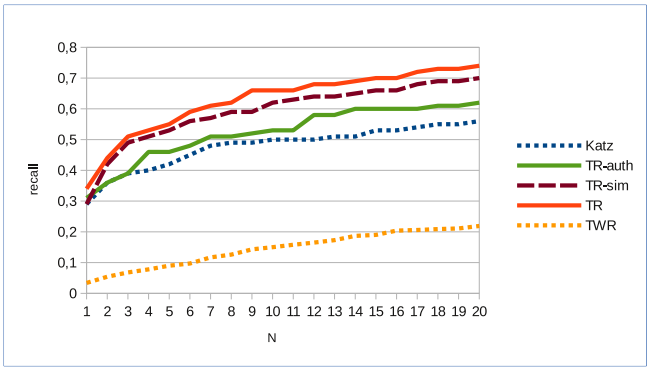

Figure 4: Recall at N (Twitter)

TwitterRank is outperformed by other algorithms. Indeed only for 4% of the recommendations the account corresponding to the removed edge is found in the top-1 for TwitterRank, while Katz provides as first recommendation the correct account in 29% of the tests and Tr in 34%. So Tr provides a 8.5 and 1.2 gain with TwitterRank and Katz respectively for the top-1. For the top-10 the improvement remains significant: 3.8 and 1.3 with TwitterRank and Katz respectively.

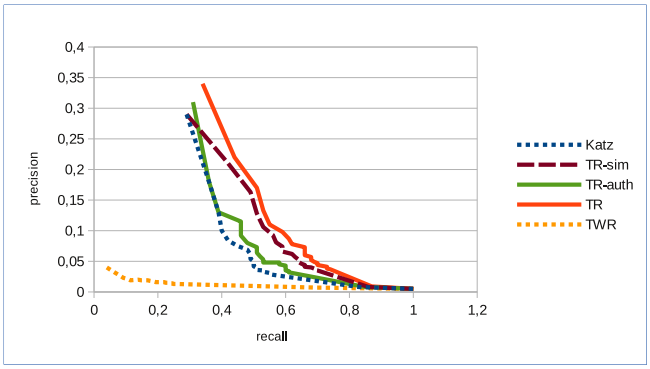

Figure 5: Precision vs recall (Twitter)

Figure 5 confirms that Tr outperforms other approaches: for a similar recall value greater than 0.4, the precision of Tr is at least twice the one of Katz and one order of magnitude higher than the one of TwitterRank.

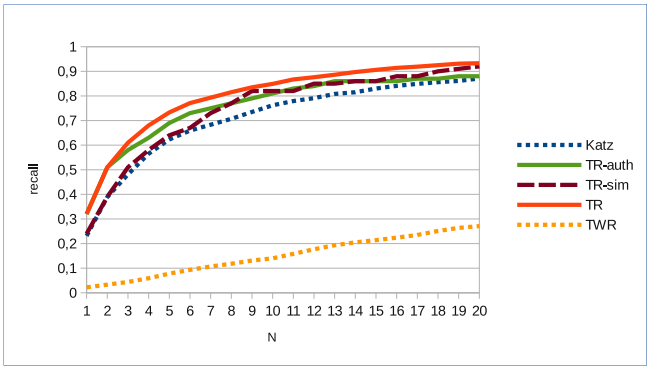

Figure 6: Recall at N (DBLP)

We observe in Figures 6 and 7 that the DBLP dataset exhibits similar results. The recall however exhibits a faster

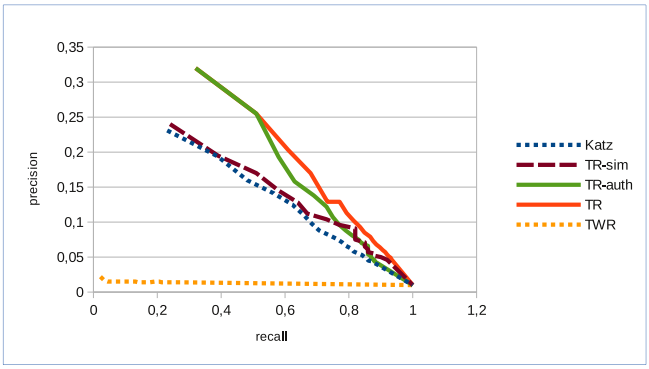

Figure 7: Precision vs recall (DBLP)

increase for TR due to the self-citations phenomenon: authors from a given paper often cite one or several of their previous papers on the topic. These papers may share some citations with the paper corresponding to the edge removed for the selected author. This also explains the faster recall increase for Katz strategy. TwitterRank whose recommendations are essentially based on the popularity (in-degree) of an account reached in the graph does not capture this phenomenon and provides slightly worse results than with the Twitter dataset.

Figure 4 also illustrates the benefit when taking into account both the edge similarity and the authority. Adding the edge similarity to Katz, which takes into account only the topology (number and length of the paths between two nodes), provides a better precision and recall (+11% for the precision and $N = 20$, see TR$-auth$). When we consider our approach without edge-similarity scores, but with topic authorities of the nodes, we improve both recall and precision (+25% compare to Katz precision, see TR$-sim$). Finally our approach which integrates the topology, the edge similarity and the topic authorities, outperforms these approaches (+32%, +19% and +6% with resp. Katz, TR$-auth$ and TR$-sim$).

However there exists a large discrepancy for accuracy when considering two dimensions of analysis: the edge removal strategy and the popularity of the topic used for the recommendation. For the top-10, Figure 8 shows that for Twitter we have a very low accuracy, *i.e.* a recall of 0.15, 0.03 and 0.18 for respectively Katz, TwitterRank and TR, when trying to retrieve an account which belongs to the top-10% less followed accounts (TW min). Conversely very popular accounts (top-10% most followed accounts) are most of time retrieved in the top-10 recommendations with a recall between 0.9 and 0.95 for all strategies. This can be explained by their path-based approach which aggregates scores on incoming paths, so accounts with numerous incoming paths got a high score. Observe that for popular accounts, TwitterRank provides the best results. Indeed most of large accounts are labeled with several topics. While TwitterRank score relies on the account popularity and on the presence or not of a label for an account (whatever the number of labels it has), TR score considers for its authority score the number of incoming edges labeled with a given topic. But the more labels an account has, the lower authority score for a given topic it may have. Oppositely an account with a low in-degree rarely has several labels. Our approach that also considers semantic similarity between topics on edges is then particularly efficient. With the DBLP dataset, authors who belong to the 10% less cited are more likely retrieved than with the Twitter dataset for Katz and TR due to the higher density of the graph. However TwitterRank based on the popularity fails to retrieve these authors. Even for the 10% most popular authors, TwitterRank does not achieve the good results obtained for the Twitter datasets, due to a different distribution of the in-degree. While the 10% most followed accounts in Twitter include few extremely popular accounts and some moderately popular, the 10% most followed authors in DBLP consist in a more uniform dataset regarding the in-degree.

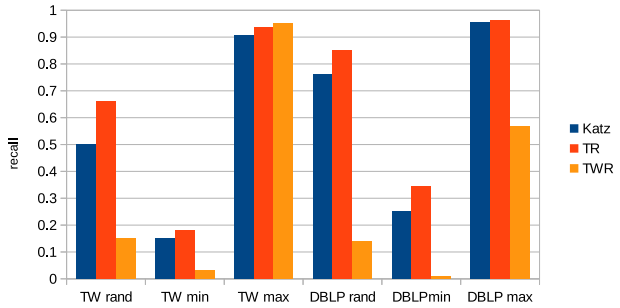

Figure 8: Recall w.r.t. popularity

Since the distribution of edge topics is very biased we also study the impact of the popularity of the topic on the recommendations. Results are depicted in Figure 9 for topics `social`, `leisure` and `technology`. Two main conclusions are underlined with this experiment. First, the less popular an account is, the better accuracy for our recommendations we get. So for an infrequent topic like `social` we get a recall-at-10 for TR, Katz and TwitterRank of respectively 0.959, 0.751 and 0.253. Oppositely for the popular topic `technology` we get respectively 0.462, 0.424 and 0.09. Indeed for a popular topic many accounts may be found in a close and connected neighborhood of the account we want to recommend, possibly with a higher score than for the account formerly linked by the removed edge. Second we observe that TR which considers the semantic similarity between topics always outperforms other strategies.

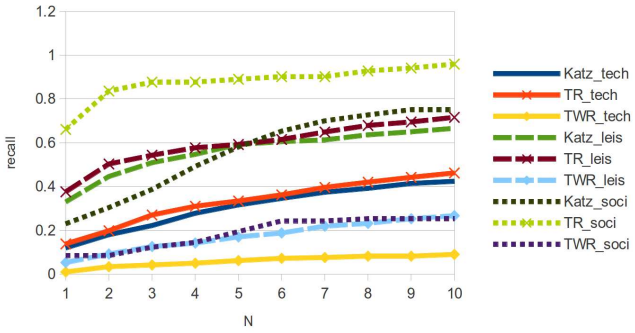

Figure 9: Recall w.r.t. topic popularity

However this experiment does not highlight the quality of the recommendations performed by each algorithm but only the ability to retrieve a removed link between two followers

(link prediction). To estimate the quality of the recommendations we rely on a user validation.

User Validation Task.
In order to evaluate the relevance of the generated recommendations, we conducted a user validation task on 54 IT users (undergraduate, postgraduate and PhD students, and academics) from which 46% are regular *Twitter* users. We set up an on-line blind test where we ask users to rate the relevance of a set of recommendations for a given topic on a scale from 1 (low relevance) to 5 (high relevance). A recommendation set consists in the top-3 recommendations given by Katz, TR and TwitterRank, so 9 recommended accounts for topics `Technology`, `Social` and `Leisure`. On the interface the recommendation list is shuffled, and for each recommendation we display a sample of 5 randomly chosen tweets from the corresponding account.

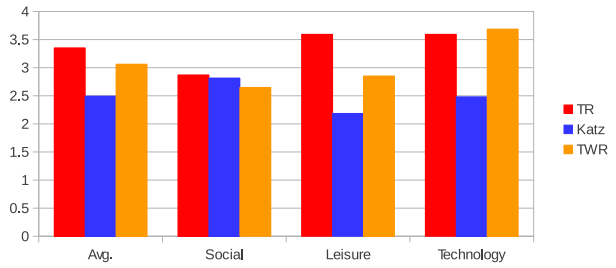

Figure 10: Relevance scores (user validation Twitter)

Figure 10 presents the results of our user validation. We observed that the user during the validation usually mark with the average 2 or 3 value all accounts when he was doubtful about the relevance or not of an account what happened usually when tweets were neutral, unclear, or when they required some knowledges about a given topic (*e.g.* few European people know who is Tom Brady so can not assert this tweet is about the `leisure` topic). So scores greater than value 3 are significant since they means users really observe the relevance of an account.

From this experiment we conclude that on average TR and TwitterRank provide more relevant recommendations according to the topic searched. However according to the popularity of the topics (see Figure 3) we have very different results. The `social` topic gave more homogeneous results with a score between 2.7 for TwitterRank, 2.8 for Katz and 2.9 for TR. The reason for this result is that posts published by these accounts are generally difficult to classify since they mix social and health, or social and politics for instance. Oppositely topics like `leisure` or `technology` are less ambiguous. For these topics, we see that TR and TwitterRank outperform Katz, which was expected since these two approaches consider the content published for their recommendation scores, unlike Katz. While TwitterRank generally recommends accounts with a large number of followers, TR can also recommend smaller account but more-specialized, which results in a better relevance score for topic with a medium popularity like `leisure`, when TwitterRank is slightly better for the most popular topic `technology`.

We also conduct a user validation for the DBLP dataset. We build a list with the top-3 recommendations returned by each method for researchers from our lab. Observe they belong to different areas (IR, DB, OR, network, software engineering, etc). To illustrate how the different methods may help to discover relevant authors we limit to 100 the number of citations of the authors returned by each algorithm (so we avoid to propose very popular and obvious authors). We propose to each researcher the randomized list with the 9 authors retrieved based on his DBLP entry. He marks each proposal between 1 and 5 according to the relevance of the proposal (*i.e.* the proposed author could have been cited regarding the past publications done by the researcher). We collect 47 answers and results are presented in Table 3.

	Katz	TR	TWR
average mark	2.38	2.47	1.51
# 4 and 5-mark	46	47	11
best answer (%)	0.38	0.50	0.12

Table 3: User validation (DBLP)

The first row shows that both Katz and TR outperform TwitterRank on this dataset. The second row shows that around a third of the recommendations proposed by Katz and TR are considered particularly relevant by our panel (4 or 5-mark)) while only 8% are so-considered with TwitterRank. A first rationale is that papers we cite, including papers from co-authors, often cite the same relevant articles within a topically-closed research community. The importance of the semantics on edge is less than with Twitter since researchers, whatever the number of articles they published, cite/are cited by mainly researchers from their community. This explains the close score for Katz and TR. The important role of the popularity in TwitterRank explains it poor results in this context since it proposed popular authors even when there exists a small number of paths between the query author and them. Last row confirms the quality of our recommendations since TR presents for 50% of the tests the best top-3 recommendations, when Katz and TwitterRank achieve respectively 38% and 12%.

5.4 Approximate computations

We perform a set of experiments to illustrate the benefits of the landmark-based approximate computations. Since results may be highly related to the choice for the landmarks selection, as underlined in [22], we decided to implement and compare recommendations based on 11 different landmark selection strategies presented in Table 4.

Size and building time of the landmark index. A first experiment highlights the important time discrepancy for the landmark selection algorithms (see Table 5). Obviously random selections of the landmarks like RANDOM, BTW-FOL and BTW-PUB are the fastest strategies (around $2ms$ per landmark), while strategies based on the centrality property are 5 orders of magnitude slower (around $17h$) due to $O(N^2.\log N + NE)$ centrality complexity (with Johnson's algorithm). Table 5 also illustrates that the recommendation computation for a given landmark is almost independent of the landmarks selection strategy (between 12 and 15 mns), which means that convergence is achieved in a similar number of steps after exploring a similar number of paths.

Comparison of the landmark selection strategies for recommendations. We evaluate our approximate approach

Algorithm	Description
RANDOM	Draw landmarks with a uniform distribution
FOLLOW	Draw landmarks with a probability depending on their # of followers
PUBLISH	Draw landmarks with a probability depending on their # of publishers
IN-DEG	Landmarks are nodes with highest in-degree
BTW-FOL	Draw landmarks among nodes with # of followers in [min_follow,max_follow]
OUT-DEG	Landmarks are nodes with highest out-degree
BTW-PUB	Draw landmarks among nodes with # of publishers in [min_publis,max_publish]
CENTRAL	Select landmarks that are reachable at a given distance from most of chosen seed nodes
OUT-CEN	Select the landmarks based on the number of different output seeds that they cover
COMBINE	Weighted combination between the CENTRAL and OUT-CEN
COMBINE2	Weighted combination between the BTW-FOL and BTW-PUB

Table 4: Landmarks selection algorithms proposed

Strategy	landmarks	
	select. (ms)	comput. (s)
RANDOM	2.4	756.7
FOLLOW	3,712.8	877.3
PUBLISH	3,614.7	868.6
IN-DEG	459.6	854.3
BTW-FOL	2.4	735.1
OUT-DEG	1,815.7	918.6
BTW-PUB	1.7	822.7
CENTRAL	61,060.2	807.8
OUT-CEN	66,862,3	816.5
COMBINE	130,461.8	818.2
COMBINE2	2.45	805.6

Table 5: Determining landmark w.r.t. strategies

presented in Section 4. We perform a BFS at depth 2 from a query node and combine scores with the ones of the landmarks encountered. (see Algorithm 2). Then we compare the recommendations retrieved with the ones provided by the exact computation with convergence. Average results for 100 landmarks are reported in Table 6.

Strategy	#lnd	time in s (gain)		L10	L100	L1000
RANDOM	2.9	0.93	(338)	0.130	0.124	0.125
FOLLOW	17.5	0.83	(379)	0.377	0.140	0.096
PUBLISH	11.7	0.58	(539)	0.349	0.136	0.100
IN-DEG	58.9	0.84	(373)	0.523	0.149	0.066
BTW-FOL	3.5	0.55	(577)	0.061	0.059	0.058
OUT-DEG	6.2	0.81	(388)	0.518	0.147	0.064
BTW-PUB	2.9	0.54	(585)	0.129	0.127	0.123
CENTRAL	5.3	0.76	(414)	0.134	0.123	0.125
OUT-CEN	4.4	0.74	(425)	0.172	0.131	0.121
COMBINE	4.2	0.71	(443)	0.180	0.125	0.118
COMBINE2	3.7	0.54	(584)	0.129	0.126	0.124

Table 6: Comparison of the landmark selection strategies

First we observe that the number of landmarks encountered during the BFS at distance 2 differs from one strategy to another and ranges from 2.9 on average for the RANDOM strategy to 58.9 for IN-DEG. Centrality approaches lead to less landmarks encountered since they select landmarks among nodes which connect connected subgraphs and a two-hop BFS is more unlikely to visit several connected subgraphs. We notice that the processing time does not depend on the number of landmarks found, what seems counterintuitive since more landmarks means more computations (score combinations) to perform. The rationale is that we perform pruning when we encounter a landmark during the BFS, to avoid considering twice paths from the BFS which pass through a landmark. Since the recommendation computation is dominated by the BFS exploration and computation, this pruning largely reduces the whole processing time. A second important result is that our approximate computation allows to get a *2-3 order of magnitude gain compared to the exact computation*. Finally observe that a strategy which allows to find more landmarks is more tolerant to a landmark departure or to a landmark with outdated recommendation values.

Finally to validate the quality of the approximate computation we report the average Kendall Tau distance between the approximate computation and the exact computation obtained at convergence when a landmark stores respectively the top-10, top-100 or top-1000 recommendations for all topics (see last 3 columns of Table 6). Keeping 1000 recommendations for the landmarks at the pre-processing allows to reach a Kendall Tau distance between 0.06 and 0.13 for the top-100 recommended accounts for a node at query time. Keeping a top-10 at landmarks leads to a higher Kendall Tau distance since a landmark may update at most 10 scores from the top-10 built at distance 2 from the BFS. Consequently an account which is ranked at the 11th place for two landmarks is not kept as a recommendation whereas its aggregate score may be higher than accounts kept as recommendations. Remark that even when storing the top-1000 for each topic, the landmarks recommendations can easily fit in memory since they require 1.4MB storage each.

6. CONCLUSION

We present the TR recommendation score which combines topology and semantic information regarding the user interest. To face prohibitive computations with very large graphs, we propose a landmark-based approach which requires a pre-computation step for a small set of identified nodes and achieves a 2-3 order of magnitude gain compare to the exact computation. The experiments and user validation show that TR outperforms other algorithms. As future work we intend to study updating strategies since many following links have a short lifespan. This graph dynamicity may impact the scores stored by the landmarks. Moreover we made the choice to handle the recommendation task in a centralized manner motivated by the current social media architectures like *Twitter Who-to-Follow* service hosted on a single server. However, with the continuous increase of the social graph sizes, distribution strategies must be considered in the future. Regarding our approach, distribution implies to split the graph by taking into account connectivity, but also to perform landmark selections and distributions that allow a node to evaluate the recommendation scores "locally" minimizing network transfer costs.

7. REFERENCES

[1] J. A. Aslam and M. Montague. Models for metasearch. In *SIGIR*, pages 276–284, 2001.

[2] S. Budalakoti and R. Bekkerman. Bimodal Invitation-Navigation Fair Bets Model for Authority Identification in a Social Network. In *WWW*, pages 709–718, 2012.

[3] V. Chaoji, S. Ranu, R. Rastogi, and R. Bhatt. Recommendations to Boost Content Spread in Social Networks. In *WWW*, pages 529–538, 2012.

[4] K. Chen, T. Chen, G. Zheng, O. Jin, E. Yao, and Y. Yu. Collaborative personalized tweet recommendation. In *SIGIR*, pages 661–670, 2012.

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

[6] P. Cremonesi, Y. Koren, and R. Turrin. Performance of Recommender Algorithms on Top-n Recommendation Tasks. In *RECSYS*, pages 39–46, 2010.

[7] E. Diaz-Aviles, L. Drumond, Z. Gantner, L. Schmidt-Thieme, and W. Nejdl. What is happening right now ... that interests me?: online topic discovery and recommendation in twitter. In *CIKM*, pages 1592–1596, 2012.

[8] S. G. Esparza, M. P. O'Mahony, and B. Smyth. Towards the Profiling of Twitter Users for Topic-Based Filtering. In *SGAI*, pages 273–286, 2012.

[9] A. Gubichev, S. J. Bedathur, S. Seufert, and G. Weikum. Fast and accurate estimation of shortest paths in large graphs. In *CIKM*, pages 499–508, 2010.

[10] P. Gupta, A. Goel, J. Lin, A. Sharma, D. Wang, and R. Zadeh. WTF: the Who to Follow Service at Twitter. In *WWW*, pages 505–514, 2013.

[11] J. Hannon, M. Bennett, and B. Smyth. Recommending twitter users to follow using content and collaborative filtering approaches. In *RECSYS*, pages 199–206, 2010.

[12] G. Jeh and J. Widom. Scaling Personalized Web Search. In *WWW*, pages 271–279, 2003.

[13] P. Kapanipathi, F. Orlandi, A. P. Sheth, and A. Passant. Personalized Filtering of the Twitter Stream. In *SPIM*, pages 6–13, 2011.

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

[15] R. Lempel and S. Moran. Salsa: The stochastic approach for link-structure analysis. *ACM Transactions on Information Systems*, 19(2):131–160, 2001.

[16] D. Liben-Nowell and J. Kleinberg. The Link Prediction Problem for Social Networks. In *CIKM*, pages 556–559, 2003.

[17] T.-Y. Liu, Y. Yang, H. Wan, H.-J. Zeng, Z. Chen, and W.-Y. Ma. Support Vector Machines Classification with a Very Large-Scale Taxonomy. *SIGKDD Explorations*, 7(1):36–43, 2005.

[18] S. A. Myers, A. Sharma, P. Gupta, and J. Lin. Information network or social network?: The structure of the twitter follow graph. In *WWW Companion Volume*, pages 493–498, 2014.

[19] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, November 1999.

[20] A. Pal and S. Counts. Identifying Topical Authorities in Microblogs. In *WSDM*, pages 45–54, 2011.

[21] M. Pennacchiotti, F. Silvestri, H. Vahabi, and R. Venturini. Making your interests follow you on twitter. In *CIKM*, pages 165–174, 2012.

[22] M. Potamias, F. Bonchi, C. Castillo, and A. Gionis. Fast shortest path distance estimation in large networks. In *CIKM*, pages 867–876, 2009.

[23] A. D. Sarma, S. Gollapudi, M. Najork, and R. Panigrahy. A sketch-based distance oracle for web-scale graphs. In *WSDM*, pages 401–410, 2010.

[24] P. Symeonidis and E. Tiakas. Transitive Node Similarity: Predicting and Recommending Links in Signed Social Networks. *WWWJ*, 17(4):743–776, 2014.

[25] K. Tretyakov, A. Armas-Cervantes, L. García-Bañuelos, J. Vilo, and M. Dumas. Fast fully dynamic landmark-based estimation of shortest path distances in very large graphs. In *CIKM*, pages 1785–1794, 2011.

[26] J. Weng, E.-P. Lim, J. Jiang, and Q. He. TwitterRank: Finding Topic-sensitive Influential Twitterers. In *WSDM*, pages 261–270, 2010.

[27] Z. Wu and M. Palmer. Verbs Semantics and Lexical Selection. In *ACL*, pages 133–138, 1994.

[28] X. Yang, H. Steck, Y. Guo, and Y. Liu. On top-k Recommendation Using Social Networks. In *RECSYS*, pages 67–74, 2012.