Tweet Recommendation with Graph Co-Ranking

Rui Yan[†]
[†]Department of Computer
Science and Technology,
Peking University,
Beijing 100871, China
`r.yan@pku.edu.cn`

Mirella Lapata[‡]
[‡]Institute for Language,
Cognition and Computation,
University of Edinburgh,
Edinburgh EH8 9AB, UK
`mlap@inf.ed.ac.uk`

Xiaoming Li[†, ♮]
[♮]State Key Laboratory of Software
Development Environment,
Beihang University,
Beijing 100083, China
`lxm@pku.edu.cn`

Abstract

As one of the most popular micro-blogging services, Twitter attracts millions of users, producing millions of tweets daily. Shared information through this service spreads faster than would have been possible with traditional sources, however the proliferation of user-generation content poses challenges to browsing and finding valuable information. In this paper we propose a graph-theoretic model for tweet recommendation that presents users with items they may have an interest in. Our model ranks tweets and their authors *simultaneously* using several networks: the social network connecting the users, the network connecting the tweets, and a third network that ties the two together. Tweet and author entities are ranked following a co-ranking algorithm based on the intuition that that there is a mutually reinforcing relationship between tweets and their authors that could be reflected in the rankings. We show that this framework can be parametrized to take into account user preferences, the popularity of tweets and their authors, and diversity. Experimental evaluation on a large dataset shows that our model outperforms competitive approaches by a large margin.

1 Introduction

Online micro-blogging services have revolutionized the way people discover, share, and distribute information. *Twitter* is perhaps the most popular such service with over 140 million active users as of 2012.[1] Twitter enables users to send and read text-based posts of up to 140 characters, known as *tweets*. Twitter users follow others or are followed. Being a follower on Twitter means that the user receives all the tweets from those she follows. Common practice of responding to a tweet has evolved into a well-defined markup culture (e.g., RT stands for retweet, '@' followed by an identifier indicates the user). The strict limit of 140 characters allows for quick and immediate communication in real time, whilst enforcing brevity. Moreover, the retweet mechanism empowers users to spread information of their choice beyond the reach of their original followers.

Twitter has become a prominent broadcasting medium, taking priority over traditional news sources (Teevan et al., 2011). Shared information through this channel spreads faster than would have been possible with conventional news sites or RSS feeds and can reach a far wider population base. However, the proliferation of user-generated content comes at a price. Over 340 millions of tweets are being generated daily amounting to thousands of tweets per second![2] Twitter's own search engine handles more than 1.6 billion search queries per day.[3] This enormous amount of data renders it infeasible to browse the entire Twitter network; even if this was possible, it would be extremely difficult for users to find information they are interested in. A hypothetical tweet recommendation system could

---

[1]For details see `http://blog.twitter.com/2012/03/twitter-turns-six.html`

[2]In fact, the peak record is 6,939 tweets per second, reported by `http://blog.twitter.com/2011/03/numbers.html`.

[3]See `http://engineering.twitter.com/2011/05/engineering-behind-twitters-new-search.html`

516

alleviate this acute information overload, e.g., by limiting the stream of tweets to those of interest to the user, or by discovering intriguing content outside the user's following network.

The tweet recommendation task is challenging for several reasons. Firstly, Twitter does not merely consist of a set of tweets. Rather, it contains many latent networks including the *following* relationships among users and the *retweeting* linkage (which indicates information diffusion). Secondly, the recommendations ought to be of interest to the user and likely to to attract user response (e.g., to be retweeted). Thirdly, recommendations should be personalized (Cho and Schonfeld, 2007; Yan et al., 2011), avoid redundancy, and demonstrate diversity. In this paper we present a graph-theoretic approach to tweet recommendation that attempts to address these challenges.

Our recommender operates over a heterogeneous network that connects the users (or authors) and the tweets they produce. The user network represents links among authors based on their *following* behavior, whereas the tweet network connects tweets based on content similarity. A third bipartite graph ties the two together. Tweet and author entities in this network are ranked simultaneously following a co-ranking algorithm (Zhou et al., 2007). The main intuition behind co-ranking is that there is a mutually reinforcing relationship between authors and tweets that could be reflected in the rankings. Tweets are important if they are related to other important tweets and authored by important users who in turn are related to other important users. The model exploits this mutually reinforcing relationship between tweets and their authors and couples two random walks, one on the tweet graph and one on the author graph, into a combined one. Rather than creating a global ranking over all tweets in a collection, we extend this framework to individual users and produce personalized recommendations. Moreover, we incorporate diversity by allowing the random walk on the tweet graph to be time-variant (Mei et al., 2010).

Experimental results on a real-world dataset consisting of 364,287,744 tweets from 9,449,542 users show that the co-ranking approach substantially improves performance over the state of the art. We obtain a relative improvement of 18.3% (in nDCG) and 7.8% (in MAP) over the best comparison system.

2   Related Work

Tweet Search   Given the large amount of tweets being posted daily, ranking strategies have become extremely important for retrieving information quickly. Many websites currently offer a real-time search service which returns ranked lists of Twitter posts or shared links according to user queries. Ranking methods used by these sites employ three criteria, namely recency, popularity and content relevance (Dong et al., 2010). State-of-art tweet retrieval methods include a linear regression model biased towards text quality with a regularization factor inspired by the hypothesis that documents similar in content may have similar quality (Huang et al., 2011). Duan et al. (2010) learn a ranking model using SVMs and features based on tweet content, the relations among users, and tweet specific characteristics (e.g., urls, number of retweets).

Tweet Recommendation   Previous work has also focused on tweet recommendation systems, assuming no explicit query is provided by the users. Collaborative filtering is perhaps the most obvious method for recommending tweets (Hannon et al., 2010). Chen et al. (2010) investigate how to select interesting URLs linked from Twitter and recommend the top ranked ones to users. Their recommender takes three dimensions into account: the source, the content topic, and social voting. Similarly, Abel et al. (2011a; 2011b; 2011c) recommend external websites linked to Twitter. Their method incorporates user profile modeling and temporal recency, but they do not utilize the social networks among users. R. et al. (2009) propose a diffusion-based recommendation framework especially for tweets representing critical events by constructing a diffusion graph. Hong et al. (2011) recommend tweets based on popularity related features. Ramage et al. (2010) investigate which topics users are interested in following a Labeled-LDA approach, by deciding whether a user is in the followee list of a given user or not. Uysal and Croft (2011) estimate the likelihood of a tweet being reposted from a user-centric perspective.

Our work also develops a tweet recommendation system. Our model exploits the information provided by the tweets and the underlying social networks in a unified co-ranking framework. Although

these sources have been previously used to search or recommend tweets, our model considers them simultaneously and produces a ranking that is informed by both. Furthermore, we argue that the graph-theoretic framework upon which co-ranking operates is beneficial as it allows to incorporate personalization (we provide user-specific rankings) and diversity (the ranking is optimized so as to avoid redundancy). The co-ranking framework has been initially developed for measuring scientific impact and modeling the relationship between authors and their publications (Zhou et al., 2007). However, the adaptation of this framework to the tweet recommendation task is novel to our knowledge.

3 Tweet Recommendation Framework

Our method operates over a heterogeneous network that connects three graphs representing the tweets, their authors and the relationships between them. Let G denote the heterogeneous graph with nodes V and edges E, and $G = (V, E) = (V_M \cup V_U, E_M \cup E_U \cup E_{MU})$. G is divided into three subgraphs, G_M, G_U and G_{MU}. $G_M = (V_M, E_M)$ is a weighted undirected graph representing the tweets and their relationships. Let $V_M = \{m_i | m_i \in V_M\}$ denote a collection of $|V_M|$ tweets and E_M the set of links representing relationships between them. The latter are established by measuring how semantically similar any two tweets are (see Section 3.4 for details). $G_U = (V_U, E_U)$ is an unweighted directed graph representing the social ties among Twitter users. $V_U = \{u_i | u_i \in V_U\}$ is the set of users with size $|V_U|$. Links E_U among users are established by observing their *following* behavior. $G_{MU} = (V_{MU}, E_{MU})$ is an unweighted bipartite graph that ties G_M and G_U together and represents tweet-author relationships. The graph consists of nodes $V_{MU} = V_M \cup V_U$ and edges E_{MU} connecting each tweet with all of its authors. Typically, a tweet m is written by only one author u. However, because of *retweeting* we treat all users involved in reposting a tweet as "co-authors". The three subnetworks are illustrated in Figure 1.

The framework includes three random walks, one on G_M, one on G_U and one on G_{MU}. A random walk on a graph is a Markov chain, its states being the vertices of the graph. It can be described by a square $n \times n$ matrix $\mathbf{M}$, where n is the number of vertices in the graph. $\mathbf{M}$ is a stochastic matrix prescribing
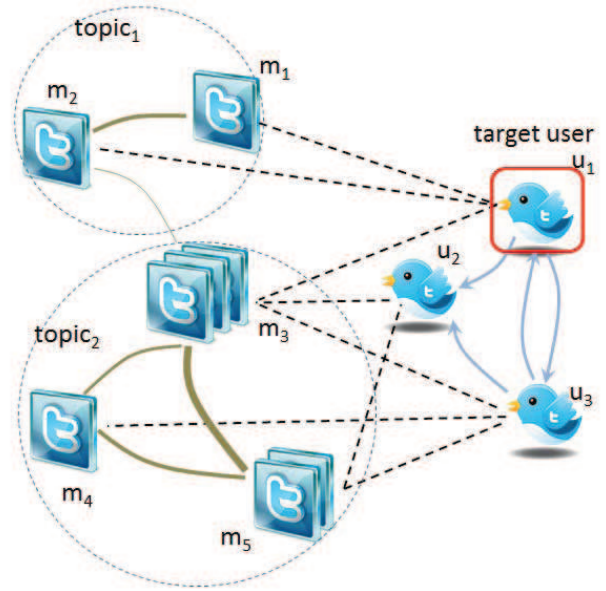

Figure 1: Tweet recommendation based on a co-ranking framework including three sub-networks. The undirected links between tweets indicate semantic correlation. The directed links between users denotes *following*. A bipartite graph (whose edges are shown with dashed lines) ties the tweet and author networks together.

the transition probabilities from one vertex to the next. The framework couples the two random walks on G_M, and G_U that rank tweets and theirs authors in isolation. and allows to obtain a more global ranking by taking into account their mutual dependence. In the following sections we first describe how we obtain the rankings on G_M and G_U, and then move on to discuss how the two are coupled.

3.1 Ranking the Tweet Graph

Popularity We rank the tweet network following the PageRank paradigm (Brin and Page, 1998). Consider a random walk on G_M and let $\mathbf{M}$ be the transition matrix (defined in Section 3.4). Fix some damping factor μ and say that at each time step with probability $(1\text{-}\mu)$ we stick to random walking and with probability μ we do not make a usual random walk step, but instead jump to any vertex, chosen uniformly at random:

$$\mathbf{m} = (1 - \mu)\mathbf{M}^{\mathrm{T}}\mathbf{m} + \frac{\mu}{|V_M|}\mathbf{1}\mathbf{1}^{\mathrm{T}} \qquad (1)$$

Here, vector $\mathbf{m}$ contains the ranking scores for the vertices in G_M. The fact that there exists a unique so-

518

lution to (1) follows from the random walk $\mathbf{M}$ being ergodic ($\mu > 0$ guarantees irreducibility, because we can jump to any vertex). $\mathbf{M}^T$ is the transpose of $\mathbf{M}$. $\mathbf{1}$ is the vector of $|V_M|$ entries, each being equal to one. Let $\mathbf{m} \in \mathbf{R}^{V_M}$, $||\mathbf{m}||_1 = 1$ be the only solution.

Personalization The standard PageRank algorithm performs a random walk, starting from any node, then randomly selects a link from that node to follow considering the weighted matrix $\mathbf{M}$, or jumps to a random node with equal probability. It produces a global ranking over all tweets in the collection without taking specific users into account. As there are billions of tweets available on Twitter covering many diverse topics, it is reasonable to assume that an average user will only be interested in a small subset (Qiu and Cho, 2006). We operationalize a user's topic preference as a vector $\mathbf{t} = [t_1, t_2, \ldots, t_n]_{1 \times n}$, where n denotes the number of topics, and t_i represents the degree of preference for topic i. The vector $\mathbf{t}$ is normalized such that $\sum_{i=1}^{n} t_i = 1$. Intuitively, such vectors will be different for different users. Note that user preferences can be also defined at the *tweet* (rather than *topic*) level. Although tweets can illustrate user interests more directly, in most cases a user will only respond to a small fraction of tweets. This means that most tweets will not provide any information relating to a user's interests. The topic preference vector allows to propagate such information (based on whether a tweet has been reposted or not) to other tweets within the same topic cluster.

Given n topics, we obtain a topic distribution matrix $\mathbf{D}$ using Latent Dirichlet Allocation (Blei et al., 2003). Let D_{ij} denote the probability of tweet m_i to belong to topic t_j. Consider a user with a topic preference vector $\mathbf{t}$ and topic distribution matrix $\mathbf{D}$. We calculate the response probability $\mathbf{r}$ for all tweets for this user as:

$$\mathbf{r} = \mathbf{t}\mathbf{D}^T \qquad (2)$$

where $\mathbf{r}=[r_1, r_2, \ldots, r_{V_M}]_{1 \times |V_M|}$ represents the response probability vector and r_i the probability for a user to respond to tweet m_i. We normalize $\mathbf{r}$ so that $\sum_{r_i \in \mathbf{r}} r_i = 1$. Now, given the observed response probability vector $\mathbf{r} = [r_1, r_2, \ldots, r_w]_{1 \times w}$, where $w < |V_M|$ for a given user and the topic distribution matrix $\mathbf{D}$, our task is estimate the topic preference vector $\mathbf{t}$. We do this using maximum-likelihood

estimation. Assuming a user has responded to w tweets, we approximate $\mathbf{t}$ so as to maximize the observed response probability. Let $\mathbf{r}(\mathbf{t}) = \mathbf{t}\mathbf{D}^T$. Assuming all responses are independent, the probability for w tweets $r_1, r_2, \ldots, r_w$ is then $\prod_{i=1}^{w} r_i(\mathbf{t})$ under a given $\mathbf{t}$. The value of $\mathbf{t}$ is chosen when the probability is maximized:

$$\mathbf{t} = \underset{\mathbf{t}}{\operatorname{argmax}} \left(\prod_{i=1}^{w} r_i(\mathbf{t}) \right) \qquad (3)$$

In a simple random walk, it is assumed that all nodes in the matrix $\mathbf{M}$ are equi-probable before the walk. In contrast, we use the topic preference vector as a prior on $\mathbf{M}$. Let $\operatorname{Diag}(\mathbf{r})$ denote a diagonal matrix whose eigenvalue is vector $\mathbf{r}$. Then $\mathbf{m}$ becomes:

$$\begin{aligned} \mathbf{m} &= (1-\mu)[\operatorname{Diag}(\mathbf{r})\mathbf{M}]^T \mathbf{m} + \mu\mathbf{r} \\ &= (1-\mu)[\operatorname{Diag}(\mathbf{t}\mathbf{D}^T)\mathbf{M}]^T \mathbf{m} + \mu\mathbf{t}\mathbf{D}^T \end{aligned} \qquad (4)$$

Diversity We would also like our output to be diverse without redundant information. Unfortunately, equation (4) will have the opposite effect, as it assigns high scores to closely connected node communities. A greedy algorithm such as Maximum Marginal Relevance (Carbonell and Goldstein, 1998; Wan et al., 2007; Wan et al., 2010) may achieve diversity by iteratively selecting the most prestigious or popular vertex and then penalizing the vertices "covered" by those that have been already selected. Rather than adopting a greedy vertex selection method, we follow DivRank (Mei et al., 2010) a recently proposed algorithm that balances popularity and diversity in ranking, based on a time-variant random walk. In contrast to PageRank, DivRank assumes that the transition probabilities change over time. Moreover, it is assumed that the transition probability from one state to another is reinforced by the number of previous visits to that state. At each step, the algorithm creates a dynamic transition matrix $\mathbf{M}^{(\cdot)}$. After z iterations, the matrix becomes:

$$\mathbf{M}^{(z)} = (1-\mu)\mathbf{M}^{(z-1)} \cdot \mathbf{m}^{(z-1)} + \mu\mathbf{t}\mathbf{D}^T \qquad (5)$$

and hence, $\mathbf{m}$ can be calculated as:

$$\mathbf{m}^{(z)} = (1-\mu)[\operatorname{Diag}(\mathbf{t}\mathbf{D}^T)\mathbf{M}^{(z)}]^T \mathbf{m} + \mu\mathbf{t}\mathbf{D}^T \qquad (6)$$

Equation (5) increases the probability for nodes with higher popularity. Nodes with high weights are

likely to "absorb" the weights of their neighbors directly, and the weights of their neighbors' neighbors indirectly. The process iteratively adjusts the matrix $\mathbf{M}$ according to $\mathbf{m}$ and then updates $\mathbf{m}$ according to the changed $\mathbf{M}$. Essentially, the algorithm favors nodes with high popularity and as time goes by there emerges a *rich-gets-richer* effect (Mei et al., 2010).

3.2 Ranking the Author Graph

As mentioned earlier, we build a graph of authors (and obtain the affinity $\mathbf{U}$) using the *following* linkage. We rank the author network using PageRank analogously to equation (1). Besides popularity, we also take personalization into account. Intuitively, users are likely to be interested in their friends even if these are relatively unpopular. Therefore, for each author, we include a vector $\mathbf{p} = [p_1, p_2, \ldots, p_{|V_U|}]_{1 \times |V_U|}$ denoting their preference for other authors. The preference factor for author u toward other authors u_i is defined as:

$$p_i^u = \frac{\#\text{tweets from } u_i}{\#\text{tweets of } u} \qquad (7)$$

which represents the proportion of tweets inherited from user u_i. A large p_i^u means that u is more likely to respond to u_i's tweets.

In theory, we could also apply DivRank on the author graph. However, as the authors are unique, we assume that they are sufficiently distinct and there is no need to promote diversity.

3.3 The Co-Ranking Algorithm

So far we have described how we rank the network of tweets G_M and their authors G_U independently following the PageRank paradigm. The co-ranking framework includes a random walk on G_M, G_U, and G_{MU}. The latter is a bipartite graph representing which tweets are authored by which users. The random walks on G_M and G_U are *intra-class* random walks, because take place either within the tweets' or the users' networks. The third (combined) random walk on G_{MU} is an *inter-class* random walk. It is sufficient to describe it by a matrix $\mathrm{MU}_{|V_M| \times |V_U|}$ and a matrix $\mathrm{UM}_{|V_U| \times |V_M|}$, since G_{MU} is bipartite. One intra-class step changes the probability distribution from $(\mathbf{m}, \mathbf{0})$ to $(M\mathbf{m}, \mathbf{0})$ or from $(\mathbf{0}, \mathbf{u})$ to $(\mathbf{0}, U\mathbf{u})$, while one inter-class step changes the probability distribution from $(\mathbf{m}, \mathbf{u})$ to $(\mathrm{UM}^T \mathbf{u}, \mathrm{MU}^T \mathbf{m})$. The

design of $\mathbf{M}$, $\mathbf{U}$, $\mathbf{MU}$ and $\mathbf{UM}$ is detailed in Section 3.4.

The two intra-class random walks are coupled using the inter-class random walk on the bipartite graph. The coupling is regulated by λ, a parameter quantifying the importance of G_{MU} versus G_M and G_U. In the extreme case, if λ is set to 0, there is no coupling. This amounts to separately ranking tweets and authors by PageRank. In general, λ represents the extent to which the ranking of tweets and their authors depend on each other.

There are two intuitions behind the co-ranking algorithm: (1) a tweet is important if it associates to other important tweets, and is authored by important users and (2) a user is important if they associate to other important users, and they write important tweets. We formulate these intuitions using the following iterative procedure:

Step 1 Compute tweet saliency scores:

$$\mathbf{m}^{(z+1)} = (1 - \lambda)([\mathrm{Diag}(\mathbf{r})\mathbf{M}^{(z)}]^T)\mathbf{m}^{(z)} + \lambda \mathrm{UM}^T \mathbf{u}^{(z)}$$
$$\mathbf{m}^{(z+1)} = \mathbf{m}^{(z+1)}/||\mathbf{m}^{(z+1)}|| \qquad (8)$$

Step 2 Compute author saliency scores:

$$\mathbf{u}^{(z+1)} = (1 - \lambda)([\mathrm{Diag}(\mathbf{p})\mathbf{U}]^T)\mathbf{u}^{(z)} + \lambda \mathrm{MU}^T \mathbf{m}^{(z)}$$
$$\mathbf{u}^{(z+1)} = \mathbf{u}^{(z+1)}/||\mathbf{u}^{(z+1)}|| \qquad (9)$$

Here, $\mathbf{m}^{(z)}$ and $\mathbf{u}^{(z)}$ are the ranking vectors for tweets and authors for the z-th iteration. To guarantee convergence, $\mathbf{m}$ and $\mathbf{u}$ are normalized after each iteration. Note that the tweet transition matrix $\mathbf{M}$ is dynamic due to the computation of diversity while the author transition matrix $\mathbf{U}$ is static. The algorithm typically converges when the difference between the scores computed at two successive iterations for any tweet/author falls below a threshold ε (set to 0.001 in this study).

3.4 Affinity Matrices

The co-ranking framework is controlled by four affinity matrices: $\mathbf{M}$, $\mathbf{U}$, $\mathbf{MU}$ and $\mathbf{UM}$. In this section we explain how these matrices are defined in more detail.

The tweet graph is an undirected weighted graph, where an edge between two tweets m_i and m_j represents their cosine similarity. An adjacency matrix $\mathbf{M}$

describes the tweet graph where each entry corresponds to the weight of a link in the graph:

$$M_{ij} = \frac{\mathcal{F}(m_i, m_j)}{\sum_k \mathcal{F}(m_i, m_k)}, \; \mathcal{F}(m_i, m_j) = \frac{\vec{m}_i \cdot \vec{m}_j}{||\vec{m}_i|| \, ||\vec{m}_j||} \quad (10)$$

where $\mathcal{F}(.)$ is the cosine similarity and $\vec{m}$ is a term vector corresponding to tweet m. We treat a tweet as a short document and weight each term with *tf.idf* (Salton and Buckley, 1988), where *tf* is the term frequency and *idf* is the inverse document frequency.

The author graph is a directed graph based on the *following* linkage. When u_i follows u_j, we add a link from u_i to u_j. Let the indicator function $I(u_i, u_j)$ denote whether u_i follows u_j. The adjacency matrix **U** is then defined as:

$$U_{ij} = \frac{I(u_i, u_j)}{\sum_k I(u_i, u_k)}, \; I(u_i, u_j) = \begin{cases} 1 \, \text{if } e_{ij} \in E_U \\ 0 \, \text{if } e_{ij} \notin E_U \end{cases} \quad (11)$$

In the bipartite tweet-author graph G_{MU}, the entry $E_{MU}(i, j)$ is an indicator function denoting whether tweet m_i is authored by user u_j:

$$\mathcal{A}(m_i, u_j) = \begin{cases} 1 & \text{if } e_{ij} \in E_{MU} \\ 0 & \text{if } e_{ij} \notin E_{MU} \end{cases} \quad (12)$$

Through E_{MU} we define **MU** and **UM**, using the weight matrices $\text{MU} = [\bar{W}_{ij}]$ and $\text{UM} = [\hat{W}_{ji}]$, containing the conditional probabilities of transitioning from m_i to u_j and vice versa:

$$\bar{W}_{ij} = \frac{\mathcal{A}(m_i, u_j)}{\sum_k \mathcal{A}(m_i, u_k)}, \quad \hat{W}_{ji} = \frac{\mathcal{A}(m_i, u_j)}{\sum_k \mathcal{A}(m_k, u_j)} \quad (13)$$

4 Experimental Setup

Data We crawled Twitter data from 23 seed users (who were later invited to manually evaluate the output of our system). In addition, we collected the data of their *followees* and *followers* by traversing the *following* edges, and exploring all newly included users in the same way until no new users were added. This procedure resulted in a relatively large dataset consisting of 9,449,542 users, 364,287,744 tweets, 596,777,491 links, and 55,526,494 retweets. The crawler monitored the data from 3/25/2011 to 5/30/2011. We used approximately one month of this data for training and the rest for testing.

Before building the graphs (i.e., the tweet graph, the author graph, and the tweet-author graph), the dataset was preprocessed as follows. We removed tweets of low linguistic quality and subsequently discarded users without any linkage to the remaining tweets. We measured linguistic quality following the evaluation framework put forward in Pitler et al. (2010). For instance, we measured the out-of-vocabulary word ratio (as a way of gauging spelling errors), entity coherence, fluency, and so on. We further removed stopwords and performed stemming.

Parameter Settings We ran LDA with 500 iterations of Gibbs sampling. The number of topics n was set to 100 which upon inspection seemed generally coherent and meaningful. We set the damping factor μ to 0.15 following the standard PageRank paradigm. We opted for more or less generic parameter values as we did not want to tune our framework to the specific dataset at hand. We examined the parameter λ which controls the balance of the tweet-author graph in more detail. We experimented with values ranging from 0 to 0.9, with a step size of 0.1. Small λ values place little emphasis on the tweet graph, whereas larger values rely more heavily on the author graph. Mid-range values take both graphs into account. Overall, we observed better performance with values larger than 0.4. This suggests that both sources of information — the content of the tweets and their authors — are important for the recommendation task. All our experiments used the same λ value which was set to 0.6.

System Comparison We compared our approach against three naive baselines and three state-of-the-art systems recently proposed in the literature. All comparison systems were subject to the same filtering and preprocessing procedures as our own algorithm. Our first baseline ranks tweets randomly (Random). Our second baseline ranks tweets according to token length: longer tweets are ranked higher (Length). The third baseline ranks tweets by the number of times they are reposted assuming that more reposting is better (RTnum). We also compared our method against Duan et al. (2010). Their model (RSVM) ranks tweets based on tweet content features and tweet authority features using the RankSVM algorithm (Joachims, 1999). Our fifth comparison system (DTC) was Uysal and Croft

(2011) who use a decision tree classifier to judge how likely it is for a tweet to be reposted by a specific user. This scenario is similar to ours when ranking tweets by retweet likelihood. Finally, we compared against Huang et al. (2011) who use weighted linear combination (WLC) to grade the relevance of a tweet given a query. We implemented their model without any query-related features as in our setting we do not discriminate tweets depending on their relevance to specific queries.

Evaluation We evaluated system output in two ways, i.e., automatically and in a user study. Specifically, we assume that if a tweet is retweeted it is relevant and is thus ranked higher over tweets that have not been reposted. We used our algorithm to predict a ranking for the tweets in the test data which we then compared against a goldstandard ranking based on whether a tweet has been retweeted or not. We measured ranking performance using the normalized Discounted Cumulative Gain (nDCG; Järvelin and Kekäläinen (2002)):

$$\text{nDCG}(k, V_U) = \frac{1}{|V_U|} \sum_{u \in V_U} \frac{1}{Z_u} \sum_{i=1}^{k} \frac{2^{r_i^u} - 1}{\log(1 + i)} \quad (14)$$

where V_U denotes users, k indicates the top-k positions in a ranked list, and Z_u is a normalization factor obtained from a perfect ranking for a particular user. r_i^u is the relevance score (i.e., 1: retweeted, 0: not retweeted) for the i-th tweet in the ranking list for user u.

We also evaluated system output in terms of Mean Average Precision (MAP), under the assumption that retweeted tweets are relevant and the rest irrelevant:

$$\text{MAP} = \frac{1}{|V_U|} \sum_{u \in V_U} \frac{1}{N_u} \sum_{i=1}^{k} P_i^u \times r_i^u \quad (15)$$

where N_u is the number of reposted tweets for user u, and P_i^u is the precision at i-th position for user u (Manning et al., 2008).

The automatic evaluation sketched above does not assess the full potential of our recommendation system. For instance, it is possible for the algorithm to recommend tweets to users with no linkage to their publishers. Such tweets may be of potential interest, however our goldstandard data can only provide information for tweets and users with *following* links.

System	nDCG@5	nDCG@10	nDCG@25	nDCG@50	MAP
Random	0.068	0.111	0.153	0.180	0.167
Length	0.275	0.288	0.298	0.335	0.258
RTNum	0.233	0.219	0.225	0.249	0.239
RSVM	0.392	0.400	0.421	0.444	0.558
DTC	0.441	0.468	0.492	0.473	0.603
WLC	0.404	0.421	0.437	0.464	0.592
CoRank	**0.519**	**0.546**	**0.550**	**0.585**	**0.617**

Table 1: Evaluation of tweet ranking output produced by our system and comparison baselines against goldstandard data.

System	nDCG@5	nDCG@10	nDCG@25	nDCG@50	MAP
Random	0.081	0.103	0.116	0.107	0.175
Length	0.291	0.307	0.246	0.291	0.264
RTNum	0.258	0.318	0.343	0.346	0.257
RSVM	0.346	0.443	0.384	0.414	0.447
DTC	0.545	0.565	0.579	0.526	0.554
WLC	0.399	0.447	0.460	0.481	0.506
CoRank	**0.567**	**0.644**	**0.715**	**0.643**	**0.628**

Table 2: Evaluation of tweet ranking output produced by our system and comparison baselines against judgments elicited by users.

We therefore asked the 23 users whose Twitter data formed the basis of our corpus to judge the tweets ranked by our algorithm and comparison systems. The users were asked to read the systems' recommendations and decide for every tweet presented to them whether they would retweet it or not, under the assumption that retweeting takes place when users find the tweet interesting.

In both automatic and human-based evaluations we ranked all tweets in the test data. Then for each date and user we selected the top 50 ones. Our nDCG and MAP results are averages over users and dates.

5 Results

Our results are summarized in Tables 1 and 2. Table 1 reports results when model performance is evaluated against the gold standard ranking obtained from the Twitter network. In Table 2 model performance is compared against rankings elicited by users.

As can be seen, the Random method performs worst. This is hardly surprising as it recommends tweets without any notion of their importance or user interest. Length performs considerably better than

System	nDCG@5	nDCG@10	nDCG@25	nDCG@50	MAP
PageRank	0.493	0.481	0.509	0.536	0.604
PersRank	0.501	0.542	**0.558**	0.560	0.611
DivRank	0.487	0.505	0.518	0.523	0.585
CoRank	**0.519**	**0.546**	0.550	**0.585**	**0.617**

Table 3: Evaluation of individual system components against goldstandard data.

System	nDCG@5	nDCG@10	nDCG@25	nDCG@50	MAP
PageRank	0.557	0.549	0.623	0.559	0.588
PersRank	0.571	0.595	0.655	0.613	0.601
DivRank	0.538	0.591	0.594	0.547	0.589
CoRank	**0.637**	**0.644**	**0.715**	**0.643**	**0.628**

Table 4: Evaluation of individual system components against human judgments.

Random. This might be due to the fact that informativeness is related to tweet length. Using merely the number of retweets does not seem to capture the tweet importance as well as Length. This suggests that highly retweeted posts are not necessarily informative. For example, in our data, the most frequently reposted tweet is a commercial advertisement calling for reposting!

The supervised systems (RSVM, DTC, and WLC) greatly improve performance over the naive baselines. These methods employ standard machine learning algorithms (such as SVMs, decision trees and linear regression) on a large feature space. Aside from the learning algorithm, their main difference lies in the selection of the feature space, e.g., the way content is represented and whether authority is taken into account. DTC performs best on most evaluation criteria. However, neither DTC nor RSVM, or WLC take personalization into account. They generate the same recommendation lists for all users. Our co-ranking algorithm models user interest with respect to the content of the tweets and their publishers. Moreover, it attempts to create diverse output and has an explicit mechanism for minimizing redundancy. In all instances, using both DCG and MAP, it outperforms the comparison systems. Interestingly, the performance of CoRank is better when measured against human judgments. This indicates that users are interested in tweets that fall outside the scope of their followers and that recommendation can improve user experience.

We further examined the contribution of the individual components of our system to the tweet recommendation task. Tables 3 and 4 show how the performance of our co-ranking algorithm varies when considering only tweet popularity using the standard PageRank algorithm, personalization (PersRank), and diversity (DivRank). Note that DivRank is only applied to the tweet graph. The PageRank algorithm on its own makes good recommendations, while incorporating personalization improves the performance substantially, which indicates that individual users show preferences to specific topics or other users. Diversity on its own does not seem to make a difference, however it improves performance when combined with personalization. Intuitively, users are more likely to repost tweets from their followees, or tweets closely related to those retweeted previously.

6 Conclusions

We presented a co-ranking framework for a tweet recommendation system that takes popularity, personalization and diversity into account. Central to our approach is the representation of tweets and their users in a heterogeneous network and the ability to produce a global ranking that takes both information sources into account. Our model obtains substantial performance gains over competitive approaches on a large real-world dataset (it improves by 18.3% in DCG and 7.8% in MAP over the best baseline). Our experiments suggest that improvements are due to the synergy of the two information sources (i.e., tweets and their authors). The adopted graph-theoretic framework is advantageous in that it allows to produce user-specific recommendations and incorporate diversity in a unified model. Evaluation with actual Twitter users shows that our recommender can indeed identify interesting information that lies outside the the user's immediate following network. In the future, we plan to extend the co-ranking framework so as to incorporate information credibility and temporal recency.

References

Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. 2011a. Analyzing temporal dynamics in Twitter profiles for personalized recommendations in the social web. In *Proceedings of the ACM Web Science Conference 2011*, pages 1–8, Koblenz, Germany.

Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. 2011b. Analyzing user modeling on Twitter for personalized news recommendations. *User Modeling, Adaptation and Personalization*, pages 1–12.

Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. 2011c. Semantic enrichment of twitter posts for user profile construction on the social web. *The Semantic Web: Research and Applications*, pages 375–389.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet aladdress. *Journal of Machine Learning Research*, 3:993–1022.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Proceedings of the 7th International Conference on World Wide Web*, 30(1-7):107–117.

Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 335–336, Melbourne, Australia.

Jilin Chen, Rowan Nairn, Les Nelson, Michael Bernstein, and Ed Chi. 2010. Short and tweet: experiments on recommending content from information streams. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems*, pages 1185–1194, Atlanta, Georgia.

Junghoo Cho and Uri Schonfeld. 2007. Rankmass crawler: a crawler with high personalized pagerank coverage guarantee. In *Proceedings of the 33rd International Conference on Very Large Data Bases*, pages 375–386, Vienna, Austria.

Anlei Dong, Ruiqiang Zhang, Pranam Kolari, Jing Bai, Fernando Diaz, Yi Chang, Zhaohui Zheng, and Hongyuan Zha. 2010. Time is of the essence: improving recency ranking using Twitter data. In *Proceedings of the 19th International Conference on World Wide Web*, pages 331–340, Raleigh, North Carolina.

Yajuan Duan, Long Jiang, Tao Qin, Ming Zhou, and Heung-Yeung Shum. 2010. An empirical study on learning to rank of tweets. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 295–303, Beijing, China.

John Hannon, Mike Bennett, and Barry Smyth. 2010. Recommending twitter users to follow using content and collaborative filtering approaches. In *Proceedings of the 4th ACM Conference on Recommender Systems*, pages 199–206, Barcelona, Spain.

Liangjie Hong, Ovidiu Dan, and Brian D. Davison. 2011. Predicting popular messages in Twitter. In *Proceedings of the 20th International Conference Companion on World Wide Web*, pages 57–58, Hyderabad, India.

Minlie Huang, Yi Yang, and Xiaoyan Zhu. 2011. Quality-biased ranking of short texts in microblogging services. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 373–382, Chiang Mai, Thailand.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20:422–446.

Thorsten Joachims. 1999. Making large-scale svm learning practical. In *Advances in Kernel Methods: Support Vector Learning*, pages 169–184. MIT press.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze. 2008. *Introduction to Information Retrieval*, volume 1. Cambridge University Press.

Qiaozhu Mei, Jian Guo, and Dragomir Radev. 2010. Divrank: the interplay of prestige and diversity in information networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1009–1018, Washington, DC.

Emily Pitler, Annie Louis, and Ani Nenkova. 2010. Automatic evaluation of linguistic quality in multi-document summarization. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 544–554, Uppsala, Sweden.

Feng Qiu and Junghoo Cho. 2006. Automatic identification of user interest for personalized search. In *Proceedings of the 15th International Conference on World Wide Web*, pages 727–736, Edinburgh, Scotland.

Sun Aaron R., Cheng Jiesi, Zeng, and Daniel Dajun. 2009. A novel recommendation framework for microblogging based on information diffusion. In *Proceedings of the 19th Annual Workshop on Information Technologies and Systems*, pages 199–216, Phoenix, Arizona.

Daniel Ramage, Susan Dumais, and Dan Liebling. 2010. Characterizing microblogs with topic models. In *International AAAI Conference on Weblogs and Social Media*, pages 130–137. The AAAI Press.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523.

Jaime Teevan, Daniel Ramage, and Meredith Ringel Morris. 2011. #Twittersearch: a comparison of microblog search and web search. In *Proceedings of the 4th ACM*

International Conference on Web Search and Data Mining, pages 35–44, Hong Kong, China.

Ibrahim Uysal and W. Bruce Croft. 2011. User oriented tweet ranking: a filtering approach to microblogs. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pages 2261–2264, Glasgow, Scotland.

Xiaojun Wan, Jianwu Yang, and Jianguo Xiao. 2007. Single document summarization with document expansion. In *Proceedings of the 22nd Conference on Artificial Intelligence*, pages 931–936, Vancouver, British Columbia.

Xiaojun Wan, Huiying Li, and Jianguo Xiao. 2010. Cross-language document summarization based on machine translation quality prediction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 917–926, Uppsala, Sweden.

Rui Yan, Jian-Yun Nie, and Xiaoming Li. 2011. Summarize what you are interested in: An optimization framework for interactive personalized summarization. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1342–1351. Association for Computational Linguistics.

Ding Zhou, Sergey A. Orshanskiy, Hongyuan Zha, and C. Lee Giles. 2007. Co-ranking authors and documents in a heterogeneous network. In *Proceedings of the 7th IEEE International Conference on Data Mining*, pages 739–744. IEEE.