

Making Your Interests Follow You on Twitter

Marco Pennacchiotti
eBay Inc., San Jose, USA
mpennacchiotti@ebay.com

Hossein Vahabi
IMT, Lucca, Italy
ISTI - CNR, Pisa, Italy
h.vahabi@isti.cnr.it

Fabrizio Silvestri
ISTI - CNR, Pisa, Italy
f.silvestri@isti.cnr.it

Rossano Venturini
Dept. of Computer Science,
University of Pisa, Italy
rossano@di.unipi.it

ABSTRACT

In this paper we introduce the task of *tweet recommendation*, the problem of suggesting tweets that match a user's interests and likes. We propose an Information-Retrieval-like model that leverages the content of the user's tweets and those of her friends, and that effectively retrieves a set of tweets that is personalized and varied in nature. Our approach could be easily leveraged to build, for example, a Twitter or Facebook timeline that collects messages that are of interest for the user, but that are not posted by her friends. We compare to typical approaches used in similar tasks, reporting significant gains in terms of overall precision, up to about +20%, on both a corpus-based evaluation and real world user study.

Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous;
H.3.4 [INFORMATION STORAGE AND RETRIEVAL]:
Systems and Software—*User profiles and alert services*

General Terms

Algorithms, Experimentation

Keywords

Twitter recommendation, Information Filtering

1. INTRODUCTION

Social media is having an unprecedented success in terms of popularity in recent years. Twitter and Facebook, the two most popular social media hubs, occupy high positions in the top-10 of the most visited sites, rivaled only by big search engines and by Wikipedia¹. In January 2012 Twitter has been visited 2.5 billion times, more than double than

¹<http://www.alexa.com/topsites>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.
Copyright 2012 ACM 978-1-4503-1156-4/12/10 ...\$15.00.

6 months before². In October 2011, Twitter revealed that 250 million tweets are posted every day, with a user base of about 300 million people.

With a rate of roughly 3,000 tweets per seconds, social media are today facing the same challenge of information overloading that the Web faced more than 10 years ago. More than that, we are today also facing the challenge of *information hiding*. By no means can a Twitter user possibly read all the messages s/he might consider important. Indeed, only a very small percentage of tweets spreads to a significant number of users [25]. It is quite likely that users miss lots of interesting tweets just because none of the people in their network retweet or mention them. Social search engines can help to solve the problem by providing a simple means to retrieve information. However, these engines cannot predict what the user may like and push interesting information for her. The solution is therefore a combination of social search and algorithms for personalizing, summarizing and recommending social content.

In this paper we focus on the latter task of social content recommendation. We propose two Information Retrieval inspired methods for recommending engaging tweets to a user, matching her interests and likes. The methods are based on the idea of building a user profile from her tweets and a weighted combination of those posted by her friends, and to match this profile on incoming tweets. Our extensive evaluation performed on both an in-house dataset and on real users, shows that our methods successfully recommend interesting tweets when compared to other approaches. Our system differs from most previous work on social recommendation, where the focus is predominantly on recommending web content using social signals, and on recommending new connections. Our primary objective here is to recommend the social content itself, i.e., tweets. Although the methods we use to approach the tweet recommendation problem are derived from information retrieval techniques, their application to our context requires, as we shall see in the rest of the paper, some adjustments and additions to make them work in social media settings.

The most direct practical application of our system is to provide a social media user with a new timeline³ that contains messages that strongly match her interests, but that have not been posted by any of her friends. The benefits of

²<http://www.quantcast.com/twitter.com>

³The user entry point in most social media hubs like Twitter and Facebook is the personal timeline, where the user is shown the recent messages posted by her friends.

having this additional timeline are twofold. Firstly, the user will not miss messages that are relevant to her, because our method will show them. Second, authors of recommended tweets may be considered relevant by the user who receives the recommendation. Our tweet recommendation system may therefore indirectly serve also as a mechanism to “*implicitly*” recommend accounts to follow.

Summing up, the original contributions of this paper are:

- We define two novel models for tweet recommendation: TWEETREC aims at recommending the most interesting tweets; INTERESTS-SPANNING TWEETREC aims at recommending a set of tweets, minimizing overlapping informative content. As a result, INTERESTS-SPANNING TWEETREC provides more diverse recommendations.
- The two proposed models define a measure of tweet *interestingness* for a user. We propose an effective measure to estimate interestingness by considering the content of tweets posted by the user herself, and also by her *most authoritative* friends. We prove the impact of the newly proposed measure by using both an automatic and user-study-based evaluation.

The paper is structured as follows. Section 2 presents an overview of the current state of the art in areas that are related to our work. Section 3 presents a formal definition of the problem, while Section 4 describes the proposed solutions. Experiments are presented in Section 5 along with a careful description of the datasets we are using and a definition of the metrics we evaluate. We conclude the paper in Section 6 with a final analysis and future work.

2. RELATED WORK

Research on recommender systems has been widely studied in the past decade (e.g., see [22, 4, 7, 21]). Recommenders have been applied to many different items including movies, videos, music, users, restaurants, news stories, journal articles, and much more (see [22] and references therein). To the best of our knowledge, tweet recommendation has not been yet explicitly defined. This task relates to four main areas of research: *tweet-based content recommendation*, *tweet ranking*, *tweet classification*, and *information spreading in social networks*.

Tweet-based content recommendation systems use Twitter for recommending other types of content. Among others, Abel *et al.* [1] propose a URL recommendation system for Twitter users. They build a linguistic model for the user, as the frequency weighted vector of the hashtags and entities that he/she mentions in his/her tweets. A similar vector is computed from the content of candidate URLs. The web pages that have the highest cosine similarity with the user are recommended [1]. Chen *et al.* present similar models for the same task, where the user profile is represented as the bag-of-words vector of all terms in the user’s tweets (or his/her friends’ tweets); and the URL model is built from the terms in the tweet that mention it [8]. Our work differs from the above two, in that we recommend tweets instead of URLs, which is a harder task since there is less linguistic material available. Yet, like them, we leverage the linguistic content of the user’s tweets and of his/her friends’ tweets.

Tweet ranking is the task of ranking tweets given an input query. Duan *et al.* [11] use user and social features to

feed a RankSVM algorithm. They report good DGC results by using a variety of information, including user authority features and cosine similarity between the texts of the retrieved tweets. Similar ranking systems have been proposed at the TREC-2011 Microblog Track⁴. In general, typical models for web document ranking such as BM25 and text similarity, can be easily adapted to tweet ranking, showing good performance. Yet, microblogging specific features, such as social feedback and annotations (e.g., number of retweets and hashtags), the authority of the user posting the tweet, tweet length and freshness also play a key role. Tweets have also been used to improve document ranking: Dong *et al.* [10] show that integrating Twitter features (e.g., user and textual properties of the tweets mentioning the URLs) significantly improve state of the art web search algorithms, in ranking documents by relevance and freshness. The task presented in this paper differs from tweet ranking, in that our task is query independent and user centric. Instead of ranking tweets for a given query, we recommend tweets to a user according to his/her interests. There are however similarities between the two tasks, e.g., in both cases there is a specific user need to answer, and similar features can be used.

Short texts classification is a relatively new problem that is receiving more and more attention. Tweet classification is a specialization of this problem where text fragments propagate through social networks. The main issue for classification of microblog messages is their small size. The same issue holds also in different domains where fragments of text are small. Take for instance search snippets, forum messages and Instant Message service texts. The main issue is that messages are small in size, and therefore difficult to classify. Several studies have approached this problem from different angles. Banerjee *et al.* [3] exploit Wikipedia as an external source for improving the accuracy of clustering short texts by enriching their representation with features coming from Wikipedia pages related to terms appearing in the short text. Schonhofen [24] presents a similar approach exploiting Wikipedia as a taxonomy for topics showing significant improvements on clustering accuracy. Phan *et al.* [20] use topic models for building short and sparse text classifiers. They build a Latent Dirichlet Allocation [5] topic model from this dataset, and then use it to classify short texts.

Information spreading in social networks is another important and well-studied problem in social networks. This research area is important as results can be immediately exploited in several ways, e.g., to spread a marketing campaign or to block spread diseases. One of the seminal papers dealing with this problem is that of Kempe *et al.* [16]. In the paper an approximation algorithm to find the most influential nodes in a network is devised. The algorithm is further refined by Leskovec *et al.* [18] and by Goyal *et al.* [14]. A similar problem, but with the opposite purpose, is defined by Budak *et al.* [6]. The goal of their algorithm is to block a “*bad*” advertisement campaign as soon as it is initiated. As in the case of Kempe *et al.* [16] authors formulate the problem as an optimization problem described by a sub-modular objective function. The standard greedy algorithm for this kind of functions [12] is used with the goal of identifying a subset of individuals that need to be convinced to adopt

⁴sites.google.com/site/microblogtrack

the competing (or “good”) campaign so as to minimize the number of people that adopt the “bad” campaign at the end of both propagation processes.

3. PROBLEM DEFINITION

We give here a definition of the notation and the problems we shall study in the rest of the paper. Let \mathcal{T} be a stream of tweets, each tweet $t_1, t_2, \dots \in \mathcal{T}$ is indexed by the timestamp at which it arrives. For each user u , we assume to know the *interestingness* of a tweet $t \in \mathcal{T}$, denoted by $\mathcal{I}_u(t)$. Hereinafter, we consider a single user u and we omit the subscript u from \mathcal{I}_u whenever the user can be inferred from the context.

We shall define two problems whose goal is to select a subset of tweets $S \subseteq \mathcal{T}$ to be recommended to a given user. The first problem aims at selecting $S \subseteq \mathcal{T}$ of size k , without considering the goodness of the set in its entirety. Namely, each tweet in S is selected independently from the already selected tweets. We refine the idea of tweet subset selection in the second problem, where the set of tweets $S \subseteq \mathcal{T}$ is further required to maximize the overall information conveyed to u . To the best of our knowledge both problems are novel and have never been proposed before in the context of social media systems such as Twitter.

We begin by introducing the first problem, namely the TweetRec problem, defined as follows.

PROBLEM 1 (TWEETREC). *Given a user u and a positive integer k , we aim at finding a set S of k tweets in \mathcal{T} maximizing the overall “interestingness”. More formally, we would like to find*

$$\max_{\substack{S \subseteq \mathcal{T} \\ |S|=k}} \sum_{t \in S} \mathcal{I}(t) \quad (1)$$

Essentially, we are looking for the set of k tweets S having the largest interestingness under the assumption that the tweets independently convey their interesting content. The nicest property of TWEETREC is that its solution is optimally discoverable in $O(|\mathcal{T}| \log k)$ time by the obvious greedy algorithm selecting the most interesting, (i.e., top- k) tweets. The drawback of this solution is that we are assuming independence of tweet content in the solution set. This, indeed, is far from being satisfactory for a user. Suppose, in fact, that all the tweets selected for being included in the set S are highly correlated to each other. This represents a feasible solution of TWEETREC but the user may be far from being satisfied. For instance, each of these two tweets “What’s new in Linux 3.2? #linux” and “New features in Linux 3.2. #linux” may be highly interesting for a user but reporting both of them is useless.

The issue in TWEETREC is given by the fact that we are not considering interdependency among selected tweets. The objective function in TWEETREC assumes the independence of all the tweets and does not consider the *overall* interestingness of the identified set S in its entirety.

In order to overcome this limitation, we restate the tweet recommendation problem in an “idependency aware” setting, where shared (i.e., duplicated) information is avoided. Given $S \subseteq \mathcal{T}$, we want to maximize the *overall* interestingness $F(S)$ of the content of tweets $t \in S$ for the user u . Formally, $F(S) = \mathcal{I}\left(\bigsqcup_{t \in S} t\right)$. In this definition *overall* means that duplicate information across tweets is not going to increase

the value of the objective function $F(S)$. We also define $\mathcal{I}\left(\prod_{t \in S} t\right)$ as the interestingness for user u in the *shared* informative content among all tweets in $t \in S$.

PROBLEM 2 (INTERESTS-SPANNING TWEETREC). *Given a user u , a positive integer k , and the stream of tweets \mathcal{T} , we aim at finding the k tweets t in \mathcal{T} maximizing the overall interestingness. More formally, we would like to identify the set \mathcal{T} such that*

$$\max_{\substack{S \subseteq \mathcal{T} \\ |S|=k}} F(S) = \sum_{j=1}^k \left((-1)^{j+1} \sum_{\substack{I \subseteq S \\ |I|=j}} \mathcal{I}\left(\prod_{t \in I} t\right) \right) \quad (2)$$

The derivation of the equation in the objective function of INTERESTS-SPANNING TWEETREC is an instantiation of the Inclusion-Exclusion principle to the set $S \subseteq \mathcal{T}$ of tweets. As in TWEETREC, we are looking for the set of k tweets S having the largest overall interestingness for the user u . However, in this case we are taking into consideration all possible dependencies, i.e., shared content among the tweets in the set.

INTERESTS-SPANNING TWEETREC is NP-Hard as proven in the following Theorem.

THEOREM 1. *The INTERESTS-SPANNING TWEETREC problem is NP-hard.*

PROOF. The reduction is from Independent Set [13]. Assume we have a graph $G = (V, E)$ with $n = |V|$ and we want to compute the maximum independent set of G . The independent set can be computed by creating an appropriate instance of our problem. Vertexes in V are considered as tweets. Each vertex is assigned an interestingness equal to $\frac{1}{n}$. Any set S has an overall interestingness equal to $\frac{1}{n} |S|$ if and only if it does not exist a pair of vertexes in S connected by an edge in E . The interestingness is set to be smaller than $\frac{1}{n} |S|$, otherwise. To conclude the reduction we observe that, fixed k , the set S identified by a solution of INTERESTS-SPANNING TWEETREC has probability $\frac{k}{n}$ if and only if G has an independent set of size k . In this case, nodes in S are exactly the nodes of one of these independent sets. \square

Despite being complex, the objective function of our problem has the property of being a non-negative *submodular* function, as stated in the following Theorem 2. This theorem requires that the function F is monotone (namely, any set S of tweets is at most as interesting as any of its subsets). Obviously, whether \mathcal{I} is monotone depends on its particular instantiation (i.e., the real formula used to its estimation). The monotonicity is a natural assumption to make and, indeed, it holds in our estimation of \mathcal{I} defined in Section 4.

THEOREM 2 (SUBMODULARITY OF $F(X)$). *Let k be a positive integer and let S be a set of k tweets. Then, the function*

$$F(S) = \sum_{j=1}^k \left((-1)^{j+1} \sum_{\substack{I \subseteq X \\ |I|=j}} \mathcal{I}\left(\prod_{t \in I} t\right) \right)$$

is submodular.

PROOF. We have to show, by the definition of submodularity, that given $A \subseteq B \subseteq \mathcal{T}$ and a tweet $c \in \mathcal{T}$, $F(\{c\} \cup A) - F(A) \geq F(\{c\} \cup B) - F(B)$.

The theorem is proven by observing that since $F(X) = \mathcal{I}\left(\bigsqcup_{t \in X} t\right)$ then $F(\{c\} \cup X) = \mathcal{I}(c) + \mathcal{I}\left(\bigsqcup_{t \in X} t\right) - \mathcal{I}\left(c \sqcap \bigsqcup_{t \in X} t\right)$.

Thus, $F(\{c\} \cup A) - F(A) = \mathcal{I}(c) - \mathcal{I}\left(c \sqcap \bigsqcup_{t \in A} t\right)$. Similarly, $F(\{c\} \cup B) - F(B) = \mathcal{I}(c) - \mathcal{I}\left(c \sqcap \bigsqcup_{t \in B} t\right)$.

The thesis easily follows by observing that $\mathcal{I}\left(c \sqcap \bigsqcup_{t \in A} t\right) \leq \mathcal{I}\left(c \sqcap \bigsqcup_{t \in B} t\right)$ since $A \subseteq B \subseteq \mathcal{T}$.

□

Given that our objective function is submodular, its maximization can be obtained by the greedy algorithm presented in [9, 19]. This algorithm incrementally builds the set S in k steps. At each step, the element that has the largest marginal gain is added. This simple algorithm guarantees to obtain a $(1 - 1/e)$ approximation. The following Theorem summarizes this very well-known result.

THEOREM 3 (SUBMODULARITY APPROXIMATION). [9, 19] *For a non-negative, monotone submodular function F , let S be a set of size k obtained by selecting elements one at a time, each time choosing an element that provides the largest marginal increase in the function value. Let S^* be a set that maximizes the value of F over all k -element sets. Then $F(S) \geq (1 - 1/e)F(S^*)$.*

4. INTERESTINGNESS ESTIMATION

The TWEETREC and INTERESTS-SPANNING TWEETREC problem definitions give a good formalization for a tweet recommendation system. Yet, these formalizations assume that for any user u and tweet $t \in \mathcal{T}$, we know the score $\mathcal{I}(t)$ measuring the interestingness of the tweet t for the user u . In order to build a functioning recommendation system we therefore need an estimator of interestingness.

In this section, we present one example of an estimator based on a linguistic model. The main assumption is that the interests of a user u are implicitly expressed in his/her tweets. By comparing the linguistic content of the user's tweets and that of a generic tweet t , we can therefore estimate the interestingness $\mathcal{I}(t)$.

In details, we estimate the interestingness as a linear combination of two distinct similarity measures between the set of tweets of the user and a candidate tweet (or set of tweets). In these two measures, sets are compared either by using *item-wise* or *pair-wise* comparisons between their enclosed terms. Our estimation is detailed in Section 4.1.

A drawback of the above-mentioned measure is that it heavily relies on the (textual content of the) set of tweets produced by the user. Indeed, there exist (a class of) users, namely *passive users*, that post a small number of tweets in their lifetime, and adopt Twitter mostly for reading news published by other users. In Section 4.2, we show how to extend the set of tweets written by a passive user u with other carefully chosen tweets that have been posted by users that are highly authoritative.

4.1 Exploiting textual similarities

In this section, we show how to estimate $\mathcal{I}(t)$ by computing the similarity between the set of user's tweets T_u and any subset of tweets from \mathcal{T} .

First, given a generic set of tweets X , we define the *bag-of-words* $\text{BT}(X) = \{\omega_1, \omega_2, \dots, \omega_n\}$ as the set of distinct terms ω_i contained in at least a tweet $t \in X$. Similarly, we define the *bag-of-pairs* $\text{BP}(X) = \{\rho_1, \rho_2, \dots, \rho_n\}$ as the set of distinct *pairs* of terms ρ_i that co-occur (possibly not consecutively) in at least a tweet $t \in X$.

We define two score functions for terms and pairs with respect to a set of tweets X .

DEFINITION 1 (TERM SCORE). *Given a term ω and a set of tweets X , we define the score of ω given X as*

$$\text{tscore}(\omega, X) = \text{TF}_\tau(\omega, X) \cdot \text{IDF}_\tau(\omega)$$

where $\text{TF}_\tau(\omega, X)$ is the number of tweets in X containing term ω , $\text{IDF}_\tau(\omega) = \log \frac{|\mathcal{T}|}{\text{DF}_\tau(\omega)}$, and DF_τ is the number of tweets in the stream \mathcal{T} containing ω .

As we will show in the experimental section, the sum of the $\text{tf} - \text{idf}$ [23] of the terms used by the user is particularly useful for tweets. Indeed, apart from increasing the importance of co-occurrences of rare terms, it privileges longer tweets with respect to shorter ones.

Similarly to what we have just done we define a score function for *pairs* of terms.

DEFINITION 2 (PAIR SCORE). *Given a pair of terms ρ and a set of tweets X , we define the score of ρ given X as*

$$\text{pscore}(\rho, X) = \text{TF}_\rho(\rho) \cdot \text{IDF}_\rho(\rho) \quad (3)$$

where TF_ρ is the number of tweets in X containing pair of terms ρ , $\text{IDF}_\rho = \log \frac{|\mathcal{T}|}{\text{DF}_\rho(\rho)}$, and $\text{DF}_\rho(\rho)$ is the number of tweets in the stream \mathcal{T} containing the pair of terms ρ .

The rationale behind the measure defined by Equation 3 is that since tweets are relatively small documents usually consisting of few sentences, there are good chances that two tweets having pairs of terms in common have higher correlation than tweets sharing only terms.

We observe that both scores implicitly consider the overlapping content between sets of tweets only once. This is an important aspect of the scoring functions that complies with the concept of overall interestingness discussed in the previous section.

Finally, by combining Definitions 1 and 2, we define a score that establishes the similarity of a set of tweets with respect to another.

DEFINITION 3 (SET SIMILARITY). *Given two sets of tweets X_1 and X_2 , we define the similarity of X_1 with respect to X_2 as*

$$\text{int}_\lambda(X_1, X_2) = (1 - \lambda) \cdot \sum_{\omega \in \text{BT}(X_1) \cap \text{BT}(X_2)} \text{tscore}(\omega, X_2) + \lambda \cdot \sum_{\rho \in \text{BP}(X_1) \cap \text{BP}(X_2)} \text{pscore}(\rho, X_2),$$

where $\lambda \in [0, 1]$ is the linear combination coefficient.

Observe that the similarity is computed by considering only terms whenever λ is equal to 0. Conversely, the similarity is computed by considering only pairs of terms when λ is equal to 1. Note that it is possible to define a score

for triples of terms or more, similar to what we have done for terms and term pairs. In that case the function int_λ is defined by a linear combination of different scores, using coefficients $\lambda_1, \lambda_2, \lambda_3, \dots$ such that $\lambda_1 + \lambda_2 + \lambda_3 + \dots = 1$.

Given the above definitions, we are now ready to estimate $\mathcal{I}(t)$. Given a user u and his set of tweets T_u , for any value $\lambda \in [0, 1]$ we estimate the interestingness of a set of tweets X for user u , by setting $\mathcal{I}_u(X) = \text{int}_\lambda(X, T_u)$.

We can now solve the TWEETREC problem by selecting the k tweets in \mathcal{T} having the largest values of int_λ with respect to T_u . Obviously, this considers the contribution of each tweet $t \in X$ to the overall interest of X for a user independently. In fact, the content of two tweets may overlap. When this overlap is non-negligible, the contribution of these two tweets to the overall interestingness of the pair is far from being the sum of the two independent contributions.

We can similarly solve the INTERESTS-SPANNING TWEETREC problem, directly applying Definition 3 to Equation 2 to address this issue. It is straightforward to prove that int_λ satisfies all the requisites of Theorem 2 and, thus, it is submodular.

4.2 Enhancing precision for passive users

The main assumption of our approach is that the number of tweets $|T_u|$ posted by a user is large enough to build a rich language model. Unfortunately a large portion of Twitter users are *passive*, i.e., they post only few or no tweets, as shown in Figure 1(a) and by other analytic studies (e.g., [15]). According to Twitter, as of September 2011, 40% of the users “sign in just to observe”. Building a language model for such users is thus prohibitive.

We therefore enhance our approach by including in the user model the tweets posted by the user’s friends. The intuition is that the user’s interests are expressed not only in his own feed, but also in the feeds of the people he follows. Indeed, contrary to other social networks, social connections in Twitter are based more on shared interests than on real world friendships [2, 17].

A further important observation is that some friends are intuitively more relevant than others when building a user model. Friends that are more authoritative should be weighted more in the model, because they provide more contentful and relevant information. We model this intuition by integrating Definition 1 with the following new formulation of tscore :

$$\text{tscore}(\omega, X) = \text{IDF}_T(\omega) \cdot \sum_{x \in X} \text{auth}(u_x) \cdot \text{in}(\omega, x) \quad (4)$$

where $\text{in}(\omega, x)$ is 1 if the tweet x contains term ω , 0 otherwise; and $\text{auth}(u_x)$ is the authoritativeness of the user that originally posted x , obtained as described next. Likewise, we also modify Definition 2 by considering pairs of terms instead of single terms.

As a side note, we experimented with another similar approach for solving the problem of passive user, where instead of considering authoritative users, we considered friends that the user often retweets, i.e., users that likely share her interests. This approach, obtained by replacing the $\text{auth}(u_x)$ score in the above formula with a *retweet score*, showed no improvements over the authority approach, and has therefore not been included in the paper for lack of space.

4.3 User authority

A clear definition of user authority in Twitter is very difficult to draw and to model being often subjective and task dependent.

In our case, authoritative users are those that post relevant information that is engaging for other users. Authoritative users, often called *information sources*, are usually characterized by a high number of followers and a high ratio between followers and friends. Although other indicators of authority may serve in this context, e.g., number of retweets or PageRank scores, we prefer to keep the computation of the score as simple as possible, given the real-time high-load nature of our application⁵.

For each user u in our corpus, we define the authority score $\text{auth}(u) \in [0, 1]$ as a linear combination of two logistic functions⁶:

$$\text{auth}(u) = A \cdot \frac{1}{1 + e^{-\frac{\text{ffRatio}(u)}{\alpha}}} + (1 - A) \cdot \frac{1}{1 + e^{-\frac{\text{foll}(u)}{\beta}}} \quad (5)$$

where $\text{ffRatio}(u) = \text{foll}(u)/\text{fri}(u)$. $\text{foll}(u)$ and $\text{fri}(u)$ are respectively the number of followers and friends of the user; $A \in [0, 1]$ is the linear parameter; α and β are logistic parameters. All the parameters have been set using a greedy search on a small manually ranked dataset of authoritative users. The resulting parameter values are as following: $A = 0.5$, $\alpha = 2$ and $\beta = 2000$.

5. EXPERIMENTS

This section shows the experiments we conducted in order to assess the effectiveness of our recommendation methods.

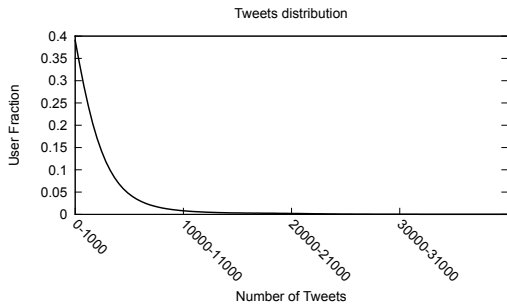
After describing the datasets in Section 5.1, we assess the effectiveness of the int_λ function as an estimator of the interestingness of a tweet for a user, and we provide a comparison against two well-known methods (Section 5.2). These comparisons are done by using both an automatic evaluation and a user study. After assessing the effectiveness of int_λ , we use it to find solutions for the TWEETREC and the INTERESTS-SPANNING TWEETREC problems. Eventually, in Section 5.3 we compare the solutions obtained by approaching these two problems, and we show that results produced by solving INTERESTS-SPANNING TWEETREC are of a higher quality.

5.1 Dataset

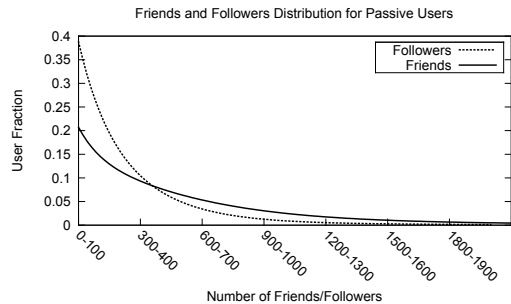
Our corpus consists of about 182,000 tweets posted between October 30 and November 4, 2011. The corpus was collected as follows. A large set of more than 14 million tweets was downloaded from Twitter using the Spritzer API, that provides access to a 1% random sample of all tweets. This set was then *pruned* to obtain our final corpus containing informative and non-junk English tweets on which we run our experiments. In details, the pruning process was as follows. First, we discarded all the tweets shorter than 30 characters and having less than 8 tokens (i.e., terms, hashtags, and usernames). Then, we removed tweets containing less than 3 English nouns and more than 5 English stopwords. To do that, we performed part of speech tagging

⁵Despite other indicators, the number of followers and the number of friends are indeed directly available from the Twitter streaming API.

⁶Normalization omitted for clarity.



(a) Distribution of the number of tweets per user in our corpus. The y-axis indicates the fraction of users in the corpus that have a given number of tweets.



(b) Distribution of Friends and Followers for users with less than 200 tweets posted since they joined Twitter.

Figure 1: Statistical data on our Tweets Stream.

using the NLTK⁷ toolkit. Finally, we discarded directed tweets (i.e., tweets starting with the @ symbol), that are usually personal in nature, and therefore not interesting for recommendation purposes.

Table 1 reports general statistics regarding our corpus. Figure 1(a) shows the distribution of the number of tweets per user. Each point on the x -axis corresponds to a bucket of 1,000 tweets. The corresponding y -axis value is the overall fraction of users that have posted that number of tweets. As expected the distribution follows a power-law. Almost 40% of users have posted less than 1,000 tweets since they joined Twitter. This confirms the need for strategies capable of handling “passive” users, as described in Section 4.2. Figure 1(b) shows the distribution of the friends and followers in Twitter for “passive” users with less than 200 tweets posted since they joined Twitter. Values on the x -axis are buckets of 100 friends or followers. We observe that about 80% has more than 100 friends, while around 40% has less than 100 followers.

Number of tweets	14, 720, 718
Number of tweets after filtering	182, 054
Number of hashtags	43, 026
Number of distinct terms	108, 295
Number of pairs with freq. ≥ 2	262, 081

Table 1: Statistics regarding our tweet corpus.

5.2 Evaluating int_λ quality

In this section, we show the effectiveness of int_λ by comparing it to two baselines: **Cosine** and **Hashtags**. Comparisons are done both using an automatic evaluation and a user-study-based methodology.

5.2.1 Automatic Evaluation

We base this set of experiments on the assumption that a user is likely to find his own tweets more interesting than random tweets from other users. Building on this assumption we randomly select a set of 250 users from our corpus. For each one of these users u we use 90% of his tweets (we call it the set T_u) for building the user profile. The remain-

ing 10% is mixed with all the tweets from other users in the corpus, forming the testing set.

Ideally, the correct recommendations for a given user correspond to tweets from the 10% of his own tweets. More formally, we divide the test set into a positive set \mathcal{P} , consisting of the 10% of the user’s tweets, and a negative set \mathcal{N} formed by all other tweets. We consider correct a tweet belonging to the positive set.

As evaluation metrics we use standard Information Retrieval measures. For each user we compute:

- **P@k**. *Precision at rank k* is the fraction of correct tweets in the top- k tweets ranked by the method.
- **S@k**. *Success at rank k* is the probability of finding at least a correct tweet on the top- k ranked ones.
- **MRR**. *Reciprocal Rank* is the inverse of the position of the first correct tweet in the ranking produced by the method.

To compute the final metrics’ scores we take the average of the above measures across all the 250 users.

Optimizing λ . The aim of our first experiment is to identify the value of λ in int_λ that provides the best performance. Figure 2 reports results of int_λ in terms of P@k (a), S@k (b), and MRR (c), for different λ values.

Results show that adopting λ values close to 1, corresponding to give more importance to **pscore**, the model achieves better performance. Maximum P@k and S@k values, in fact, are obtained when $\lambda = 1$. In addition, in Figure 2(c) we observe that a value of $\lambda = 0.9$ corresponds to a significant MRR decrease, i.e., the rank of the best tweets in the positive set \mathcal{P} significantly increases.

We can overall conclude that by setting $\lambda = 0.9$ the int_λ metric performs very well in terms of both precision and recall, and ranks interesting tweets high in the ranked list. Hereinafter, we therefore adopt the $\text{int}_{0.9}$ metric to estimate the interestingness of tweets for a user.

Comparing the different methods. We are now ready to compare $\text{int}_{0.9}$ against the following two baselines.

Cosine. A baseline model based on cosine similarity. For each tweet t in \mathcal{T} , interestingness is estimated as the cosine similarity between the term vectors of t and T_u , where each vector is obtained by using the classical information retrieval **tf – idf** [23] score.

⁷<http://www.nltk.org/>

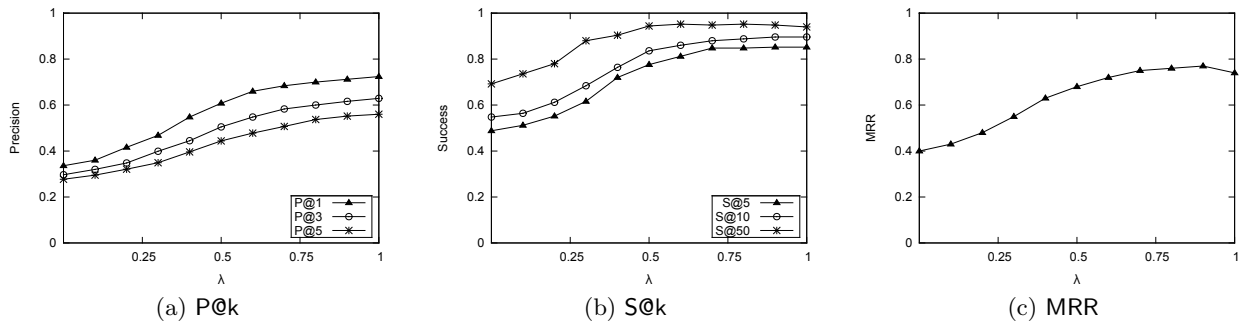


Figure 2: P@k (a), S@k (b), and MRR (c) of int_λ by varying λ .

Hashtags. A baseline similar to the above, where the vectors are not filled with terms, but with hashtags, similar to what proposed by Abel *et al.* [1]. The intuition is that hashtags explicitly summarize the topic(s) of a tweet (e.g., “I am flying to #NY today! #travel”).

We also run experiments to compare our models with those generated by Latent Dirichlet Allocation (LDA) [5]. Surprisingly, despite being one at the state of the art model, the LDA method performs very poorly. An analysis of the topics discovered by LDA reveals that terms clustered in the same topic do not have a well defined semantic relation. For example one of the topics has as the top-most representative words “art , box, winner, soul, alabama, adam, master, live”. We conclude that the hypothesis that LDA topics can be mapped to user’s interests does not hold in our specific case. One possible explanation is that our dataset is probably not big enough to correctly estimate the model parameters. Another issue is that LDA may be very sensitive to the short nature of the tweets. While a whole set of user’s tweets may suffice to infer the user’s topics of interest, a single tweet is too short to be reliably assigned topics. We will not report results of LDA given its extremely poor performance. However, we acknowledge that a deeper investigation is necessary to better understand the reasons behind poor LDA performances. We defer this to an extended version of this paper.

	P@1	P@3	P@5	S@5	S@10	S@50	MRR
$\text{int}_{0.9}$	0.71	0.62	0.55	0.85	0.90	0.95	0.77
Cosine	0.30	0.25	0.23	0.39	0.46	0.68	0.48
Hashtags	0.32	0.28	0.24	0.52	0.59	0.71	0.50

Table 2: Comparison of $\text{int}_{0.9}$ with Cosine, and Hashtags.

Table 2 shows the comparative results among the tested methods. We observe that $\text{int}_{0.9}$ is the best performing method for all measures. In particular, for the P@1 metric, $\text{int}_{0.9}$ ranks a correct tweet in first position in 71% of the cases. Hashtags, which is the second best performing method, instead ranks a correct tweet in the first position in only 32% of cases.

$\text{int}_{0.9}$ is the most effective method also in terms of S@k, with a 0.85 accuracy in finding correct results in the top-5 ranked tweets. The good performance of $\text{int}_{0.9}$ is confirmed also for the MRR metric. Correct tweets are ranked higher on average by $\text{int}_{0.9}$ than the other two methods.

	P@1	P@3	P@5	S@5	S@10	S@50	MRR
$\text{int}_{0.9}$	0.40	0.37	0.44	0.70	0.70	0.98	0.48
Cosine	0.22	0.30	0.31	0.33	0.33	0.78	0.31
HashTags	0.20	0.20	0.12	0.29	0.30	0.35	0.28

Table 3: Comparison of $\text{int}_{0.9}$ with Cosine, and Hashtags on a subset of highly related users.

The high results for $\text{int}_{0.9}$ may have been obtained because of bias in the data, i.e., a user might use a very peculiar writing style that makes his tweets highly distinguishable. In order to remove this bias we manually selected 20 pairs of users having very similar interests. The selection was done by using the Twitter user recommender system. The evaluation for this experiment works as follows: for each pair of users we add all the tweets of the second user in the stream; the task consists to re-retrieving them by using the set of tweets of the first user as the user profile.

Table 3 shows the results of this experiment, with a comparison against Cosine, and Hashtags. Once again, it is confirmed that $\text{int}_{0.9}$ performs better than the baselines. This suggests that the bias induced by the user’s writing style is very limited.

5.2.2 User Study Evaluation

In addition to the automatic evaluation just shown, we also test the effectiveness of $\text{int}_{0.9}$ by running a user study. We experiment with the following methods: Cosine⁸, $\text{int}_{0.9}$, and $\text{int}_{0.9}$ enhanced with the authoritative friends strategy described in Section 4.2 (hereafter referred as $\text{int}_{0.9}+\text{auth}$).

The assessment is conducted by a group of 7 professional assessors that regularly post tweets (Active users, in our terminology), and a group of 5 professional assessors that are using Twitter mainly for reading tweets (Passive users).

For each assessor and each method, we generate the top-20 tweet suggestions. Each assessor is provided with a random combination of tweets selected by the different methods, and is asked to state his personal interest on each tweet, using the following scale:

- **Excellent:** if the tweet is “*very interesting/very informative/very funny with respect to his/her interests*”;
- **Good:** if the tweet is “*interesting/informative/funny with respect to his/her interests*”;

⁸To reduce the load on human assessors we did not evaluate Hashtags since in the most conservative experiments in Table 3, they were performing worse than Cosine.

- **Fair**: if the tweet is “*somehow interesting, but nothing bad if he/she would have skipped it*”;
- **Bad**: if the tweet is “*not interesting, and he/she would have preferred not to have it in his/her timeline*”.

All the experimented models were built using a maximum of 200 tweets from the assessor’s Twitter stream. As evaluation metrics we use absolute Discounted Cumulative Gain (DCG), Precision computed over all assessed tweets of each method, and MRR.

Active users			
	DCG	Precision	MRR
int _{0.9}	23.29	0.67	0.57
int _{0.9} +auth	24.42	0.67	0.86
Cosine	19.28	0.56	0.80

Passive users			
	DCG	Precision	MRR
int _{0.9}	25.50	0.67	0.75
int _{0.9} +auth	36.60	0.88	1.00
Cosine	14.60	0.47	0.90

Active + Passive users			
	DCG	Precision	MRR
int _{0.9}	24.09	0.67	0.64
int _{0.9} +auth	29.50	0.76	0.92
Cosine	17.33	0.52	0.84

Table 4: User study results for int_{0.9} and Cosine for Active and Passive users separately. We consider a tweet annotated as E, G, or F as useful.

Table 4 reports results for Active users, Passive users, and both classes combined. Table 5 reports more in detail the percentage of Excellent, Good, Fair and Bad judgments for each method. For Active users, both our methods outperform Cosine in DCG and Precision, consolidating the results obtained in the automatic evaluation. As expected, int_{0.9}+auth is the best performing system. The addition of the authoritative friends’ tweets thus proves to be principled, as well as the criterion we used to select authoritative users, described in Section 4.3. For Active users, int_{0.9} only reports 0.57 in MRR, i.e., the top recommended tweet is often not correct. A closer look at the errors shows that this happens when the user’s model shares with a bad tweet a term pair with a high p_{score} (thus boosting int_{0.9}), but the term pair is not representative of the user interests (e.g., ‘have’-‘today’). This effect disappears in int_{0.9}+auth, where the larger size of the tweet set smoothes the impact of idiosyncratic term pairs.

In general, we verified that the most common errors across all methods are recommendations of ‘status update’ tweets, e.g., “Almost home! God can’t wait to get in my bed” or “None of my dog friends around here raw feed”. Even if these tweets may contain terms that define an interest (e.g., ‘raw feed’ and ‘dog’), they are personal in nature and therefore not interesting. In future work we will explore heuristics to classify and discard status updates.

top-5 tweets				
	Excellent	Good	Fair	Bad
Cosine	15%	22%	26%	37%
int _{0.9}	22%	16%	27%	35%
int _{0.9} +auth	30%	33%	15%	22%

top-10 tweets				
	Excellent	Good	Fair	Bad
Cosine	10%	17%	27%	46%
int _{0.9}	21%	21%	30%	28%
int _{0.9} +auth	27%	28%	18%	27%

top-20 tweets				
	Excellent	Good	Fair	Bad
Cosine	8%	19%	27%	46%
int _{0.9}	16%	21%	31%	32%
int _{0.9} +auth	20%	25%	25%	30%

Table 5: Percentage of Excellent, Good, Fair and Bad judgments in the top-*k* tweets suggested to Passive and Active users.

	TWEETREC	Equal	INTERESTS-SPANNING TWEETREC
int _{0.9}	18%	45%	37%
int _{0.9} + auth	8%	58%	34%

Table 6: Results of pairwise comparison of TWEETREC and INTERESTS-SPANNING TWEETREC, as judged by the assessors.

int_{0.9}+auth stands out even more for Passive users than Active users. This proposes int_{0.9}+auth as a promising robust solution for tweet recommendation, for Active users as well as Passive users. In both cases, it is important to leverage the tweets of the users’ authoritative friends, that mostly post informative tweets.

The overall precision of our baseline Cosine is 0.52, which is relatively higher than expected w.r.t. our int_{0.9}. We further investigate this issue by reporting in Table 5 the results of the user study for different scores. int_{0.9} is almost always 1.5 times better than Cosine in ranking Excellent results. The difference is even higher if we consider the effectiveness of the top-10 Excellent or Good results. As a matter of fact, we have that only 27% of Cosine are Excellent or Good w.r.t. 66% of int_{0.9}+auth.

5.3 TWEETREC vs. INTERESTS-SPANNING TWEETREC

In this section, we compare the quality of the solutions of TWEETREC versus INTERESTS-SPANNING TWEETREC. As described in previous sections, a solution for the latter problem should recommend a more heterogeneous, and therefore interesting, set of tweets to the user. As in the previous section, we both conduct an automatic and a user-based assessment.

Automatic Evaluation. Goal of the automatic evaluation is to assess if the INTERESTS-SPANNING TWEETREC solution successfully recommends a set of tweets that spans a larger set of user’s interests, with respect to TWEETREC.

top-5 TWEETREC	top-5 INTERESTS-SPANNING TWEETREC
1. President Obama LinkedIn Town Hall About Jobs (VIDEO)...	1. President Obama LinkedIn Town Hall About Jobs (VIDEO)...
2. #WWLP President Obama has declared a state of emergency for Massachusetts	2. I've set my email filters so the President's messages go straight to my spam folder.
3. Let me state it bluntly and clearly, Obama sucks as president and is quite the amateur!	3. UN leader Ban Ki-moon arrived on Wednesday in Libya for a ...
4. I've set my email filters so the President's messages go straight to my spam folder.	4. Jerry Sandusky, faces charges of sex crimes ...
5. Do you have questions on Obama's plan to help Federal student ...	5. New issue will be in our office in about an hour! re-tweet ...

Table 7: An example of pairwise comparison, where the top-5 tweets from INTERESTS-SPANNING TWEETREC were preferred over TWEETREC. Tweets are shortened for lack of space.

We proceed by randomly selecting 15 sets of 20 lists⁹ each. The assumption is that each list represents a specific user interest. For each list we download 800 tweets. We then create 15 *virtual users* with 8,000 ($= 20 \times 400$) tweets, one per each set, by selecting 400 tweets per list. We use the remaining 400 tweets per list to build a single virtual stream of tweets. The resulting dataset is a set of 15 virtual users spanning 20 different interests and having produced 8,000 tweets.

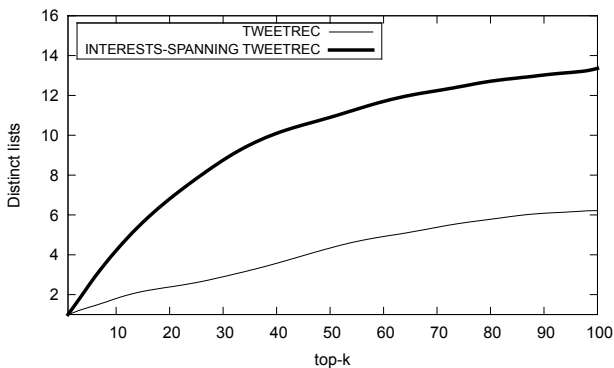


Figure 3: Number of distinct interests (i.e., lists) present in the top- k recommended tweets.

We now evaluate how many different lists (i.e., interests) are present in the top- k tweets recommended by our functions. Figure 3 shows the number of distinct lists in the top- k recommended tweets by using the solutions of TWEETREC and INTERESTS-SPANNING TWEETREC. Results show that, given a user, the set of tweets reported by solving INTERESTS-SPANNING TWEETREC spans multiple different interests, whereas the set returned by solving TWEETREC is more biased towards a very small set (usually one) of interests. This experimentally proves the value of solving INTERESTS-SPANNING TWEETREC.

User study Evaluation. We further investigate the difference between TWEETREC and INTERESTS-SPANNING TWEETREC, by evaluating which of the two solutions is preferred by real, i.e., human, users. We use the same group of assessors as in Section 5.2.2. For each assessor and for each method, we generate the set of top-5 recommended tweets. We then ask the assessors to perform a pairwise comparison of the top-5 sets from two different methods. We ask

⁹A *Twitter list* aggregates the tweets of users that share a common interest.

them to select the most interesting set or, alternatively, to indicate that the two sets are equally interesting.

Results are reported in Table 6, for two different configurations: $\text{int}_{0.9}$ and $\text{int}_{0.9} + \text{auth}$. For both configurations, the top-5 recommended tweets are equally interesting in about half of the cases. If they are not, preference is mostly given to INTERESTS-SPANNING TWEETREC, because in most cases it selects a set of tweets that spans several user's interests.

For example in Table 7, the tweets selected by TWEETREC are mostly focused on Barack Obama, while those selected by INTERESTS-SPANNING TWEETREC are about a variety of political topics.

6. CONCLUSIONS AND FUTURE WORK

We presented different methods for recommending tweets to Twitter users. Results over a large experimental study show that our proposed solutions outperform existing baselines, and that selecting tweets that are about a variety of the user's interests, improves the user experience.

There is a large avenue for future work. First, we are interested in improving the precision of our methods by devising automatic strategies to filter out tweets that are uninteresting 'status updates'. We also plan to improve the selection power of INTERESTS-SPANNING TWEETREC by providing tweet deduping at higher levels of semantics (e.g., adopting Textual Entailment Recognition techniques).

Our recommendation system could work in several application scenarios, e.g., providing to users a more interesting timeline than the one currently available in Twitter. We plan to carry out simulation trials to test the computational cost of our methods when facing a real-time load of thousands of tweets per second.

7. ACKNOWLEDGMENTS

This research has been partially funded by the EU CIP PSP-BPN ASSETS Project, Grant Agreement no. 250527, and the by the EU CIP-ICT PSP 2011.4.1 InGeoCloudS Project, Grant Agreement no. 297300. This work was completed before Marco Pennacchiotti joined eBay Inc..

References

- [1] F. Abel, Q. Gao, G.-J. Houben, and K. Tao. Analyzing Temporal Dynamics in Twitter Profiles for Personalized Recommendations in the Social Web. In *Proceedings of ACM WebSci '11, 3rd International Conference on Web Science, Koblenz, Germany*. ACM, June 2011.
- [2] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts. Everyone's an influencer: quantifying influence

- on Twitter. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 65–74, New York, NY, USA, 2011. ACM.
- [3] S. Banerjee, K. Ramanathan, and A. Gupta. Clustering short texts using Wikipedia. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 787–788, New York, NY, USA, 2007. ACM.
- [4] N. J. Belkin and W. B. Croft. Information filtering and information retrieval: two sides of the same coin? *Commun. ACM*, 35:29–38, December 1992.
- [5] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.
- [6] C. Budak, D. Agrawal, and A. El Abbadi. Limiting the spread of misinformation in social networks. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 665–674, New York, NY, USA, 2011. ACM.
- [7] R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, Nov. 2002.
- [8] J. Chen, R. Nairn, L. Nelson, M. Bernstein, and E. Chi. Short and tweet: experiments on recommending content from information streams. In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, pages 1185–1194, New York, NY, USA, 2010. ACM.
- [9] G. Cornuejols, M. L. Fisher, and G. L. Nemhauser. Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms. *Management Science*, 23(8):789–810, 1977.
- [10] A. Dong, R. Zhang, P. Kolari, J. Bai, F. Diaz, Y. Chang, Z. Zheng, and H. Zha. Time is of the essence: improving recency ranking using twitter data. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 331–340, New York, NY, USA, 2010. ACM.
- [11] Y. Duan, L. Jiang, T. Qin, M. Zhou, and H.-Y. Shum. An empirical study on learning to rank of tweets. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 295–303, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.
- [12] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey. An analysis of approximations for maximizing submodular set functions – II. In *Polyhedral Combinatorics*, volume 8 of *Mathematical Programming Studies*, pages 73–87. Springer Berlin Heidelberg, 1978. 10.1007/BFb0121195.
- [13] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.
- [14] A. Goyal, W. Lu, and L. V. Lakshmanan. Celf++: optimizing the greedy algorithm for influence maximization in social networks. In *Proceedings of the 20th international conference companion on World wide web*, WWW '11, pages 47–48, New York, NY, USA, 2011. ACM.
- [15] P. Judge. Barracuda labs, annual report, 2009.
- [16] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, pages 137–146, New York, NY, USA, 2003. ACM.
- [17] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *WWW '10: Proceedings of the 19th international conference on World wide web*, pages 591–600, New York, NY, USA, 2010. ACM.
- [18] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, pages 420–429, New York, NY, USA, 2007. ACM.
- [19] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14:265–294, 1978. 10.1007/BF01588971.
- [20] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceeding of the 17th international conference on World Wide Web*, WWW '08, pages 91–100, New York, NY, USA, 2008. ACM.
- [21] P. Resnick and H. R. Varian. Recommender systems. *Commun. ACM*, 40(3):56–58, Mar. 1997.
- [22] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor. Recommender systems handbook. *Media*, 54(3):217–253, 2011.
- [23] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18:613–620, November 1975.
- [24] P. Schonhofen. Identifying Document Topics Using the Wikipedia Category Network. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, WI '06, pages 456–462, Washington, DC, USA, 2006. IEEE Computer Society.
- [25] S. Ye and S. F. Wu. Measuring message propagation and social influence on Twitter.com. In *Proceedings of the Second international conference on Social informatics*, SocInfo'10, pages 216–231, Berlin, Heidelberg, 2010. Springer-Verlag.