Ranking Twitter Discussion Groups

James Cook
UC Berkeley
jcook@cs.berkeley.edu

Abhimanyu Das
Microsoft Research
abhidas@microsoft.com

Krishnaram Kenthapadi
Microsoft Research
krisken@microsoft.com

Nina Mishra
Microsoft Research
ninam@microsoft.com

ABSTRACT

A discussion group is a repeated, synchronized conversation organized around a specific topic. Groups are extremely valuable to the attendees, creating a sense of community among like-minded users. While groups may involve many users, there are many outside the group that would benefit from participation. However, finding the right group is not easy given their quantity and given topic overlap. We study the following problem: given a search query, find a good ranking of discussion groups. We describe a random walk model for how users select groups: starting with a group relevant to the query, a hypothetical user repeatedly selects an authoritative user in the group and then moves to a group according to what the authoritative user prefers. The stationary distribution of this walk yields a group ranking. We analyze this random walk model, demonstrating that it enjoys many natural properties of a desirable ranking algorithm. We study groups on Twitter where conversations can be organized via pre-designated hashtags. These groups are an emerging phenomenon and there are at least tens of thousands in existence today according to our calculations. Via an extensive collection of experiments on one year of tweets, we show that our model effectively ranks groups, outperforming several baseline solutions.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval; J.4 [**Computer Applications**]: Social and Behavioral Sciences

Keywords

discussion groups; group chats; Twitter groups; online communities; ranking groups; group search; group preference

1. INTRODUCTION

Many venues exist for holding group conversations on the Internet. For example, forum sites, Internet Relay Chat, newsgroups and Yahoo groups have been widely studied. In this paper, we study *discussion groups*, which are groups that repeatedly meet at a mutually agreed upon time with the goal of discussing a particular topic.

We specifically study Twitter discussion groups where prior work shows that some discussions may be organized via pre-designated hashtags [13]. For example, wine aficionados append the hashtag **#winechat** during their conversations. Those interested follow the hashtag to listen during the pre-agreed upon time. The topics of these chats span multiple categories, from arts to education, entertainment and hobbies.

Our goal is to develop a method that will enable new users to search for discussion groups. A key question is given a search query, how can we rank discussion groups according to where the query is best discussed?

A natural approach is to treat the question as a classical web page ranking problem. In other words, treat the content of all messages exchanged in a discussion group as a web page and order the web pages using traditional information retrieval metrics such as TFIDF [42]. We study such approaches as a baseline upon which our methods should improve. But in the case of Twitter, and other social networks, we have more information at our disposal beyond the content of the message. For example, we know who contributed to which discussion group, as well as some indication of the authoritativeness of a user. We study how these additional signals can be used to generate an improved ranking.

We are not aware of any existing work on ranking discussion groups. There is related work in the area of ranking threads within a discussion, where the goal is to prevent new users from posting the same question twice [17] but, to our knowledge, nothing in the area of ordering groups.

In this paper, we begin by defining a discussion group, which is a generalization of a group chat [13]. We seek groups that have repeatedly met in a short window of time, where a group meets if a significant fraction of the traffic generated by the group takes place in a narrow window of time. This more general definition will give users of a group ranking engine the ability to search over a larger collection of groups.

We describe a new model for ranking groups called the group preference model: for a given search query, a hypothetical user starts with a group where the topic is discussed

COSN'14, October 1–2, 2014, Dublin, Ireland.
Copyright 2014 ACM 978-1-4503-3198-2/14/10 ...$15.00.
http://dx.doi.org/10.1145/2660460.2660473.

and repeatedly finds an authoritative user in the group and walks to a random group according to what the authoritative user prefers. With small probability the hypothetical user jumps to a random group. The model resembles PageRank [4] where a random surfer repeatedly follows outgoing links and jumps to a random node with small probability. A key difference is that in our model the hypothetical user bounces back and forth between groups and authoritative participants.

The technical exploration of this paper is devoted to understanding how our ranking algorithm responds to small perturbations in the input. Observe that the data that feeds our group preference model is constantly changing: *e.g.*, meeting attendance patterns may shift over time and a user's authoritativeness may rise and fall over time. We still want good groups to remain near the top of the ranking, particularly if the underlying data continues to support it. One notion of such good behavior is *rank stability* [37], but this turns out to be a very strong requirement: for example, the well-known PageRank and HITS algorithms are not rank-stable. Instead, we make a more specific list of desirable properties, and show that our algorithm satisfies them. For example, if one group is universally preferred to another according to a dataset and we add a new user to the dataset who holds the same preference, then our algorithm will also retain the preference. In a similar vein, if a user has an exclusive preference for some group, then increasing that user's authority cannot hurt that group's ranking.

We conduct an experiment on one year of tweets. We identify a collection of 27K discussion groups (hashtags) from this data. We create a set of group queries based on queries posed to Yahoo groups and a ground truth ranking of hashtags for these queries. We compare the performance of our algorithm with the performance of several natural baseline algorithms in terms of precision, recall, mean average precision, and NDCG and show that our algorithm outperforms the baselines on all of these metrics.

2. RELATED WORK

Our work sits in the context of other work in online forums, ranking, recommendation, group membership and group chats. We describe findings in these areas, as well as how our work relates.

Search in Online Forums While we are not aware of previous work on ranking discussion groups, there is work addressing related search problems. Online forums are similar to discussion groups, but generally involve parallel, asynchronous discussion threads instead of synchronized meetings. There is work on finding forum threads relevant to a query [17], as well as matching questions to answers [12]. The goal of this line of research is to prevent people from posting the same question twice. In contrast, the unit of retrieval in our work is a discussion group, rather than a thread. Also, we seek to a connect a user to a like-minded community of individuals where they can engage in repeated conversations, rather than find a closely matching question.

Ranking Models Given a search query, our group preference model describes a user (the *seeker*) who starts at a random group where the query is discussed, then repeatedly finds an authoritative participant in that group and then a group where that person discusses the query. The model is related to the Random Surfer Model [4] where a random walk repeatedly follows outgoing links on a directed graph. Our model differs in that we are bouncing back and forth between two kinds of nodes (groups and authoritative participants). Also, the transition probabilities depend on the query, and are determined by social interactions instead of links between documents. Both models include a *teleport* probability that the walk jumps to a completely random node, and in both models, the walk's stationary distribution is used to rank nodes. Our model is similar to personalized PageRank [25] in that the probability of teleporting to a group can depend on features such as how often the query is discussed in the group. Some of the mathematics developed for PageRank regarding how small changes to a graph affect the stationary distribution [10] are useful in our work (§5.1).

The group preference model is also related to HITS [32] which assigns hub and authority values to each node on a graph. The hub and authority scores complement each other in much the same way that the group preference seeker spends more time on participants with authority in highly ranked discussion groups, and groups preferred by highly-ranked participants. One important difference is that the HITS algorithm computes each new hub or authority score as a sum of neighboring values, whereas our model, since it follows a random walk, averages the values. Averaging has the advantage that a discussion group with very many participants but only marginally related to a query can be ranked lower than a collection of groups very related to a query that comes from a community of groups and participants who reinforce each other with evidence of preference and authority (§5.2). Another difference is that we allow the group preference seeker, when jumping from a person to a group, to use the previous group visited to inform the decision. For example, it is within the scope of our model for the seeker to only jump to groups that the person prefers to the previous group.

Implementing our model to serve a large number of queries would introduce scalability challenges similar to those faced by the HITS algorithm. For example, both models use a different graph for every query. There is past work on improving the efficiency of the HITS and related MAX and SALSA algorithms [36, 39, 47]. There has also been work on the similarly challenging problem of pre-computing personalized PageRank results for every starting node [14, 25].

The random shopper model [23] was developed in the context of online shopping and is also related. Each feature is represented as a directed graph over products with an edge from one product to another if it is better according to that feature. For example, if the feature is "lower price", then the user will walk to a cheaper product. The process of selecting a product starts at a random one, and then repeatedly selects a feature according to its importance and walks to a better product according to that feature. The principle goal is to learn the relative importance of each feature. One can view the features as authoritative participants and the walk within a feature as selecting a group according to the participants' preferences. The random choice of feature to select is independent of which product the random shopper has reached. In our work, the group that the seeker walks to intentionally depends on which authoritative participant was selected. In our work, the technical emphasis is in demonstrating that under reasonable changes to the underlying data, the ranking will remain unchanged, while in that work the emphasis was on showing how the ordering can and should flip [48] depending on which other products are shown.

Learning to Rank There is a large body of work in the learning to rank literature [6, 7, 11, 27] that on the surface seems relevant to the discussion group ranking problem. These techniques learn a function that given a search query and URL produce a score for how well the URL matches the query. However, these techniques require training data indicating how well a URL matches a given query. This training data can either be editorially judged or inferred from click activity [1, 28]. We note that obtaining training data for our search problem is quite challenging. For a human judge who is external to a search query and a group to evaluate the relevance of a group to a query is quite difficult and time-consuming, as we ourselves discovered as candidate judges. Furthermore, since no discussion group ranking system is in existence today, no click activity exists for inferring relevance. Instead, we use meeting attendance patterns, message content and user authority to drive a model of group preference.

Recommending Hashtags and Groups The general problem of recommending hashtags has been previously studied where given a tweet, the goal is to find a relevant hashtag. In one approach, the text of the tweet is used to identify similar tweets, and then a hashtag is recommended based on those found in similar tweets [34]. In other methods, the users who tweet about the subject may be used to find a relevant hashtag [20]. Note that arbitrary hashtags may never meet again. Indeed, prior work shows that 86% of hashtags have been used less than five times [52]. Such hashtags are not relevant to our problem of helping a new user find a future conversation. Further, since prior techniques are applied to arbitrary hashtags, the work does not take advantage of the fact that some of these hashtags are discussion groups — whose richer structure can be exploited to deduce higher quality rankings. We are motivated by applications where a new user seeks a future conversation. The hashtag prediction problem — given a user, predict which hashtags they will use in the future — has also been studied [51]. Many interesting characteristics of a hashtag are identified as useful for effective prediction, such as the prestige of a hashtag. These characteristics could also be used to create richer models of group preference.

There is also past work on recommending groups in online social networks [45], based on a user's existing social links and without any query. In our work, we hope to introduce a user with a topic of interest (query) to relevant discussion groups, even if the user is not yet a member of the system that hosts the groups.

Group Membership There is a substantial body of work in understanding why people join and remain in online communities. The size of a group is known to affect whether a user joins a group. Too many messages drive people away [8, 29], while having too few inhibits community responsiveness [38]. The level of moderation also plays a role [44]. The more friends a user has in a group, the more likely they are to join [33], and this likelihood increases if their friends are in turn connected [2]. If a user receives a response to their first message to a community, it increases the likelihood that they will subsequently interact with the community [3, 30]. A first response is also known to increase the speed at which a second message is posted [35]. Linguistic complexity reduces the chance of a response [50], and linguistic discrepancy can signal a user's departure from a group [15]. Our work differs in that we seek to connect a user to a group that was previously unknown to them. Our goal is to rank the best groups for discussing a particular topic. Richer contextual clues (friends in the group, linguistic coherence, etc.) that are known to drive group membership could lead to better and more personalized rankings.

Group Chats A group chat [13] is defined by three properties that we state for completeness. (1) Regular: In a group, people who share an interest meet on a regular basis over a prolonged period of time. (2) Synchronized: In a group, meetings occur for a fixed duration at a specified time. (3) Cohesive: Members in a group communicate with each other over the course of many meetings. In contrast, the definition of a discussion group explored in this paper is looser and focuses on the second property. We replace the notion of a regular meeting, *e.g.*, once a week, with one where the group meets multiple times. We also remove the requirement that the group be cohesive. The resulting set of groups is much larger – 1500 group chats versus 27K discussion groups, though still not as large as the number of Yahoo groups (~6M according to [3]). Where the previous work was concerned with measuring the number and variety of group chats, and therefore called for a more conservative definition, the goal of the current work is to provide a comprehensive discussion group ranking algorithm, and is best served by a broader definition, providing a large number of candidates as input. The goal of [13] was to design algorithms that could automatically find group chats on Twitter. The emphasis of our work is on ranking discussion groups so that we can connect a new user to a group.

3. PROBLEM FORMULATION

Consider any setting where many groups $g_1, \ldots, g_n$ meet often to discuss various topics. Our goal is to help a user with a topic of interest (the *query*) to find a relevant discussion group in which to participate. We hope that such an algorithm will help people to find others with similar interests, and give them a place to ask questions and share stories.

Discussion Groups We begin by describing the kind of group we seek. Since we wish to find a place for the user to have discussions, we restrict our attention to groups that have proved themselves by holding meetings in the past:

Definition 3.1 (Discussion group). *A* meeting *is a span of time at most w hours long during which at least a γ fraction of all of the group's interactions in a specified time period happen. A collection of meetings constitutes a candidate* discussion group *if there have been at least m different meetings.*

In other words, a discussion group should have many discussions that last for some short period of time, typically one or two hours.

Problem Statement Given a query topic q that a user is interested in, we have two closely related goals. First, to understand which discussion group the user would choose to attend after spending some time on their own exploring groups related to topic q. Second, to develop an algorithm to predict these preferences, in order to save time or to suggest discussion groups to a user who would not otherwise embark on such an exploration.

PROBLEM 3.2. *Given a query topic q, we wish to find a set of discussion groups $g_1, \ldots, g_r$ relevant to q, together with a ranking on those groups: we say $g_i >_q g_j$ if our algorithm determines that group g_i is preferable to g_j in the context of topic q.*

We also seek to understand what characteristics influence a user's decision to prefer one discussion group over another. To this end, we will investigate a variety of characteristics.

Twitter Interpretation To interpret Definition 3.1 in the context of Twitter, we say that a *meeting* is a w-hour window of time that contains at least a γ fraction of all tweets sent during that week, and a set of tweets forms a *chat* if there are at least m weeks that contain a meeting. We make the simplifying assumption that every chat has a hashtag that is not used by any other chat — this is usually the case in our experience. In this work, we set out to solve Problem 3.2 via Twitter discussion groups.

4. MODEL

To solve Problem 3.2, we propose a model called the *group preference model* for the process a user (the *seeker*) interested in a topic q might follow to choose among the relevant discussion groups. The seeker begins by finding an arbitrary relevant group g_0. They then find a participant p_0 who holds some degree of *authority* in the group g_0. By looking at p_0's profile page, they look at the other discussion groups that p_0 participates in, and choose a group g_1 that p_0 shows a *preference* for. The seeker continues alternating between discussion groups and people $g_0, p_0, g_1, p_1, \ldots$ and eventually stops on one of the discussion groups.

An important feature of this model is that it makes use of social signals. This allows a community of discussion groups and people around a topic to be promoted through a feedback effect (§5.2). The model also satisfies several desirable properties described in §5.1. See §2 for a comparison to some similar ranking models.

We begin our precise description of the model by describing in more detail the steps of jumping from a discussion group to a participant and from a participant to a group.

4.1 Authority Score $A_{q,g}(p)$

After arriving at a discussion group g, the seeker chooses a participant to jump to according to their *authority score*. The authority of different participants p within a group g is quantified by authority scores $A_{q,g}(p)$ which form a probability distribution. The scores could be determined in many different ways. For example, we could assign equal weight to every person who has participated in g. Alternatively, we could assign weight proportional to the number of followers, or the number of @-mentions received by the person.

4.2 Preference Score $P_{q,p,g}(g')$

After arriving at a participant p from a group g, the seeker looks at various discussion groups p has participated in and jumps to one according to the *preference score*. For a query q, participant p, and last group g, the preference scores $P_{q,p,g}(g')$ of participant p for different groups g' form a probability distribution (that is, $\sum_{g'} P_{q,p,g}(g') = 1$). One simple way to determine $P_{q,p,g}(g')$ is to make it proportional to the number of meetings of group g' that p took the time to attend. We may also wish the preference score to depend on the last group g that the seeker visited. In particular, our implementation (described in §6.2) requires that the seeker never jump to a group g' if p is less active in g' than in g.

4.3 Teleport Distribution D_q

After each step, with some probability $\lambda \in (0, 1)$ the seeker decides to cut short their current exploration, and chooses a new random discussion group to start from. For example, in the context of Twitter, the seeker might use Twitter's search feature to find a new potential group. This is analogous to the teleportation step of PageRank, where the surfer sometimes jumps to a uniformly random web page. The probability distribution the seeker uses to jump to a new discussion group is a parameter of our model, called the *teleport distribution* D_q, and is over discussion groups relevant to the topic q. D_q plays the same role as the preference vector in personalized PageRank [25, 40]. As with PageRank, one simple choice is to set $D_q(g) = \frac{1}{n}$ for every relevant group g, where n is the number of such groups. Alternatively, we may wish to capture the notion that the seeker is more likely to start at discussion groups which are more strongly relevant to the topic q. In the context of Twitter, we could set the teleport probability $D_q(c)$ of a chat c to be proportional to the number of tweets in chat c where q is mentioned divided by the total number of tweets in chat c. However D_q is determined, it should be normalized so that the sum of probabilities is one. We require $D_q(g) > 0$ for every relevant group g in order to ensure that the model gives a well-defined solution, in a sense that will become clear when we describe our algorithm in §5.

4.4 The Group Preference Model

Given a query q, the seeker follows this process, which is parameterized by a teleportation parameter $\lambda \in (0, 1)$.

1. Choose an arbitrary starting group g.
2. Select a participant p at random using the probability distribution $A_{q,g}(p)$.
3. Select a group g' at random using the probability distribution $P_{q,p,g}(g')$.
4. With probability λ, sample a discussion group g from the teleport distribution D_q, and go to step 2.
5. Otherwise, go to step 2 using g' as the new g.

Eventually, the seeker stops and chooses the discussion group that they most recently jumped to. Figure 1 illustrates the first three steps of the process in the context of Twitter. We are not claiming that real users follow this process, only that it may model part of the behavior we observe.

5. ALGORITHM AND ANALYSIS

We now describe our algorithm for solving Problem 3.2 using the group preference model. The key observation is that even though the seeker visits both discussion groups

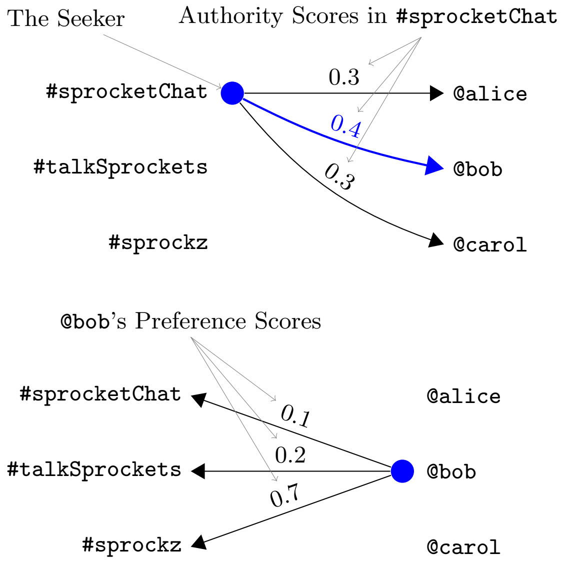


Figure 1: The first steps of the group preference model in the context of Twitter. Starting from a random discussion group (#sprocketChat), the seeker jumps to a user randomly using authority scores in that group, then to a new group according to that user's preferences over the current group.

and participants, the model can be represented by the following Markov process over just the groups with the matrix of transition probabilities $M(q)$ computed as follows:

$$M(q)_{g_1,g_2} = \lambda D_q(g_2) + (1-\lambda)\sum_{p\in U} A_{q,g_1}(p)P_{q,p,g_1}(g_2) \quad (1)$$

Where U is the set of people who participate in any relevant group. Each transition probability $M(q)_{g_1,g_2}$ in (1) is then equal to the probability that the seeker lands on g_2 given that the last group they landed on was g_1. To understand why this is true, note that the seeker can land on g_2 either by (a) landing on a participant with a positive preference for g_2, or (b) teleporting directly. Case (b) happens with probability $\lambda D_q(g_2)$, where $D_q(g_2)$ is the teleport distribution parameter of the model. To compute the probability of (a), note that the probability of arriving at g_2 through a participant p is $(1-\lambda)A_{q,g_1}(p) \cdot P_{q,p,g_1}(g_2)$, and sum over all participants p. Notice that every query q gives rise to a different Markov process, and that $M(q)$ is regular so long as $\lambda > 0$. (If we generalize to an arbitrary teleport distribution D_q, this is why we require (§4.3) that every probability is positive.)

Given a query q, Algorithm 1 (GroupPreference) computes the stationary distribution of $M(q)$ and ranks the discussion groups by their stationary probabilities.

5.1 Properties of Algorithm GroupPreference

To help understand the behavior of Algorithm 1 (GroupPreference), we study how changes in the input data can affect the ranking. The stationary distribution of a Markov process can change in unintuitive ways as a result of changes to the transition probabilities. For example, increasing a transition probability to one state can increase the stationary probabilities of many other states, and when λ is near 0, a small change can have a large effect. In this section, we show that our algorithm has many simple and desirable properties: for example, if a participant shows an increased preference for a discussion group g, then g's ranking will not be negatively affected (Theorem 5.5).

```
Algorithm 1 GroupPreference
Parameters: Authority score function $A_{q,g}(.)$; Preference
    score function $P_{q,p,g}(.)$; Teleport parameter $\lambda \in (0,1)$;
    Teleport distribution $D_q$.
Input: A set of candidate discussion groups (Def. 3.1); A
    dataset of group interactions; A query $q$.
Output: A ranking of discussion groups relevant to topic
    $q$.
Find all groups $g_1, \ldots, g_n$ where the topic $q$ is mentioned
    in some group interaction.
Compute the authority and preference scores and tele-
    port probabilities $A_{q,g}(p)$, $P_{q,p,g}(g')$, $D_q(g)$ for every $g$,
    $g'$, $p$.
Compute the stationary distribution $\pi$ of the Markov
    process $M(q)$ defined in (1).
return Groups ranked so $g_1 >_q g_2$ iff $\pi(g_1) > \pi(g_2)$.
```

Our first property describes what happens when every participant prefers one group g_1 over another group g_2. The property holds when the teleport distribution is uniform, or at least does not favor g_2 over g_1.

Theorem 5.1. *If for topic q, every participant always assigns a higher preference score to group g_1 than g_2, and g_2 does not have a higher teleport probability, then $g_1 >_q g_2$.*

Proof. The proof is guided by the intuition that whenever the seeker is at a participant, the next group they jump to is more likely to be g_1 than g_2. Looking at (1), we see that for every group g, $M(q)_{g,g_1} > M(q)_{g,g_2}$. It follows that after one step of the Markov process, the seeker is more likely to end up at group g_1 than g_2 — in particular, taking π to be the stationary distribution, we have $(\pi M(q))(g_1) > (\pi M(q))(g_2)$. Since $\pi = \pi M(q)$, we have $\pi(g_1) > \pi(g_2)$, so the algorithm will rank $g_1 >_q g_2$. □

Instead of comparing two groups, we can describe what happens if every user's preference for one group g_1 is high. This property holds if the teleport distribution is uniform.

Theorem 5.2. *Suppose that for topic q, every participant has a preference of at least α for group g_1, regardless of the previous group g'. If the teleport distribution D_q is uniform, then no more than $1/\alpha - 1$ other groups will be ranked higher than g_1.*

Proof. First, notice that the stationary probability of g_1 is at least $\gamma = \lambda\frac{1}{n} + (1-\lambda)\alpha$. This is true because, looking at (1), $M(q)_{g,g_1} \geq \gamma$ for every group g. Hence, $\pi(g_1) = \sum_g(\pi(g) \cdot M(q)_{g,g_1}) \geq (\sum_g \pi(g)) \cdot \gamma = \gamma$. Using a similar argument, for every group g, $\pi(g) \geq \lambda\frac{1}{n}$ since $M(q)_{g',g} \geq \lambda\frac{1}{n}$ for any g' and g. Based on these two lower bounds on the stationary probabilities, we next obtain an upper bound on the number of groups with large stationary probability. It is not possible for more than $1/\alpha$ groups to have a stationary probability as high as γ: otherwise, the sum of all stationary probabilities would be more than $n\lambda\frac{1}{n} + (1/\alpha)(1-\lambda)\alpha = 1$. □

The remaining properties restrict how the algorithm's ranking can change if the input data changes. In each case, we will consider two datasets T and T' of discussion group interactions. We will assume the preference or authority scores which result from these datasets (§4.1, §4.2) differ in some small way. Notationally, we will add T as a parameter to the authority and preference scores $A_{T,q,g}(p)$ and $P_{T,q,p,g}(g')$; the teleport distribution $D_{T,q}$; the transition matrix $M(T,q)_{g_1,g_2}$; and the resulting judgments $g_1 >_q^T g_2$.

Next, we show that if we add to the dataset a new participant who shares a preference with all existing participants, that preference will still be reflected in the new ranking. Also, if we add a participant with a preference of α for a group g_1 to a dataset where all existing participants have such a preference, then g_1 will still be ranked in the top $1/\alpha$.

Corollary 5.3. *Suppose that in T, every participant assigns a higher preference score to g_1 than g_2 and g_2 does not have a higher teleport probability. If the only change from T to T' is the addition of a new person p_* who also prefers g_1 to g_2 (that is, teleport probabilities, and preference scores as well as the proportions between authority scores not involving p_*, are unchanged), then $g_1 >_q^{T'} g_2$.*

Similarly, suppose that in T, every participant assigns a preference of at least α to g_1, and the teleport distribution is uniform. If the only change from T to T' is the addition of a new person who also has a preference of at least α for g_1, then g_1 will be ranked in the top $1/\alpha$ groups.

This corollary follows because the hypotheses of Theorem 5.1 and Theorem 5.2 are still respectively true in dataset T'.

Our next two theorems will make use of a result by Chien et al. [10] about Markov processes, that increasing the transition probability to a state at the expense of other states cannot negatively affect that state's ranking. We re-formulate their result to be more immediately applicable to our setting.

Theorem 5.4 (Chien et al. [10, Theorem 2.9]). *Consider a regular Markov chain M. Fix some state s_1 in M. Let M' be a regular Markov chain over the same set of states as M, obtained by modifying M as follows. Transition probabilities to states other than s_1 are either decreased or kept unchanged in M', compared to M. Correspondingly, transition probabilities to s_1 are either increased or kept unchanged (so that the transition probabilites out of any state sum to 1). In other words, for every $s_2 \neq s_1$ and every s_3, $M'_{s_3,s_2} \leq M_{s_3,s_2}$ (and since the transition probabilities out of s_3 sum to 1, $M'_{s_3,s_1} \geq M_{s_3,s_1}$).*

Let π and π' be the stationary distributions of M and M' respectively. Then, for any state s_4, if $\pi_{s_1} > \pi_{s_4}$, then $\pi'_{s_1} > \pi'_{s_4}$.

Theorem 5.4 allows us to understand the consequences of various changes by studying their effects on the transition matrix $M(q)$. Our next two properties say that the algorithm is monotonic in ways that one would expect: the rank of a discussion group g must not decrease when a participant's demonstrated preference for it increases (for example, because they attended more meetings) or when an avid fan of the group gains authority.

Theorem 5.5. *Suppose that the only change from T to T' is that participant p_1 shows an increased preference for group g_1 and a decreased preference for other groups for a given query q. That is: $P_{T',q,p_1,g'}(g_1) \geq P_{T,q,p_1,g'}(g_1)$ for all g'; $P_{T',q,p_1,g'}(g) \leq P_{T,q,p_1,g'}(g)$ for all $g \neq g_1$ and all g'; and all other authority and preference scores and teleport probabilities are unchanged. Then for any group g_2, if $g_1 >_q^T g_2$, then $g_1 >_q^{T'} g_2$.*

Proof. Since the authority scores and teleport probabilities are unchanged, and preference scores for participants other than p_1 are also unchanged, we can express the change in the Markov transition matrix (1) as: $\forall h_1, h_2$,

$$\begin{aligned}&M(T',q)_{h_1,h_2} - M(T,q)_{h_1,h_2}\\ =&(1-\lambda)A_{T,q,h_1}(p_1)(P_{T',q,p_1,h_1}(h_2) - P_{T,q,p_1,h_1}(h_2)).\end{aligned}$$

This change is nonnegative when $h_2 = g_1$ and nonpositive otherwise. So by Theorem 5.4, for any group g_2, if $g_1 >_q^T g_2$, then $g_1 >_q^{T'} g_2$. □

Finally, increasing the authority of group's fan cannot negatively impact the group's ranking.

Theorem 5.6. *Suppose that participant p_1 has an exclusive preference for group g_1: $P_{T,q,p_1,g'}(g_1) = 1$ for all g'. Assume that the only change from T to T' is that p_1 gains authority. That is: for every group g, $A_{T',q,g}(p_1) \geq A_{T,q,g}(p_1)$; for every group g and participant $p \neq p_1$, $A_{T',q,g}(p) \leq A_{T,q,g}(p)$; and all other authority and preference scores and teleport probabilities are unchanged. Then for any group g_2, if $g_1 >_q^T g_2$, then $g_1 >_q^{T'} g_2$.*

Proof. Notice that for any groups g and g' where $g' \neq g_1$, and any participant p,

$$A_{T',q,g}(p)P_{T',q,p}(g') \leq A_{T,q,g}(p)P_{T,q,p}(g').$$

(For user p_1, this is true because p_1's preference for g' is zero.) It follows that $M(T',q)_{g,g'} \leq M(T,q)_{g,g'}$. So by Theorem 5.4, for any group g_2, if $g_1 >_q^T g_2$, then $g_1 >_q^{T'} g_2$. □

In §5.2, we describe a scenario showing an advantage of Algorithm 1 over simpler approaches. In §6, we evaluate the algorithm experimentally.

5.2 Comparing to the Naive Approach

Instead of using Algorithm 1, one could rank the discussion groups relevant to a topic q based simply on the number of people who attend meetings, the number of interactions in the groups, or some similar metric. One problem with such naïve rankings is that very popular groups which are not about topic q, but where topic q arises incidentally, can dominate smaller groups whose main focus is q. For example, if q is a disease and a celebrity is diagnosed with it, then a Twitter chat about celebrities might see a surge of messages about q that is much greater in volume then any discussion on the Twitter chats that are focused on topic q.

To understand the advantage of Algorithm 1, consider the following scenario, illustrated in Figure 2:

Scenario 5.7. *There is a set of discussion groups G_{pop}, which we think of as being* popular, *but barely related to the topic q. There is a very large set F of participants whom we think of as* fans *of groups in G_{pop} and uninterested in q. We assume the following two properties:*

- *Non-fans $p \notin F$ who mention topic q give small preference scores to G_{pop}: $\forall g'$, $\sum_{g \in G_{\text{pop}}} P_{q,p,g'}(g) < \epsilon$.*

- *Fans $p \in F$ do not have a strong interest in q, so they have small authority scores in the groups that are focused on the topic: $\forall g \notin G_{\text{pop}}$, $\sum_{p \in F} A_{q,g}(p) < \epsilon$.*

THEOREM 5.8. *In Scenario 5.7, let $D_{\text{pop}} = \sum_{g \in G_{\text{pop}}} D_q(g)$ be the total teleport probability of the non-relevant groups. Suppose that there is some relevant group $g_* \notin G_{\text{pop}}$ for which every non-fan $p \notin F$ has a preference of at least $\frac{8}{7}(\beta + \frac{\lambda}{1-\lambda} D_{\text{pop}})/(1-\beta)$, where $\beta = D_{\text{pop}} + \frac{2\epsilon}{\lambda}$. Then, if $\epsilon < \frac{1}{8}$ and $\lambda < 1$, then Algorithm 1 will rank group g_* above every group in G_{pop}. (This holds true even if there are many more fans than non-fans and the groups in G_{pop} have many more tweets than the other groups.)*

PROOF. Let π be the stationary distribution of $M(q)$. For a group g, let π_g denote its probability under distribution π, and for a set of groups G let $\pi_G = \sum_{g \in G} \pi(g)$. We will first show that $\pi_{G_{\text{pop}}}$ is small and then show that π_{g_*} is large.

For any $g_1 \notin G_{\text{pop}}$ and $g_2 \in G_{\text{pop}}$,

$$\begin{aligned} M(q)_{g_1,g_2} \leq& \lambda D_q(g_2) + (1-\lambda)\big(\sum_{p \in F} A_{q,g}(p) + \sum_{p \notin F} A_{q,g}(p)\epsilon\big) \\ \leq& \lambda D_q(g_2) + (1-\lambda)(2\epsilon), \end{aligned}$$

and similarly, for $g_1, g_2 \in G_{\text{pop}}$,

$$M(q)_{g_1,g_2} \leq \lambda D_q(g_2) + (1-\lambda).$$

For any distribution d and $g_2 \in G_{\text{pop}}$, it follows that

$$(dM(q))_{g_2} \leq \lambda D_q(g_2) + (1-\lambda)(d_{G_{\text{pop}}} + 2\epsilon(1 - d_{G_{\text{pop}}})),$$

and so $(dM(q))_{G_{\text{pop}}} \leq \lambda D_{\text{pop}} + (1-\lambda)(d_{G_{\text{pop}}} + 2\epsilon(1 - d_{G_{\text{pop}}}))$.

In particular, the total stationary probability of G_{pop} satisfies

$$\pi_{G_{\text{pop}}} = (\pi M(q))_{G_{\text{pop}}} \leq \lambda D_{\text{pop}} + (1-\lambda)(\pi_{G_{\text{pop}}} + 2\epsilon(1 - \pi_{G_{\text{pop}}}))$$

so $\pi_{G_{\text{pop}}}(1 - (1-\lambda)(1-2\epsilon)) \leq \lambda D_{\text{pop}} + (1-\lambda)2\epsilon$, and so

$$\pi_{G_{\text{pop}}} \leq D_{\text{pop}} + 2\epsilon\tfrac{1-\lambda}{\lambda} = \beta - 2\epsilon. \qquad (2)$$

Now, let's show that the stationary probability of g_* is high. For any discussion group $g \notin G_{\text{pop}}$, we have

$$\begin{aligned} M(q)_{g,g_*} \geq& (1-\lambda) \sum_{p \notin F} A_{q,g}(p) P_{q,p,g}(g_*) \\ \geq& \tfrac{8}{7}(1-\lambda) \tfrac{\beta + \lambda D_{\text{pop}}/(1-\lambda)}{1-\beta} \sum_{p \notin F} A_{q,g}(p) \end{aligned}$$

(Recall that $\sum_{p \notin F} A_{q,g}(p) \geq 1 - \epsilon > \frac{7}{8}$.)

$$> (1-\lambda) \tfrac{\beta + \lambda D_{\text{pop}}/(1-\lambda)}{1-\beta}$$

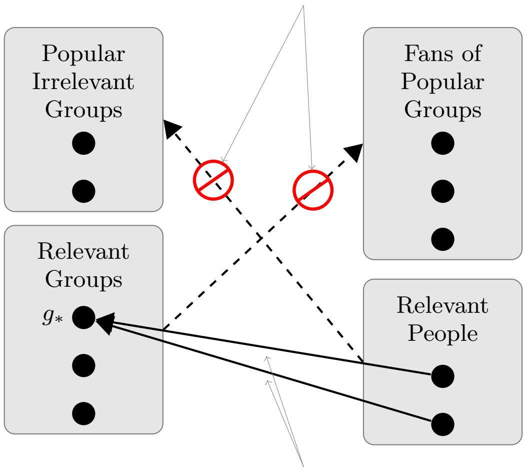
and so for any distribution d,

$$(dM(q))_{g_*} \geq (1-\lambda)(1 - d_{G_{\text{pop}}}) \tfrac{\beta + \lambda D_{\text{pop}}/(1-\lambda)}{1-\beta}.$$

We have:

$$\begin{aligned} \pi_{g_*} = (\pi M(q))_{g_*} \geq& (1-\lambda)(1 - \pi_{G_{\text{pop}}}) \tfrac{\beta + \lambda D_{\text{pop}}/(1-\lambda)}{1-\beta} \\ >& (1-\lambda)(1-\beta) \tfrac{\beta + \lambda D_{\text{pop}}/(1-\lambda)}{1-\beta} \\ =& \beta - 2\epsilon \geq \pi_{G_{\text{pop}}}. \quad \square \end{aligned}$$

People interested in q don't prefer popular but irrelevant groups, and fans of popular groups don't have authority in relevant groups.



People interested in q support the best group g_*, and have high authority scores in relevant groups.

Figure 2: Illustration of Scenario 5.7. A set of popular but irrelevant groups has many fans. The most relevant group g_* has fewer supporters, but the whole community of relevant groups gives authority to them. Under the right conditions, g_* will be ranked at the top (Thm. 5.8).

6. EXPERIMENTS

We present the results of running our algorithm on one year of tweets. We begin with the experimental setup and data description, and then explain our evaluation methodology. We show empirically that our algorithm performs significantly better than the baseline with respect to different performance measures. We also present qualitative results.

6.1 Experimental Setup

We obtained the set of all English language tweets posted in a 12 month time period starting from 6/2012. Given the scale of this data (several petabytes), we implemented our algorithm in the SCOPE language [9] and ran it offline over a large distributed computing cluster. We extracted the set of all distinct hashtags used in this timeframe, and the tweets associated with each hashtag, along with their corresponding users and timestamps. We processed each tweet message to extract the hashtags and the noun phrases present in the tweet (using a Part-of-Speech Tagger). The noun phrases serve to capture the potential queries that the tweet contains. Underutilized hashtags were removed (present in less than 60 tweets or used by less than 10 people), as were underutilized queries (less than 100 tweets).

Identifying Twitter Discussion Groups The set of Twitter Discussion Groups was determined as per §3. We consider the activity for the hashtag during each week, and analyze the fraction of the activity occurring during every possible duration of a short window of time each. In our implementation, we used $w = 2$ hours as the window length, and considered discrete time windows starting at every hour and half hour (since participants are likely to agree to meet at a round time such as 3:30 or 4:00). We then check if there is significant activity in the window with the largest activity during the week. We denote the window with the largest activity during the week as a "meeting" if at least $\gamma = 20\%$ of the activity for the hashtag in that week occurred during this window. We only consider the window with the largest activity during the week under the reasonable assumption that a large group of people are unlikely to have time to participate in multiple meetings in the same week. For a hashtag to be considered a discussion group, there should have been at least $m = 10$ weeks containing valid meetings. We obtained a total of $27K$ discussion groups using the above process.

Selecting candidate queries for ranking Since our algorithm is query-specific, we need to identify a set of representative queries against which to perform our evaluation. The union of all the noun phrases in the tweets gave us a set of 27 million potential queries, but a large fraction of them were phrases that were unrealistic as real queries (for example, phrases such as "someone", "next week" or "great day"). We sought a list of queries that capture how seekers query for groups, and queries posed to Yahoo Groups provided such a collection. We collected queries posed to Yahoo Groups based on five months of browsing behavior. After intersecting these queries with the set that we gathered from tweets, we were left with 2K queries.

A limitation of using Yahoo Group queries is that there may be Yahoo Groups not present on Twitter, and Twitter Discussion groups not present in Yahoo. Note that since no Twitter Discussion Group search engine exists, we are unable to use an existing query log for the purpose of our experiment. Rather, we are using Yahoo Group queries as a proxy for how people seek online communities.

Ground Truth Creation To evaluate the performance of our algorithm, we next need to obtain a ground truth ranked hashtag list for each query. However, given the number of candidate hashtags, this is clearly impossible to create manually — even for a few of the 2K candidate queries. Instead, we rely on an approach to obtain a (noisy) list of ground truth hashtags for each of a small set of queries, and then manually clean the list. For each candidate query, we identify a list of Twitter self-declared enthusiasts by selecting people who mention the query phrase in their Twitter profile. This is a simple approach and quite prone to error, *e.g.*, Jimmy Fallon (comedian) claims to be an astrophysicist in his Twitter profile [19]. We note that better techniques exist for identifying true experts, *e.g.*, [16, 19, 21, 26, 31, 41, 43] and these could perform better. We leave this as a potential direction for future work. Nevertheless, given the limited space allowed for a Twitter profile, people who explicitly mention the query (for example, "camera") in their Twitter profile, are more likely to be enthused (enjoy photography) than a random person who has merely used the query in a few tweets. For each query, we then rank hashtags based on their popularity among the tweets of the enthusiasts corresponding to the query. More specifically, we obtain a ranked list of hashtags for each query, where the ranking is based on the number of enthusiasts that have written tweets containing the query and the hashtag.

From among the $2K$ candidate queries, we were able to obtain this enthusiast ranking for only around 600 queries (for the remaining queries, we could not find enough enthusiasts who mentioned the query in their profile). Note that this coverage issue is another critical shortcoming of this method, and is the main reason why this cannot be a candidate algorithm for the discussion group ranking problem, even though it is used in creating the ground truth and (as seen later) has very good performance on the queries for which it returns an answer.

A manual evaluation of the enthusiast based ranking revealed that while the ranking had good precision for most queries, it had two shortcomings: first, it did not have sufficient recall and failed to report hashtags that we manually found to be very relevant to the query (*e.g.*, `#photography-chat` for the query, "camera" and `#tl_chat` for the query "travel", both of which are highly relevant Twitter discussion groups) and second, there were some queries on which its precision was quite poor.

To resolve these issues, we resorted to the pooling method in information retrieval [49] and manually created the final ground truth as follows: for each query, we pooled together the top 10 hashtags output by the above enthusiast ranking, the baselines, and our algorithm. We then asked a human assessor to consider each of these candidate hashtags in the pool, and manually annotate the hashtag (by scanning through the set of tweets corresponding to the hashtag, and performing a web search for information related to the hashtag) on a four-point graded relevance scale (with 3-being most relevant to the query, and 0 being irrelevant). Note that the human assessor did not have access to any information about which algorithm(s) generated the candidate hashtag in the pool. Since this process is extremely labor-intensive, we considered only the top 10 results from each algorithm for a given query, and restricted the set of queries for which we generated ground-truth rankings by sampling 50 queries from among the 600 candidate queries.

6.2 Implementation choices

Next we list the various implementation choices related to our model.

Authority Score We consider four different methods for assigning participants an authority score to capture how authoritative they are with respect to the discussion group and the query. (1) Noun-Frequency based Authority (NounFreqWeights): For each query and hashtag, we compute the authority score of a participant according to how many of their tweets contain both the query and the hashtag. A participant that tweets a lot about the query in the context of that hashtag is considered more authoritative than a participant with only a few tweets containing the (query, hashtag) pair. (2) @-mention Authority (@-mentionWeights): For each query and hashtag, we compute a participant's authority score according to the number of times the participant is @-mentioned in the context of the query and the hashtag. A participant that is @-messaged frequently in tweets containing the (query, hashtag) pair is considered more authoritative. (3) Follower Authority (FollowerWeights): We compute a participant's authority score according to how many followers they have on Twitter. A snapshot of the complete Twitter follower group was used to obtain follower counts. (4) Equal Authority (EqualWeights): For each query and hashtag, we give equal weights to all participants.

We report the performance of our algorithm with respect to each of these authority scores.

Teleport Distribution As described in §4, the teleport distribution for the random jumps in our group preference model can be either unweighted, or weighted according to the hashtag to which we are teleporting. We experiment with both options. For the unweighted case, the probability is divided among all hashtags equally. For the weighted case, we divide this probability among hashtags based on the fraction of tweets of this hashtag that contain the specific query. That is, the teleportation process is biased towards hashtags in which the query occurs more frequently. The intuition behind weighing the teleportation process is that if the input graph for the PageRank computation contains a few disjoint connected components, then ranking the hashtags across these two clusters would normally (in the unweighted case) depend only on the relative sizes of the components. By weighing the teleport distribution, we can factor in the query-specific popularity of hashtags when comparing hashtags from different connected components. As we will observe in the experimental results, weighting the teleport process significantly improves the quality of our rankings.

Preference Score As described in §4, a key component of our GroupPreference algorithm is the computation of *preference scores*. For any fixed query, participant, and last group g, these scores form a probability distribution representing the participant's preference for different groups g'. Hence, for a fixed query and a fixed participant, these scores can be viewed as a probability transition matrix over groups. For computing a participant's preference between hashtags (groups) g and g', we wish to only use Twitter data corresponding to the time when the participant was "aware" of *both* the hashtags. We define the participant's awareness-time for a hashtag as the first time when they tweeted with that hashtag. Using this definition, we then restrict the tweets of the participant to the time-period starting from the *later* of the awareness time for g and g'. For this time-period, we compute the number of meetings of g and g' attended by the participant for the given query (*i.e.*, the number of two-hour windows within which the participant posted at least one tweet containing the hashtag and the query). We define the transition from g to g' (resp. g' to g) to be valid, if the participant has attended "significantly more" (we use a relative difference threshold of 0.1 for estimating significance) meetings of g' compared to g (resp. g compared to g'). The combined transition probability of 1 from g is then equally divided among all valid transitions from g. If the participant does not have a significant preference for any group g' over g (and hence there is no valid transition from g), we assign a transition probability of 1 from g to itself. Formally, for query q and participant p, let $G^{pref}_{q,p,g}$ denote the set of groups for which p has significant preference over group g. If $G^{pref}_{q,p,g}$ is non-empty, then we set $P_{q,p,g}(g') = 1/|G^{pref}_{q,p,g}|$ for $g' \in G^{pref}_{q,p,g}$, and $P_{q,p,g}(g') = 0$ for $g' \notin G^{pref}_{q,p,g}$. If $G^{pref}_{q,p,g}$ is empty, then we set $P_{q,p,g}(g) = 1$ and $P_{q,p,g}(g') = 0$ for any g' different from g.

6.3 Baseline Algorithms

We compared our GroupPreference algorithm against the following baselines, all of which correspond to various intuitive notions of the popularity of a discussion group on Twitter with respect to a given query.

User Frequency-based Ranking Algorithm (UFA): For each query, we assign a score to each hashtag based on the number of distinct participants that have posted tweets containing the given hashtag and query.

TFIDF Algorithm (TFIDF): We treat all tweets corresponding to a hashtag as a document. For each query, we rank hashtags by their TFIDF scores [42].

Tweet Ratio-based Ranking Algorithm (TRA): For each query, we assign a score to each hashtag based on the ratio of the number of tweets containing that hashtag divided by the number of tweets containing both the hashtag and the query.

In addition to the above three baselines, we also compare our algorithm against the enthusiast ranking (EnthusiastPreference) algorithm mentioned previously, that was used for creating the ground truth. As mentioned previously, while this is not a practical algorithm due to its extremely low coverage of queries, we still use it as an upper bound for a practical ranking algorithm and compare our algorithms against the performance of EnthusiastPreference.

6.4 Evaluation Metrics

For our evaluation, we compute metrics for each algorithm by comparing it with the ground truth ranking. For a given query, let A and G be the ranked list of discussion groups identified by an algorithm and by the ground truth respectively, with $A[i]$ (resp. $G[i]$) being the i^{th} discussion group. For every discussion group p, let $R(p) \in [0, 3]$ be the ground truth relevance rating provided by the human assessor. We define the following metrics [46]:

Weighted Precision: The WeightedPrecision @K of the algorithm at the top K rank is $\frac{\sum_{i=1}^{K} R(A[i])}{3K}$.

Weighted Recall: The WeightedRecall @K of the algorithm at the top K rank is $\frac{\sum_{i=1}^{K} R(A[i])}{\sum_{p \in G} R(p)}$

Weighted Mean Average Precision: The WeightedMAP of the algorithm is $\frac{1}{|G|} \cdot \sum_{p \in (G \cap A)} \mathrm{WeightedPrecision\ @r_{p,A}}$, where $r_{p,A}$ is the rank of group p in A.

NDCG: The NDCG@K of the algorithm at the top K rank is $\frac{\mathrm{DCG(A)}}{\mathrm{DCG(G)}}$, where $\mathrm{DCG(A)} = R(A[1]) + \sum_{i=2}^{K} \frac{R(A[i])}{\log_2 i}$.

In addition, we also compute the unweighted versions of the above metrics corresponding to precision (Precision @K), recall (Recall @K) and Mean Average Precision (MAP).

For the unweighted metrics, the relevance rating of a group is rounded to 1 if $R(p) \geq 2$ and 0 otherwise. We set $K = 5$.

6.5 Results of Implementation Choices

Teleport Distribution We first study the effect of varying the teleport probability from 0 to 1, with NounFreqWeights as the authority score. From Table 1, we first observe the significant benefit of having a non-zero teleport probability. This observation can be explained by the presence of several disjoint connected components of varying sizes in the graph formed over hashtags. For example, the graph over hashtags for the query "photography" consists of two large connected components: the first component consists of highly relevant groups such as #photographytips, #phototips, #photog and #photochat, while the second component consists of several less relevant hashtags such as #northeasthour, #yorkshirehour, #bathhour and #devonhour. In the absence of the option to teleport, the surfer may get stuck in the less relevant component. Even with a small teleport probability, the surfer is able to explore components containing relevant hashtags, and consequently, our algorithm is able to rank such hashtags higher.

As the teleport probability is increased, the performance improves initially, maximizing at 0.25, and then drops because the surfer teleports too often instead of moving towards better hashtags. Hence, we chose 0.25 as the teleport probability for further analysis. We next validate the benefit of having a biased teleport distribution (Table 2), confirming that it is desirable to factor in the query-specific popularity of hashtags instead of teleporting uniformly.

Authority Score We present a comparison of different authority scores in Table 3. We were at first surprised to observe similar performance across different authority scores, since these scores correspond to orthogonal signals. In fact, giving equal weight to all participants performed slightly better than the other three authority scores. A possible explanation is that for a given query, the signal to discriminate highly relevant hashtags from highly irrelevant hashtags are spread across many participants, and the aggregate preference captures this signal irrespective of the weights given to the participants. The participants may differ in their finer preferences over relevant hashtags (*e.g.*, between #rosechat and #gardenchat for the query "garden"), and hence, while the authority scores can influence the final relative ordering of two highly relevant hashtags, our metrics are unaffected if the positions of two such groups are swapped. Even though the authority scores did not significantly influence the performance measures at the aggregate level, we did observe relatively large variance in performance at the level of individual queries.

6.6 Performance Results

We next compare the performance of our algorithm (with NounFreqWeights as the authority score) with the three baselines, and the enthusiast-based ranking in Table 4. We observe that our algorithm significantly outperforms the best baseline, TFIDF along all seven metrics. Our algorithm improves TFIDF by 30% with respect to mean average precision (0.437 vs 0.336), about 25% with respect to weighted mean average precision (0.309 vs 0.246), and about 20% with respect to NDCG (0.488 vs 0.404). With respect to these three metrics, our algorithm achieves about 70% of the performance of EnthusiastPreference, which, as noted earlier, is not a practical algorithm but can serve as an upper bound.

Qualitative Evaluation of Rankings

To provide qualitative insights into the ranking algorithms, we next highlight the top-3 Twitter hashtags retrieved by the different algorithms for 4 representative queries, in Table 5. (We omitted the UFA baseline due to its poor performance.) A quick scan on Twitter of the tweets related to the retrieved hashtags will reveal that for most of these queries, the GroupPreference algorithm clearly retrieves more relevant groups compared to the baselines, and performs almost as well as EnthusiastPreference. For example, for the query "garden", both GroupPreference and EnthusiastPreference retrieve a weekly Twitter group about gardening enthusiasts (#gardenchat) as the top hashtag (though EnthusiastPreference also retrieves another related Twitter group related to roses (#rosechat). On the other hand, the baselines results are not very relevant. Indeed, TFIDF returns a hashtag related to Justin Bieber's "Believe Tour" at Madison Square Garden, simply due to the sheer number of tweets containing both "#believetour" and "garden". Similarly, for the query "resume", GroupPreference returns three relevant weekly Twitter groups about jobs and hiring (#omcchat, #animalchat and #hfchat), and outperforms all the other algorithms that return at least one group that is not a discussion group (for example, #jobfair or #forbesgreatesthits). For the query "hotels", both GroupPreference and EnthusiastPreference return a travel-related weekly Twitter group as the top-ranked hashtag (#tni and #ttot respectively), whereas the baselines' top hashtag is not as relevant (#dimiami is a Miami-specific travel hashtag).

7. DISCUSSION AND FUTURE WORK

Group selection is admittedly more complicated than combining preference with authoritativeness. For example, among two groups that equally discuss a topic, the group that is more open to outsiders may be more preferable. The age and size of a group may also play a role in that mature, sizeable groups may be less welcome to newbies than younger, smaller groups. There are other potential factors [24]: for example, the quality of the relationships in the group (both online and offline), whether participant privacy is respected, and how conflict is handled (netiquette). Our work implicitly uses these signals by following the trail of participation left by authoritative users, but explicit use of such signals may lead to better solutions.

Personalized group ranking is another potential direction. For example, the demographic makeup of a group (race, gender, age) may be used to match a user's demographic. The language/vocabulary of a group is known to impact further participation [5, 18] and consequently may be used to improve ranking. The nature of groups that a user already participates in may also be an indication of the kinds of groups the user wishes to join. Richer graph structure signals such as the number of friends that a person has in the group and how connected their friends are could also be useful [2].

Furthermore, varying query types may call for varying groups. If the query suggests a user seeking knowledge or new expertise about a subject, then groups that frequently invite outside experts to answer questions may be more desirable. Other queries may suggest users seeking groups for humor or entertainment, and this could be another factor that improves ranking.

Moreover, the dynamic nature of Twitter implies that the best venue to discuss a topic may shift over time. In addition, participant authoritativeness may rise and fall over time. It would be valuable to understand and characterize when and how often these changes occur. If the ground truth changes repeatedly, then methods may be needed to quickly detect these shifts and rerank groups accordingly.

Our goal in ranking groups is to connect a new user to a group of like-minded users. The true test of whether our algorithm works is if users positively respond to our ranking. To that end, a practical direction is to build a system that runs our ranking algorithm periodically, and allows users to obtain the most recently computed ranked lists of discussion groups corresponding to their issued queries. One concern that needs to be addressed prior to deployment is determining which groups want to be found. Even though these conversations take place in "public", some participants may hide in obscurity [22], *e.g.*, believe that their conversations are invisible to current search engines, or communicate with fake accounts. Another concern is the potential consequences of overwhelming these groups with new members, *e.g.*, message overload [8, 29]. Historically, groups have found ways to cope with large membership, *e.g.*, by splitting into smaller groups, by geography or subtopic.

8. REFERENCES

[1] R. Agrawal, A. Halverson, K. Kenthapadi, N. Mishra, and P. Tsaparas. Generating labels from clicks. In *WSDM*, 2009.

[2] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group formation in large social networks: Membership, growth, and evolution. In *KDD*, 2006.

[3] L. Backstrom, R. Kumar, C. Marlow, J. Novak, and A. Tomkins. Preferential behavior in online groups. In *WSDM*, 2008.

[4] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30, 1998.

[5] C. Budak and R. Agrawal. Participation in group chats on Twitter. In *WWW*, 2013.

[6] C. Burges, R. Ragno, and Q. V. Le. Learning to rank with nonsmooth cost functions. In *NIPS*, 2006.

[7] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *ICML*, 2005.

[8] B. S. Butler. Membership size, communication activity, and sustainability: A resource-based model of online social structures. *Info. Sys. Research*, 12(4), Dec. 2001.

[9] R. Chaiken, B. Jenkins, P. Larson, B. Ramsey, D. Shakib, S. Weaver, and J. Zhou. SCOPE: Easy and efficient parallel processing of massive data sets. *PVLDB*, 1(2), 2008.

[10] S. Chien, C. Dwork, R. Kumar, D. R. Simon, and D. Sivakumar. Link evolution: Analysis and algorithms. *Internet Mathematics*, 1(3), 2003.

[11] W. Cohen, R. Schapire, and Y. Singer. Learning to order things. *JAIR*, 10, 1999.

[12] G. Cong, L. Wang, C.-Y. Lin, Y.-I. Song, and Y. Sun. Finding question-answer pairs from online forums. In *SIGIR*, 2008.

[13] J. Cook, K. Kenthapadi, and N. Mishra. Group chats on Twitter. In *WWW*, 2013.

[14] K. Csalogány, D. Fogaras, B. Rácz, and T. Sarlós. Towards scaling fully personalized pagerank: Algorithms, lower bounds, and experiments, 2005.

[15] C. Danescu-Niculescu-Mizil, R. West, D. Jurafsky, J. Leskovec, and C. Potts. No country for old members: User lifecycle and linguistic change in online communities. In *WWW*, 2013.

[16] K. Ehrlich, C.-Y. Lin, and V. Griffiths-Fisher. Searching for experts in the enterprise: Combining text and social network analysis. In *GROUP*, 2007.

[17] J. L. Elsas and J. G. Carbonell. It pays to be picky: An evaluation of thread retrieval in online forums. In *SIGIR*, 2009.

[18] D. Forsyth. *Group dynamics*. Wadsworth, 2009.

[19] S. Ghosh, N. K. Sharma, F. Benevenuto, N. Ganguly, and P. K. Gummadi. Cognos: Crowdsourcing search for topic experts in microblogs. In *SIGIR*, 2012.

[20] F. Godin, V. Slavkovikj, W. D. Neve, B. Schrauwen, and R. V. de Walle. Using topic models for Twitter hashtag recommendation. In *WWW (Companion Volume)*, 2013.

[21] P. Gupta, A. Goel, J. Lin, A. Sharma, D. Wang, and R. Zadeh. WTF: The who to follow service at Twitter. In *WWW*, 2013.

[22] W. Hartzog and F. Stutzman. The case for online obscurity. *California Law Review*, 101(1), 2013.

[23] S. Ieong, N. Mishra, and O. Sheffet. Predicting preference flips in commerce search. In *ICML*, 2012.

[24] A. Iriberri and G. Leroy. A life-cycle perspective on online community success. *CSUR*, 41(2), 2009.

[25] G. Jeh and J. Widom. Scaling personalized web search. In *WWW*, 2003.

[26] J. Jiao, J. Yan, H. Zhao, and W. Fan. Expertrank: An expert user ranking algorithm in online communities. In *NISS*, 2009.

[27] T. Joachims. Optimizing search engines using clickthrough data. In *KDD*, 2002.

[28] T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Trans. Inf. Syst*, 25(2), 2007.

[29] Q. Jones, G. Ravid, and S. Rafaeli. Information overload and the message dynamics of online interaction spaces: A theoretical model and empirical exploration. *Info. Sys. Research*, 15(2), 2004.

[30] E. Joyce and R. Kraut. Predicting continued participation in newsgroups. *J. Comput. Mediat. Comm*, 11(3), 2006.

[31] H. Kautz, B. Selman, and M. Shah. Referral web: Combining social networks and collaborative filtering. *Commun. ACM*, 40(3), 1997.

[32] J. Kleinberg. Authoritative sources in a hyperlinked environment. In *SODA*, 1998.

[33] G. Kossinets and D. J. Watts. Empirical analysis of an evolving social network. *Science*, 311(5757), 2006.

[34] S. M. Kywe, T. A. Hoang, E. P. Lim, and F. Zhu. On recommending hashtags in Twitter networks. In *SocInfo*, 2012.

[35] C. Lampe and E. Johnston. Follow the (slash) dot: Effects of feedback on new members in an online community. In *GROUP*, 2005.

[36] R. Lempel and S. Moran. Salsa: The stochastic approach for link-structure analysis. *ACM TOIS*, 19(2), 2001.

[37] R. Lempel and S. Moran. Rank-stability and rank-similarity of link-based web ranking algorithms in authority-connected graphs. *Information Retrieval*, 8(2), 2005.

[38] M. L. Markus. Toward a "critical mass" theory of interactive media universal access, interdependence and diffusion. *Communication research*, 14(5), 1987.

[39] M. Najork, S. Gollapudi, and R. Panigrahy. Less is more: Sampling the neighborhood graph makes SALSA better and faster. In *WSDM*, 2009.

[40] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[41] A. Pal and S. Counts. Identifying topical authorities in microblogs. In *WSDM*, 2011.

[42] A. Rajaraman and J. D. Ullman. *Mining of massive datasets*. Cambridge University Press, 2011.

[43] T. Reichling, K. Schubert, and V. Wulf. Matching human actors based on their texts: Design and evaluation of an instance of the expertfinding framework. In *GROUP*, 2005.

[44] Y. Ren and R. E. Kraut. A simulation for designing online community: Member motivation, contribution, and discussion moderation. *Info. Sys. Research*, 2011.

[45] B. Saha and L. Getoor. Group proximity measure for recommending groups in online social networks. In *SNA–KDD Workshop*, 2008.

[46] T. Sakai. On the reliability of information retrieval metrics based on graded relevance. *Inform. Process. Manag*, 2007.

[47] P. Tsaparas. Using non-linear dynamical systems for web searching and ranking. In *PODS*, 2004.

[48] A. Tversky. Elimination by aspects: A theory of choice. *Psychological review*, 79(4), 1972.

[49] E. M. Voorhees and D. K. Harman. *TREC: Experiment and evaluation in information retrieval*. MIT Press, 2005.

[50] S. Whittaker, L. Terveen, W. Hill, and L. Cherny. The dynamics of mass interaction. In *From Usenet to CoWebs*. Springer London, 2003.

[51] L. Yang, T. Sun, M. Zhang, and Q. Mei. We know what @you #tag: Does the dual role affect hashtag adoption? In *WWW*, 2012.

[52] E. Zangerle and W. Gassler. Recommending #-tags in Twitter. In *CEUR Workshop*, 2011.

Teleport Probability	Precision	Recall	MAP	Weighted Precision	Weighted Recall	Weighted MAP
0.00	0.091	0.110	0.083	0.092	0.117	0.068
0.15	0.395	0.486	0.425	0.347	0.437	0.303
0.25	0.395	0.491	0.437	0.350	0.447	0.309
0.50	0.382	0.463	0.423	0.332	0.412	0.297
0.75	0.364	0.451	0.414	0.323	0.408	0.288
1.00	0.350	0.440	0.391	0.306	0.395	0.271

Table 1: Effect of Varying Teleport Probability

Teleport Bias	Precision	Recall	MAP	Weighted Precision	Weighted Recall	Weighted MAP
Uniform	0.177	0.224	0.169	0.171	0.211	0.131
Biased	0.395	0.491	0.437	0.350	0.447	0.309

Table 2: Benefit of Non-uniform Teleport Distribution

Authority Score	Precision	Recall	MAP	Weighted Precision	Weighted Recall	Weighted MAP
NounFreqWeights	0.395	0.491	0.437	0.350	0.447	0.309
FollowerWeights	0.377	0.467	0.461	0.345	0.433	0.330
@-mentionWeights	0.382	0.467	0.467	0.341	0.423	0.332
EqualWeights	0.400	0.485	0.479	0.359	0.446	0.340

Table 3: Empirical Analysis of Different Authority Scores

Algorithm	Precision	Recall	MAP	Weighted Precision	Weighted Recall	Weighted MAP	NDCG
UFA	0.236	0.280	0.232	0.212	0.277	0.168	0.301
TRA	0.273	0.377	0.313	0.245	0.348	0.217	0.362
TFIDF	0.309	0.362	0.336	0.288	0.366	0.246	0.404
GroupPreference	0.395	0.491	0.437	0.350	0.447	0.309	0.488
EnthusiastPreference	0.532	0.706	0.611	0.480	0.636	0.446	0.691

Table 4: Performance of Different Algorithms

Query	TFIDF	TRA	GroupPreference	EnthusiastPreference
garden	#believetour #beastmode #knicks	#fuego #joedirt #count	#gardenchat #fuego #joedirt	#gardenchat #growyourown #rosechat
hotels	#dimiami #united #ttot	#dimiami #tunehotelquiz #dolcehotels	#tl_chat #traveltuesday #tni	#ttot #traveltuesday #barcelona
photographers	#photog #togchat #phototips	#photographychat #togchat #thegridlive	#photographychat #phototips #togchat	#photog #scotland #sbs
resume	#forbesgreatesthits #hfchat #sctop10	#momken #resuchat #hfchat	#omcchat #animalchat #hfchat	#hfchat #jobhuntchat #jobfair

Table 5: Sample Discussion Group Rankings using Different Algorithms