From the Telegraph to Twitter Group Chats

James Cook

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2014-59
http://www.eecs.berkeley.edu/Pubs/TechRpts/2014/EECS-2014-59.html

May 9, 2014

From the Telegraph to Twitter Group Chats

by

James Alexander Cook

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Satish Rao, Chair
Professor John Canny
Professor Jasjeet Sekhon

Spring 2014

From the Telegraph to Twitter Group Chats

Abstract

From the Telegraph to Twitter Group Chats

by

James Alexander Cook

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Satish Rao, Chair

Communication now is easier than ever before. One consequence of this is the emergence of virtual communities, unconstrained by physical proximity. We perform two investigations into changing social trends. We study a corpus of 100 years of newspaper articles to see if we can find evidence to support the popular intuition that as news cycles have sped up, the public's attention span has gotten shorter. We find no such evidence: to the contrary, we find that the typical length of time that a person's name stays in the news has not changed over time, and celebrities now stay in the news for longer than ever before. We also investigate a new kind of community on Twitter called a group chat, where members have regular meetings to discuss a broad range of topics, from medical conditions to hobbies. We find that the phenomenon is growing over time, and paint a broad picture of the topics which one could find a group chat to discuss. With a view to helping connect new participants to group chats they may not have been able to find or might not have been aware of, we design an algorithm to rank group chats in the context of a topic given as a query.

To Mom and Dad.

Some material in this thesis was written in collaboration with others, and appears in or was submitted to conferences:

- James Cook, Atish Das-Sarma, Alex Fabrikant, Andrew Tomkins. *Your Two Weeks of Fame and Your Grandmother's*. Proceedings of the 2012 International World Wide Web Conference. ACM.
- James Cook, Nina Mishra, Krishnaram Kenthapadi. *Group Chats on Twitter*. Proceedings of the 2013 International Word Wide Web Conference. ACM
- James Cook, Abhimanyu Das, Krishnaram Kenthapadi, Nina Mishra. *Ranking Discussion Groups*. Submitted to the 2014 SIGKDD conference.

Contents

List of Figures

List of Tables

Acknowledgments

I was a young and lost graduate student only in my fourth year when Alex Fabrikant encouraged me to do an internship at Google which began the story of this thesis. Then it was Nina Mishra who brought me over to Microsoft Research, where the other part of this work was born. Alex and Nina, together with Atish Das Sarma, Abhimanyu Das, Krishnaram Kenthapadi and Andrew Tomkins, welcomed me, showed me how it all works, and worked hard to make the projects a success.

My advisors Luca Trevisan and Satish Rao have the patience of saints. They gave me the freedom to wander far from the usual domain of the Theory group, and my quals and dissertation committees — Satish, John Canny, Marti Hearst, Christos Papadimitriou and Jasjeet Sekhon — guided my wanderings toward a thesis.

I would be remiss not to mention Geoffrey Hinton, with whom I did my first research, and who deserves substantial credit for launching me into grad school, along with Faith Ellen and Dror Bar-Natan, who must have exaggerated in their letters for Berkeley to have admitted me.

Berkeley's CS department is known for its friendly community of graduate students. I'll miss all my fellow students. A random selection: Anand, who likes learning languages, Anindya, who explained politics to me and wrote an exemplary acknowledgements section, Di, who was very patient when I thought I had cool ideas about linear algebra, Greg, Omid, who showed me how to think clearly about strange new things, Rishi, who introduced me to bouldering, my new hobby, Piyush, who has access to an oracle, Siuman, Siuon, Thomas, who kept everyone organized, Tom, whose meticulous CS70 preparation I will make good use of this summer, and Urmila, who brought Satish to my dissertation talk and made me write this section.

This thesis concludes 25 years of school. I would have escaped earlier if it hadn't been for all my teachers along the way: in public school, at the University of Toronto, and at Berkeley, and Jeannette Zingg and Marshall Pynkoski of the School of Atelier Ballet who taught me discipline and presentation. I might have decided to finish more quickly if not for my friends along the way who made Toronto and then Berkeley such nice places to stay.

And I couldn't have gotten started without my parents, who taught me the basics, and my brother Gordon, who taught me how to count past ten or twenty.

Chapter 1

Introduction

Beginning in the 19th century, long-distance communication transitioned from foot to telegraph on land, and from sail to steam to cable by sea. Each new form of technology began with a limited number of dedicated routes, then expanded to reach a large fraction of the population, eventually resulting in near-complete deployment of digital electronic communication. Each transition represented an opportunity for news to travel faster, break more uniformly, and reach a broad audience closer to its time of inception.

Meanwhile, a change has been taking place in the way that people form communities. The term "community" has classically been associated with local physical meetings of groups of people such as the Lions Club and Rotary Club. Participation in these groups is primarily for social capital: for example, mutual support, cooperation, trust, good will, fellowship and sympathy. Over time, many of these physical communities have dissipated due to factors such as urban sprawl, families with two working parents, and time pressures [55]. In contrast, the term *virtual community* was first coined in 1993 by Howard Rheingold [58] who described them as "social aggregations that emerge from the Net when enough people carry on public discussions long enough, with sufficient human feeling to form webs of personal relationships in cyberspace". Virtual communities include forums, chat rooms, discussion boards, Usenet groups and Yahoo groups [4].

We perform two studies related to these changing communication media and shifting forms of community. In Parts I and II, we investigate a popular and growing phenomenon called a Twitter group chat, and develop an algorithm to search for group chats relevant to a user's interests. In Part III, we investigate the phenomenon of fame in newspaper articles from 1895 to 2010, inspired by popular intuition that the public's attention span is getting shorter as news cycles get faster.

1.1 Group Chats on Twitter (Parts I and II)

We report on a new kind of community that meets on Twitter: periodic, synchronized conversations focused on specific topics, called *group chats*. The chats cover a broad range of

topics: for example, there are support groups for post-partum depression and mood disorders, and groups of hobbyists meet to discuss skiing, photography, wine, and food. Members of a group conversation communicate using an agreed upon *hashtag* (a short string preceded by a '`#`' sign). For example, in a group of passionate movie-goers, members agree to include `#mtos` in every tweet. In addition to a hashtag, members also agree on a day and time: for example, every Sunday evening at 20:00 GMT, hence the name "Movie Talk on Sunday". Many of these groups are moderated[1] to ensure that each meeting has a focused subject: for example, suspense movies. Both active participants who tweet and passive users who follow the conversation benefit from the excitement of live communication. The topics of these chats span many categories, from health support groups to arts and entertainment.

In Part I, we present an algorithm based on a new quantitative definition to determine what group chats exist on Twitter, and examine the topics they cover and the growth of the group chat phenomenon over time. Note that group chats are not explicitly registered with Twitter; if they were, our task of listing all group chats would be much easier. While we did find a crowdsourced list [64], it was incomplete and contained some things that are not group chats.

Upon extensive observation, it is clear that people derive immense value from Twitter group chats. Like other kinds of community, group chats serve as a place to exchange knowledge, to share experiences, to provide empathy and to be included. For example, people living with diabetes have a means to share their difficulties with glucose monitoring. Caregivers of Alzheimer's patients have an opportunity to benefit from the knowledge and experiences of other Alzheimer's caregivers. People coping with addiction find a way to discuss their daily battles resisting addiction. People with shared hobbies also meet to discuss their passion for wine in `#winechat`, for skiing in `#skichat` and for photography in `#photog`. Without these Internet discussions, it is unclear whether these people, often in geographically spread out locations, would have another way to communicate in real-time.

Even though these groups are known to their participants, there are far more people not in the discussion that we believe would benefit from listening and participating. In order to help the next person in need of a community, we are motivated in Part II to develop a search engine that can rank groups: given a search query, we seek to find an ordering of groups where the topic of the query is best discussed.

The problem would be easier if the subjects of the groups were disjoint, but there is topic overlap. For example, irrigation strategies are discussed in landscape, gardening and ground care groups. In fact, for some topics we found sister groups with identical subjects (watch movies in real-time via a shared link), where we do not know if one group is aware of the other. Note that our intent is to connect new users to groups of their interest. We are aware that if a group grows too large, people may depart due to message overload [11, 38]. Historically, groups have found ways of coping with an increased number of users, e.g, splitting into

[1]The moderator cannot stop people from sending messages to the group by tweeting using the group hashtag. Instead, the moderator takes an active role in suggesting topics for discussion and repeating interesting tweets.

smaller groups by subtopic or by geography.

There are many other kinds of online group to which we can compare Twitter group chats. Online forums provide a place for communities to form around particular subjects, but lack the real-time nature of Twitter group chats. Chat rooms allow real-time conversation, but do not usually have regular scheduled meetings: instead, participants drop in and out at will. Another key difference is that group chats are implicit and therefore not easily discoverable. In contrast, the list of Yahoo! groups or the rooms on an IRC (Internet Relay Chat) server can be searched and browsed. The website Reddit often features "IAmA" conversations featuring an interesting person — often a celebrity or an expert in some area — who answers questions from other users of Reddit. This could be compared to the appearance of guest stars that often happens in Twitter group chat meetings.

That Twitter is used to organize such discussions is perhaps surprising. The 140 character limit imposes a succinctness that seems unsuitable for group discussions. But the already large-scale adoption of Twitter, together with its real-time nature, has enabled these group conversations to grow to a massive scale. Also surprising are support group chats. It is hard to imagine how support can be given or received in 140 characters. But people have found a way!

While the existence of group chats is certainly known to members who participate in these chats, we have not seen any work in the published literature reporting the number and variety of these groups. Our work is focused on algorithms for automatically finding groups at scale.

1.1.1 Contributions — Part I (Group Chats on Twitter)

We begin by presenting a definition for a group chat, based on key properties of groups abstracted from the sociology literature. The key components of the definition are regular bounded-length meetings and cohesion among active group members. We found meetings with a fixed period and duration, such as "every Wednesday at 2-3pm PST", to be quite common among groups on Twitter. A predictable, agreed-upon meeting time may help people plan their schedule and focus for a bounded time on a particular subject. However, we found that defining a group chat to be any group with periodic bounded-duration meetings was not enough. For example, hashtags associated with weekly television shows, such as `#dwts` for "Dancing With the Stars", can show increased activity when the show is on the air. This behaviour shows extremely periodic, fixed duration meetings, but these are not group conversations: users broadcast their thoughts on the show, but are not participating in a single conversation. The last part of our definition addresses this: group chats must have cohesion, which we measure by looking at communication among active group members.

Next, we propose an algorithm for finding group chats. We begin with a large collection of candidate groups and repeatedly remove those that do not satisfy our definition. Every hashtag is an initial candidate. In the first step, the algorithm removes candidates that do not have routine meetings. To identify these, for each candidate, the time series of exchanged messages is computed and a method based on the Fourier transform is applied to the time

series to identify periodic candidates. At this point, the candidates may contain those that "meet" all day: for example, the hashtag `#ff` stands for "Follow Friday" and is used throughout the day to recommend Twitter accounts worth following. In order to eliminate these, we restrict ourselves to groups where at least 20% of the messages are exchanged within a short span of time. At this point, the remaining candidates may still include those that meet but never really engage in a conversation, so in the final step we remove candidates that are not cohesive. What remain are the group chats.

We prove that hashtags that have certain properties of group chats are accepted by our algorithm, and that non-group chat hashtags, under certain natural models of how tweets with such hashtags could be generated, are rejected. Specifically, if a group meets sufficiently many times, group meetings are well-separated, and a reasonable number of tweets are exchanged per meeting, then we prove that our algorithm will accept the hashtag, under certain assumptions. On the other hand, if a hashtag is generated randomly from a daily activity cycle that is not concentrated in time, or is generated in meetings that happen at uniformly random times, or the hashtag does not have cohesion, then our algorithm will reject the hashtag.

Finally, we run our algorithm over two years of Twitter data. We find 1.4 thousand groups involving 2.3 million users. To provide a glimpse into these groups, we show the distribution of the periods of these groups. Most groups meet weekly. To check the quality of the groups discovered, we randomly sample 10% of the groups and report on the categories of groups represented in the sample, finding that most are interest-driven groups such as music enthusiasts, sports lovers, and foodie communities. We also find many support and self-help groups. Finally, we compute the times of birth and death of each group in order to find the cumulative number of living groups over time. The data suggest that group chats are a growing phenomenon. We hope that this discovery inspires others to study group chats on Twitter.

1.1.2 Contributions — Part II (Ranking Discussion Groups)

In Part II, we begin by broadening the set of chats we consider, from the notion of a group chat to a more general notion we call a *discussion group*, or a *chat* in the context of Twitter. The definition of a group chat is conservative, since our goal in Part I is to count and categorize them, whereas this more general definition of a discussion group will give users of a search 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 and repeatedly finds an authoritative user in the group and walks to a random group according to what the authoritative user prefers. The algorithm to solve this problem involves computing the stationary distribution of a matrix. Since the stationary distribution is unstable in the sense that small changes to a matrix can alter the final ordering, we analytically show that a variety of natural changes to the underlying data still yield the expected ranking. 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 one group is preferred to another and a particular user agrees with this preference, then increasing their preference or authority in a new dataset will also retain this preference. The goal of this analysis is to build confidence that the algorithm will not wildly change the ranking under reasonable changes to the underlying data.

Naïve solutions, such as ordering groups by how often the query appears in the discussion, also turn out to satisfy the properties that we consider. Thus, we describe a scenario (based on what we observed in practice) and analytically show that our group preference model will succeed where naïve solutions fail. These findings are also borne out in our experiments. For example, when it became known a former prime minister suffered from dementia, a large number of tweets were generated in news-related discussion groups about dementia. These news-related groups are not a good place to discuss dementia. On the other hand, we prove (and experimentally show) that if we order groups according to our model that a group where dementia is actively discussed will be ranked higher than a news-related group.

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 and mean average precision and show that our algorithm outperforms the baseline on all of these metrics.

1.2 Fame in News (Part III)

The increasing speed of the news cycle is a common theme in discussions of the societal implications of technology. Stories break sooner, and news sources cover them in less detail before quickly moving on to other topics. Online and cable outlets aggressively search for novelty in order to keep eyeballs glued to screens. Popular non-fiction dedicates significant coverage to this trend, which by 2007 prompted *The Onion*, a satirical website, to offer the following commentary on cable news provider CNN's offerings: "CNN is widely credited with initiating the acceleration of the modern news cycle with the fall 2006 debut of its spin-off channel CNN:24, which provides a breaking news story, an update on that story, and a news recap all within 24 seconds."

With this speed-up of the news cycle comes an associated concern that, whether or not causality is at play, attention spans are shorter, and consumers are only able to focus for progressively briefer periods on any one news subject. Stories that might previously have occupied several days of popular attention might emerge, run their course, and vanish in a single day. This popular theory is consistent with a suggestion by Herbert Simon [61] that as the world grows rich in information, the attention necessary to process that information becomes a scarce and valuable resource.

The speed of the news cycle is a difficult concept to pin down. We focus our study on the most common object of news: the individual. An individual's fame on a particular day might

be thought of as the probability with which a reader picking up a news article at random would see their name. From this idea we develop two notions of the duration of the interval when an individual is in the news. The first is based on falloff from a peak, and intends to capture the spike around a concrete, narrowly-defined news story. The second looks for a period of sustained public interest in an individual, from the first time the public notices that person's existence until the public loses interest and the name stops appearing in the news. We study the interaction of these two notions of "duration of fame" with the radical shifts in the news cycle we outline above. For this purpose, we employ Google's public news archive corpus, which contains over sixty million pages covering 250 years, and we perform what we believe to be the first study of the dynamics of fame over such a time period.

Data within the archive is heterogeneous in nature, ranging from directly captured digital content to optical character recognition employed against microfilm representations of old newspapers. The crawl is not complete, and we do not have full information about which items are missing. Rather than attempt topic detection and tracking in this error-prone environment, we instead directly employ a recognizer for person names to all content within the corpus; this approach is more robust, and more aligned with our goal of studying fame of individuals.

Based on these different notions of periods of reference to a particular person, we develop at each point in time a distribution over the duration of fame of different individuals.

Our expectation upon undertaking this study was that in early periods, improvements to communication would cause the distribution of duration of coverage of a particular person to shrink over time. We hypothesized that, through the 20th century, the continued deployment of technology, and the changes to modern journalism resulting from competition to offer more news faster, would result in a continuous shrinking of fame durations, over the course of the century into the present day.

1.2.1 Summary of Findings

We did indeed observe fame durations shortening in the early 20th century, in line with our hypothesis about accelerating communications. However, from 1940 to 2010, we saw quite a different picture. Over the course of 70 years, through a world war, a global depression, a two order of magnitude growth in (available) media volume, and a technological curve moving from party-line telephones to satellites and Twitter, both of our fame duration metrics showed that neither the typical person in the news, i.e. the median fame duration, nor the most famous, i.e. high-volume or long-duration outliers, experienced any statistically significant decrease in fame durations.

As a matter of fact, the bulk of the distribution, as characterized by median fame durations, stayed constant throughout the entire century-long span of the news study and was also the same through the decade of Blogger posts on which we ran the same experiments. As another heuristic characterization of the bulk of the distribution, both news and Blogger data produced roughly comparable parameters when fitted to a power law: an exponent of around -2.5, although with substantial error bars, suggesting that the fits were mediocre.

Furthermore, when we focused our attention on the very famous, by various definitions, all signs pointed to a slow but observable growth in fame durations. From 1940 onward, on the scale of 40-year intervals, we found statistically significant fame duration growth for the "very famous", defined as either:

- people whose fame lasts exceptionally long: 90th and 99th percentiles of fame duration distributions; or
- exceptionally highly-discussed people: using distributions among just the top 1000 people or the top 0.1% of people by number of mentions within each year.

In the case of taking the 1000 most-often-mentioned names in each year, the increasing could be explained as follows: as the corpus increases in volume toward later years, a larger number of names appear, representing more draws from the same underlying distribution of fame durations. The quantiles of the distribution of duration for the top 1000 elements will therefore grow over time as the corpus volume increases. On the other hand, our experiments that took the top 0.1% most-often-mentioned names, or the top quantiles of duration, still showed in increasing trend. We therefore conclude that the increasing trend is not completely caused by an increase in corpus volume.

To summarize, we find that the most famous figures in today's news stay in the limelight for longer than their counterparts did in the past. At the same time, however, the average newsworthy person remains in the limelight for essentially the same amount of time today as in the past.

Chapter 2

Related Work

2.1 Group Chats on Twitter (Parts I and II)

2.1.1 Twitter group chats.

In the published literature, we found very little discussion of Twitter group chats. There are articles discussing the benefits of a single education group chat called `#edchat` [18, 30], and Budak and Agrawal [10] investigate characteristics of education group chats that lead to continued individual participation, but we found nothing reporting the number and variety of group chats overall.

Our aim in this work is to develop an algorithm for automatically discovering group chats. While we did find a crowdsourced spreadsheet [64] of group chats, we found many chats listed in the spreadsheet that are now defunct, many chats that are missing (possibly because the moderator of the group chat was not aware of the spreadsheet), some that do not have predictable meetings, and others that are not truly group conversations, e.g, one-time chats.

2.1.2 Definition of groups.

Many different definitions of groups have been proposed in the sociology literature. For example, a group is: "a collection of individuals who have relations to one another" [12]; "a bounded set of patterned relations among members" [2]; or "two or more individuals who are connected by and within social relationships" [28]. In the context of online groups, *virtual communities* have been defined as "social aggregations that emerge from the Net when enough people carry on public discussions long enough, with sufficient human feeling to form webs of personal relationships in cyberspace" [58]. An overview of more definitions of online communities that have arisen in various disciplines is presented in the survey by Iriberri and Leroy [33]. A common theme among the definitions is that groups are driven by, and exist because of, their members. We provide a definition of a group which is geared toward deciding when a set of people and their interactions constitute a group. Whereas

the previous definitions are qualitative, our definition is quantitative and is useful for finding groups whose existence is not known of in advance.

2.1.3 Nature and formation of groups.

There is extensive literature on the formation of groups, the nature and purpose of groups that exist, and the causes of their success or failure. Forsyth [21] provides a comprehensive treatment of group dynamics. We have already mentioned the work of Iriberri and Leroy [33]. Besides its main focus on the life-cycles of online communities and factors that contribute to their success, their work also discusses the "importance and benefits" of online communities, and the types of communities that exist, both of which we study in the more narrow context of Twitter. Backstrom et al [3] study group formation using LiveJournal and DBLP data. Backstrom et al [4] examine the nature of the communities that exist in Yahoo! groups, and explore factors influencing whether a user will stay with a group. Bateman et al. [6] also study the factors that motivate participants to stay. An early paper by McGrath and Kravitz [49] gives a survey of group research in psychology from 1950 to 1982. Of particular relevance to our work is the discussion of communication or "patterning of interaction" in group settings.

Surveys of the public studying how many people participate in online groups and for what purpose are instructive: for example, a 2001 study found that 84% of Internet users participated in online groups, and a survey conducted in 2010 found that 23% of Internet users living with a chronic ailment have looked for support online, and people with rare conditions are even more likely to do so [22, 31, 36]. Much work has also been done studying why animals form groups and of what size; for example, see [51]. In contrast to this line of work, our focus is on algorithmically identifying groups in Twitter. Investigating the nature and formation of Twitter groups is a promising direction for future work.

2.1.4 Participation in Groups

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 [11, 38], while having too few inhibits community responsiveness [48]. The level of moderation also plays a role [57]. The more friends a user has in a group, the more likely they are to join [43], and this likelihood increases if their friends are in turn connected [3]. First impressions are important [37, 4, 39, 45]. 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 [4, 39]. A first response is also known to increase the speed at which a second message is posted [45]. Linguistic complexity reduces the chance of a response [67], and linguistic discrepancy can signal a user's departure from a group [17]. Brandtzæg and Heim study the causes of group attrition in the context of online communities [8]. In Part II, we solve a different but related problem: connect a user to a group that was previously unknown to them. We seek to rank the best groups for discussing a

particular topic. The only information we use is a search query, akin to web search. Richer contextual clues (friends in the group, linguistic coherence, etc.) could lead to better and more personalized rankings.

2.1.5 Determining periodicity.

Our algorithm for detecting group chats begins by determining which hashtags have regular meetings. We considered an algorithm by Kleinberg [41] for detecting bursts of activity. Many different approaches have been proposed for periodicity detection in time-series data: for example, using Fourier analysis [65] and Wavelet transforms [53]. In the end, we adapted the *autoperiod* method proposed in [65] as described in Section 4.1.

2.1.6 Other applications of Fourier analysis to group dynamics.

Gottman [26] applied Fourier analysis to conversations, but whereas we seek to understand the timing of the meetings themselves, Gottman looked for cyclic structure within individual sessions. Gottman also examined cross-correlations between time series to understand interaction between participants in a conversation. He applied this work to the study of marital conflict, as referred to by later work with Krokoff [27]. Dabbs [16] used Fourier analysis to understand the rhythm of conversations, and was able to distinguish "low cognitive load" conversations, where people were asked to talk about themselves, from "high cognitive load" conversations, where people were asked to discuss current events.

2.1.7 Ranking Models

While we are not aware of previous work on ranking groups, much work has been devoted to other ranking problems.

Given a search query, our group preference model describes a user (the *seeker*) who starts with 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 [9] 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 node (groups and authoritative participants). Also, the transition probabilities depend on the query, and are computed by measuring social interactions instead of links between documents. In both models, the stationary distribution of the walk is used to rank nodes. Also, in both models, with some probability a user teleports to a completely random node. Our model is more similar to personalized PageRank [34] 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 [14] are useful in our work (Section 10.1).

The group preference model is also related to HITS [42] 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 will spend more time on participants who have authority in highly ranked discussion groups, and groups that are preferred by highly-ranked participants. One important difference is that the HITS algorithm computes each new hub or authority score as a sum of neighbouring values, whereas our model, since it follows a random walk, averages the values. Averaging has the advantage that a discussion group marginally related to a query with a very large community of participants 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 (Section 10.2). 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.

The random shopper model [32] 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 product, 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 how important the group is for the participants. The random choice of which 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 [63] depending on which other products are shown.

2.1.8 Recommending Hashtags and Experts

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 [44]. In other methods, the users who tweet about the subject may be used to find a relevant hashtag [25]. Note that arbitrary hashtags may never meet again. Indeed, prior work shows that 86% of hashtags have been used less than five times [72]. 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 [71]. Many interesting characteristics of a chat 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. Expert finding is another problem that is related to our work. Different issues arise in expert finding since it is also important to find an expert willing to answer your question [20]. Our method of finding authoritative users draws upon insights from prior work, e.g, the importance of network as well as textual signals [20, 35, 40, 56].

2.2 Fame in News (Part III)

Michel et al. [50] study a massive corpus of digitized books in an attempt to study cultural trends. In particular, they study the rate at which famous names appear in books over time. Where our study focuses on periods of fame on the scale of news coverage — usually on the scale of weeks — their work measures long-term fame in years and decades. The corpus they study is even larger than ours in volume and in temporal extension.

Leetaru [46] presents evidence that sentiment analysis of news articles from the past decade could have been used to predict the revolutions in Tunisia, Egypt and Libya.

Our spike method for identifying periods of fame is motivated in part by the work of Yang and Leskovec [70] on identifying patterns of temporal variation on the web. Szabo and Huberman [62] also consider temporal patterns, in their case regarding consumption of particular content items. Kleinberg studies other approaches to identification of bursts [41].

Numerous works have studied the propagation of topics through online media. Leskovec et al. [47] develop techniques for tracking short "memes" as they propagate through online media, as a means to understanding the news cycle. Adar and Adamic [1], and Gruhl et al. [29] consider propagation of information across blogs.

Finally, a range of tools and systems provide access to personalized news information; see Gabrilovich et al [23] and the references therein for pointers.

Part I

Group Chats on Twitter

In which we study the phenomenon of group chats on Twitter; develop an algorithm to find group chats with some mathematical justification; run the algorithm on a Twitter dataset; and study the set of group chats found.

This part includes material from *Group Chats on Twitter*, co-authored with Nina Mishra and Krishnaram Kenthapadi, and appearing in the proceedings of the 2013 International Word Wide Web Conference (ACM). Work done while I was an intern at Microsoft Research.

Chapter 3

Preliminaries

In this section, we work towards a definition of a group chat. We are not aware of any quantitative definitions of a group in the sociology literature. Instead, there are many qualitative definitions with no convergence towards a single definition [21]. We begin by describing three key properties of a group chat. Group chats are:

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: The members of a group communicate with each other over the course of many meetings.

These properties can be instantiated in many ways. In this paper, we interpret them in one way that leads to our definition of a group chat.

Definition 3.1. *A set of people form a* group chat *if every τ days some subset of them meet for a duration of at most l hours where at least a ν fraction of pairs of people exchange messages during each meeting. l and ν are parameters of the definition, and τ may be different for different groups.*

Observe that this definition captures all of the key properties that we outlined above. Meetings are regular because they occur every τ days. They are synchronized because they last l hours and they are cohesive because at least a ν fraction of pairs communicate.

To motivate each component of the definition in the specific context of Twitter, we describe a few crucial examples that steered our thinking. We started with every hashtag on Twitter as a candidate group, whose members are all users who have tweeted with the hashtag. The first component of our definition requires a group chat to have periodic meetings. The justification for this property is that in order for group connections to form, members must meet predictably over the course of many meetings. (See Bateman et al. [5] for a study of the reasons people repeatedly return to groups.) We observed this property in the

group chat hashtags that we discovered such as Movie Talk on Sunday, which meets every Sunday (right side of Figure 3.1). Offline meetings of local support groups also often occur on a weekly or monthly basis. Note that in order to satisfy Property 1, meetings need not necessarily occur every τ days, but periodic meetings are the focus of the present work.

Requiring periodic meetings is not enough: there are hashtags such as `#monday` that surge once a week but are not group conversations (left side of Figure 3.1). Users simply append `#monday` to their tweets on Mondays. A key difference between such hashtags and group conversations is that `#monday` surges all day Monday, while group meetings surge for only a short time (Figure 3.2). This is how we arrived at the second component of our definition: the group must meet for at most some maximum duration l. Offline meetings also follow such a time-bounded pattern.

Even these two requirements are not enough, since there are hashtags tied to TV shows, such as `#dwts` for Dancing with the Stars, which surge for one hour every week when the TV show airs. Users are not communicating with each other, but are tweeting with the goal of seeing their message broadcast on live television. This motivates the third aspect of our definition: cohesion. The difference between Dancing with the Stars and a real group conversation is that users communicate with each other during a group conversation. In the context of Twitter, we observe this behaviour via `@`-mentions between group members.

We note that our definition of a group could be improved in a number of ways. For example, there may be groups that meet regularly but are not periodic: for example, they might schedule each meeting at the end of the previous meeting. We miss such groups since they do not have a periodic structure. Other groups may meet, but not use a hashtag. Again, our work misses such groups. We leave the question of alternate group definitions as a subject for future work.

3.1 Notation

Let H denote the set of distinct hashtags contained in the text of all tweets. For a hashtag $h \in H$, let t_h denote the *timeline* for h: the multiset consisting of the time of every tweet that contains h. Denote by $\alpha_h = |t_h|$ the total number of tweets containing h. Denote the period of a periodic hashtag h by τ_h. Denote the number of meetings associated with a periodic and synchronized hashtag h by m_h. We omit the subscript when the hashtag is clear from the context. Denote the maximum allowed duration of a meeting by l. While our definition of a group refers to the period in days and the meeting duration in hours, we henceforth assume that τ and l are in the same units.

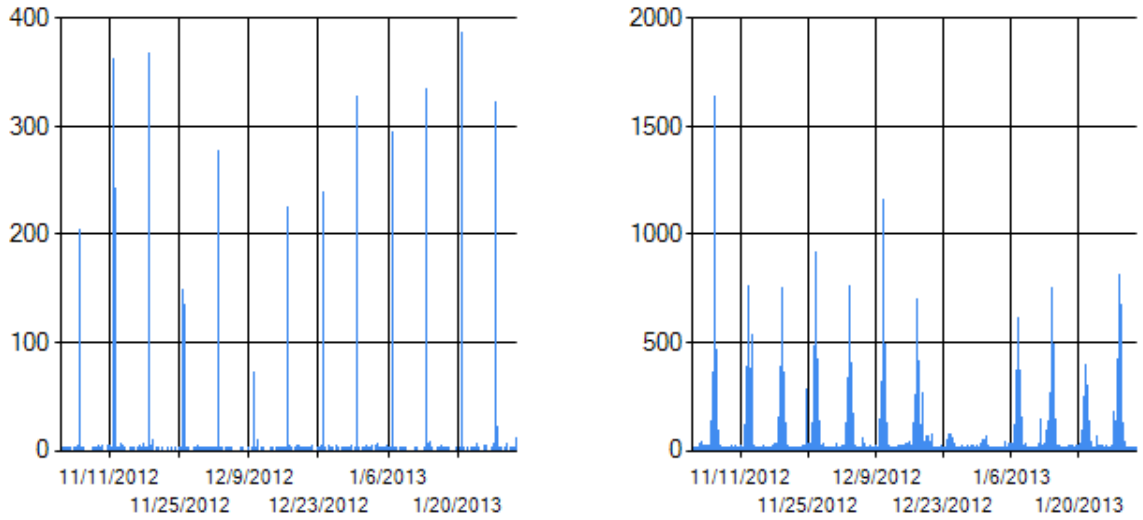

Figure 3.1: Number of tweets per hour for the hashtags `#mtos` (left) and `#monday` (right) over a three month timeframe (November 2011 to January 2013). Both hashtags are periodic, but `#monday` is not a group conversation.

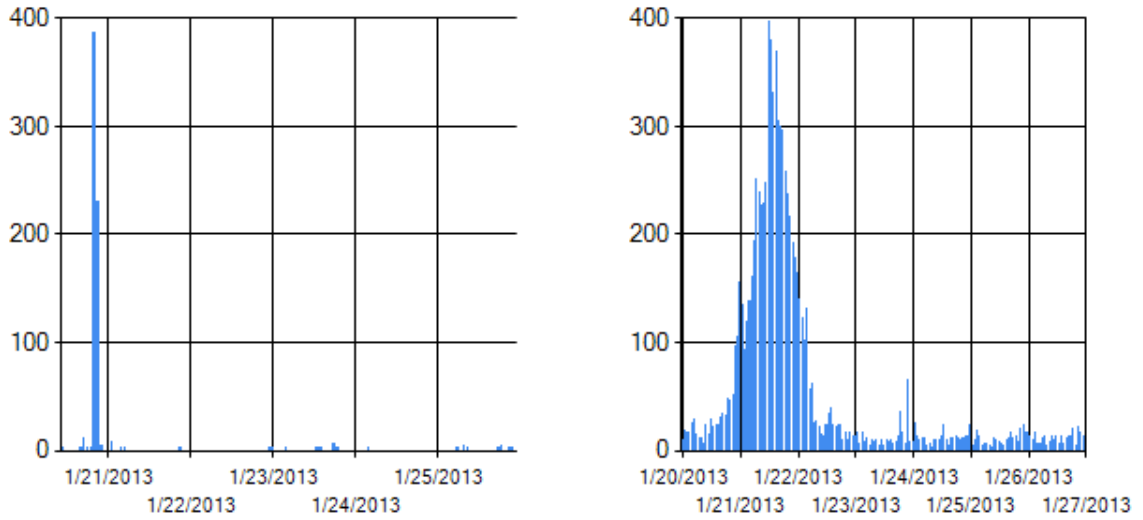

Figure 3.2: Number of tweets per hour for the hashtags `#mtos` (left) and `#monday` (right) over a one week timeframe (last week of January, 2013). `#monday` is active all day, while `#mtos` is active for one hour.

Chapter 4

Finding Group Chats on Twitter

Our method for determining the set of group chats on Twitter is outlined in Algorithm 1. We first identify the set of periodic hashtags on Twitter. Next, for each periodic hashtag, we check whether meetings using the hashtag occur at a predefined time during each period. If such meetings exist, we compute the meeting time, and otherwise, we exclude the hashtag from further consideration. We then check if there is sufficient communication among the top users of the hashtag, and if not, exclude the hashtag. Finally, we eliminate hashtags with a very small number of meetings. The remaining hashtags are considered to be group chats.

4.1 Regular

The main technical component of our approach is to determine whether a hashtag is periodic (Property 1). We first motivate the design of the algorithm through an example.

The left side of Figure 3.1 shows part of the timeline for the hashtag `#mtos`. The group corresponding to this hashtag meets at a predefined time every week. We observe that the hashtag is mentioned frequently during the weekly meetings and infrequently between sessions, resulting in a timeline that is visually periodic. However, the number of occurrences of the hashtag varies widely across meetings. Further, some meetings may not occur (for example, around Christmas) and there may be some occurrences of the hashtag between meetings, (for example, generating awareness about upcoming meetings). Our algorithm for detecting periodicity begins by taking a Fourier transform, which is robust to these factors.

Given a timeline t_h and a frequency ξ, the Fourier transform produces a *Fourier coefficient* $\mathcal{F}(t_h)(\xi) \in \mathbf{C}$. The Fourier transform satisfies the following property: *If t_h is periodic over a large interval with period $\tau = 1/\xi$, then the magnitude of the corresponding Fourier coefficient, $|\mathcal{F}(t_h)(\xi)|$, is large.* This property is robust to variations in the intensity of f from cycle to cycle and addition of a small amount of noise. Hence, a first attempt would be to check whether the largest Fourier coefficient is comparable in magnitude to the total

Algorithm 1 Find Twitter group chats.

Parameters: Periodicity threshold $\delta < \frac{1}{2}$ (§4.1); maximum meeting duration l and synchronization threshold γ (§4.3); number of top users k and minimum density ψ for cohesion (§4.4); minimum number of meetings μ.

Input: The author and timestamp of every tweet in a certain range of time, together the set of hashtags and @-mentioned users in each of those tweets.

Output: The set of hashtags that are group chats, and the meeting time and time between meetings for each one.

1: Identify the set H of distinct hashtags contained in text of all tweets.
2: **for** every hashtag $h \in H$ **do**
3: Determine the timeline t_h, consisting of the timestamps of all tweets containing h.
4: (REGULAR) Determine whether t_h is a periodic timeline, and if so, its period τ. If t_h is not periodic, stop processing h. (§4.1)
5: Using the determined period τ, determine whether at least μ meetings have occurred, and if not, stop processing h.
6: (SYNCHRONIZED) Check whether meetings in the timeline t_h occur at a predefined time during each period. If yes, compute the meeting time, and otherwise, stop processing h. (§4.3)
7: (COHESIVE) Determine if there is sufficient communication among the top users of hashtag h. If not, stop processing h. (§4.4)
8: **return** the set of hashtags that passed all four tests, along with the meeting time and time between meetings (period) for each one.

Algorithm 2 Find periodic hashtags.

Parameter: Periodicity threshold $\delta < \frac{1}{2}$.

Input: A timeline (multiset of timestamps) t_h.

Output: Whether the hashtag is periodic, and if so, the period τ.

1: Compute the Fourier coefficients $\mathcal{F}(t_h)(\xi)$ for a large set of frequencies $\{\xi_j\}$. (§4.1.1)
2: Compute the autocorrelations $\tilde{A}(t_h)(\tau)$ for a large set of periods $\{\tau_k\}$. (§4.1.2)
3: Define the periodicity score for each period T_k as $S(T_k) := \frac{|\mathcal{F}(t_h)(\xi_k^*)|}{|\mathcal{F}(t_h)(0)|} \cdot \frac{|\tilde{A}(t_h)(\tau_k^*)|}{|\tilde{A}(t_h)(0)|}$, for $1 \le k \le r$. Here, ξ_k^* is the closest computed frequency to $1/T_k$ and τ_k^* is the closest computed period to T_k. (§4.1.3)
4: Determine the candidate period T_h with the largest periodicity score, that is, $T_h := \arg\max_{1 \le k \le r} S(T_k)$, and output $\tau = T_h$ if $S(T_h) \ge \delta$. (§4.1.3)

number of tweets containing a hashtag and if so, declare the hashtag to be periodic with the corresponding period.

However, the converse of the above property is not true: in particular, $|\mathcal{F}(t_n)(\xi')|$ may also be large when $\xi' = k\xi$ is an integer multiple of ξ. For example, a group chat that occurs at noon every alternate Monday would have a strong Fourier coefficient at the once-per-two-week frequency, but also at a frequency of once per week and even once per day. To distinguish the base frequency ξ from multiples $k\xi$, we measure the *autocorrelation* of a hashtag's timeline. Given a function f, the autocorrelation $A(t_h)(\tau)$ is a measure of the similarity between t_h and the same timeline t_h shifted by τ.

The autocorrelation satisfies a property similar to that of the Fourier transform: *if t_h is periodic over a large interval with period τ, then $A(t_h)(\tau)$ is large.* Intuitively, a periodic function shifted by its period (or integer multiples of period) aligns well with the original function, resulting in a large autocorrelation comparable to the autocorrelation at $t = 0$ (the function with itself). In other words, the autocorrelation is large for integer multiples $k\tau$ of the base period, rather than periods $1/(k\xi) = \tau/k$ associated with integer multiples of the frequency ξ as is the case with the Fourier transform. The only periods that have both large Fourier coefficients and autocorrelations should be close to the true period τ.

Algorithm 2 formalizes the above intuition. Given the timeline of a hashtag, we compute its Fourier transform for a large set of frequencies as well as autocorrelation for a large set of periods. The periodicity score for each candidate period is computed as the product of two ratios: the ratio of the corresponding Fourier coefficient to the total number of tweets and the ratio of the autocorrelation for this period to the autocorrelation of the function with itself. The algorithm checks if the largest periodicity score exceeds a given threshold, and if so, outputs the period that achieves that score. See Sections 4.1.1-4.1.3 for details.

The idea of combining the Fourier transform with autocorrelation was explored by Vlachos et al. [65] in their work on detecting periodicity. Our method differs in two respects. First, while they use the discrete Fourier transform (DFT), we obtain samples from the continuous Fourier transform. At the cost of requiring more computation, this choice allows us to measure Fourier coefficients for all frequencies of interest to us, for the whole data set at once. The DFT only produces frequencies which are multiples of the inverse total window length: for example, when examining six weeks of data, the DFT could measure frequencies of once every 0.75 weeks, once every 1.5 weeks or once every 3 weeks, but not once per week or once per two weeks. Second, they distinguish between 'hills' and 'valleys' of the autocorrelation, likely to compensate for the lack of precision in their choice of Fourier coefficients. In our implementation, we simply multiply Fourier coefficients and autocorrelations to produce a score for each candidate period.

To detect periodic hashtags, we also tried using Kleinberg's burst detection algorithm [41], which detects periods of high activity using a generative model that switches between a low-activity state and a high-activity "bursty" state. However, we chose the Fourier analysis based method instead, for two reasons. First, Kleinberg's model has a parameter that determines how easily the underlying model switches to a bursty state, and we had trouble finding a value which worked for all group chats. Second, Kleinberg's method does not detect

whether or not the bursts are of a periodic nature, nor does it produce the period. Both of these are natural outputs of the Fourier analysis based method. Naïve attempts to measure the period as an average time between bursts produced by Kleinberg's method are thwarted by missing meetings or short bursts that occur between meetings, although it is possible that some adaptation of Kleinberg's algorithm could overcome these limitations.

4.1.1 The Fourier Transform

Given a function $f : \mathbf{R} \to \mathbf{C}$, the *Fourier transform* $\mathcal{F}(f) : \mathbf{R} \to \mathbf{C}$ applied to a frequency $\xi \in \mathbf{R}$ is defined as:

$$\mathcal{F}(f)(\xi) = \int_{-\infty}^{\infty} f(t) e^{-2\pi i \xi t} \mathrm{d}t. \tag{4.1}$$

$t_h = \{t_{h,1}, \ldots, t_{h,\alpha_h}\}$ is a multiset of timestamps rather than a function, but we can consider it to be a measure with a point mass for each timestamp. The Fourier transform then becomes:

$$\mathcal{F}(t_h)(\xi) = \sum_{j=1}^{\alpha_h} e^{-2\pi i \xi t_{h,j}}. \tag{4.2}$$

The largest-magnitude Fourier coefficient is achieved at $\xi = 0$ and equals the number of tweets in t_h: that is, $\forall \xi,\ |\mathcal{F}(t_h)(\xi)| \leq \mathcal{F}(t_h)(0) = \alpha_h$.

4.1.2 Autocorrelation

The *autocorrelation* of a function $f : \mathbf{R} \to \mathbf{C}$ with respect to a period τ is defined as:

$$A(f)(\tau) = \int_{-\infty}^{\infty} f(t+\tau)\overline{f}(t) \mathrm{d}t.$$

The magnitude of the autocorrelation $|A_f(\tau)|$ is always highest at $\tau = 0$. The autocorrelation can also be expressed in terms of the Fourier transform as:

$$A(f)(\tau) = \int_{-\infty}^{\infty} |\mathcal{F}(f)(\xi)|^2 e^{-2\pi i \xi \tau} \mathrm{d}\xi. \tag{4.3}$$

In the same spirit, we can define the autocorrelation of a timeline or multiset $t_{h,1}, \ldots, t_{h,\alpha_h}$ as the number of pairs of tweets that are separated by a duration of τ:

$$A(t_h)(\tau) = |\{(i,j) | t_{h,j} - t_{h,i} = \tau\}|. \tag{4.4}$$

Unfortunately, this leads to a problem. If the resolution of timestamps is sufficiently granular (say, seconds), then it is very likely that the f_h shifted by its period will not align with itself: for example, if a tweet is emitted three seconds after 10am in the first meeting, but the closest tweets in the next meeting are at 9:59:55 and 10:00:11, then the 10:00:03 tweet will contribute nothing to the autocorrelation.

To address this, we appeal to (4.3) and approximate the autocorrelation using a set of computed Fourier coefficients $\mathcal{F}(t_h)(\xi_j)$, $j = 1, \ldots, r$:

$$\tilde{A}(t_h)(\tau_k) = \sum_{j=1}^{r} |\mathcal{F}(t_h)(\xi_j)|^2 e^{-2\pi i \xi_j \tau_k}.$$

If we bound the sampled frequencies ξ_j so that the maximum frequency is at most two cycles per day, then the approximate autocorrelations $\tilde{A}(t_h)(\cdot)$, loosely speaking, can only see the approximate time at which a tweet happened, to within half a day or so. Computing the autocorrelation in terms of the Fourier transform may also be faster, since the runtime depends on the number of Fourier coefficients and not on the number of tweets.

4.1.3 The Periodicity Score

Given the timeline t_h for a hashtag, we first compute the Fourier coefficients $\mathcal{F}(t_h)(\xi_j)$ for N_F equally spaced frequencies ξ_j in a fixed range $[-1/\tau_F, 1/\tau_F]$ using (4.2). (N_F stands for Number of Fourier coefficients. In our implementation, τ_F was twelve hours.) Then, we use the computed Fourier coefficients to compute the approximate autocorrelation $\tilde{A}(t_h)(\tau_k)$ for a large but fixed set of periods τ_k.

Now, define the *periodicity score* for a candidate period T as:

$$S(\tau) := \frac{|\mathcal{F}(t_h)(\xi^*)|}{|\mathcal{F}(t_h)(0)|} \cdot \frac{|\tilde{A}(t_h)(\tau^*)|}{|\tilde{A}(t_h)(0)|},$$

where ξ^* and τ^* are the closest available frequency and period to $1/T$ and T in the sets $\{\xi_j\}$ and $\{\tau_k\}$, respectively. Note that $S(\tau)$ is always between 0 and 1. Then, we determine the period τ with the largest periodicity score out of a set of candidate periods. If $S(\tau)$ exceeds the periodicity threshold δ, the algorithm returns τ as the period, and otherwise the algorithm declares the timeline not to be periodic.

4.1.4 Incremental Updates

As more tweets arrive, it is possible to recompute the periodicity scores and the estimated period of a hashtag in time proportional to the number of new tweets. Suppose $t_h^{\text{all}} = t_h^{\text{old}} \cup t_h^{\text{new}}$, where t_h^{old} includes all the tweets up to time $\mathcal{T}$, and $\cup t_h^{\text{new}}$ includes only tweets after time $\mathcal{T}$. If we have already computed a Fourier coefficients $\mathcal{F}(t_h^{\text{old}})(\xi)$, the updated Fourier coefficient $\mathcal{F}(t_h^{\text{all}})(\xi)$ can be computed as:

$$\begin{aligned}\mathcal{F}(t_h^{\text{all}})(\xi) = \sum_{t \in t_h^{\text{all}}} e^{-2\pi i \xi t} &= \sum_{t \in t_h^{\text{old}}} e^{-2\pi i \xi t} + \sum_{t \in t_h^{\text{new}}} e^{-2\pi i \xi t} \\ &= \mathcal{F}(t_h^{\text{old}})(\xi) + \sum_{t \in t_h^{\text{new}}} e^{-2\pi i \xi t}.\end{aligned}$$

The remaining steps of recomputing the autocorrelation and finding the highest-scoring period have a running time that depends on the number of candidate periods (r) but not the total number of tweets.

4.2 Minimum Number of Meetings

The final step of our algorithm is to remove from consideration hashtags with a very small number of meetings. The minimum number of meetings μ is a parameter of the algorithm. This helps to ensure that the periodic behaviour of the hashtag is intentional. For example, any hashtag with two short bursts of cohesive activity would be considered by the other steps of the algorithm to be a group chat, with period equal to the time between the two bursts. A chat that did not intend to meet at a regular time might still chance to have three meetings that are exactly nine days apart.

4.3 Synchronized

Algorithm 3 Detect synchronized meetings.

Parameters: Maximum meeting duration l; synchronization threshold γ.
Input: A timeline (multiset of timestamps) t_h and its period τ.
Output: Whether h has synchronized meetings, and if so, the Meeting start time $\tilde{t}$.

1: Given a potential meeting offset t, we evaluate the hypothesis that each meeting starts t after the start of its period: so every meeting lies in $[t, t+l] \cup [\tau+t, \tau+t+l] \cup [2\tau+t, 2\tau+t+l], \ldots$. Define the score $\beta(t)$ as the fraction of tweets that lie in this set:
2: $\beta(t) := \frac{1}{\alpha}|\{s \in t_h : s \in [j\tau+t, j\tau+t+l] \text{ for some integer } j\}|$.
3: Determine the candidate meeting start time $\tilde{t}$ with the largest score: that is, $\tilde{t} := \arg\max_{t\in[0,\tau)} \beta(t)$. If $\beta(\tilde{t})$ exceeds the threshold γ, then output $\tilde{t}$.

As described in Chapter 3, hashtags like `#monday` have a periodic timeline, but do not have an agreed-upon time of day when users meet to have a conversation. Algorithm 3 eliminates these hashtags from consideration by noticing that there is no short window of time each week within which a substantial fraction of the tweets are emitted. The underlying intuition is that most people simply do not have the time to participate and listen to others for very long, so synchronized meetings cannot last for more than a few hours.

Given the timeline t_h and period τ of a periodic hashtag, Algorithm 3 looks for a window with duration l that repeats every period, such that at least a γ fraction of the tweets lie in that window. Note that this requires the period τ to exactly match the true period of the hashtag: if one meeting falls in the window $[t, t+l]$, Algorithm 3 expects every other meeting will fall within $[j\tau+t, j\tau+t+l]$ for some integer multiple $j\tau$ of the period. Unfortunately, the period computed by Algorithm 2 may deviate slightly from the true period. We make use

of the fact that group chats on Twitter have routine meetings whose period is in multiple of days, and address this issue by rounding the period computed by Algorithm 2 to the nearest day.

4.4 Cohesive

In the final step, our algorithm excludes hashtags that are not cohesive. To measure this quantity, we estimate the communication among the k most active members who use the hashtag h during regular, synchronized meetings. Our intuition is that members of a healthy group will look forward to communicating with one another during each meeting. Let V be the set of k users who participated in the most meetings of the hashtag. For $1 \leq i \leq m$, let E_i capture the directed communication edges among users in V during the i^{th} meeting. For example, if user $u \in V$ @-mentions two other top users v and w during the i^{th} meeting, we include the two directed edges (u, v) and (u, w) in E_i. We define the cohesion score of a hashtag by the average number of edges across all meetings: that is, $\text{cohesion}(h) := \frac{1}{m}\sum_{i=1}^{m} |E_i|$, where m is the number of meetings for hashtag h. The larger this average interaction, the more cohesive the group. We exclude hashtags with cohesion score less than a given threshold ψ.

Chapter 5

Theoretical Analysis of Algorithm 1

In this section, we prove that group chats that are sufficiently "well-behaved" are accepted by our algorithm and hashtags that are far from being group chats are rejected. Specifically, if a group meets sufficiently many times, if group meetings are well-separated, and if a reasonable number of messages are exchanged per meeting, then we prove that our algorithm will accept the hashtag, under certain assumptions (§5.1). On the flip side, we prove that our algorithm will reject hashtags that are not cohesive, or are generated from models of not-regular or not-synchronized hashtags (§5.2).

5.1 Algorithm 1 accepts group chats

We next define the notion of a *well-behaved group chat* and show that our algorithm will accept a well-behaved group chat under certain assumptions. The following definition is motivated by our qualitative observations of a group chats: see for example the timeline of `#mtos` in Figure 3.1.

Definition 5.1. *A set of tweets forms a* well-behaved group chat *if all of the following are true.*

- *There are m meetings of duration l separated by a period τ, for some m, l and τ. (The jth meeting interval is $[j\tau, j\tau + l]$.)*
- *During each of the meeting intervals, at least ψ pairs of the top k group members exchange messages*[1]*.*
- *Tweets are sent at a higher rate within meetings than outside of meetings. That is: at least $\frac{\gamma}{1-\gamma} n_-$ tweets are sent during every meeting, where $\gamma \geq l/\tau$ is the threshold of Algorithm 3 and n_- is the average number of tweets sent between two adjacent meetings. No tweets are sent before the first meeting or after the last. We denote by*

[1]Here, ψ and k are the parameters of the cohesion algorithm.

n_j the number of tweets in the j-th meeting, and by $n_{\min}$ and $n_{\max}$ the smallest and largest values n_j.

Theorem 5.2 (Completeness of Algorithm 1)**.** *Consider any well-behaved group chat that also satisfies the following properties:*

- *(Technical conditions.) There are at least* $\max\{3, \mu\}$ *meetings, where μ is the minimum number of meetings parameter of Algorithm 1. The period τ is not shorter or longer than the range of periods considered by Algorithm 2 (periodicity). The duration l is shorter than half the shortest period considered by Algorithm 2, and also no longer than the duration l of Algorithm 3 (synchronization).*

- *(Quantifying well-behavedness.) The following inequality holds, where $\eta = n_{\max}/n_{\min}$, $\rho = n_-/n_{\min}$ and $\delta < \frac{1}{2}$ is the threshold used by Algorithm 2.*

$$\begin{aligned} &\frac{1}{2\eta^2}\left(1 - \frac{2\pi l}{\tau} - 2\rho\right) \\ &> \max\left\{\delta, 3\rho, \frac{2\pi l}{\tau} + \frac{\eta^2}{4}\left(1 + 6\frac{\eta + \rho - 1}{\rho + 1}\right)\right\} \end{aligned} \tag{5.1}$$

- *(*) The timeline behaves like a step function, in the following way: each tweet happens at the beginning of a second; the number of tweets at each second of the j-th meeting is $\lambda_j = n_j/l$; and the number of tweets at any second which is between meetings is exactly $\lambda_- = n_-/(\tau - l)$.*

Consider a modified (**) version of Algorithm 2 which computes autocorrelations exactly (using Equation (4.4), §4.1.2). Call this Algorithm 2', and the resulting group chats algorithm Algorithm 1'. Then, Algorithm 1' will accept this chat as a group chat. It will also return the correct meeting start time to within l, and will report the correct period τ with error of at most $l + \epsilon$, where ϵ is the largest difference between two adjacent periods considered by either the Fourier or autocorrelation parts of Algorithm 2.

Note 1. *The parts marked (*) and (**) are added to simplify the proof. We believe a version could be proved which does not have the step function condition (*) and applies to the true Algorithm 1 (**). (§5.3.1 contains the only results that depend on (*) and (**).) Although modifying the algorithm to compute autocorrelation exactly makes the proof simpler, it also forces us to impose a strong condition on the timeline of tweet rates for the given hashtag. As noted in Section 4.1.3, the timeline of a typical chat will have an exact autocorrelation which is very close to zero, because the tweets in adjacent meetings will rarely be emitted exactly one period apart. This is why we are forced to assume the timeline follows an unnaturally rigid structure. The approximate autocorrelations computed by Algorithm 2 do not suffer from this problem, because they are computed using Fourier coefficients only for frequencies of at most twice per day, and are therefore insensitive to variations on the order of half a day or less.*

Note 2. *To interpret (5.1), note that that for a very strongly-structured group chat, we can expect η to be close to 1 (meaning all meetings have similar attendance), ρ close to 0 (the group's hashtag is rarely used outside meetings), and l/τ to be quite small (meetings are short compared to the time between). Therefore the left side of the equation will be about $1/2$, and the right side will be about $\max\{\delta, 1/4\}$.*

We defer the proof to Section 5.3.

5.2 Algorithm 1 rejects non-group chats

We next show that, with high probability, our algorithm will reject hashtags that are not cohesive, or are obtained from generative models representing non-synchronized or non-regular hashtags. As our algorithm explicitly rejects non-cohesive hashtags, we focus on the two generative models.

5.2.1 Non-Synchronized Hashtags

We begin by describing a random process that captures the rhythm of a typical non-synchronized hashtag. Our observations suggest that a typical hashtag has a daily cycle where its usage increases during waking hours and declines during sleeping hours. Such a hashtag has no short (3 hour) time window where a large fraction of weekly messages are exchanged — for example, see the charts of average Twitter activity compiled by Pingdom [54]. We call the timeline of such a hashtag *diffuse*:

Definition 5.3 ((γ, l)-diffuse)**.** *We say a probability distribution p over an interval $[a, b]$ is* (γ, l)-diffuse *if there is no length-l sub-interval $I = [t, t + l]$ such that $\Pr_{x\sim p}[x \in I] > \gamma$. A function $\mu : \mathbf{R} \to \mathbf{R}^{\geq 0}$ is (γ, l)-diffuse over an interval $[a, b]$ if the normalized distribution $\mu / \left(\int_a^b \mu(t)\mathrm{d}t\right)$ is. Similarly, we say a timeline $t_1, \ldots, t_m \in [a, b]$ is (γ, l)-diffuse if at most γm messages lie in any length-l subinterval $[t, t + l]$.*

For example, when the period τ is longer than the time between the first and last tweets, Algorithm 3 accepts exactly those timelines that are not (γ, l)-diffuse.

Definition 5.4. *In the* non-synchronized model*, messages are generated according to a Poisson process with a varying rate $\lambda(t)$. We assume that $\lambda(t)$ is periodic over an interval $[0, T]$ with a period of one day, and is (γ^*, l^*)-diffuse over every 1-day-long subinterval of $[0, T]$, for any $\gamma^* < \gamma/2$ and $l^* > l$.*

5.2.2 Non-Regular Hashtags

The next random process captures non-regular hashtags that do not have periodic meetings but have a fixed rate of meetings on average.

Definition 5.5. *In the* non-regular model*, meeting times $\mu_1, \ldots, \mu_m \in [0, T]$ are sampled according to a Poisson process with a fixed rate λ. Within the j-th meeting, N messages are emitted in the range $[\mu_j, \mu_j + l]$ (the exact times are allowed to be arbitrary and non-random).*

5.2.3 Theorem Statement

We state the theorem below and defer the proof to Section 5.4.

Theorem 5.6 (Soundness of Algorithm 1)**.** *Suppose a set of messages for a hashtag is generated from the non-synchronized model (Definition 5.4), from the non-regular model (Definition 5.5), or does not satisfy cohesion (§4.4). Then Algorithm 1 will reject it with probability approaching 1 as $T \to \infty$, so long as the smallest candidate period $\tau_{\min}$ is at least one half day and the duration l is less than $\gamma\tau_{\min}/3$.*

5.3 Proof of Theorem 5.2 (Completeness of Algorithm 1)

Notice that Definition 5.1 requires immediately that the any well-behaved group chat pass the cohesion test, and the theorem statement requires that the number of meetings is at least the minimum of μ required by Algorithm 1. Further, any well-behaved group chat will pass the test for being synchronized: Algorithm 3 will accept the chat because $\beta(0) \geq mn_{\min}/(mn_{\min} + (m-1)n_{-}) \geq \gamma$. The only real difficulty is in showing that (the modification of) Algorithm 2 will accept the timeline of tweets. Lemma 5.7 completes this last step, and the remainder of this section is devoted to its statement and proof.

Lemma 5.7 (Correctness of Algorithm 2)**.** *Let t_h be a timeline satisfying the hypotheses of Theorem 5.2. Then the periodicity score of the correct period is at least*

$$S(\tau) \geq (2\eta^2)^{-1}(1 - 2\pi l/\tau - 2\rho) \tag{5.2}$$

and for any period $\tau' \geq l$ for which $|\tau' - \tau| \geq l$, the periodicity score is at most

$$S(\tau') \leq \max\left\{3\rho, \frac{2\pi l}{\tau} + \frac{\eta^2}{4}\left(1 + 6\frac{\eta + \rho - 1}{\rho + 1}\right)\right\}. \tag{5.3}$$

Proof. The periodicity score of a period $S(\tau)$ is the product of two terms, which we analyze in Sections 5.3.2 (Fourier) and 5.3.1 (autocorrelation). The lower bound (5.2) on the true period's score follows from Lemmas 5.8 and 5.13. The bound (5.3) on the scores of other periods τ' is proved in cases. For $\tau' \in [l, \tau - l] \cup [\tau + l, 2\tau - l]$, Lemma 5.9 gives $S(\tau') \leq |A(t_h)(\tau')|/|A(t_h)(0)| \leq 3\rho$.

The remaining case is that $\tau' \geq 2\tau - l$. We begin by approximating the term $s = 2|\sin \pi\tau\xi|$ from the statement of Lemma 5.14, where $\xi = 1/\tau'$. By the concavity of the sine function

on $[0, \pi]$, we have $\sin \psi\pi \geq \psi$ for $\psi \in [0, \frac{2}{3}]$. So for $\tau' \geq 2\tau - l \geq 3\tau/2$, we have $s \geq 2\tau/\tau'$. If $\tau' > m\tau$, then by Lemma 5.10 the periodicity score is zero. Otherwise,

$$\left(\frac{m - \tau'/\tau}{m}\right) s^{-1} \leq \left(\frac{m - \tau'/\tau}{m}\right) \frac{\tau'}{2\tau} \leq \frac{m}{8}.$$

since the middle quantity takes its maximum value at $\tau' = (m/2)\tau$. So by Lemmas 5.10 and 5.14,

$$\begin{aligned}\frac{|\mathcal{F}(t_h)(1/\tau')||A(t_h)(\tau')|}{|\mathcal{F}(t_h)(0)||A(t_h)(0)|} \leq & \frac{2\pi l}{\tau} + \eta^2 \left(\frac{m - \tau'/\tau}{m}\right)\left(\frac{2s^{-1}}{m} + \frac{3}{2} \cdot \frac{\eta + \rho - 1}{\rho + 1}\right) \\ \leq & \frac{2\pi l}{\tau} + \eta^2 \left(\frac{1}{4} + \frac{3}{2} \cdot \frac{\eta + \rho - 1}{\rho + 1}\right).\end{aligned}$$

□

5.3.1 Lemmas about Autocorrelation

Here we state results which say that the autocorrelation of the true period is high, and that the autocorrelations of certain other periods are low. We assume t_h satisfies the hypotheses of Theorem 5.2. Recall that Theorem 5.2 assumes the autocorrelation is computed using Equation (4.4) of Section 4.1.2.

Lemma 5.8.

$$\frac{|A(t_h)(\tau)|}{|A(t_h)(0)|} \geq \frac{1}{2\eta^2}.$$

Proof. Note that

$$|A(t_h)(0)| \leq ml\lambda_{\max}^2 + (m-1)(\tau - l)\lambda_-^2$$

and

$$|A(t_h)(\tau)| \geq (m-1)l\lambda_{\min}^2 + (m-2)(\tau - l)\lambda_-^2,$$

and that both $(m-1)l\lambda_{\min}^2/ml\lambda_{\max}^2$ and $(m-2)(\tau - l)\lambda_-^2/(m-2)(\tau - l)\lambda_-^2$ are at most $1/2\eta^2$. □

Lemma 5.9. *Let $\tau' > 0$ and assume that if we shift the timeline t_h by τ', then no meeting in the unshifted t_h will overlap any meeting in the shifted t_h. Then*

$$\frac{|A(t_h)(\tau')|}{|A(t_h)(0)|} \leq \frac{2m\rho}{m-1}.$$

Proof.

$$A(t_h)(\tau') \leq \sum_{j=1}^{m} 2l\lambda_j\lambda_- + (m-1)(\tau - 2l)\lambda_-^2 \leq \frac{\sum_{j=1}^{m} 2n_j n_-(1 + n_-/n_j)}{\tau - l}$$

but

$$A(t_h)(0) = \sum_{j=1}^{m} l\lambda_j^2 + (m-1)(\tau - l)\lambda_-^2 \geq \left(\frac{m-1}{m}\right) \frac{\sum_{j=1}^{m} n_j^2(1 + n_-/n_j)}{\tau - l}. \quad \square$$

Lemma 5.10. *For any $\tau' > 0$, $\frac{|A(t_h)(\tau')|}{|A(t_h)(0)|} \leq \max\left\{0, \eta^2\left(\frac{m-\tau'/\tau}{m}\right)\right\}$.*

Proof. The intuition here is that only a $(m - \tau'/\tau)/m$ fraction of meetings can overlap between the timeline t_h and the same timeline shifted by τ'.

Let $t_h^{\tau'} = t_{h,1} + \tau', \ldots, t_{h,\alpha_h} + \tau'$ be the timeline t_h shifted by τ'. Express τ' as $\tau' = k\tau + \epsilon$, where k is an integer and $-\tau/2 \leq \epsilon \leq \tau/2$.

Note that the first k meetings in t_h do not overlap $t_h^{\tau'}$ at all, and that when a meeting in t_h overlaps a meeting in $t_h^{\tau'}$, they overlap for a duration of $\max\{0, l - |\epsilon|\}$. Let t_- be a timeline with exactly λ_- tweets at the start of each second from 0 to $(m-1)\tau + l$: then we can decompose $t_h = t_- + (t_h - t_-)$, where $t_h - t_-$ has tweets at the rate of $\lambda_j - \lambda_-$ during the j-th meeting and no tweets between meetings. The autocorrelation of t_h is

$$A(t_h)(\tau') = A(t_-)(\tau') + A(t_h - t_-)(\tau') + C(t_h - t_-, t_-)(\tau') + C(t_-, t_h - t_-)(\tau'),$$

where $C(a, b)(\tau')$ is the correlation between a shifted by τ' and b. Then it can be shown separately that each of the four terms is less than or equal to $\max\{0, \eta^2(m - \tau'/\tau)/m\}$, which completes the proof. $\square$

5.3.2 Lemmas about Fourier Coefficients

Here we state analogous results to those in Section 5.3.1, but about Fourier coefficients. We assume t_h satisfies the hypotheses of Theorem 5.2[2].

Given two timelines s_1 and s_2, define their sum $s_1 + s_2$ to be the timeline consisting of all tweets in both timelines. (If the timelines are disjoint sets, then their sum is simply their union. If s_1 has three tweets and s_2 has four tweets all at the exact same time t, then $s_1 + s_2$ has seven tweets at time t.) Note that for any timeline s, $\mathcal{F}(s)(0)$ is the total number of tweets in s.

Proposition 5.11 (Effect of noise)**.** *Suppose a timeline s is the union of two timelines $s = s_+ + s_-$. Then, for any frequency ξ, we have $|\mathcal{F}(s)(\xi) - \mathcal{F}(s_+)(\xi)| \leq |\mathcal{F}(s_-)(\xi)| \leq |\mathcal{F}(s_-)(0)|$.*

Lemma 5.12 (Effect of timing within meetings)**.** *Consider a timeline s consisting of α tweets at times $s_0, \ldots, s_\alpha$, and a distorted version s' with tweets at times $s'_0, \ldots, s'_\alpha$, where $\forall j\, |s_j - s'_j| < l$. Note that $\mathcal{F}(s)(0) = \mathcal{F}(s')(0)$. (Think of s with tweets at starts of meetings, and s' with tweets throughout meetings.) Then, for any frequency ξ, $|\mathcal{F}(s)(\xi) - \mathcal{F}(s')(\xi)| < 2\pi l\xi\alpha$.*

[2]Just within this section, we need not assume the hypothesis (*) that the timeline t_h looks like a step function: the n_j tweets within the j-th meeting may be distributed through the duration-l interval in any way, and the tweets between meetings may be distributed in any fashion as long as there are n_- of them.

Proof. Changing the time of a single tweet from s to s' changes its contribution to the Fourier coefficient $\mathcal{F}(s)(\xi)$ from $e^{-2\pi i s\xi}$ to $e^{-2\pi i s'\xi}$, an absolute difference of at most $2\pi|s-s'|\xi$. The total difference is at most the number of tweets α times this. □

Combining Proposition 5.11 with Lemma 5.12, we have:

Lemma 5.13 (The correct Fourier coefficient)**.**

$$|\mathcal{F}(t_h)(1/\tau)| \geq (1 - 2\pi l/\tau - 2\rho)\alpha$$

where α is the number of tweets.

Proof. Let t_{h+} consist of just the tweets during meetings, and t_{h-} the tweets between meetings. Now, construct a new timeline t'_{h+} as follows. Start with t_{h+}, and move each tweet to the start of its meeting: if a tweet happened at time $j\tau+\epsilon$, where $0 \leq \epsilon \leq l$, move it to time $j\tau$ instead. Then $|\mathcal{F}(t'_{h+})(1/\tau)|$ is the number of tweets that occurred during meetings. Call this number mn_+. By Lemma 5.12, $|\mathcal{F}(t_{h+})(1/\tau)| \geq mn_+ - 2\pi l n_+/\tau = (1-2\pi l/\tau)mn_+$. By Proposition 5.11, $|\mathcal{F}(t_h)(1/\tau)| \geq (1-2\pi l/\tau)mn_+ - mn_-$. Note that $mn_+ \geq \alpha/(1+\rho) \geq \alpha(1-\rho)$ and $mn_- \leq \rho\alpha$. □

Finally, we show:

Lemma 5.14 (The wrong Fourier coefficients)**.** *Let ξ be any frequency, and let $s = 2|\sin \pi\tau\xi|$. Then,*

$$|\mathcal{F}(t_h)(\xi)| < n_{\min}(2\pi l\xi m + 2s^{-1}) + m(n_{\max} - n_{\min}) + (m-1)n_-.$$

In particular, if $m \geq 3$, then

$$\frac{|\mathcal{F}(t_h)(\xi)|}{|\mathcal{F}(t_h)(0)|} < 2\pi l\xi + 2s^{-1}m^{-1} + 3(\eta+\rho-1)/2(\rho+1).$$

Proof. We will replace t_h by a simpler version t_h^{simple}. This version will have no tweets between meetings. Every meeting will have the same number of tweets $n_{\min}$, and all every tweets will happen at the start of its meeting. The Fourier coefficients of t_h^{simple} behave well, and we can relate t_h^{simple} to t_h using Proposition 5.11 and Lemma 5.12.

Like in the proof of Lemma 5.13, let t_{h+} consist of tweets during meetings and t_{h-} the other tweets. Now, change t_{h+} to have $n_{\min}$ tweets in every meeting, by arbitrarily removing tweets from meetings that have more. Call the resulting timeline t^*_{h+}, and let $t^{\text{extra}}_{h+} = t_{h+} - t^*_{h+}$. Notice $|\mathcal{F}(t^{\text{extra}}_{h+})(0)| \leq m(n_{\max} - n_{\min})$. Now, let t_h^{simple} consist of $n_{\min}$ tweets at the start of each of the m meetings. By Lemma 5.12, $|\mathcal{F}(t^*_{h+})(\xi) - \mathcal{F}(t_h^{\text{simple}})(\xi)|$

$< 2\pi l\xi m n_{\min}$. Fourier coefficients of t_h^{simple} are geometric series:

$$|\mathcal{F}(t_h^{\text{simple}})(\xi)| = \left|\sum_{k=1}^{m} n_{\min} e^{-2\pi i(k\tau)\xi}\right| = n_{\min}\left|\sum_{k=1}^{m}(e^{-2\pi i\tau\xi})^k\right|$$
$$= n_{\min}\frac{|(e^{-2\pi i\tau\xi})^{m+1} - 1|}{|e^{-2\pi i\tau\xi} - 1|} \le \frac{2n_{\min}}{|e^{-2\pi i\tau\xi} - 1|} = \frac{2n_{\min}}{s}$$

Then $|\mathcal{F}(t^*_{h+})(\xi)| < n_{\min}(2\pi l\xi m + 2s^{-1})$. By Proposition 5.11, $|\mathcal{F}(t_h)(\xi)| < n_{\min}(2\pi l\xi m + 2s^{-1}) + m(n_{\max} - n_{\min}) + (m-1)n_-$. □

5.4 Proof of Theorem 5.6 (Soundness of Algorithm 1)

Algorithm 1 explicitly rejects hashtags that do not satisfy the coherence property. What remains is to show is that the algorithm rejects with high probability a set of messages generated from the non-synchronized model or from the or non-regular model. The following lemma is key. For a rate function $\lambda(t)$, let $|\lambda|_1 = \int_{-\infty}^{\infty} \lambda(t)\mathrm{d}t$.

Lemma 5.15. *Suppose $\lambda(t)$ is (γ_1, l_1)-diffuse on an interval $[0,T]$, and $\gamma_2 > \gamma_1$ and $l_2 < l_1$. If a set of events $\{t_1, \ldots, t_\alpha\}$ is sampled with a Poisson process of varying rate $\lambda(t)$, then the set is (γ_2, l_2)-diffuse with probability $1 - O(1)2^{-\Omega(|\lambda|_1)}$. (The coefficients in the O and Ω depend on γ_1, γ_2, l_1/T and l_2/T.)*

We will first prove the theorem from the lemma, and then prove the lemma.

5.4.1 Proof of Theorem 5.6 given Lemma 5.15

We wish to show that Algorithm 1 will reject with high probability any chat generated from the non-synchronized or non-regular model. It is enough to show that Algorithm 3, which checks for synchronized meetings, rejects such chats. Once Algorithm 2 has decided on a period $\tau' \ge \tau_{\min}$, we will think of Algorithm 3 as having two parts. First, a hashtag's timeline is *compressed* with period τ': each message time t_i is replaced with a message time $0 \le t'_i < \tau'$ by subtracting an integer multiple of the period τ'. Second, the algorithm checks whether the resulting timeline (t'_i) is (γ, l)-diffuse, and if so, classifies the hashtag as a non-group chat. We begin with a lemma quantifying the effect of this compression step on the diffuseness of a timeline.

Lemma 5.16. *Let $t_1, \ldots, t_\alpha$ be a timeline which is $(\gamma/\lceil T/\tau' \rceil, l)$-diffuse over the interval $[0,T]$. Given a period τ', consider the compressed timeline which replaces $t_i \in [0,T]$ by $t'_i \in [0,\tau')$, so that $t_i - t'_i$ is a multiple of τ'. Then the compressed timeline $t'_1, \ldots, t'_\alpha$ is (γ, l)-diffuse.*

Proof. Given any length-l interval I in $[0, \tau']$, any tweet that lands in I after the compression step must have come from an interval equal to I shifted by an integer multiple of τ'. There are at most $\lceil T/\tau' \rceil$ such intervals which overlap $[0, T]$, and each such interval contains at most $\gamma/\lceil T/\tau' \rceil \alpha$ tweets. □

Non-synchronized model To see that Algorithm 3 rejects chats generated from the non-synchronized model, first notice that the rate function $\lambda(t)$ is $(\gamma^*/\lfloor s \rfloor, l^*)$-diffuse where s is the number of days that the interval $[0, T]$ overlaps. Then apply Lemma 5.15 to see that with high probability, the timeline sampled from $\lambda(t)$ is $(\gamma^{**}/\lfloor s \rfloor, l)$-diffuse, taking γ^{**} to be some value strictly between $\gamma/2$ and γ^*. Then by Lemma 5.16, for large enough T the timeline is (γ, l)-diffuse after the compression step, since the period τ' used by the algorithm is at least one half day. Therefore Algorithm 3 rejects the timeline.

Non-regular model To see that Algorithm 3 rejects chats generated from the non-regular model, notice that if a set of meeting start times is $(\gamma, 2l)$-diffuse, then the set of tweets in the meetings themselves must be (γ, l)-diffuse. This is because any tweet that falls in an interval $[a, a + l]$ must come from a meeting that started in the interval $[a - l, a + l]$. Since the constant function is $((2l + \epsilon)/T, 2l + \epsilon)$-diffuse on $[0, T]$ (take $\epsilon = 0.1$), Lemma 5.15 gives us that with high probability, the meeting start times are $((2l + 2\epsilon)/T, 2l)$-diffuse, and so the tweets themselves are $((2l + 2\epsilon)/T, l)$-diffuse. Since by assumption $\gamma > l/3\tau_{\min}$, Lemma 5.16 gives that for large enough T the compressed timeline is (γ, l)-diffuse, so Algorithm 3 rejects the timeline.

5.4.2 Background for proving Lemma 5.15

Proposition 5.17. *Consider a non-homogeneous Poisson process with rate parameter $\lambda(t)$. Let $I \subseteq [0, T]$ be an interval. Let N_1 be the number of events that occur in I and N_2 the number that land in $[0, T] \setminus I$. Then N_1 and N_2 are independent Poisson-distributed random variables, of rates $\int_I \lambda$ and $\int_{[0,T]\setminus I} \lambda$, respectively.*

Proof. For example, Ross [59] states that a non-homogeneous Poisson process has an independent number of events in disjoint intervals, and that the number of events in an interval $(a, b]$ is a Poisson-distributed random variable with rate $\int_a^b \lambda(t)\mathrm{d}t$ □

Lemma 5.18. *Let N_1 and N_2 be independent Poisson-distributed random variables of rate λ_1 and λ_2, respectively. Let $N = N_1 + N_2$ and $\lambda = \lambda_1 + \lambda_2$. Take any $\epsilon > 0$. Then $\Pr[N_1/N \geq \lambda_1/\lambda + \epsilon] < e^{-\epsilon^2\lambda} + (2/e)^{\lambda/2}$.*

Proof. Consider the following random process. First, sample a random variable N from the Poisson distribution with rate λ. Then, flip a biased coin N times: each time, with probability λ_1/λ, the coin says "type 1", and otherwise (probability λ_2/λ) it says "type 2". Let N_1 be the number of type-1s, and N_2 be the number of type-2s: in other words,

$N_1 \sim \text{Binom}(N, \lambda_1/\lambda)$ and $N_2 = N - N_1$. As argued in [24, Section 2.3.1], the resulting N_1 and N_2 are independent Poisson random variables with rate λ_1 and λ_2, which is exactly what we assumed in the hypothesis of the lemma: so henceforth we will assume N_1 and N_2 were generated through this process.

Now, conditioned on any particular value $N = n$, we can apply a Chernoff bound: $\Pr[N_1/n \geq \lambda_1/\lambda + \epsilon | N = n] \leq e^{-2\epsilon^2 n}$. This implies $\Pr[N_1/N \geq \lambda_1/\lambda + \epsilon | N \geq \lambda/2] \leq e^{-\epsilon^2 \lambda}$. To complete the proof, note that $\Pr[N \leq \lambda/2] \leq (2/e)^{(\lambda/2)}$ [69, Other Properties]. □

5.4.3 Proof of Lemma 5.15

The proof is in three steps

1. Given any length-l_1 interval $I \subseteq [0, T]$, we show that with high probability the fraction of events that land in I is less than γ_2.

2. We define a sequence of overlapping intervals $I_0, \ldots, I_k \subseteq [0, T]$, each of length l_1. By a union bound, the fraction of events in any of these intervals is less than γ_2 with high probability.

3. Every possible length-l_2 subinterval of $[0, T]$ is contained in one of the intervals I_i, and thus contains less than a γ_2 fraction of events.

Step 1 Given a length-l_1 interval $I \subseteq [0, T]$, let $J = [0, T] \setminus I$ and $\lambda_I = \int_I \lambda$ and $\lambda_J = \int_J \lambda = |\lambda|_1 - \lambda_I$. By Lemma 5.17, the number of events that land in I and J are independent Poisson-distributed random variables with rates λ_I and λ_J, respectively. Since $\lambda(t)$ is (γ_1, l_1)-diffuse, $\lambda_I/|\lambda|_1 \leq \gamma_1$. Apply Lemma 5.18, taking $\epsilon = \gamma_2 - \gamma_1$: then with probability $2^{-\Omega(|\lambda|_1)}$, the fraction of all tweets that land in I is less than $\lambda_I/|\lambda|_1 + \epsilon \leq \gamma_2$.

Step 2 Let $\delta = l_1 - l_2$ and $n = \lceil T/\delta \rceil$. For $i = 0, \ldots, n$, let $I_i = [i\delta, i\delta + l_1]$. By a union bound, with probability $O(1)2^{-\Omega(|\lambda|_1)}$, no interval gets more than a γ_2 fraction of all tweets (if we allow the constants in the $O(\cdot)$ and the $\Omega(\cdot)$ to depend on l_1/T and l_2/T.

Step 3 Now, given any length-l_2 interval $I = [a, a + l_2] \subseteq [0, T]$, let $i^* = \lfloor a/\delta \rfloor$. Notice that interval I_{i^*} begins before a but after $a - \delta$. Since I_{i^*} is longer than I by δ, it follows that $I \subseteq I_{i^*}$. Thus in the (high probability) event that every interval I_i gets less than a γ_2 fraction of tweets, we see that every length-l_2 interval gets less than a γ_2 fraction of tweets, so the sampled timeline is (γ_2, l_2)-diffuse. □

Chapter 6

Qualitative Observations

In the course of our study, we read the transcripts of many meetings of various group chats. Here are some things we learned.

6.1 Progression of a Group Chat

Many group chats on Twitter have a common structure. Every meeting has a moderator who keeps the discussion on track. The nature of Twitter does not allow the moderator to stop people from posting messages, as is the case in some other contexts. Instead, the moderator plays an active role in the conversation. They will often ask questions at regular intervals through the chat, to keep the conversation moving and on topic — the moderator's first question begins "q1: ..." and participants answer with "a1: ...". Some of the more successful moderators are able to attract guest tweeters, who are typically celebrities in the group's area of interest. For example, a wine chat meeting featuring the celebrity Rodney Strong was well attended. Some chats have web pages announcing the topic of the next meeting, together with archives of previous chats. Moderators also remind frequent members of each meeting before it starts, remind participants to add the group's hashtag to each tweet, and close by thanking everyone for a successful chat and announcing the subject of the next chat.

6.2 Support Groups

We found the existence of support groups quite surprising. One typically associates support groups with a small number of people sitting around a circle, announcing their name and telling their story. It is hard to imagine giving or receiving support in 140 characters! But, with the same people meeting week after week, getting to know each other better, the platform has proven to be a place for support. Paraphrasing from a mental health chat, users state that social media enables them to access a support network, both those they know in

real life, as well as online contacts. They add that having a child with autism spectrum disorder is very isolating.

Support groups are successful for a variety of reasons. Empathy is the driving force, with users stating that "there is a family of us out there". In contrast to offline groups, the success of some online support groups can be attributed to the pseudo-anonymous online communication that is more comfortable than physical group meetings. For example, people say that they find group sessions hard, and that they hate opening up in front of others. In our experience as observers, we often felt that we had accidentally walked into a room full of people sharing personal experiences, but no one seemed bothered by the fact that anyone could hear what they were saying. In some cases finding others who are in a similar situation is challenging. For example, one user knew no one else in the same state who was transgendered, but was able to find a community online.

6.3 Hobbies

Passion-related groups are also quite fascinating. In Movie Talk on Sunday, a moderator pre-selects a theme — for example, suspense movies — and posts 10 questions ahead of time on a website. The moderator tweets a question every ten minutes, and participants tweet answers to the questions. The moderator and members retweet the answers they like. In this group, participants derive value from the discussion. For example, we found evidence of a user who decided to give the movie *Cabin in the Woods* a second try because so many in the group felt it was the best movie of the summer of 2012. We also found evidence of two users now dating after meeting in a group chat.

Chapter 7

Experiments

We present the results of running our algorithm on more than two years of Twitter data. We begin with a description of our data set and pre-processing (§7.1). We then describe the parameters we chose for the group chats algorithm (§§7.2,7.3). In the following sections, we present our experimental results. We study how often groups typically meet and how many members attend each meeting, and to understand the topics of discussion, we sample a subset of group chats and report on the distribution of categories using a popular taxonomy (§7.4). We observe that the number of living group chats has grown over time (§7.5). We conclude with some limitations of our method (§7.6).

7.1 Experimental Setup

Our experiments are based on more than 28 months of English-language tweets starting in September of 2010. To work with data at this scale (several petabytes) we implemented our algorithm in the SCOPE language [13] and used a large distributed computing cluster. We first obtained the set of all distinct hashtags used in this timeframe, and the timeline of tweets associated with each hashtag. We removed hashtags which were used in less than 20 tweets or by less than 10 users over the duration of the experiment. Then we used Algorithm 1 to determine which hashtags were group chats, as described in the next sections.

7.2 Determining the Periodicity Threshold

Algorithm 2 (§4.1) produces a period and periodicity score for every hashtag, and drops the hashtag from further consideration if the score is below a threshold δ. In order to determine this threshold, we manually labelled several hashtags as periodic and non-periodic by looking at the timeline of tweets for each one. A uniformly random sample of hashtags would favour hashtags with scores between 0 and 0.1, because most hashtags are not periodic. We therefore drew a stratified sample with five hashtags with score in between 0 and 0.1, five with scores between 0.1 and 0.2, and so on. The result of this manual tagging is shown in Figure 7.1

where the binary label of the hashtag is shown on the x-axis (1 denotes periodic) and the periodicity score is shown on the y-axis. There is a good separation between the periodic and not periodic hashtags when the score is set to $\frac{1}{4}$. We also computed the F-measure (harmonic mean of precision and recall) for different choices of the threshold, and confirmed that the maximum is achieved at $\delta = \frac{1}{4}$. We then kept all hashtags with a periodicity score $\geq \frac{1}{4}$.

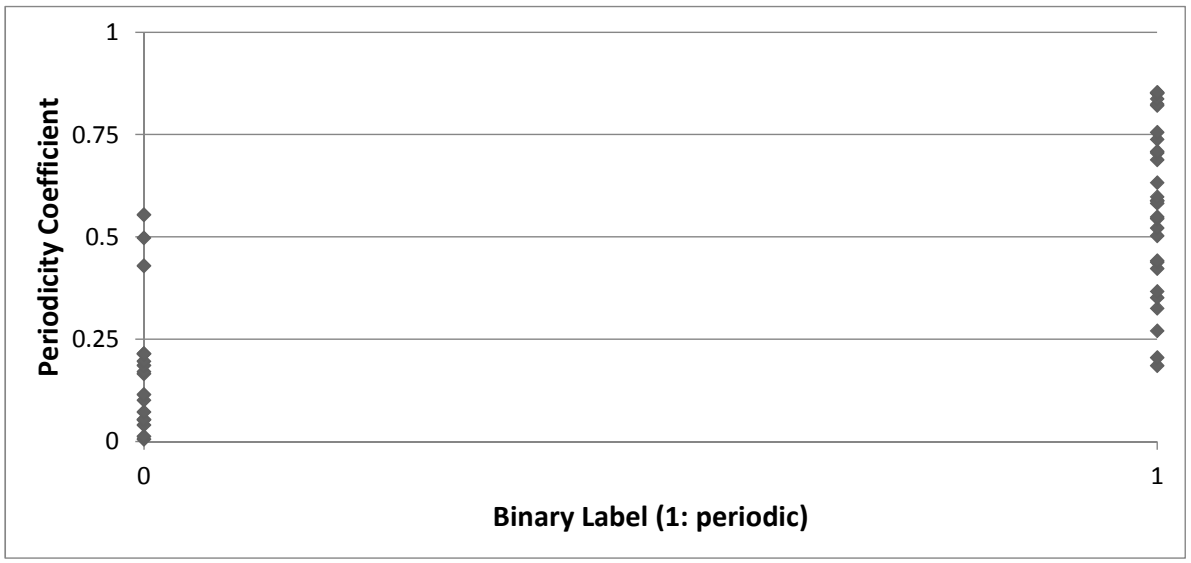

Figure 7.1: Result of manually labelling a stratified sample of hashtags: 1 denotes periodic and 0 not periodic.

7.3 Finding Group Chats

Next we describe how we found the group chats.

We selected all hashtags with periodicity threshold at least $\frac{1}{4}$, as justified above, and removed those that met less that $\mu = 5$ times. For each remaining hashtag, we checked whether there were meetings which took place at a consistent time (§4.3). We set the maximum allowed duration of a meeting l to be two hours and the synchronization threshold γ to be 0.2. These choices require a sizable fraction of tweets to appear within two hours of the meeting start time.

Finally, we eliminated hashtags without cohesion (§4.4). For each candidate hashtag, we found the $k = 5$ users who participated in the most meetings, and determined whether the average number of pairs among those k users who interacted in a meeting was at least $\psi = k - 1$. This structure is realized by, for example, a moderator that routinely converses with $k-1$ other members. Note that $\psi \geq k-1$ does not guarantee connectivity, but $\psi < k-1$ guarantees disconnectivity — for example, it could reflect a group that is just forming or dissolving or simply not cohesive.

After running Algorithm 1, we were left with 1.4K groups involving 2.3M users (counting each user multiple times if they participated in multiple group chats). These are the subject of our study.

7.4 Group Chat Analysis

We show the distribution of periods for group chats in Figure 7.2. About 80% of the group chat hashtags had a period of one week. However, there are some that meet every day (e.g, those tied to daily radio shows) and some that meet biweekly. For the rest of this section, we restrict our attention to group chats with a period of one week.

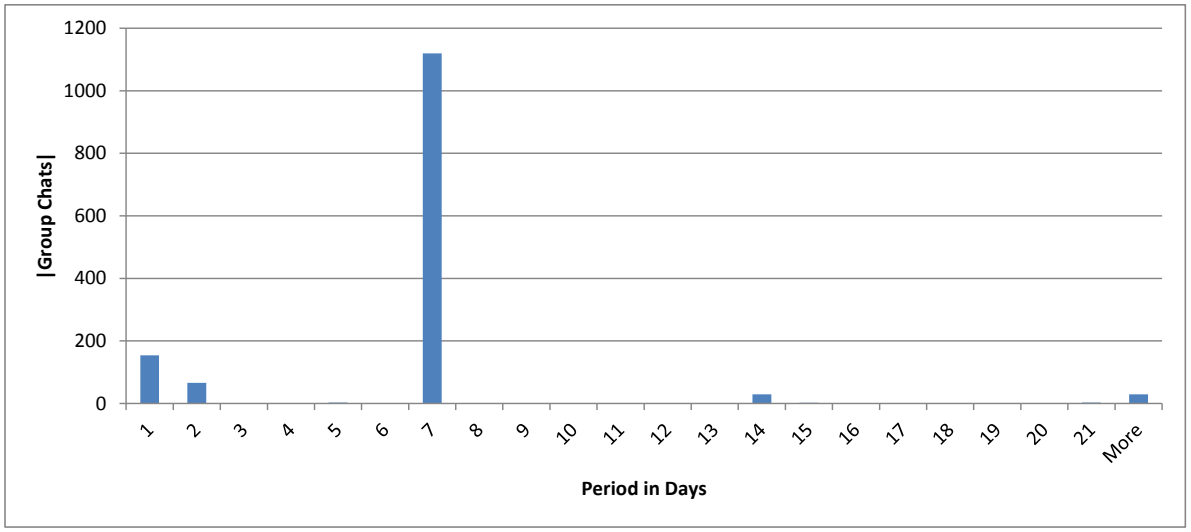

Figure 7.2: Distribution of periods of hashtags with periodicity score at least $\frac{1}{4}$.

Next we show the average number of users per meeting in Figure 7.3. For each group, we computed the average meeting attendance over the course of our data collection. The histogram shows the number of groups that have an average number of users in the range 10-20 users, 20-30 users, etc. Interestingly, we find that most groups have less than 30 members per meeting. Generally speaking, when a group grows too large, the real-time chat becomes more difficult to follow. This is not to say that large groups do not exist. There are cohesive group conversations tied to TV shows that have a very large number of users.

To understand the types of groups we found, we randomly sampled 10% of the recent group chat hashtags and manually categorized the hashtags into the top-level of the Open Directory Project (ODP) taxonomy. We were able to find categories for 95% of the group chats based on recent tweets. Table 7.1 shows the fraction of group chats that we assigned to each category. A large fraction of the groups are Arts related, including groups tied to weekly TV shows, radio shows, book clubs, craft clubs, and so on. Science is the next

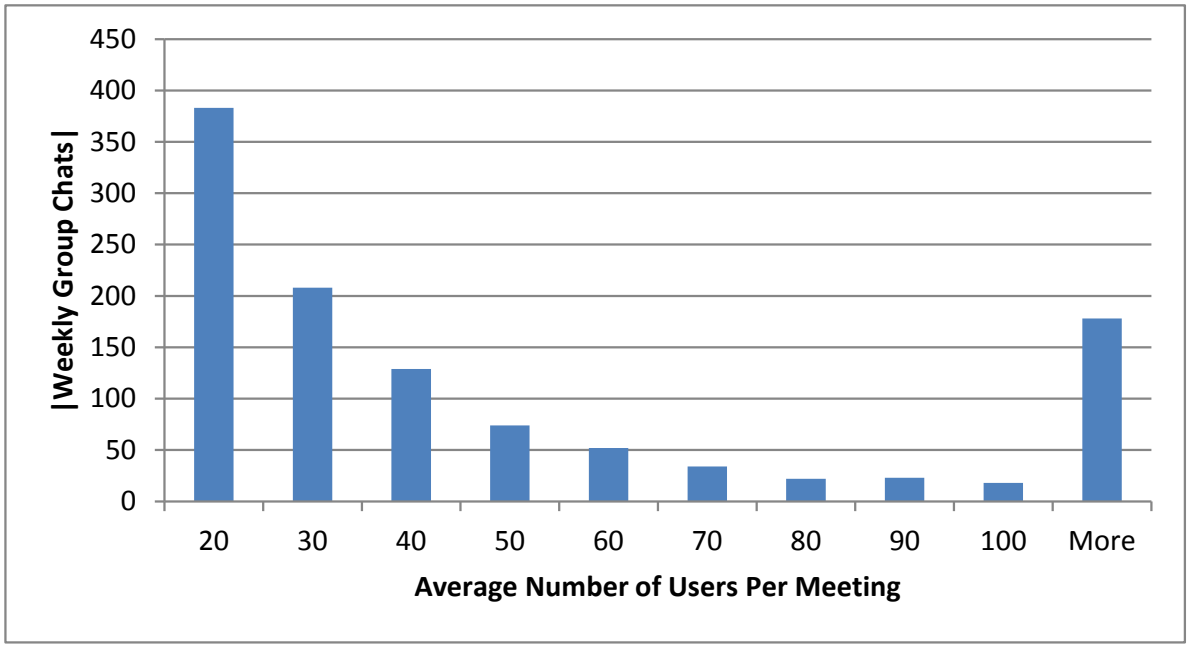

Figure 7.3: Meeting size: Average number of users per meeting. Most group meetings have a small number of participants.

largest category, with education chats dominating the category. In these chats, teachers discuss ways to be more effective educators. There is also a large number of Health-related groups including support groups for coping with addiction and borderline personality disorder. There are group chats in the Business category discussing ways to prevent fraud (for example, against seniors) and how to be a digital leader. A small number of groups conversed in languages other than English; this was one way to end up in our Don't Know category. (We restricted to English tweets only, but some groups communicate in multiple languages). In the remaining categories, there are sports enthusiasts and foodie groups. Finally, there are contest-driven chats where the goal is to give away a prize to the person who can answer the most trivia questions.

7.5 The Number of Group Chats over Time

Finally, we ask whether group chats are a growing or shrinking phenomenon on Twitter. To answer, for each weekly group, we computed the birth date of the group by finding the first weekly meeting during which at least 10 members tweeted, and similarly the death death of the group by the last weekly meeting where at least 10 members tweeted. We computed the cumulative number of births and deaths over time, and considered the difference to be the number of living groups. The chart is shown in Figure 7.4. We found that the rates of births and deaths have both increased over time, and that the net number of living groups

Category	% Groups	**Examples**
Arts	47%	TV/Radio, Writing, Music, Crafts
Science	12%	Education, Agricultural
Health	10%	Addiction, Self-help, Mood Disorder
Business	9%	Preventing Fraud, Digital Leaders
Don't Know	5%	
World	5%	Foreign
Sports	3%	Basketball, Soccer
Society	3%	Better blogger, Rights activist
Recreation	3%	Foodie
Games	2%	Prize-driven

Table 7.1: Category distribution of 10% random sample of Twitter weekly group chats.

has grown. We do not understand what causes a group to grow or to die, but it is a good subject for future work.

Since our data is restricted to a time window, we could not accurately find the birth date of a group that was born before the window, and if a group was born later and did not have five meetings before the end of the window, we did not find the group. To account for this, we threw out the first and last few months of our computed timelines of births, deaths, and net living groups.

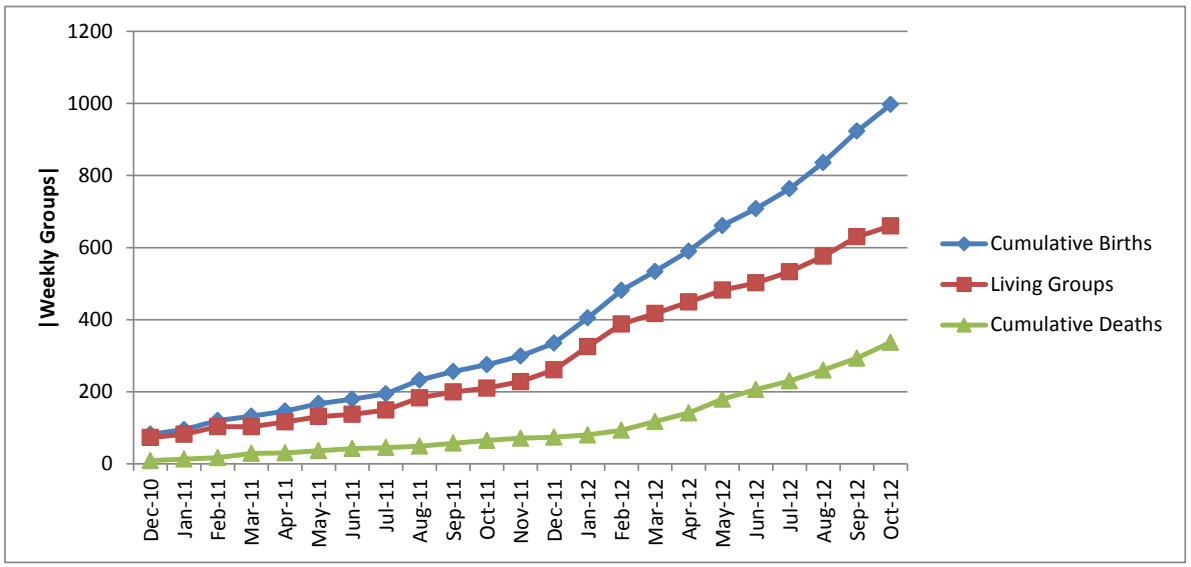

Figure 7.4: The top curve shows the number of weekly group chats born over time, the bottom curve shows the number that died over time, and the middle curve shows the number of weekly living groups over time.

7.6 Limitations

While our methods do indeed find group chats, we know that there are some group chats that we miss. For example, groups that meet the first Monday of the month are not found by our method because the separation between meetings could be either four or five weeks. If a group misses many meetings, it may be difficult for our method to find. For example, ski chats are typically on hiatus over the summer. Other group chats that we miss have irregularly spaced meetings: for example, users might agree on each meeting time at the end of the previous meeting. Also, since our algorithm ignores tweets without the candidate group chat hashtag, we overlook conversations that happen between users outside the context of the group chat. Finally, analyzing an ambiguous group chat hashtag such as `#tchat` would require teasing out the group chat uses from alternate uses of the hashtag.

Part II

Ranking Discussion Groups

In which we develop a search algorithm to rank discussion groups relevant to a query topic, based on a model of a user browsing a set of related discussion groups; and in which we evaluate the performance of this algorithm both theoretically and experimentally. We hope this algorithm will help users interested in discussing a topic find discussion groups, such as Twitter chats, of which they might not have been aware.

This part includes material from *Ranking Discussion Groups*, co-authored with Abhimanyu Das, Krishnaram Kenthapadi and Nina Mishra, and submitted to the 2014 SIGKDD conference.

Chapter 8

Problem Formulation

Consider any setting where there are many groups $g_1, \ldots, g_n$ which 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. It is our 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.

8.1 Discussion Groups

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

Definition 8.1 (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 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.

8.2 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 8.2. *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.

8.3 Twitter Interpretation

To interpret Definition 8.1 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 8.2, taking Twitter chats as our set of discussion groups.

Chapter 9

Model

To solve Problem 8.2, 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 boosted upward through a feedback effect (§10.2). The model also satisfies several desirable properties described in Section 10.1. See Section 2.1.7 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 jumping from a participant to a group.

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

After the seeker arrives at a discussion group g, they choose a participant to jump to according to their *authority score*. The authority of different participants p within a group g is quantified with authority scores $A_{q,g}(p)$ which form a probability distribution. The scores could be determined in many different ways. As a first 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.

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

After the seeker arrives at a participant p, they look at other discussion groups that person has participated in and jump to one according to its *preference score*. For a query q and person p, the preference scores $P_{q,p}(g)$ of person p for different groups g form a probability

distribution. One straightforward way to determine $P_{q,p}(g)$ 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 which case we add g' as a subscript to the notation $P_{q,p,g'}(g)$. For example, in Section 11.2 we describe our final implementation, where the seeker never jumps to a group g if p is less active in g than g'. However preference scores are determined, it should be the case that $\sum_g P_{q,p,g'}(g) = 1$ for every p and g'.

9.3 Teleport Distribution D_q

After each step, with some probability $\lambda \in (0, 1)$ the seeker will decide to cut short their current exploration, and choose 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 generally a probability distribution over discussion groups relevant to the topic q. D_q plays the same role as the preference vector in personalized PageRank [52, 34]. 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 Chapter 10.

9.4 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.

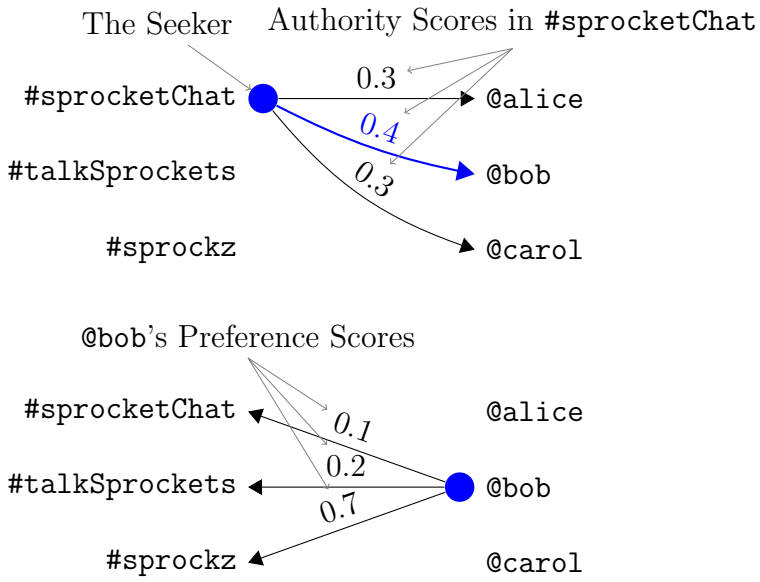

Figure 9.1: The first steps of the group preference model in the context of Twitter. Starting from a random chat (`#sprocketChat`), the seeker jumps to a random user according to authority scores in that chat, and then to a random chat according to that user's preferences. Whether or not real users follow this process, we find it useful for ranking chats.

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 9.1 illustrates the first three steps of the process in the context of Twitter. We do not know whether real users follow this process, but we find the model useful as a basis for our algorithm.

Chapter 10

Algorithm and Analysis

We now describe our algorithm for solving Problem 8.2 using the group preference model. The key observation is that even though the seeker visits both discussion groups and participants, the model can be represented by the following Markov process over just the groups with transition probabilities $M(q)_{g_1,g_2}$ computed as follows:

$$M(q)_{g_1,g_2} = \lambda D_q(g_2) + (1-\lambda)\sum_{p \in U} A_{q,g_1}(p) P_{q,p,g_1}(g_2) \tag{10.1}$$

where n is the number of relevant groups and U is the set of people who participate in any such group. Each transition probability $M(q)_{g_1,g_2}$ in (10.1) 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 $(1-\lambda)A_{q,g_1}(p)P_{q,p,g_1}(g_2)$, 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 (§9.3) that every probability is positive.)

Given a query q, our algorithm (see Algorithm 4) is then to compute the stationary distribution of $M(q)$ and rank the discussion groups by their stationary probabilities.

10.1 Properties of Algorithm 4

At first glance, it is not obvious that Algorithm 4 will behave in a reasonable way. 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 a state s can increase the stationary probabilities of many other states, and when λ is close to 0, it is possible for a small change to have a large effect. In this section, we show that our

Algorithm 4 Rank discussion groups for a query topic q.

Parameters: Teleport parameter $\lambda \in (0, 1)$; authority and preference score functions $A_{q,g}, P_{q,p,g'}$; teleport distribution D_q.

Input: A set of candidate discussion groups (Def. 8.1); a dataset of group interactions; a query q.

Output: A ranking of groups relevant to topic q.

1: Find all groups $g_1, \ldots, g_n$ where the topic q is mentioned in some group interaction.
2: Compute the authority and preference scores and teleport probabilities $A_{q,g}(p)$, $P_{q,p,g'}(g)$, $D_q(g)$ for every g, g', p.
3: Compute the stationary distribution π of the Markov process $M(q)$ defined in (10.1).
4: **return** The groups ranked so $g_1 >_q g_2$ iff $\pi(g_1) > \pi(g_2)$.

algorithm has several simple properties desirable of any ranking algorithm: for example, if a participant shows an increased preference for a discussion group g, then g's ranking will not be negatively affected (Theorem 10.5). We omit some proofs because of space constraints.

Our first property describes what happens when every participants prefers one group g_1 over another g_2. The property holds when the teleport distribution is uniform, or at least doesn't favour g_2 over g_1.

Theorem 10.1. *If for topic q, every participant always assigns a higher preference score to group g_1 than g_2, and g_2 doesn't 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 (10.1), 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 when every user's preference for a single group g_1 is high. This property holds when the teleport distribution is uniform.

Theorem 10.2. *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 (10.1), $M(q)_{g,g_1} \geq \gamma$ for every group g. 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 (§§9.1,9.2) 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 the if we add to the dataset a new participant who shares a preference with all the existing participants, that preference will continue to 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 continue to be ranked in the top α.

Corollary 10.3. *Suppose that in T, every participant always assigns a higher preference score to g_1 than g_2 and g_2 doesn't 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 (teleport probabilities, and preference and 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 follows because the hypotheses of Theorem 10.1 or Theorem 10.2 are still true in dataset T'.

Our next two theorems will make use of a result by Chien et al. [14] 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 10.4 (Chien et al. [14, Theorem 2.9])**.** *Fix some state s_1. Consider two regular Markov chains M and M', and suppose that transition probabilities to states other than s_1 are not increased. That is, for every $s_2 \neq s_1$ and s_3, $M'_{g_3,g_2} \leq M_{g_3,g_2}$ (and since rows sum to one, $M_{g_3,g_1} \geq M_{g_3,g_1}$).*

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

Proof. For $i = 0, \ldots, n$, let P_i be the transition matrix whose first i rows are taken from $M(T',q)$ and whose remaining rows are taken from $M(T,q)$. Let π_i be the stationary distribution of P_i, so π_0 is the stationary distribution of $M(T,q)$ and π_n is that of $M(T',q)$. All of these matrices form regular Markov chains because of the positive term $\lambda\frac{1}{n}$ in (10.1). The matrix P_{i+1} can be obtained from P_i by increasing the entry at (i,g_1) and decreasing the others — so by Theorem 10.4, if π_i assigns a higher probability to g_1 then g_4, then so does π_{i+1}. Since this property is true of π_0, by induction, it is true for π_n. □

Theorem 10.4 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 10.5. *Suppose the only change from T to T' is that participant p_1 shows an increased preference for a discussion 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 discussion 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, the Markov transition matrix (10.1) changes as $\forall h_1, h_2$,

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

Now, $P_{T',q,p,h_1}(h_2) - P_{T,q,p,h_1}(h_2) = 0$ whenever $p \neq p_1$, so

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

This quantity is nonnegative when $h_2 = g_1$ and nonpositive otherwise. So by Theorem 10.4, 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 10.6. *Suppose participant p_1 has an exclusive preference for discussion group g_1: $P_{T,q,p_1,g'}(g_1) = 1$ for all g'. Assume the only change from T to T' is that p_1 gains authority. That is: $A_{T',q,g}(p_1) \geq A_{T,q,g}(p_1)$ for every group c; 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 10.4, if $g_1 >_q^T g'$, then $g_1 > {T'}_q g'$. □

There are simpler algorithms than Algorithm 4 which also satisfy the properties in this section. For example, the naïve algorithm that ranks chats by the number of tweets containing the query q works. However, in Section 10.2, we describe a scenario showing an advantage of Algorithm 4 over the naïve algorithm. In Chapter 11, we evaluate the algorithm experimentally.

10.2 Comparing to the Naïve Approach

Instead of using Algorithm 4, 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 4, consider the following scenario, illustrated in Figure 10.1:

Scenario 10.7. *There is a set of discussion groups G_{pop} which we imagine as being not relevant the topic q. However, the set includes some extremely* pop*ular groups. There is a group $g_* \notin G_{\text{pop}}$ which is relevant to topic q, and supported by participants who are interested in q. There is a very large set F of participants who we think of as* fans *of groups in G_{pop} and uninterested in q. We assume the following two properties:*

- *Every non-fan $p \notin F$ who mentions topic q assigns small preference scores to G_{pop}: $\forall g'$, $\sum_{g \in G_{\text{pop}}} P_{q,p,g'}(g) < \epsilon$.*
- *On the other hand, 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 10.8. *In Scenario 10.7 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 every non-fan $p \notin F$ has a preference of at least $\frac{8}{7}(\beta + \frac{\lambda}{1-\lambda} D_{\text{pop}})/(1-\beta)$ for the group g_*, where $\beta = D_{\text{pop}} + \frac{2\epsilon}{\lambda}$. Then if $\epsilon < \frac{1}{8}$ and $\lambda < 1$, then Algorithm 4 will rank group g_2 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 big.

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

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

It follows that for any d, $(dM(q))_{G_{\text{pop}}} \leq \lambda D_{\text{pop}} + (1-\lambda)(d_{G_{\text{pop}}} + 2\epsilon(1 - d_{G_{\text{pop}}}))$. 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

$$\pi_{G_{\text{pop}}} \leq D_{\text{pop}} - 2\epsilon\tfrac{1-\lambda}{\lambda} = \beta - 2\epsilon. \tag{10.2}$$

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

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

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

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

and so for any distribution d,

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

We have:

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

$\square$

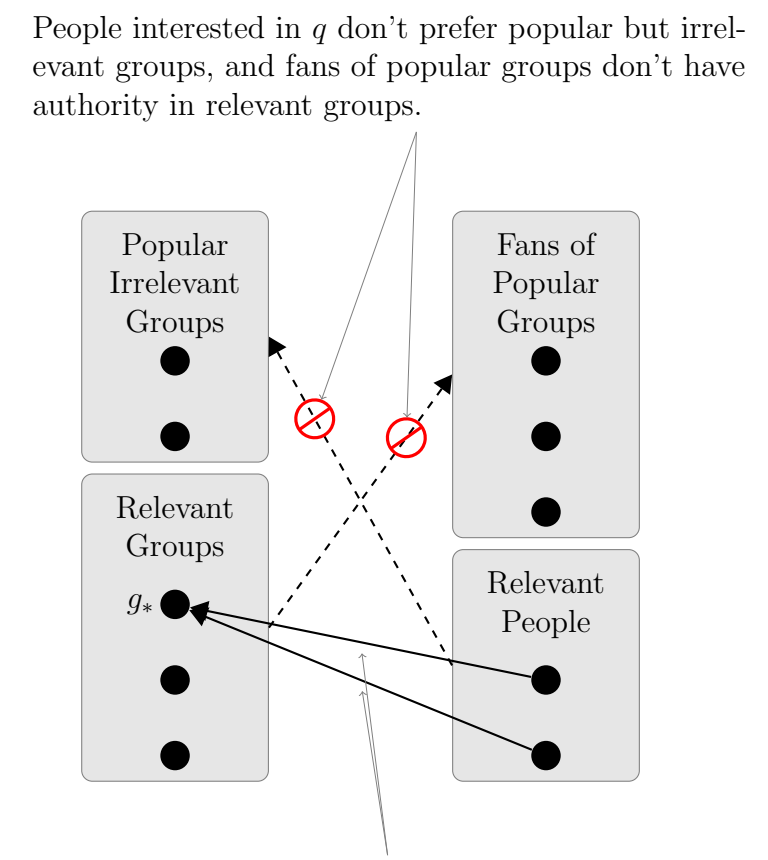

Figure 10.1: An illustration of Scenario 10.7. There is a set of popular but irrelevant groups with many fans. Although the most relevant group g_* has fewer supporters, the whole community of relevant groups gives authority to those supporters. Under the right conditions, g_* will be ranked at the top (Theorem 10.8).

Chapter 11

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.

11.1 Experimental Setup

We gathered one year of tweets and extracted noun phrases (using a part-of-speech tagger) to capture potential queries that a tweet may contain. Underutilized hashtags were removed (present in 60 tweets or used by less than 10 people), as were underutilized queries (less than 100 tweets).

11.1.1 Identifying Twitter chat hashtags

The set of Twitter chats was determined as per Chapter 8. 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 users 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 chat, there should have been at least $m = 10$ weeks containing valid meetings. We obtained a total of $27K$ chats using the above process.

11.1.2 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 users queries 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 behaviour. After intersecting these queries with the set that we gathered from tweets, we were left with 2K queries.

11.1.3 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 a novel 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 users or "experts" by selecting users who mention the query phrase in their Twitter profile. We believe that given the limited space allowed for a user's Twitter profile, users who explicitly mention the query (for example, "camera" or "diabetes") in their Twitter profile, are more likely to be a "subject-matter expert" (a photographer or a patient) than a random user 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 experts 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 experts that have written tweets containing the query and the hashtag.

From among the $2K$ candidate queries, we were able to obtain this "experts-based ranking" for only around 600 queries (for the remaining queries, we could not find enough experts who mentioned the query in their user profiles). Note that this coverage issue is a 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 expert based ranking revealed that while the ranking had good precision for most queries, it had two shortcomings: firstly, it did not have sufficient recall and failed to report hashtags that we manually found to be very relevant to the query (e.g. `#photographychat` for the query, "camera" and `#tl_chat` for the query "travel", both of which are highly relevant Twitter chats) and secondly, there were some queries on which its precision was quite poor.

To resolve these issues, we resorted to a pooling methodology in information retrieval [66] and manually created the final ground truth as follows: for each query, we pooled together the top 10 hashtags output by the above experts-based 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 labour-intensive, we restricted the set of queries for which we generated ground-truth rankings by sampling 50 queries from among the 600 candidate queries.

11.2 Implementation choices

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

11.2.1 Authority Score

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

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

11.2.2 Teleport Distribution

As described in Chapter 9, 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.

11.2.3 Preference Score

As described in our model, a key component of our GroupPreference algorithm is the computation of the transition probability matrix for a (query, user) pair. We next describe our approach for obtaining the transition edge probabilities for each user, query and pair of hashtags (say, h1 and h2). For computing the user's preference between h1 and h2, we wish to only use Twitter data corresponding to the time when the user was "aware" of *both* the hashtags. We define the user's awareness-time for a hashtag as the first time when the user tweeted with that hashtag. Using this definition, we then restrict the tweets of the user to the time-period starting from the greater of the user's awareness time for h1 and h2. For this time-period, we compute the number of meetings of h1 and h2 attended by the user for the given query (i.e. the number of two-hour windows within which the user has written at least one tweet containing the hashtag and query). We define a transition probability of 1 from h1 to h2 (resp. h2 to h1), if the user has attended "significantly more" (we use a relative difference threshold of 0.1 for estimating significance) meetings of h2 compared to h1 (resp. h1 compared to h2).

11.3 Baseline Algorithms

We compared our ChatPreference algorithm against the following baselines, all of which correspond to various intuitive notions of the popularity of a chat 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 users that have posted tweets containing the given hashtag and query.

Tweet Frequency-based Ranking Algorithm (TFA): For each query, we assign a score to each hashtag based on the total number of tweets containing the given hashtag and query.

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. Note that this is reminiscent of the tf-idf metric in information retrieval: the numerator corresponds to a notion of term-frequency and the denominator acts as a discounting factor.

In addition to the above three baselines, we also compare our algorithm against the experts-based ranking (ExpertsPreference) 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 ExpertsPreference.

11.4 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 chats identified by an algorithm and by the ground truth respectively, with $A[i]$ (resp. $G[i]$) being the i^{th} chat. For every chat p, let $R(p) \in [0, 3]$ be the ground truth relevance rating provided by the human assessor. We define the following metrics [60]:

Weighted Precision: The WeightedPrecision @K of the algorithm at the top K rank is $\frac{\sum_{i=1}^{K} R(A[i])}{3K}$ [60].

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 $\frac{1}{|G|} \cdot \sum_{p \in (G \cap A)}$ WeightedPrecision @$\mathrm{r_{p,A}}$, where $r_{p,A}$ is the rank of chat p in A.

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 chat is rounded to 1 if $R(p) \geq 2$ and 0 otherwise. We set $K = 5$.

11.5 Results of Implementation Choices

11.5.1 Teleport Distribution

We first study the effect of varying the teleport probability from 0 to 1, with NounFreqWeights as the authority score. From Table 11.1, 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 chats 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 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 11.2), confirming that it is desirable to factor in the query-specific popularity of hashtags instead of teleporting uniformly.

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 11.1: 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 11.2: Benefit of Non-uniform Teleport Distribution

11.5.2 Authority Score

We present a comparison of different authority scores in Table 11.3. 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 users 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 users, and the aggregate preference captures this signal irrespective of the weights given to the users. The users 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 chats 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.

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 11.3: Empirical Analysis of Different Authority Scores

11.6 Performance Results

We next compare the performance of our algorithm (with NounFreqWeights as the authority score) with the three baselines, and the experts-based ranking in Table 11.4. We observe that our algorithm significantly outperforms the best baseline, TFA along all six metrics. Our algorithm improves TFA by 30% with respect to mean average precision (0.437 vs 0.336), and about 25% with respect to weighted mean average precision (0.309 vs 0.246). With respect to these two metrics, our algorithm achieves about 70% of the performance of ExpertsPreference, which, as noted earlier, is not a practical algorithm but can serve as an upper bound.

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

Table 11.4: Performance of Different Algorithms

11.6.1 Qualitative Evaluation of Chat Rankings

To provide qualitative insights into the ranking algorithms, we next highlight the top-3 Twitter hashtags retrieved by the algorithm/baselines for 6 representative queries, in Table 11.5. (Due to lack of space, we omitted the poorest performing baseline of UFA). 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 chats compared to the baselines, and performs almost as well as ExpertsPreference. For example, for the query "garden", both GroupPreference and ExpertsPreference retrieve a weekly Twitter chat about gardening enthusiasts (`#gardenchat`) as the top hashtag (though ExpertsPreference also retrieves another related Twitter chat related to roses (`#rosechat`).

On the other hand, the baselines results are not very relevant. Indeed, TFA 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 chats about jobs and hiring (`#omcchat`, `#animalchat` and `#hfchat`), and outperforms all the other algorithms that return at least one chat that is not a discussion group (for example, `#jobfair` or `#forbesgreatesthits`). For the query "hotels", both GroupPreference and ExpertsPreference return a travel-related weekly Twitter chat 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).

Query	**TFA**	**TRA**	**GroupPreference**	**ExpertsPreference**
diabetes	#dsma #ozdoc #herecomeshoneybooboo	#dsma #ozdoc #tipkes	#dsma #ozdoc #gbdoc	#dsma #gbdoc #ozdoc
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
nurse	#upgradedrappernames #spon #michigan	#nursejackie #rnfmradio #nurseshift	#wenurses #nursejackie #rnfmradio	#wenurses #nurseshift #nttwitchat
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 11.5: Sample Chat Rankings using Different Algorithms

Part III

Your Two Weeks of Fame and Your Grandmother's

In which, inspired by common intuition that the public's attention span has been getting shorter as communication technology improves and news cycles speed up, we perform a study of the phenomenon of personal fame across a century of news articles.

This part includes material from *Your Two Weeks of Fame and Your Grandmother's*, co-authored with Atish Das-Sarma, Alex Fabrikant and Andrew Tomkins, and appearing in the proceedings of the 2012 International Word Wide Web Conference (ACM). Work done while I was an intern at Google.

Chapter 12

Working with the news corpus

We perform our main study on a collection of the more than 60 million news articles in the Google archive that are both (1) in English, and (2) searchable and readable by Google News users at no cost. In Chapter 15, we cross-validate our observations against the corpus of public blog posts on Blogger, which is described there.

The articles of the news corpus span a wide range of time, with the relative daily volume of articles over the range of the corpus shown in Figure 12.1. There are a handful of articles from the late 18th century onward, and the article coverage grows rapidly over the course of the 19th century. From the last decade of the 19th century through the end of the corpus (March 2011), there is consistently a very substantial volume of articles per day, as well as a wide diversity of publications. For the sake of statistical significance, our study focuses on the years 1895–2011.

The news corpus contains a mix of modern articles obtained from the publisher in the original digital form, as well as historical articles scanned from archival microform and OCRed, both by Google and by third parties. For scanned articles, per-article metadata such as titles, issue dates, and boundaries between articles are also derived algorithmically from the OCRed data, rather than manually curated.

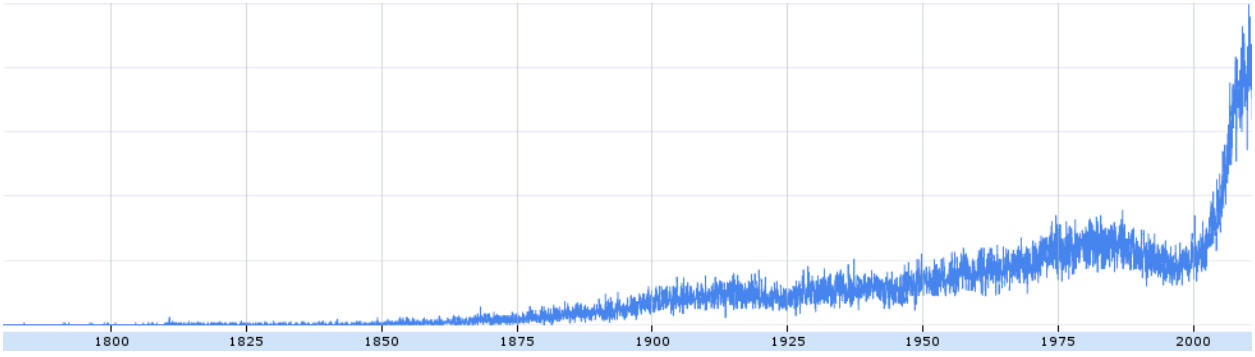

Figure 12.1: The volume of news articles by date.

Our study design was driven by several features that we discovered in this massive corpus. We list them here to explain our study design. Also, data mining for high-level behavioural patterns in a diachronous, heterogeneous, partially-OCRed corpus of this scale is quite new, precedented on this scale perhaps only by [50] (which brands this new area as "culturomics"). But, with the rapid digitization of historical data, we expect such work to boom in the near future. We thus hope that the lessons we have learned about this corpus will also be of independent interest to others examining this corpus and other similar archive corpora.

12.1 Corpus features, misfeatures, and missteps

12.1.1 News mentions as a unit of attention

Our 116-year study of the news corpus aims to extend the rich literature studying topic attention in online social media like Twitter, typically over the span of the last 3–5 years. Needless to say, 100-year-old printed newspapers are an imperfect proxy for the attention of individuals, which has only recently become directly observable via online behaviour. Implicit in the heart of our study is the assumption that news articles are published to serve an audience, and the media makes an effort, even if imperfect, to cater to the audience's information appetites. We coarsely approximate a unit of attention as one occurrence in a Google News archive article, and we leave open a number of natural extensions to this work, such as weighting articles by historical publication subscriber counts, or by size and position on the printed page.

Due to the automated OCR process, not every "item" in the corpus can be reasonably declared a news article. For example, a single photo caption might be extracted as an independent article, or a sequence of articles on the same page might be misinterpreted as a single article. Rather than weighting each of these corpus items equally when measuring the attention paid to a name, we elected to count multiple mentions of a name within an item separately, so that articles will tend to count more than captions, and there is no harm in mistakenly grouping multiple articles as one.

We manually examined (A) a uniform sample of 50 articles from the whole corpus (which, per Fig. 12.1, contains overwhelmingly articles from the last decade), and (B) a uniform sample of 50 articles from 1900–1925. We classified each sample into:

- News articles: timely content, formatted as a stand-alone "item", published without external sponsorship, for the benefit of part of the publication's audience,
- News-like items: non-article text chunks where a name mention can qualify as that person being "in the news" — e.g. photo captions or inset quotes,
- Non-news: ads and paid content, sports scores, recipes, news website comments miscategorized as news, etc.

The number of items of each type in the two samples are given in the following table.

	full corpus sample	1900–1925 sample
news articles	31	28
news-like items	3	2
non-news items	16	20

We expect that the similarity in these distributions should result in minimal noise in the cross-temporal comparisons, and leave to future work the task of automatically distinguishing real news stories from non-news.

12.1.2 Compensating for coverage

Even once we discard the more sparsely covered 18th and 19th centuries, there is still more than an order of magnitude difference between article volume in 1895 and 2011. We address these coverage differences by downsampling the data down to the same number of articles for each month in this range. We address the nuanced effects of this downsampling on our methodology in Section 13.3.

12.1.3 Evolution of discourse and media — why names?

We set out originally to understand changes in the public's attention as measured by news story topics. There are a myriad heuristics to define a computationally feasible model of a "single topic" that can be thought to receive and lose the public's attention. But over the course of a century, the changes in society, media formatting, subjects of public discourse, writing styles, and even language itself are substantial enough that neither sophisticated statistical models trained on plentiful, well-curated training data from modern media nor simple generic approaches like word co-occurrence in titles are guaranteed to work well. Very few patterns connect articles from 1910 newspapers' "social" sections (now all but forgotten) about tea at Mrs. Smith's, to 1930 articles about the arrival of a trans-oceanic liner, to 2009 articles about a viral YouTube video.

After trying out general proper noun phrases produced inconclusively noisy results, we decided to focus on occurrences of personal names, detected in the text by a proprietary state-of-the-art statistical recognizer. Personal names have a relatively stable presence in the media: even with high OCR error rates in old microform, over 1/7th of the articles even in the earliest decades since 1900 contain recognized personal names (see Figure 12.2).

But personal names are not without historical caveats, either. A woman appearing in 2005 stories as "Jane Smith" would be much more likely to be exclusively referenced as "Mrs. Smith", or even "Mrs. John Smith", in 1915. Also, the English-speaking world was much more Anglo-centric in 1900 than now, with much less diversity of names. An informal sample suggests that most names with non-trivial news presence 100 years ago referred overwhelmingly to a single bearer of that name for the duration of a particular news topic, but many names are not unique when taken across the duration of the whole corpus — for instance, "John Jacob Astor", appearing in the news heavily over several decades (Fig. 13.1),

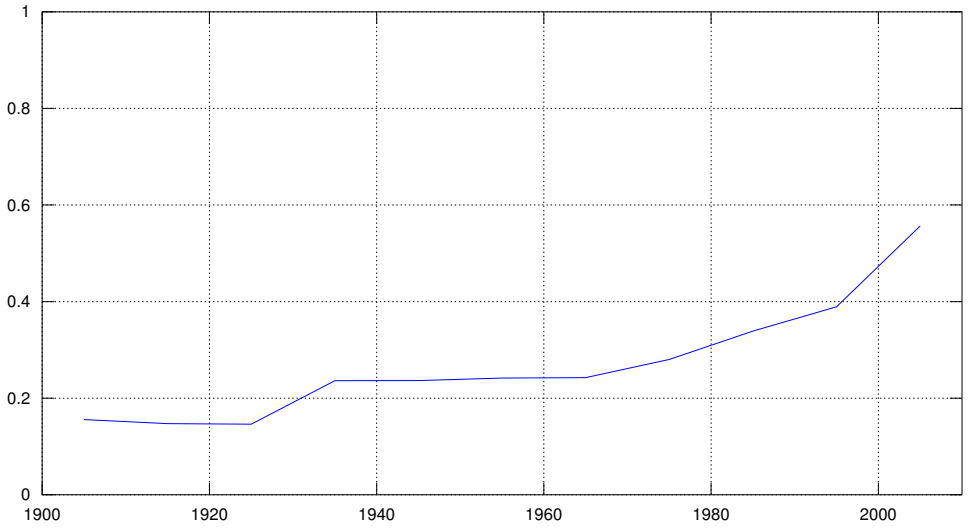

Figure 12.2: Articles with recognized personal names per decade

in reference to a number of distinct relatives. On account of both of these phenomena, among others, we aim to focus on name appearance patterns that are most likely to represent a single news story or contiguous span of public attention involving that person, rather than trying to model the full media "lifetime" of individuals, as we had considered doing at the start of this project.

12.1.4 OCR errors in data and metadata

We empirically discovered another downfall of studying long-term "media lifetimes" of individuals. In an early experiment, we measured, for each personal name, the 10th and 90th percentiles of the dates of that name's occurrence in the news. We then looked at the time interval between 10th and 90th percentiles, postulating that a large enough fraction of names are unique among newsworthy individuals that the distribution of these *inter-quantile gaps* could be a robust measure of media lifetime. After noticing a solid fraction of the dataset showing inter-quantile gaps on the scale of 10-30 years, we examined a heat map of gap durations, and discovered a regular pattern of gap durations at exact-integer year offsets, which, other than for Santa Claus, Guy Fawkes, and a few other clear exceptions, seemed an improbable phenomenon.

This turned out to be an artifact of OCRed metadata. In particular, the culprit was single-digit OCR errors in the *scanned article year*. While year errors are relatively rare, every long-tail name that occurred in fewer than 10 articles (often within a day or two of each other), and had a mis-OCRed error for one of those occurrences contributed probability mass to integral-number-of-years media lifetimes. As extra evidence, the heat map had a

distinct outlier segment of high probability mass for inter-quantile range of exactly 20 years, starting in the 1960s and ending in the 1980s — the digits 6 and 8 being particularly easy to mistake on blurry microfilm. Note that short-term phenomena are relatively safe from OCR date errors, thanks to the common English convention of written-out month names, and to the low impact of OCR errors in the day number.

OCR errors in the article text itself are ubiquitous. Conveniently, the edit distance between two recognizable personal names is rarely very short, so by agreeing to discard any name that occurs only once in the corpus, we are likely to discard virtually all OCR errors as well, with no impact on data on substantially newsworthy people. We should note that OCR errors are noticeably more frequent on older microfilm, but the reasonable availability of recognizable personal names even in 100-year-old articles, per Fig. 12.2, suggests that this problem is not dire. A manually-coded sample of 50 articles with recognized names from the first decade of the 1900s showed only 8 out of 50 articles having incorrectly recognized names (including both OCR errors and non-names mis-tagged as names).

12.1.5 Simultaneity and publishing cycles

There are also pitfalls with examining short timelines. In the earliest decades we examine, telegraph was widely available to news publishers, but not fully ubiquitous, with rural papers often reporting news "from the wire" several days after the event. An informal sample seems to suggest that most news by 1900 propagated across the world on the scale of a few days. Also, many publications in the corpus until the last 20 years or so were either published exclusively weekly or, in the case of Sunday newspaper issues, had substantially higher volume once a week, resulting in many otherwise obscure names having multiple news mentions separated by one week — a rather different phenomenon than a person remaining in the daily news for a full week. On account of both of these, we generally disregard news patterns that are shorter than a few days in our study design.

Chapter 13

Measuring Fame

We begin by producing a list of names for each article. To do this, we extract short capitalized phrases from the body text of each article, and keep phrases recognized by an algorithm to be personal names.

For every name that appears in the input, we consider that name's *timeline*, which is the multiset of dates at which that name appears, including multiple occurrences within an article. We intend the timeline to approximate the frequency with which a person browsing the news at random on a given day would encounter that name. The accuracy of this approximation will depend on the volume of news articles available. In order to avoid the possibility that any trends we detect are caused by variations in this accuracy caused by variations in the volume of the corpus, we randomly choose an approximately equal number of articles to work with from each month. We describe and analyze this process in Section 13.3.

In general, our method can be applied to any collection of timelines. In Chapter 15, we apply it to names extracted from blog posts.

13.1 Finding Periods of Fame

Once we have computed a timeline for each name that appears in the corpus, we select a time during which we consider that name to have had its period of fame, using one of the two methods described below. In order to compare the phenomenon of fame at different points in time, we consider the joint distribution of two variables over the set of names: the *peak date* and the *duration* of the name's period of fame. We try the following two methods to compute a peak date and duration for each timeline.

- **Spike method**. This method intends to capture the spike in public attention surrounding a particular news story. We divide time into one-week intervals and consider the name's rate of occurrence in each interval. The week with the highest rate is considered to be the peak date, and the period extends backward and forward in time as long as the rate does not drop below one tenth its maximum rate. Yang and Leskovec [70]

used a similar method in their study of digital media, using a time scale of hours where we use weeks.

- **Continuity method**. This method intends to measure the duration of public interest in a person. We define a name's period of popularity to be the longest span of time within which there is no seven-day period during which it is not mentioned. The peak date falls halfway between the beginning and the end of the period. We find, in Chapter 14, that durations are short compared to the time span of the study, so using any choice of peak date between the beginning and end will produce similar distributions.

To demonstrate the distinction between these two methods, Figure 13.1 shows the occurrence timeline for Marilyn Monroe. The "continuity method" picks out the bulk of her fame — 1952-02-13 ("A") through 1961-11-15 ("D"), by which point her appearance in the news was reduced to a fairly low background level. The "spike method" picks out the intense spike in interest surrounding her death, yielding the range 1962-7-18 ("E") – 1962-8-29 ("H").

Very often these two methods identify short moments of fame within a much longer context. For example, in Figure 13.1, we see the timeline for the name "John Jacob Astor", normalized by article counts. The spike method identifies as the peak the death of John Jacob Astor III of the wealthy Astor family, with a duration of 38 days (March 8 to February 15, 1890). The continuity method identifies instead the death of his nephew John Jacob Astor IV, who died on the Titanic, with a period of five months [68]. The period begins on March 23, 1912, three weeks before the Titanic sank, and ends August 31. Many of the later occurrences of the name are historical mentions of the sinking of the Titanic.

13.2 Choosing the Set of Names

13.2.1 Basic filtering

In all our experiments, to reduce noise, we discard the names which occurred less than ten times, or whose fame durations are less than two days. We also remove peaks that end in 2011 or later, since these peaks might extend further if our news corpus extended further in the future.

13.2.2 Top 1000 by year

For each peak type, we repeat our experiment with the set of names restricted in the following way. We counted the total number of times each name appeared in each year (counting repeats within an article). For each year, we produced the set of the 1000 most frequently mentioned names in that year. We took the union of these sets over all years, and ran our experiments using only the names in this set. Note that a name's peak of popularity need not be the same year in which that name was in the top 1000: so if a name is included in

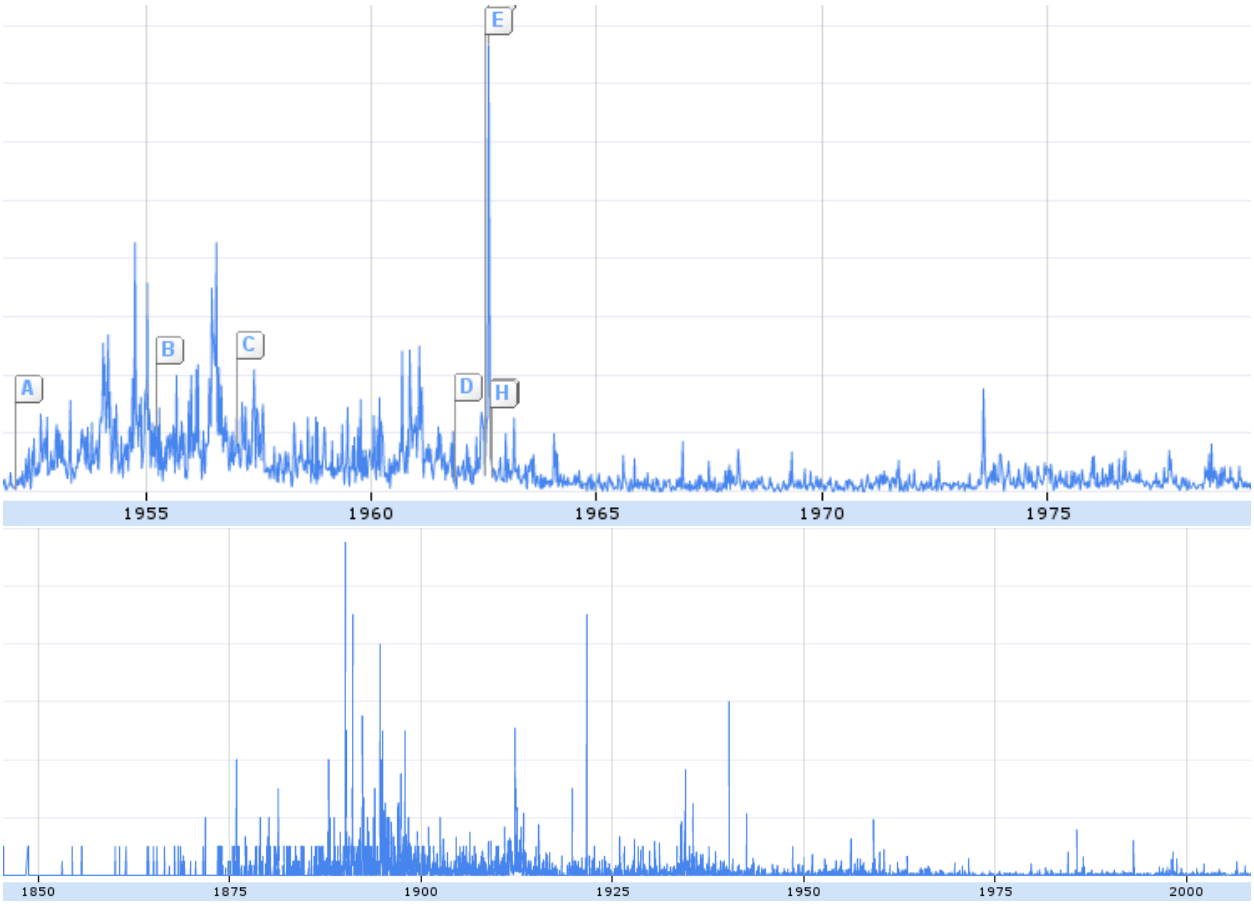

Figure 13.1: Timelines for "Marilyn Monroe" (top) and "John Jacob Astor" (bottom).

the top-1000 set because it was popular in a certain year, we may yet consider that name's peak date to be a different year.

13.2.3 Top 0.1% by year

We consider that filtering to the top 1000 names in each year might introduce the following undesirable bias. Suppose names are assigned peak durations according to some universal distribution, and later years have more names, perhaps because of the increasing volume of news. If a name's frequency of occurrence is proportional to its duration, then selecting the top 1000 names in each year will tend to produce names with longer durations of fame in years with a greater number of names. With this in mind, we considered one more restriction on the set of names. In each year y, we considered the total number of distinct names n_y mentioned in that year. We then collected the top $n_y/1000$ names in each year y. We ran our experiments using only the names in the union of those sets. As with the top-1000 filtering, a name's peak date will not necessarily be the same year for which it was in the top 0.1% of

names.

13.3 Sampling for Uniform Coverage

The spike and continuity methods for identifying periods of fame may be affected by the volume of articles available in our corpus. For example, suppose a name's timeline is generated stochastically, with every article between February 1 and March 31 containing the name with a 1% probability. If the corpus contains 10000 articles in every week, then both the spike and continuity methods will probably decide that the article's duration is two months. However, if the corpus contains less than 100 articles in each week, then the durations will tend to be short, since there will be many weeks during which the name is not mentioned.

We propose a model for this effect. Each name ν has a "true" timeline which assigns to each day t a probability $f_\nu(t) \in [0, 1]$ that an article on that day will mention ν.[1] For each day, there is a total number of articles n_t; we have no knowledge of the relation between n_t and ν, except that there is some lower bound $n_t > n_{\min}$ for all t within some reasonable range of time. Then we suppose the timeline for name ν is a sequence of independent random variables $X_{\nu,t} \sim \text{Binom}(f_\nu(t), n_t)$. Our goal is to ensure that any measurements we take are independent of the values n_t.

To accomplish this independence of news volume, we randomly sampled news articles so that the expected number in each month was $n_{\min}$. Let $X'_{\nu,t}$ be the number of sampled articles containing name ν. If we were to randomly sample $n_{\min}$ articles without replacement, then we would have $X'_{\nu,t} \sim \text{Binom}(f_\nu(t), n_{\min})$. Notice that the joint distribution of the random variables $X'_{\nu,t}$ is unaffected by the article volumes n_t. Any further measurement based on the variables $X'_{\nu,t}$ will therefore also be unrelated to the sequence n_t. In practice, instead of sampling exactly $n_{\min}$ articles without replacement, we flipped a biased coin for each of the n_t articles at time t, including each article with probability $n_{\min}/n_t$. For a large enough volume of articles, the resulting measurements will be the same.

We removed all articles published before 1895, since the months before 1895 had less than our target number $n_{\min}$ of articles. We also removed articles published after the end of the year 2010, to avoid having a month with news articles at the beginning but not the end of the month, but with the same number of sampled articles.

As an example of the effect of downsampling, the blue dotted lines in Figure 13.7 show the 50th, 90th and 99th percentiles of the distribution of fame durations using the continuity method. We see that they increase suddenly in the last ten years, when our coverage of articles surges with the digital age. The red lines show the same measurement after downsampling: the surge no longer appears.

[1] In fact, articles could mention the name multiple times, but in the limit of a large number of articles, this will not affect our analysis.

13.4 Graphing the Distributions

We graph the joint distribution of peak dates and durations in two different ways. We consider the set of names which peak in successive five-year periods. Among each set of names, we graph the 50th, 90th and 99th percentile durations of fame. These appear as darker lines in the graphs; for example, the top of Fig. 13.4 shows the distribution for the spike method. The lighter solid red lines show the same three quantiles for shorter three-month periods. For comparison, the dashed light blue lines show the same results if the article sampling described in Sec. 13.3 is not performed (and articles before 1895 and after 2010 are not removed). Fig. 13.7 shows the same set of lines using the continuity method. All the later figures are produced in the same way, except they do not include the non-sampled full distributions.

The second type of graph focuses on one five-year period at a time. The bottom of Fig. 13.4 shows a cumulative plot showing the number of names with duration greater than that shown on the x-axis. This is plotted for many five-year periods. The graphs of measurements using the spike method look more like step functions because that method measures durations in seven-day increments, whereas the longest-stretch method can yield any number of days. (Recall that peaks that last less than two days are removed.)

13.5 Estimating Power Law Exponents

We test the hypothesis that the tail of the distribution of fame durations follows a power law. For a given five-year period, we collect all names which peak in that period, and consider 20% of the names with the longest fame durations – that is, we set $d_{\min}$ to be the 80th percentile of durations, and consider durations $d > d_{\min}$. Among those 20%, we compute a maximum likelihood estimate of the power law exponent $\hat{\alpha}$, predicting that the probability of a duration $d > d_{\min}$ is $p(d) \propto d^{\hat{\alpha}}$. Clauset et al [15] show that the maximum likelihood estimate $\hat{\alpha}$ is given by $\hat{\alpha} = 1 + (\sum_{i=1}^{n} \ln(d_i/d_{\min}))$. We include a line on each plot of cumulative distributions of fame durations, of slope $\hat{\alpha}+1$ on the log-log graph because we plot cumulative distributions rather than density functions. The $\hat{\alpha}$ values we measure are discussed in the following sections, and summarized in Figure 13.2 for the news corpus and Figure 13.3 for the blog corpus.

13.6 Statistical Measurements

We used bootstrapping to estimate the uncertainty in the four statistics we measured: the 50th, 90th and 99th percentile durations and of the best-fit power law exponents. For selected five-year periods, we sampled $|S|$ names with replacement from the set S of names that peaked in that period of time. For each statistic, we repeated this process 25000 times, and reported the range of numbers within which 99% of our samples fell. The results are presented in Figures 13.2 (for the news corpus) and 13.3 (for the blog corpus).

method	filtering	period	50th %ile (days)	90th %ile (days)	99th %ile (days)	power law exponent
spike	all	1905-9	7 (7 .. 7)	28 (28 .. 28)	91 (78 .. 106)	-2.45 (-2.55 .. -2.21)
spike	all	1925-9	7 (7 .. 7)	28 (28 .. 28)	65 (63 .. 78)	-2.63 (-2.74 .. -2.33)
spike	all	1945-9	7 (7 .. 7)	21 (21 .. 28)	56 (49 .. 63)	-2.44 (-2.50 .. -2.38)
spike	all	1965-9	7 (7 .. 7)	21 (21 .. 28)	63 (56 .. 70)	-2.37 (-2.44 .. -2.31)
spike	all	1985-9	7 (7 .. 7)	21 (21 .. 28)	70 (63 .. 78)	-2.32 (-2.36 .. -2.27)
spike	all	2005-9	7 (7 .. 7)	28 (28 .. 28)	84 (78 .. 91)	-2.48 (-2.53 .. -2.43)
spike	top 1000	1905-9	21 (21 .. 21)	63 (56 .. 70)	155 (133 .. 192)	-2.75 (-3.15 .. -2.56)
spike	top 1000	1925-9	21 (14 .. 21)	49 (46 .. 56)	91 (78 .. 113)	-3.22 (-3.74 .. -2.99)
spike	top 1000	1945-9	21 (14 .. 21)	49 (42 .. 49)	91 (70 .. 130)	-3.33 (-3.73 .. -2.89)
spike	top 1000	1965-9	21 (21 .. 21)	56 (49 .. 63)	119 (99 .. 164)	-2.90 (-3.54 .. -2.65)
spike	top 1000	1985-9	21 (21 .. 28)	63 (56 .. 78)	161 (121 .. 366)	-2.85 (-3.19 .. -2.57)
spike	top 1000	2005-9	35 (28 .. 35)	99 (84 .. 119)	309 (224 .. 439)	-2.64 (-2.96 .. -2.44)
spike	top 0.1%	1905-9	35 (28 .. 42)	122 (91 .. 155)	289 (161 .. 381)	-2.82 (-3.96 .. -2.36)
spike	top 0.1%	1925-9	28 (21 .. 35)	63 (56 .. 82)	145 (91 .. 218)	-3.49 (-4.82 .. -2.92)
spike	top 0.1%	1945-9	21 (21 .. 28)	56 (49 .. 67)	133 (84 .. 161)	-3.35 (-4.32 .. -2.78)
spike	top 0.1%	1965-9	28 (21 .. 35)	70 (63 .. 99)	162 (119 .. 494)	-2.90 (-3.77 .. -2.47)
spike	top 0.1%	1985-9	35 (28 .. 35)	90 (70 .. 113)	327 (140 .. 443)	-2.66 (-3.13 .. -2.35)
spike	top 0.1%	2005-9	35 (35 .. 42)	119 (99 .. 140)	338 (263 .. 557)	-2.76 (-3.10 .. -2.44)
continuity	all	1905-9	7 (7 .. 7)	20 (19 .. 21)	70 (64 .. 79)	-2.67 (-2.76 .. -2.59)
continuity	all	1925-9	7 (7 .. 7)	18 (17 .. 19)	64 (56 .. 71)	-2.64 (-2.72 .. -2.53)
continuity	all	1945-9	7 (7 .. 7)	16 (15 .. 16)	53 (49 .. 58)	-2.74 (-2.82 .. -2.66)
continuity	all	1965-9	7 (7 .. 7)	17 (16 .. 18)	66 (58 .. 75)	-2.58 (-2.69 .. -2.52)
continuity	all	1985-9	7 (7 .. 7)	18 (17 .. 18)	77 (71 .. 83)	-2.48 (-2.56 .. -2.44)
continuity	all	2005-9	7 (7 .. 7)	21 (20 .. 21)	101 (96 .. 108)	-2.43 (-2.46 .. -2.40)
continuity	top 1000	1905-9	24 (23 .. 26)	69 (62 .. 76)	166 (136 .. 229)	-3.01 (-3.35 .. -2.70)
continuity	top 1000	1925-9	22 (21 .. 24)	58 (53 .. 66)	176 (131 .. 338)	-3.01 (-3.39 .. -2.67)
continuity	top 1000	1945-9	27 (25 .. 29)	66 (57 .. 80)	211 (169 .. 332)	-2.92 (-3.32 .. -2.59)
continuity	top 1000	1965-9	34 (32 .. 35)	92 (81 .. 104)	262 (203 .. 622)	-2.75 (-3.11 .. -2.48)
continuity	top 1000	1985-9	52 (49 .. 56)	135 (118 .. 147)	312 (231 .. 739)	-3.20 (-3.62 .. -2.83)
continuity	top 1000	2005-9	87 (80 .. 91)	229 (211 .. 250)	649 (532 .. 752)	-2.97 (-3.32 .. -2.75)
continuity	top 0.1%	1905-9	66 (59 .. 79)	146 (126 .. 176)	968 (209 .. 4287)	-3.29 (-5.20 .. -2.24)
continuity	top 0.1%	1925-9	53 (47 .. 61)	125 (104 .. 161)	476 (258 .. 2498)	-2.67 (-3.72 .. -2.20)
continuity	top 0.1%	1945-9	57 (52 .. 66)	150 (123 .. 194)	419 (218 .. 1089)	-3.19 (-4.26 .. -2.52)
continuity	top 0.1%	1965-9	69 (61 .. 79)	168 (143 .. 214)	713 (261 .. 874)	-3.01 (-4.01 .. -2.45)
continuity	top 0.1%	1985-9	85 (78 .. 94)	187 (158 .. 216)	732 (276 .. 892)	-3.40 (-4.30 .. -2.80)
continuity	top 0.1%	2005-9	113 (107 .. 119)	271 (246 .. 306)	681 (614 .. 874)	-3.16 (-3.59 .. -2.85)

Figure 13.2: Percentiles and best-fit power-law exponents for five-year periods of the news corpus. Each entry shows the estimate based on the corpus, and the 99% bootstrap interval in parentheses, as described in Section 13.6. Results discussed in Chapter 14.

method	filtering	period	50th %ile (days)	90th %ile (days)	99th %ile (days)	power law exponent
spike	all	2000-4	7 (7 .. 7)	35 (28 .. 35)	123 (84 .. 189)	-2.37 (-2.52 .. -2.23)
spike	all	2005-9	7 (7 .. 7)	28 (21 .. 28)	75 (63 .. 84)	-2.34 (-2.76 .. -2.27)
spike	top 1000	2000-4	21 (14 .. 21)	56 (49 .. 63)	265 (148 .. 479)	-2.51 (-2.83 .. -2.18)
spike	top 1000	2005-9	14 (14 .. 21)	49 (42 .. 54)	109 (91 .. 151)	-2.74 (-3.03 .. -2.41)
spike	top 0.1%	2000-4	39 (28 .. 56)	189 (106 .. 305)	717 (286 .. 840)	-2.26 (-3.05 .. -1.85)
spike	top 0.1%	2005-9	28 (25 .. 35)	88 (74 .. 102)	213 (113 .. 1674)	-3.29 (-5.40 .. -2.23)
continuity	all	2000-4	7 (7 .. 7)	22 (20 .. 23)	114 (95 .. 160)	-2.38 (-2.49 .. -2.28)
continuity	all	2005-9	6 (6 .. 7)	18 (17 .. 19)	80 (66 .. 93)	-2.62 (-2.72 .. -2.53)
continuity	top 1000	2000-4	20 (18 .. 21)	71 (59 .. 83)	387 (237 .. 819)	-2.32 (-2.54 .. -2.12)
continuity	top 1000	2005-9	21 (20 .. 22)	59 (53 .. 73)	408 (211 .. 1057)	-2.37 (-2.62 .. -2.18)
continuity	top 0.1%	2000-4	102 (89 .. 123)	372 (236 .. 768)	2010 (768 .. 2238)	-2.24 (-3.15 .. -1.86)
continuity	top 0.1%	2005-9	83 (70 .. 93)	302 (193 .. 617)	2083 (954 .. 2991)	-2.12 (-2.75 .. -1.79)

Figure 13.3: Percentiles and best-fit power-law exponents for five-year periods of the blog corpus. Each entry shows the estimate based on the corpus, and the 99% bootstrap interval in parentheses, as described in Section 13.6. Results discussed in Chapter 15.

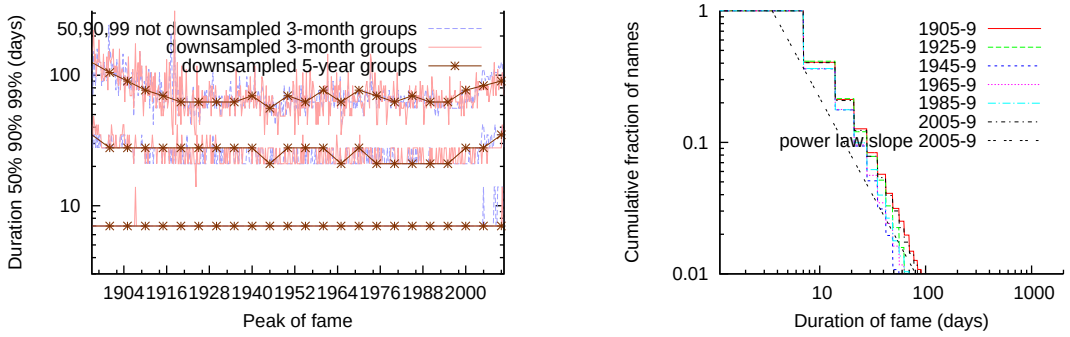

Figure 13.4: Fame durations measured using the spike method, plotted as the 50th, 90th and 99th percentiles over time (top) and for specific five-year periods (bottom). The bottom graph also includes a line showing the max-likelihood power law exponent for the years 2005-9. (The slope on the graph is one plus the exponent from Fig. 13.2, since we graph the cumulative distribution function.) To illustrate the effect of sampling for uniform article volume, the first graph includes measurements taken before sampling; see Sec. 13.3. Section 13.4 describes the format of the graphs in detail.

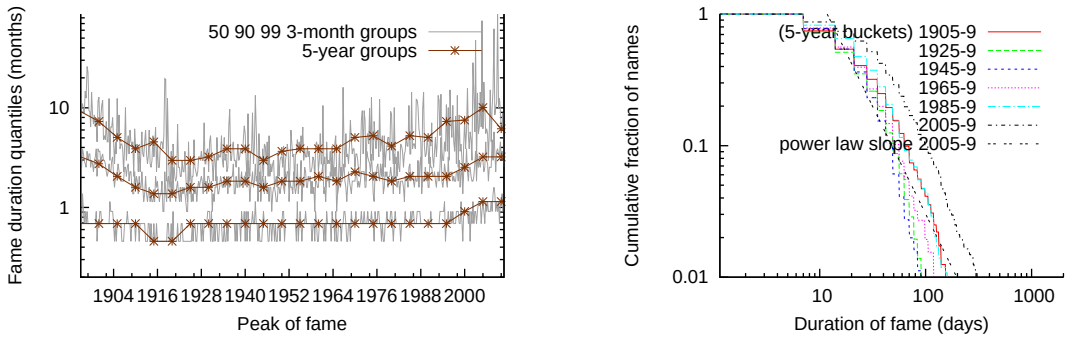

Figure 13.5: Fame durations, restricting to the union of the 1000 most-mentioned names in every year, using the spike method to identify periods of fame.

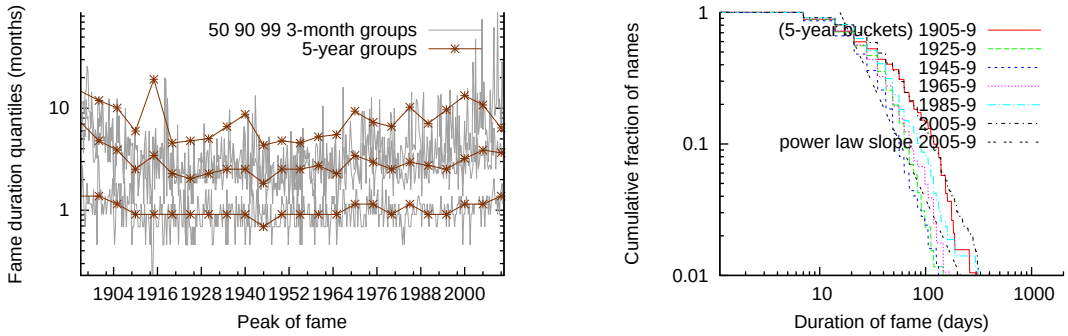

Figure 13.6: Fame durations, restricting to the union of the 0.1% most-mentioned names in every year, measured using the spike method.

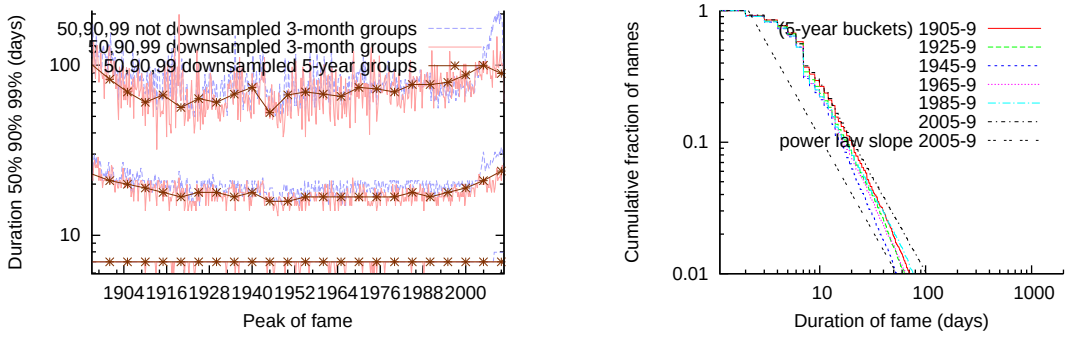

Figure 13.7: Fame durations measured using the continuity method, plotted as the 50th, 90th and 99th percentiles over time (top), and for specific five-year periods (bottom). To illustrate the effect of sampling, the first graph includes measurements taken before sampling; see Section 13.3. Section 13.4 describes the format of the graphs in detail.

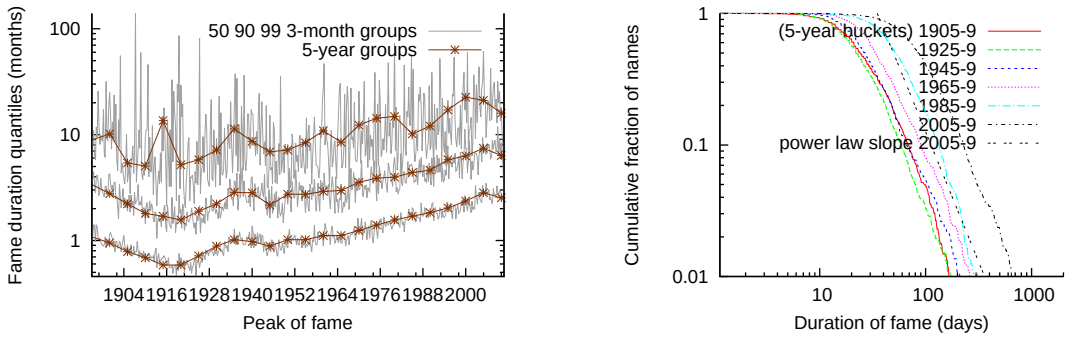

Figure 13.8: Fame durations, restricting to the union of the 1000 most-mentioned names in every year, measured using the continuity method.

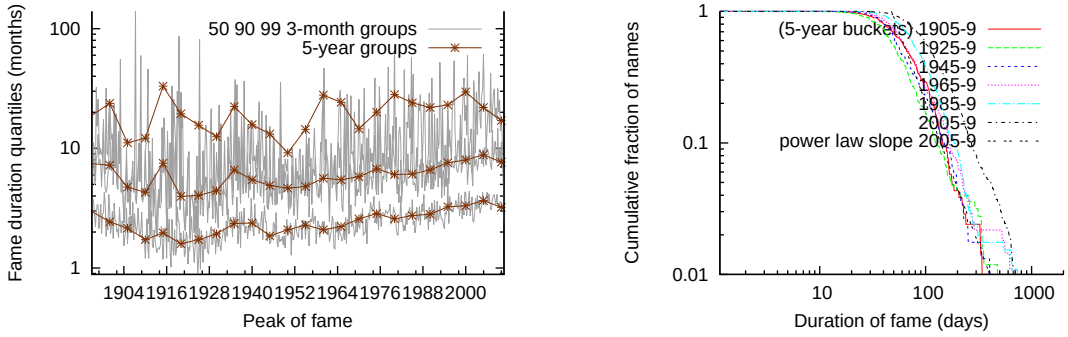

Figure 13.9: Fame durations, restricting to the union of the 0.1% most-mentioned names in every year, measured using the continuity method.

Chapter 14

Results: News Corpus

We measure periods of popularity using the spike and continuity methods described in Chapter 13, and in each case plot the distribution of duration as it changes over time.

Figures 13.4 and 13.7 show the evolution of the distribution of fame durations for the full set of names in the corpus (after the basic filtering described in Section 13.2) using the spike and continuity methods, respectively. (Section 13.4 describes the format of the graphs in detail.)

14.1 Median durations

For the entire period we studied, the median fame duration did not decrease, as we had expected, but rather remained completely constant at exactly 7 days, for both the spike and the continuity peak measurement methods. For the spike method alone, this would not have been surprising. Peaks measured by the spike method are discretized to multiples of weeks, so a perennial median of 7 days just shows that multi-week durations have never been common. On the other hand, the continuity method freely admits fame durations in increments of 1 day, with only 1-day-long peaks filtered out. Yet, the median has remained at exactly 7 days for all the years studied, and, per the full-corpus "50th percentile" measurements, shown in blue in Figure 13.2, for all decades where we've tried bootstrapping, 99% of bootstrapped samples also matched the 7-day measurement exactly (for the continuity method and, less surprisingly, for the spike method). This gives strong statistical significance to the claim that 7 days is indeed a very robust measurement of typical fame duration, which has not varied in a century.

14.2 The most famous

We next consider specially the fame durations of the most famous names, in two correlated, but distinct senses of "most famous":

- "Duration outliers" — people whose **fame lasts much longer than typical**, as measured by the 90th and 99th percentiles of fame durations within each year. These correspond to the top two lines in the timelines of Figures 13.4 and 13.7, and the columns "90 %ile" and "99 %ile" of the first and fourth blocks of Figure 13.2.

- "Volume outliers" – the names **which appear the most frequently** in the news, by being either in the top 1000 most frequent names in some year, or, separately, names in the top 0.1%, as per Section 13.2. The graphs for these subsets of names are shown in Figures 13.5 and 13.6 for the spike method, and Figures 13.8 and 13.9 for the continuity method, and the statistical measurements appear in blocks 2, 3, 5 and 6 of Figure 13.2.

From the 1900's to the 1940's, the fame durations in both categories of outliers do tend to decrease, with the decreases across that time interval statistically significantly lower-bounded by 1-2 weeks via 99% bootstrapping intervals. Heuristically, this seems consistent with our original hypothesis that accelerating communications shorten fame durations: 1-2 weeks is a reasonable delay to be incurred by sheer communications delay before the omnipresence of telegraphy and telephony. We note with curiosity that this effect applies only to the highly-famous outliers rather than the typical fame durations. We posit that this is perhaps due to median fame durations being typically attributable to people with only geographically localized fame, which does not get affected by long communication delays. We leave to further work a more nuanced study to test these hypotheses around locality and communication delays affecting news spread in the early 20th century.

After the 1940's, on the other hand, we see no such decrease. On the contrary, the durations of fame for both the duration outliers and the volume outliers reverse the trend, and actually begin to slowly increase. Using the bootstrapping method, per Section 13.6, we get the results marked in red in Figure 13.2: in almost all of the outlier studies[1], we see that the increase in durations is statistically significant over 40-year gaps for both categories of fame outliers. For example, the median fame duration according to continuity peaks for the top 1000 names (50th percentile column of row 5) appears as "27 (25 .. 29)" in the period 1945-9 and "52 (49 .. 56)" for the period 1985-9: with 99% confidence, the median duration was less than 29 days in the former period, but greater than 49 days in the latter.

We also ran experiments for names that have outlier durations *within* the subset of names with outlier volumes. The same general trends were seen there as with the above outlier studies, but, with a far shallower pool of data, the bootstrapping-based error bars were generally large enough to not paint a convincing, statistically significant picture.

[1] 7 out of the 8 outlier studies show statistically significant increases between the 1940's and the 1980's, and between the 1960's and the 2000's. The sole exception is the 90th percentile of the spike method. Given that the bootstrap values in that experiment, discretized to whole weeks, range between 3 and 4 weeks, we don't consider it surprising that the increases there were not measured to be significant by 99% bootstrap intervals.

14.3 Power law fits

The column titled "power law exponent" in Figure 13.2 shows the maximum likelihood estimates of the power law exponents for various five-year-long peak periods. We focus on rows 1 and 4, which show the estimates for the full set of names for the spike method and the continuity method respectively.

For both peak methods, the fitted power law exponents remain in fairly small ranges — between -2.77 and -2.45 for continuity peaks, and between -2.63 and -2.32 for spike peaks. In Figures 13.7 and 13.4 we show the actual distributions, and, for reference, comparisons with the power-law fit for the 2005-2009 data (a straight line on these log-log plots).

Furthermore, the continuity peaks fits also support the above observation of slowly-growing long-tail fame durations from 1940 onward. That is, power-law exponents from 1940 onward slowly move toward zero, with statistically significant changes when compared at 40-year intervals. The fluctuations and the error bars for both methods are rather noticeable, though, suggesting that power laws make for only a mediocre fit to this data.

Chapter 15

Results: Blog Posts

We also ran our experiments on a second set of data consisting of public English-language blog posts from the Blogger service. We began by sampling so that the number of blog posts in each month in our data set was equal to the number of news articles we sampled in each month, as per Sec. 13.3. The cumulative graphs of fame duration from six experiments are shown in Fig. 15.1. We combine the two methods for identifying periods of fame with three sets of names described in Section 13.2. The respective distributions from the news corpus are superimposed for comparison.

The graphs of fame duration measured using the continuity method are much smoother for the blog corpus than for the news corpus. This happens because whereas we only know which day each news article was written, we know the time of day each blog entry was posted.

The continuity-method graphs (bottom of Figure 15.1) had a distinctive rounded cap which surprised us at first. We believe it is caused by the following effect. Peaks with only two mentions in them are fairly common, and have a simple distinctive distribution that is the difference between two sample dates conditioned on being less than a week apart. Since two dates that are longer than one week apart cannot constitute a longest-stretch peak, the portion of the graph with durations longer than one week does not include any names from this two-sample distribution, and so it looks different. Our estimates of power-law exponents only consider the longest 20% of durations, so they ignore this part of the graph.

The estimates we computed for the power-law exponents of the duration distributions for blog data are shown in Figure 13.3, and can be compared to the figures for news articles in Figure 13.2.

The medians for both blogs and news for both methods are remarkably the same, with no statistically significant differences. The power law fits are also quite similar, although they show enough variation to produce statistically significant differences. Qualitatively, we take these as evidence that the fame distributions in news and blogs are coarsely similar, and that it is not unreasonable to consider these results as casting some light on more fundamental aspects of human attention to and interest in celebrities, rather than just on the quirks of the news business.

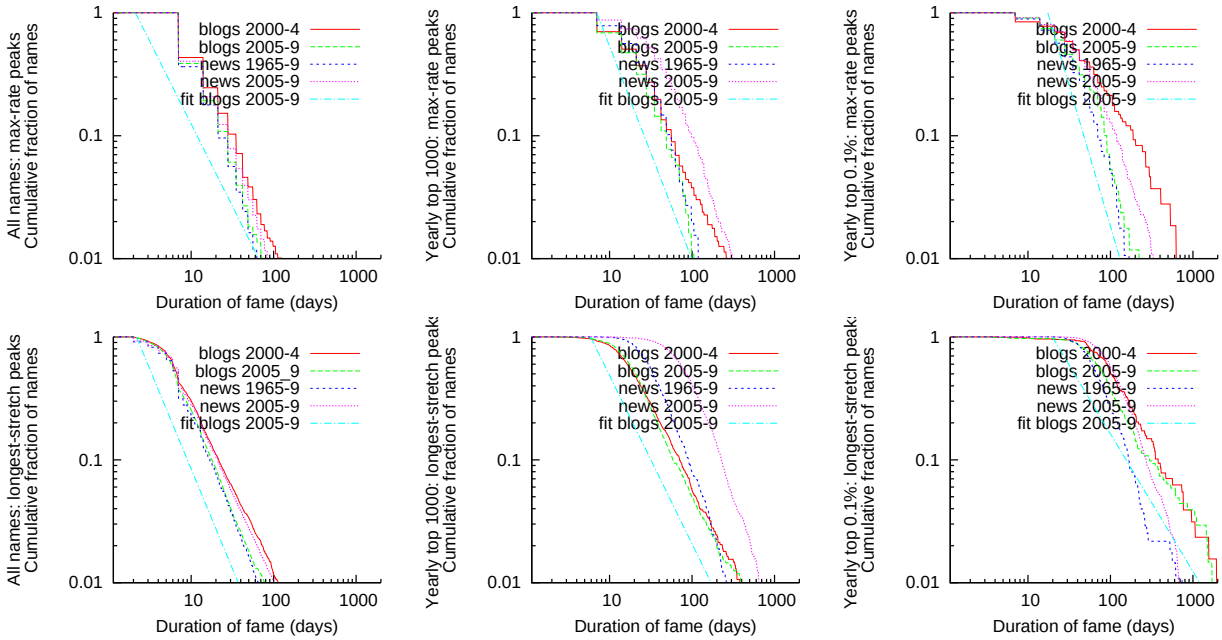

Figure 15.1: Cumulative duration-of-fame graphs for the blog corpus. The graphs at the top show the spike method results (for all names, top 1000, and top 0.1%), and those at the bottom show the continuity method results.

We do leave open the question of accounting for the occasionally significant distinctions between outlier results for blogs, as compared to news, especially for outlier-volume continuity peaks.

Part IV

Conclusion and Future Work

Chapter 16

Conclusion

We performed two studies related to changing social trends. We studied the phenomenon of fame in newspapers, inspired by popular intuition that shorter news cycles are fuelling a shortening of attention spans, but we found no evidence of this: to the contrary, the typical length of time for which a person's name appears in a newspaper has not changed in a century, and among the most famous names, the duration of news coverage has increased. We found similar results in a side study of ten years of blog posts, hinting at the existence of a more general phenomenon.

We also studied a new kind of community on Twitter called a group chat. We developed an algorithm to generate a list of group chats on Twitter, and provided theoretical arguments for its effectiveness. Using this algorithm, found that the phenomenon of group chats has grown over time, and we investigated the typical topics discussed. We developed a second algorithm which searches a broader set of Twitter "chats" and ranks them for relevance to a topic given as a query. This algorithm is based on a new model of user browsing relevant discussion groups and their participants, and we provide mathematical arguments that it has desirable properties. We found experimentally that our ranking algorithm performs noticeably better than natural baselines, and provided some theoretical justification for its We hope that in the future it will prove useful for introducing users to discussion groups that are relevant to their interests but which they would not otherwise have been aware of.

Chapter 17

Future Work

17.1 Twitter Group Chats

17.1.1 The Nature of Group Chats

There has been much work devoted to understanding how groups are formed and die (§2.1.3) and why people participate in them (§2.1.4). It would be good to gain such an understanding of Twitter group chats. To understand how group chats are created, a good starting point could be to simply ask the founders and moderators, since they have public personas. Ceren and Rakesh [10] have done work in understanding participation in group chats related to education. We believe that users attend group chats for a variety of reasons: for example, to speak, to listen, to learn, or to be viewed as subject matter experts. Understanding what drives participation as a function of both the type of group, as well as the role (e.g. leader, information-seeker, sympathizer) of the user in the group is an interesting direction.

The value that these chats bring to the individuals that participate as well as to the community as a whole is not well-understood. We believe that the benefit that users derive cannot be found without the group. For example, the information that users learn from passion-oriented groups may be hard to find without the group. Similarly, the support that a user receives from a support group may be hard to find without the group. However, it is not clear why Twitter, with its 140-character limit on message text, is the chosen platform.

Some of the users who participate in these chats are quite knowledgeable about the subject matter they are discussing. Understanding and quantifying their level of expertise is a promising direction for future work. In addition, it is useful to find ways to summarize a meeting of a group chat [19] in a manner that takes advantage of the structure of a typical conversation: this could be used to help a new user decide whether to a join a group, or help an existing user to catch up on a recent missed meeting.

During our analysis of Twitter Group Chats, we attempted to recruit workers on Amazon's Mechanical Turk (`www.mturk.com`) to determine whether hashtags were group chats. We were not able to get useful answers this way, possibly because the workers did not have appropriate resources at hand, or because the questions were not clear. In the past, clever

ways of coordinating the work of many workers have produced results far beyond what could be done by asking a single worker directly to do what is asked. For example, Bernstein et al. [7] created a text editor enhancement which coordinates workers in order to proofread a document and suggest ways to make it shorter, if desired. It would be interesting to see if a more careful approach could allow Mechanical Turk workers to produce useful data about Twitter group chats.

17.1.2 Ranking

Our goal in ranking groups is to connect a new user who has an interest in a topic to a group where that topic is regularly discussed. However, group selection is a more complex task. 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 many other potential factors: 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). Such factors are known to influence membership in a group [33]. 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 [21] 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 signal such as the number of friends that a person has in the group and how connected their friends are could also be useful [3].

Finally, different types of query may call for different types of groups. For example, a user seeking an online health support group may desire a group with a history of exchange, interaction and sharing of medical experiences [36]; these factors may be less important to a user looking for help promoting a local business. 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 suggest users seeking groups for humour or entertainment, and this could be yet another factor that improves ranking.

17.2 Fame in News

We feel that our study of fame in newspapers has barely scratched the surface of what is possible. For example, instead of personal names, the object of study could be news stories, clothing fashions or topics in a field of research. Instead of measuring changes in attention

span across time, it would be interesting to measure changes across age groups, geography or level of education.

News companies are generally based in particular locations, but we did not use location data at all in our study. For example, we could ask whether communication across long distances was a factor behind the duration of news stories. It would be interesting if the proximity to telegraph lines could be inferred from the delay before a particular news outlet publishes a story. More generally, more careful analysis and modelling of factors such as locations and availability of news sources in our corpus might shed more light on the factors behind our observations.

Bibliography

[1] E. Adar and L. A. Adamic. Tracking information epidemics in blogspace. WI 2005, pages 207–214.

[2] Holly Arrow, Joseph E McGrath, and Jennifer L Berdahl. *Small groups as complex systems: Formation, coordination, development, and adaptation*. Sage Publications, 2000.

[3] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group formation in large social networks: Membership, growth, and evolution. In *KDD*, pages 44–54, 2006.

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

[5] P. Bateman, P.H. Gray, and B.S. Butler. Community commitment: How affect, obligation, and necessity drive online behaviors. In *Twenty-Seventh International Conference on Information Systems*, 2006.

[6] P. Bateman, P.H. Gray, and B.S. Butler. The impact of community commitment on participation in online communities. *Information Systems Research*, 22(4):841–854, 2011.

[7] Michael S Bernstein, Greg Little, Robert C Miller, Björn Hartmann, Mark S Ackerman, David R Karger, David Crowell, and Katrina Panovich. Soylent: a word processor with a crowd inside. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 313–322. ACM, 2010.

[8] P.B. Brandtzæg and J. Heim. User loyalty and online communities: Why members of online communities are not faithful. In *INTETAIN*, 2008.

[9] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30:107–117, 1998.

[10] Ceren Budak and Rakesh Agrawal. Participation in group chats on Twitter. In *WWW*, 2013.

[11] Brian S. Butler. Membership size, communication activity, and sustainability: A resource-based model of online social structures. *Info. Sys. Research*, 12(4):346–362, December 2001.

[12] Dorwin Cartwright and Alvin Zander. *Group dynamics: Theory and research*. New York: Harper & Row, 1968.

[13] R. Chaiken, B. Jenkins, P.Å. Larson, B. Ramsey, D. Shakib, S. Weaver, and J. Zhou. SCOPE: Easy and efficient parallel processing of massive data sets. *Proceedings of the VLDB Endowment*, 1(2):1265–1276, 2008.

[14] Steve Chien, Cynthia Dwork, Ravi Kumar, Daniel R. Simon, and D. Sivakumar. Link evolution: Analysis and algorithms. *Internet Mathematics*, 1(3):277–304, 2003.

[15] A. Clauset, C. R. Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703, 2009.

[16] J.M. Dabbs Jr. Fourier analysis and the rhythm of conversation. ERIC, 1982. Paper presented at the Annual Meeting of the American Psychological Association (Washington, DC, August 23-27, 1982).

[17] Cristian Danescu-Niculescu-Mizil, Robert West, Dan Jurafsky, Jure Leskovec, and Christopher Potts. No country for old members: User lifecycle and linguistic change in online communities. In *WWW '13*, pages 307–318, 2013.

[18] M.R. Davis. Social networking goes to school. *Education Digest*, 76(3):14, 2010.

[19] YaJuan Duan, ZhuMin Chen, FuRu Wei, Ming Zhou, and Heung Yeung Shum. Twitter topic summarization by ranking tweets using social influence and content quality. In *COLING*, pages 763–780, 2012.

[20] Kate Ehrlich, Ching-Yung Lin, and Vicky Griffiths-Fisher. Searching for experts in the enterprise: Combining text and social network analysis. In *GROUP*, pages 117–126, 2007.

[21] D.R. Forsyth. *Group dynamics*. Wadsworth Publishing Company, 2009.

[22] S. Fox. Peer-to-peer healthcare. *Pew Internet & American Life Project*, Feb. 2011. `http://www.pewinternet.org/Reports/2011/P2PHealthcare.aspx`, accessed on April 26, 2012.

[23] E. Gabrilovich, S. Dumais, and E. Horvitz. Newsjunkie: providing personalized newsfeeds via analysis of information novelty. WWW 2004, pages 482–490.

[24] Robert Gallager. 6.262 Discrete stochastic processes, Spring 2011. (Massachusetts Institute of Technology: MIT OpenCourseWare) http://ocw.mit.edu (Accessed 23 Mar, 2013). License: Creative Commons BY-NC-SA.

[25] Fréderic Godin, Viktor Slavkovikj, Wesley De Neve, Benjamin Schrauwen, and Rik Van de Walle. Using topic models for Twitter hashtag recommendation. In *WWW (Companion Volume)*, pages 593–596, 2013.

[26] J.M. Gottman. Detecting cyclicity in social interaction. *Psychological Bulletin*, 86(2):338, 1979.

[27] J.M. Gottman and L.J. Krokoff. Marital interaction and satisfaction: a longitudinal view. *Journal of consulting and clinical psychology*, 57(1):47, 1989.

[28] John D Greenwood. *The disappearance of the social in American social psychology*. Cambridge University Press, 2003.

[29] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. WWW 2004, pages 491–501.

[30] M. Herbert. Why all the chatter about# edchat?. *District Administration*, 48(4):n4, 2012.

[31] J. Horrigan. Online communities. *Pew Internet & American Life Project*, Oct. 2001. `http://www.pewinternet.org/Reports/2001/Online-Communities.aspx`, accessed November 26, 2012.

[32] Samuel Ieong, Nina Mishra, and Or Sheffet. Predicting preference flips in commerce search. In *ICML*, 2012.

[33] A. Iriberri and G. Leroy. A life-cycle perspective on online community success. *ACM Computing Surveys (CSUR)*, 41(2):11, 2009.

[34] Glen Jeh and Jennifer Widom. Scaling personalized web search. In *WWW*, pages 271–279, 2003.

[35] Jian Jiao, Jun Yan, Haibei Zhao, and Weiguo Fan. Expertrank: An expert user ranking algorithm in online communities. In *NISS*, pages 674–679, 2009.

[36] G.J. Johnson and P.J. Ambrose. Neo-tribes: The power and potential of online communities in health care. *Communications of the ACM*, 49(1):107–113, 2006.

[37] S. Johnson. Should I stay or should I go? Continued participation intentions in online communities. In *Proceedings of Academy of Management Annual Conference*, 2010.

[38] Quentin Jones, Gilad Ravid, and Sheizaf Rafaeli. Information overload and the message dynamics of online interaction spaces: A theoretical model and empirical exploration. *Info. Sys. Research*, 15(2):194–210, June 2004.

[39] E. Joyce and R.E. Kraut. Predicting continued participation in newsgroups. *Journal of Computer-Mediated Communication*, 11(3):723–747, 2006.

[40] Henry Kautz, Bart Selman, and Mehul Shah. Referral web: Combining social networks and collaborative filtering. *Commun. ACM*, 40(3):63–65, 1997.

[41] J. Kleinberg. Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery*, 7(4):373–397, 2003.

[42] Jon Kleinberg. Authoritative sources in a hyperlinked environment. In *SODA*, 1998.

[43] Gueorgi Kossinets and Duncan J Watts. Empirical analysis of an evolving social network. *Science*, 311(5757):88–90, 2006.

[44] Su Mon Kywe, Tuan-Anh Hoang, Ee-Peng Lim, and Feida Zhu. On recommending hashtags in Twitter networks. In *SocInfo*, pages 337–350, 2012.

[45] Cliff Lampe and Erik Johnston. Follow the (slash) dot: Effects of feedback on new members in an online community. In *GROUP*, pages 11–20, 2005.

[46] K. Leetaru. Culturomics 2.0: Forecasting large-scale human behavior using global news media tone in time and space. *First Monday*, 16(9-5), 2011.

[47] J. Leskovec, L. Backstrom, and J. Kleinberg. Meme-tracking and the dynamics of the news cycle. KDD 2009, pages 497–506.

[48] M Lynne Markus. Toward a "critical mass" theory of interactive media universal access, interdependence and diffusion. *Communication research*, 14(5):491–511, 1987.

[49] J.E. McGrath and D.A. Kravitz. Group research. *Annual Review of Psychology*, 33(1):195–230, 1982.

[50] J-B. Michel, Y. K. Shen, A. P. Aiden, A. Veres, M. K. Gray, The Google Books Team, J. P. Pickett, D. Hoiberg, D. Clancy, P. Norvig, J. Orwant, S. Pinker, M. A. Nowak, and E. L. Aiden. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182, 2011.

[51] C. Packer, D. Scheel, and A.E. Pusey. Why lions form groups: food is not enough. *American Naturalist*, pages 1–19, 1990.

[52] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. *Technical report, Stanford University, Stanford, CA*, 1998.

[53] Götz E Pfander and John J Benedetto. Periodic wavelet transforms and periodicity detection. *SIAM Journal on Applied Mathematics*, 62(4):1329–1368, 2002.

[54] Pingdom. In-depth study of Twitter: How much we tweet, and when. `http://royal.pingdom.com/2009/11/13/in-depth-study-of-twitter-how-much-we-tweet-and-when`, November 2009. Accessed March 17, 2013.

[55] R.D. Putnam. *Bowling alone: The collapse and revival of American community*. Simon & Schuster, 2001.

[56] Tim Reichling, Kai Schubert, and Volker Wulf. Matching human actors based on their texts: Design and evaluation of an instance of the expertfinding framework. In *GROUP*, pages 61–70, 2005.

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

[58] H. Rheingold. *The virtual community: Homesteading on the electronic frontier*. MIT press, 2000.

[59] Sheldon M. Ross. *Simulation*. Academic Press, fourth edition, 2006.

[60] Tetsuya Sakai. On the reliability of information retrieval metrics based on graded relevance. *Inform. Process. Manag*, pages 531–548, 2007.

[61] H. A. Simon. Designing organizations for an information-rich world. 1971.

[62] G. Szabo and B. A. Huberman. Predicting the popularity of online content. *Commun. ACM*, 53:80–88, August 2010.

[63] Amos Tversky. Elimination by aspects: A theory of choice. *Psychological review*, 79(4):281, 1972.

[64] Twitter chat schedule. `https://docs.google.com/spreadsheet/ccc?key=0AhisaMy5TGiwcnVhejNHWnZlT3NvWFVPT3Q4NkIzQVE` accessed July 4, 2012.

[65] M. Vlachos, P. Yu, and V. Castelli. On periodicity detection and structural periodic similarity. In *SDM*, 2005.

[66] Ellen M Voorhees, Donna K Harman, et al. *TREC: Experiment and evaluation in information retrieval*, volume 63. MIT press Cambridge, 2005.

[67] Steve Whittaker, Loen Terveen, Will Hill, and Lynn Cherny. The dynamics of mass interaction. In *From Usenet to CoWebs*, Computer Supported Cooperative Work, pages 79–91. Springer London, 2003.

[68] Wikipedia. Astor family — Wikipedia, the free encyclopedia, 2011. [Online; accessed 10-August-2011].

[69] Wikipedia. Poisson distribution, 2013. [Online; accessed 28-March-2013].

[70] J. Yang and J. Leskovec. Patterns of temporal variation in online media. WSDM 2011, pages 177–186.

[71] Lei Yang, Tao Sun, Ming Zhang, and Qiaozhu Mei. We know what @you #tag: Does the dual role affect hashtag adoption? In *WWW*, pages 261–270, 2012.

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