MONITORANDO INFORMAÇÕES NAS MÍDIAS SOCIAIS ONLINE

TIAGO RODRIGUES DE MAGALHÃES

MONITORANDO INFORMAÇÕES NAS MÍDIAS SOCIAIS ONLINE

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Orientador: Virgílio Augusto Fernandes de Almeida
Co-Orientador: Ponnurangam Kumaraguru

Belo Horizonte
Junho de 2012

TIAGO RODRIGUES DE MAGALHÃES

MONITORING INFORMATION IN THE ONLINE SOCIAL MEDIA SPHERE

Dissertation presented to the Graduate Program in Ciência da Computação of the Universidade Federal de Minas Gerais in partial fulfillment of the requirements for the degree of Master in Ciência da Computação.

Advisor: Virgílio Augusto Fernandes de Almeida
Co-Advisor: Ponnurangam Kumaraguru

Belo Horizonte
June 2012

Magalhães, Tiago Rodrigues de
M188m Monitoring Information in the Online Social Media Sphere / Tiago Rodrigues de Magalhães. — Belo Horizonte, 2012
xxvi, 70 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de Minas Gerais
Orientador: Virgílio Augusto Fernandes de Almeida

Co-Orientador: Ponnurangam Kumaraguru

1. information diffusion. 2. online social media. 3. tracking. I. Título.

CDU 519.6*04(043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Monitorando informações nas mídias sociais online

TIAGO RODRIGUES DE MAGALHÃES

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

Prof. Virgílio Augusto Fernandes Almeida - Orientador
Departamento de Ciência da Computação - UFMG

Prof. Ponnurangam Kumaraguru - Co-orientador
Departamento de Ciência da Computação - IIITD

Prof. Adriano Alonso Veloso
Departamento de Ciência da Computação - UFMG

Fabrício Benevenuto
Prof. Fabrício Benevenuto de Souza
Departamento de Computação - UFOP

Wmeira Jr
Prof. Wagner Meira Júnior
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 01 de junho de 2012.

Dedico este trabalho a todos os leitores. Espero que seja útil e contribua para o avanço da ciência e para o desenvolvimento de novas tecnologias e serviços, facilitando a vida de algumas pessoas.

Acknowledgments

Gostaria de agradecer a todos que me ajudaram durante estes 2 anos no mestrado. A todos os funcionários das instituições que visitei (Max Planck Institute for Software Systems - Saarbrücken - Alemanha, Indraprastha Institute of Information Technology - Delhi - India, e Indian Institute of Science - Bangalore, India). Aos responsáveis diretos por idealizarem estas visitas (orientadores, funcionários, secretaria do DCC-UFMG, pesquisadores com quem trabalhei), pois sem eles nada teria acontecido. Foram momentos e experiências inesquecíveis em lugares que jamais imaginei estar algum dia. A todos os amigos do CAMPS, pelos bons momentos passados e pelo aprendizado adquirido desde a iniciação científica, no segundo semestre de 2007. A todos com quem trabalhei diretamente durante o mestrado (orientadores, co-autores), pelo apoio, aprendizado e contribuição durante todo o trabalho desenvolvido. A toda minha família pelo apoio, mesmo que alguns não fizessem idéia do que estava estudando ou porque passei mais 2 anos na universidade depois de ter me formado. Não poderia me esquecer daqueles que me deram suporte financeiro: pais, CAPES, MPI-SWS.

Preferi não citar nomes para não esquecer de ninguém e cometer injustiça, mas tenho certeza de que quem leu esta seção e se identificou teve um papel importante na minha jornada.

"O caminho batido não leva a novas pastagens"
(Indira Ghandi)

Resumo

As mídias sociais online (blogs, redes sociais) são ferramentas de comunicação cada vez mais presentes e importantes no mundo moderno. Tradicionalmente, as pessoas encontram informações na Web navegando ou buscando. Com o recente sucesso das mídias sociais online, as pessoas passaram a receber novas informações através de conversas com seus amigos. Uma quantidade enorme e diversa de informação é gerada diariamente por milhões de pessoas em todo o mundo. Por exemplo, através do Twitter os usuários compartilham e repassam diversas informações; pelo Facebook os usuários ficam sabendo sobre o que os amigos fizeram no último fim de semana; e pelo Foursquare descobrem onde estiveram recentemente. O grande volume de dados, a diversidade de informação, a dinâmica dos sistemas, e a necessidade de informações em tempo real levam a grandes desafios para se processar e analisar os dados gerados pelos usuários. Entretanto, um melhor entendimento sobre o processo de difusão de informação nas mídias sociais online é necessário, não apenas para os usuários, mas também para empresas e pesquisadores. Exploração de propagandas, melhor organização de conteúdo e criação de novas ferramentas são exemplos de benefícios possíveis através de um melhor estudo do problema. Neste trabalho propomos o MIOSphere, um sistema para monitoramento, em tempo real, da propagação de informação por importantes mídias sociais online. Além disso, foi feita uma ampla caracterização da propagação de URLs no Twitter, analisando não somente a propagação interna, mas também como URLs geradas em outras mídias sociais (vídeos do YouTube, por exemplo) se propagam no Twitter.

Palavras-chave: mídias sociais online, difusão de informacão, boca-a-boca, monitoramento.

Abstract

Online Social Media services (blogs, social networks) are important communication tools in the modern world. Traditionally, people find information in the Web by browsing or searching. With the recent success of Online Social Media, people started to receive information from conversations with their friends. A huge and diverse amount of information is generated everyday by millions of people all over the world. As an example, on Twitter users share and forward various information; on Facebook, users get to know about what their friends did on the last weekend; and on Fourquare users discover where their friends visited recently. The huge volume of data, the diversity of information, the dynamic of the systems, and the need for real-time information leads to big challenges while processing and analyzing the user generated data from Online Social Media services. However, a better understanding about the information diffusion process in Online Social Media services is necessary, not only for users, but also for companies and researchers. Advertising, better content organization, and creation of new tools are examples of possible benefits from a better understanding of the problem. In this work we propose MIOSphere, a system to monitor, in real-time, the diffusion of information through popular Online Social Media services. Furthermore, a wide characterization of the diffusion of URLs on Twitter was done, analyzing not only the internal diffusion, but also how URLs generated in external services (YouTube videos, for example) diffuses on Twitter.

Keywords: online social media, information diffusion, word-of-mouth, monitoring.

List of Figures

1.1 High level view of MIOSphere's architecture 3

2.1 Basics: dynamics of Online Social Media Sphere 6

2.2 Basics: modeling information cascades terminology 8

4.1 Word-of-mouth: URLs on Twitter over time 15

4.2 Word-of-mouth: popular domains 18

4.3 Word-of-mouth: audience size 19

4.4 Word-of-mouth: the size of word-of-mouth 22

4.5 Word-of-mouth: the number of initiators per URL 23

4.6 Word-of-mouth: role of initiators 24

4.7 Word-of-mouth: the shape of word-of-mouth 25

4.8 Word-of-mouth: cascade size distributions 26

4.9 Word-of-mouth: size per content type 27

4.10 Word-of-mouth: physical distance of friendship and propagation links . . . 28

4.11 Word-of-mouth: locality across content types 29

4.12 Word-of-mouth: distance between initiators and spreaders 30

5.1 Cross-Pollination: data collection framework 32

5.2 Cross-Pollination: meme creation over time 34

5.3 Cross-Pollination: user participation 35

5.4 Cross-Pollination: diffusion delay 36

5.5 Cross-Pollination: cascade size 36

6.1 MIOSphere: architecture and workflow 42

6.2 MIOSphere: tweets collected versus total number of tweets 51

6.3 MIOSphere: snapshot of interface - home page 55

6.4 MIOSphere: snapshot of interface - memes page 55

6.5 MIOSphere: snapshot of interface - meme detail page 56

6.6 MIOSphere: snapshot of interface - about page 57
6.7 MIOSphere: top 5 URLs collected 58

List of Tables

4.1 Word-of-mouth: statistics of the datasets 15

4.2 Word-of-mouth: top 10 URL shortening services 16

4.3 Word-of-mouth: top 5 domains 17

4.4 Word-of-mouth: categories of URLs 20

4.5 Word-of-mouth: top 10 URLs 21

5.1 Cross-Pollination: statistics of the datasets 33

5.2 Cross-Pollination: information cascade statistics 37

6.1 MIOSphere: traces for the simulations 46

6.2 MIOSphere: dead threshold results for MPQ scheduler 50

6.3 MIOSphere: dead threshold results for baseline scheduler 50

6.4 MIOSphere: waiting time queue results 51

6.5 MIOSphere: importance sliding window size results 52

6.6 MIOSphere: number of priority queues and thresholds results 53

Contents

Acknowledgments **xi**

Resumo **xv**

Abstract **xvii**

List of Figures **xix**

List of Tables **xxi**

1 Introduction **1**

1.1 Context 2

1.2 Purpose 2

1.3 Solution 3

1.4 Organization 4

2 Basics of Online Social Media and Definitions **5**

2.1 Online Social Media Sphere 5

2.1.1 Cross-Pollination of Information 6

2.2 Modeling Information Cascades 7

2.2.1 Hierarchical Tree Model 7

3 Literature Review **9**

3.1 Information Diffusion in OSM Services 9

3.2 Information Diffusion Across OSM Services 10

3.3 Tools and Applications for OSM Services 11

3.4 Information Retrieval 12

4 On Word-of-Mouth Based Discovery of the Web **13**

4.1 Methodology 13

4.1.1 The Twitter Dataset 13
4.1.2 URLs in Tweets 14
4.2 Which Content is Popular? 15
4.2.1 URL Shortening Services 16
4.2.2 Popularly Linked Web Domains 17
4.2.3 Popular Individual URLs 18
4.2.4 Content Types 19
4.3 The Shape of Word-of-mouth 21
4.3.1 How Large is the Largest Word-of-mouth? 21
4.3.2 The Role of Initiators 22
4.3.3 The Shape of Word-of-mouth 24
4.4 Content Distribution 26
4.4.1 Content Producers and Consumers 27

5 Cross-Pollination of Information in Online Social Media: a Case Study on Popular Social Networks 31

5.1 Methodology 31
5.1.1 Data Collection 31
5.1.2 Datasets 32
5.2 Cross-Pollination Characteristics 33
5.2.1 Sharing Activity 33
5.2.2 User Participation 34
5.2.3 Diffusion Delay 34
5.2.4 Information Cascades 35
5.3 Relationship Between OSM Services in a Cross-Pollinated Network . . 37

6 Developing a System to Monitor Information in Online Social Media 39

6.1 Lessons Learned 39
6.2 Architecture 41
6.2.1 Scheduler 43
6.2.2 Algorithm 44
6.3 Simulations 46
6.3.1 Traces 46
6.3.2 Assumptions 47
6.3.3 Evaluation Metrics 47
6.3.4 Parameters 48
6.3.5 Analysis 49

6.3.6 Discussion 53

6.4 Prototype 54

6.4.1 Collected Data 57

7 Conclusions and Future Work 61

7.1 Main Contributions 61

7.2 Future Work 62

7.3 Limitations 62

Bibliography 63

Chapter 1

Introduction

Recently, Online Social Media (OSM) services (i.e., blogs, social networks) have emerged as a popular way of discovering information on the World Wide Web. In contrast to traditional methods of content discovery such as browsing or searching, content sharing in OSM services occurs via conversations between users, which is known as *word-of-mouth*. For instance, users share links with personal recommendations like "Check out this great photo of the concert last night!"

While such word-of-mouth based content discovery existed long before in the form of emails and web forums, OSM services have made this phenomenon extremely popular and globally reaching. In fact, today social networking sites are known to be a major driver of traffic to many Web sites [Campbell, 2009]. For certain Web sites, Facebook and Twitter drive, respectively, 44% and 29% of the traffic [Schonfeld, 2010]. These OSM sites are sharing tens of millions of Web links every day, and the amount of information exchanged by word-of-mouth in OSM services is expected to grow over time [Rao, 2010].

With users having accounts in different OSM services, there is a tendency to exchange information across OSM services [Broxton et al., 2010]. Users usually post URLs on Twitter and Facebook to announce to their friends about a new blog post or a new uploaded video (on YouTube). Studying the dynamics and characteristics of this *information diffusion process* across OSM services is important for various reasons, like help marketers to explore the rich environment for advertisement purposes, as well as help social media providers to improve their systems and develop tools to facilitate the information exchange across networks.

1.1 Context

Recently, researchers contributed significantly in understanding several aspects of OSM services, including diffusion of information. Blogs [Leskovec et al., 2007b], YouTube [Cha et al., 2007], Facebook [Nazir et al., 2008], and Digg [Szabo and Huberman, 2010] are some of the OSM services which have been extensively studied. Most of these studies are focused in only one OSM service or in one specific aspect of interest, like understanding temporal and topological patterns, or the interplay between the social network structure and the information flow. Some researchers developed tools to visualize and analyze the diffusion of information in OSM services, but these tools are usually based on only one OSM service or in some specific memes [1] related to specific themes (for example, politics and rumors) [Leskovec et al., 2009; Truthy, 2012].

Popular OSM services (e.g., Facebook and Twitter) are characterized by a highly dynamic environment, with a huge volume of traffic generated everyday by the sharing of several types of content (i.e., photos, videos, news), as well as by the increasing interest of users in real-time information. Each unit of information (a *meme*, like URLs or keywords) has a different life cycle in these environments. For instance, while the majority of the Web links (URLs) are unpopular, a few of them became extremely popular, shared by millions of users. Moreover, some URLs attract interest at a single period in time and die quickly, while others are periodic and attract interest of users for a long period. A recent blog post showed that, on average, the time that a URL receive half of its clicks on Twitter is only 2.8 hours, and on Facebook is only 3.2 hours [Bit.ly, 2011].

1.2 Purpose

Service providers usually do not give users a good manner to track how the information they have shared is diffused among other users and services. Developing a system to track the diffusion of information in and across OSM services is certainly useful, not only for users, but also for companies and researchers. In this work, we propose **MIOSphere**, a system to Monitor Information in the Online Social Media Sphere (i.e., blogosphere, social networks, content sharing systems). MIOSphere uses a distributed approach with a master controlling several slave machines to monitor the memes (i.e., URLs and users), as showed in Figure 1.1. The master is responsible for controlling and organizing the whole process. When slaves request information to track, the master

[1] A meme is an element of a culture or system of behavior passed from one individual to another by imitation or other non-genetic means, taken from *http://oxforddictionaries.com/*

selects which memes should be collected at that time, in order to optimize the usage of resources.

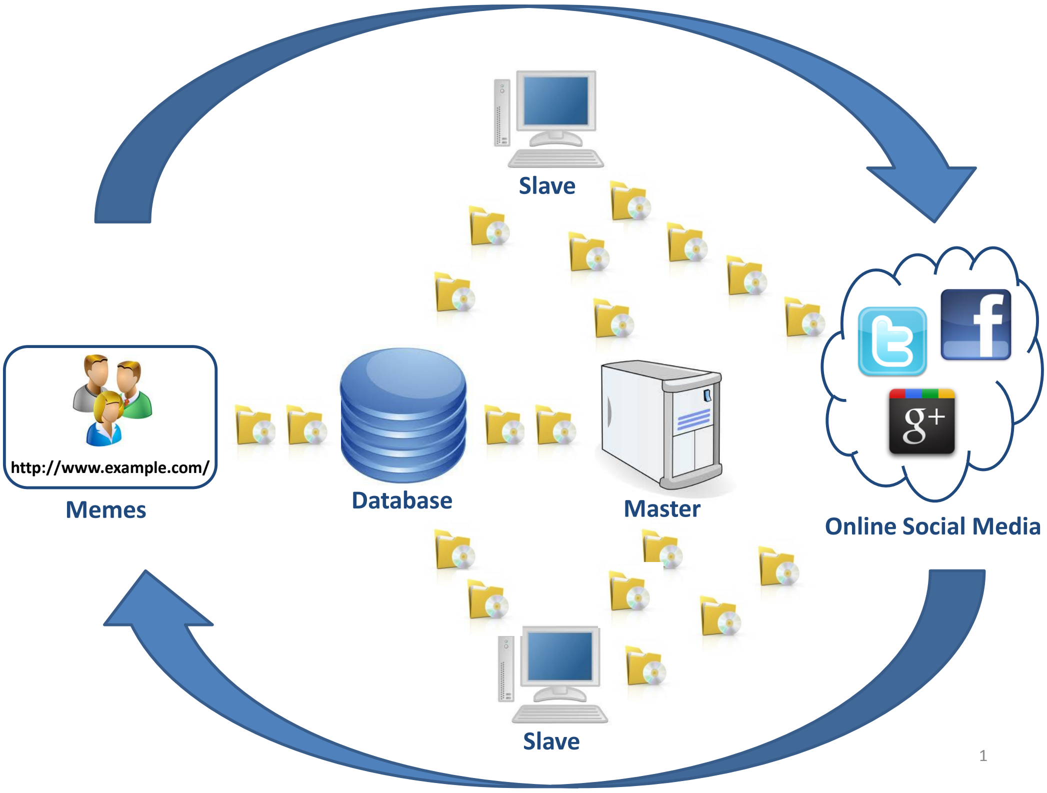


Figure 1.1. High level view of MIOSphere's architecture

As a proof of concept, a prototype which monitors the diffusion of URLs in and across popular OSM services (Twitter, Facebook, and Google+) was implemented.

1.3 Solution

Before developing MIOSphere, a wide characterization about the diffusion of URLs in and across OSM services was done. Two publications in important venues emerged from this initial step [Rodrigues et al., 2011; Jain et al., 2011]. The findings from these research efforts were extremely important for the development of MIOSphere, as will be highlighted during this dissertation. In both works the diffusion of URLs on Twitter were analyzed, but with different goals. In Rodrigues et al. [2011] the focus was to study how users discover content through word-of-mouth on Twitter and what are the characteristics of the propagation cascades, while in Jain et al. [2011], the focus was to understand how URLs from an external OSM service (YouTube videos, for example) diffuses on Twitter and how is the relationship between the OSM services involved. Both works and the relevant findings for the development of MIOSphere are detailed in Chapters 4 and 5.

After this characterization step, MIOSphere was developed. Based on some findings discussed in Chapters 4 and 5, the architecture of the system was proposed. Any Web crawler has several limitations (bandwidth, overload of traffic in the downloaded Web servers, etc), so choosing the most important URLs to visit at a given time is extremely important for achieving a good performance, especially if the goal is a real-time system in a highly dynamic environment [Cho et al., 1998]. As a tool to help in the development of MIOSphere's architecture, a trace-driven simulator was implemented and used, which was useful to understand the behavior of the parameters and how the performance of the system was affected. After running the simulations, a real prototype was implemented [2] to validate the proposed architecture. This development step is detailed in Chapter 6.

1.4 Organization

This dissertation is organized as follows. Chapter 2 explains some basic concepts and the terminology used. Chapter 3 presents the literature review. In Chapters 4 and 5, a wide characterization about the diffusion of URLs in OSM services is showed. Then, in Chapter 6, MIOSphere's architecture is explained in details. Finally, this dissertation is concluded in Chapter 7.

[2] MIOSphere prototype is available on `http://miosphere.speed.dcc.ufmg.br/`

Chapter 2

Basics of Online Social Media and Definitions

In this chapter, a discussion about some basic aspects of Online Social Media and an explanation of the terminology used in this dissertation is given.

2.1 Online Social Media Sphere

Several definitions for *Online Social Media* (OSM) can be found on the Web [Guide, 2012; Econsultancy, 2012]: (1) “Media is an instrument on communication, like a newspaper or a radio, so social media would be a social instrument of communication.” (2) “Social media essentially is a category of online media where people are talking, participating, sharing, networking, and bookmarking online.” (3) “A category of sites that is based on user participation and user-generated content. They include social networking sites like LinkedIn, Facebook, or My Space, social bookmarking sites like Del.icio.us, social news sites like Digg or Simpy, and other sites that are centered on user interaction.” (4) “ Digital word of mouth.” (5) “A term used to describe a variety of Web-based platforms, applications and technologies that enable people to socially interact with one another online.”

We refer to the entire environment around Online Social Media as *Online Social Media Sphere.* Users, providers, technologies, applications, and services, for example, are all components of the Online Social Media Sphere. Among the services, there are social networking sites like Facebook and Google+; blogs and micro-blogging services like Blogger, Tumblr and Twitter; video, photo and music sharing services like YouTube, Flickr, and Last.fm; and geo-location services like Foursquare. Each of these Web applications are referred as *OSM services.*

2.1.1 Cross-Pollination of Information

Information is exchanged among users inside an OSM service and between different OSM services. A basic unit of information is defined as a *meme*. A video on YouTube and a tweet on Twitter are the examples of memes. Memes can be divided into two categories: *foreign* and *local*. All posted URLs embedding meme belonging to another OSM service are defined as a *foreign meme*. URLs embedding YouTube videos or Flickr photos, when shared on Twitter, are examples of foreign memes. All other types of memes generated and diffused within one OSM service are *local memes*. Hashtags (term starting with # to represent the topic of the tweet, e.g., #BestDad) and mentions (internal link to another user in the form of @username) are examples of local memes on Twitter. Figure 2.1 illustrates the dynamics of exchange of memes from one OSM service to another one. OSM service where a foreign meme originates is termed as *source OSM*, and the OSM service in which the foreign meme diffuses is termed as *diffusion OSM*. Users create a meme in a source OSM, embed the meme in a URL and diffuse it on the diffusion OSM.

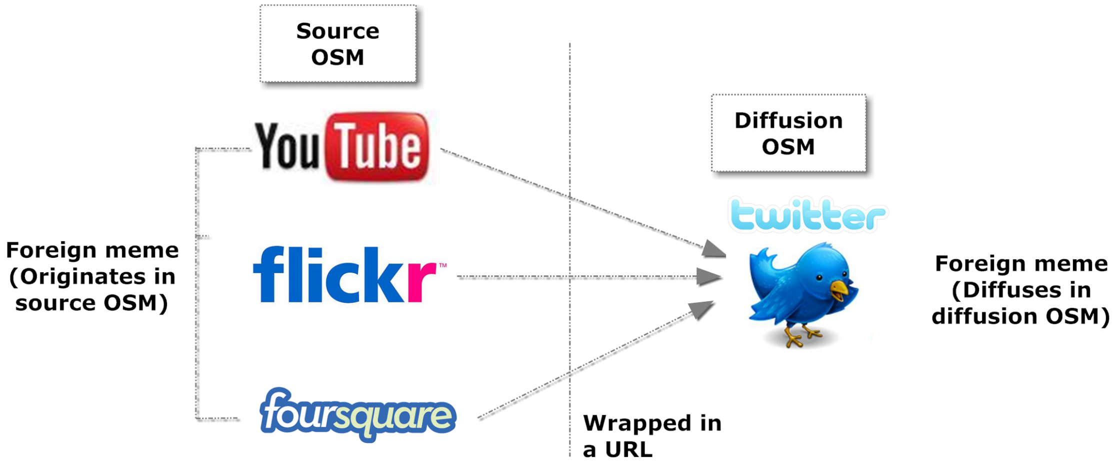

Source OSM
Diffusion OSM
Foreign meme (Originates in source OSM)
YouTube
flickr
foursquare
twitter
Foreign meme (Diffuses in diffusion OSM)
Wrapped in a URL

Figure 2.1. Dynamics of exchange of information in the Online Social Media Sphere

The information diffusion process across OSM services is analogous to a process in biology, termed as *Cross-Pollination*. In this process, pollen is delivered to a flower from a different plant, with the plants being different in their genesis [Darwin, 1900]. Following the same analogy, we term the information diffusion process across OSM services as *Cross-Pollination*. A unit of information is analogous to pollen, and different OSM services are analogous to plants having different genesis. A network formed by users who participate in Cross-Pollination and content produced in the network is termed as *Cross-Pollinated network*.

2.2 Modeling Information Cascades

Below, a description of the model of URL propagation is provided.

2.2.1 Hierarchical Tree Model

Information propagation paths are built based on Krackhardt's hierarchical tree model [Hanneman and Riddle, 2005]. A hierarchical tree is a directed graph where all nodes are connected and all but one node, namely the *root*, have in-degree of one. This means that all nodes in the graph (except for the root) have a single parent. Hence, an edge from node A to node B is added to the tree only when B is not already a part of the tree. An edge from node A to node B means that a piece of information was passed from A to B. While each hierarchical tree has a single root, there may be multiple users who independently share the same URL. In this case, the propagation pattern of a single URL will contain *multiple* trees and form a *forest*.

While we assume there is only a single parent for any intermediate nodes, in real life there may be more than one source that passed the same piece of information to a given user. On Twitter, for example, more than 80% of the users who received an information from multiple sources cite their last source [Rodrigues et al., 2011]. This pattern might be attributed to the timeline interface of Twitter, which works as a stream, showing the last 200 tweets to the user, chronologically ordered. Facebook and Google+ also provide users a timeline interface. Hence, it is assumed that each user received a URL from the most recent source. This model is different from that proposed by Sun et al. [2009], where all friends who joined the same group on Facebook within the last 24 hours were considered as valid sources.

There are two major ways to reconstruct information propagation paths. One approach is to rely solely on explicit evidences about who passed the information to whom. On Twitter, a retweet typically contains the original tweet content by someone else along with a "RT @username" or "via @username" text that cites the person who posted the tweet. Citation sometimes extends to the intermediary persons who also retweeted the same message. Other popular OSM services also have similar explicit evidences, like the "share" button on Facebook. Identifying information propagation based on explicitly links has been a popular approach in previous work [Kwak et al., 2010; Yang et al., 2010]. Another approach is to consider implicit evidences as well. As opposed to the former approach, implicit URL propagation occurs when a user A publishes a URL after receiving it from a user B, but user A does not explicitly cite user B. In this case, we have to infer that the information passed from user B to user A. Both approaches were used in this dissertation.

Following the definitions proposed in Wang et al. [2011], all users in the root of a hierarchical tree are called *initiators*. These users are the ones who independently shared URLs. All other nodes who participated in URL propagation are called *spreaders*. Initiators and spreaders make up the hierarchical tree. Users who simply received a URL but did not forward it to others are called *receivers*. Later when we refer to the hierarchical tree structure, we do not include these users. For convenience, we collectively call all three types of users who potentially read the URL as its *audience*. Figure 2.2 depicts this relationship.

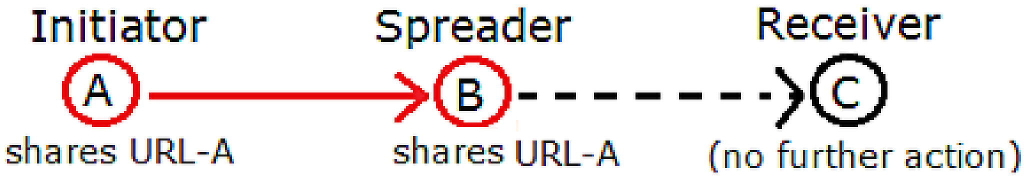


Figure 2.2. Modeling information cascades terminology

Chapter 3

Literature Review

Information diffusion in OSM is a very active research area, with a rich set of work. There are several theoretical research work which attempt to model some aspects of the diffusion, and a lot of characterizations focused on a diversity of aspects and services. There are also proposals of tools and algorithms exploring information diffusion in OSM.

3.1 Information Diffusion in OSM Services

A rich set of theoretical work explains the interplay between the social network structure and information flow. Granovetter [1978] proposed a linear threshold model, where someone will adopt an innovation only if a large enough proportion of his neighbors have previously adopted the same innovation. Dodds and Watts [2005] studied this model in the field of disease spreading in an epidemiological setting. Watts [2002] proposed a mathematical model of global cascades based on sparse Erdős-Rényi random networks and found that global-scale cascade could occur even with few early adopters. Watts examined the conditions for when such cascade happens under homogenous thresholds of user susceptibility. Karsai et al. [2010] followed the time evolution of information propagation in small-word networks and showed that the slowing down of spreading is found to be caused mainly by weight-topology correlations and the bursty activity patterns of individuals. Steeg et al. [2011] analyzed information cascades on Digg and concluded that the highly clustered structure of the Digg network limits the final size of cascades observed, as most people who are aware of a story have been exposed to it via multiple friends. Cha et al. [2012] studied information cascades on Flickr and found that popular photos do not spread as quickly as one might expect, but show a steady linear growth of popularity over several years. They concluded that burstiness of user login times and content aging can explain how the small-world network's ability to spread information quickly and widely is affected.

Another set of work contributed significantly in understanding several aspects of OSM services, including diffusion of information. Sun et al. [2009] found long chains by studying cascades on Facebook pages, and also showed that these diffusion chains on Facebook are typically started by a substantial number of users. Gomez Rodriguez et al. [2010] investigated the problem of tracing paths of diffusion and influence and proposed an algorithm to decide a near-optimal set of directed edges that will maximize influence propagation. Ghosh and Lerman [2010] compared a number of influence metrics over Digg data and suggested that a centrality-based measure is the best predictor of influence. Scellato et al. [2011] studied how geographic information extracted from social cascades can be exploited to improve caching of multimedia files in a Content Delivery Network (CDN). Their evaluation showed that cache hits can be improved with respect to cache policies without geographic and social information. Wang et al. [2011] found that social and organizational context significantly impacts to whom and how fast people forward information. Bakshy et al. [2009] studied content propagation in the context of the social network existent in Second Life, a multi-player virtual game. By examining cascade trees they find that the social network plays a significant role in the adoption of content.

Blogging and micro-blogging networks are shown to have temporal and topological patterns which largely exhibit power-law behavior [Leskovec et al., 2007b; Galuba et al., 2010; Kwak et al., 2010]. Krishnamurthy et al. [2008] presented a detailed characterization of Twitter, and De Choudhury et al. [2010] analyzed how user similarities (homophily) along various attributes can affect the information diffusion process on Twitter. Liben-Nowell and Kleinberg [2008] reconstructed the propagation of massively circulated Internet chain letters and showed that their diffusion proceeds in a narrow but very deep-like pattern.

3.2 Information Diffusion Across OSM Services

A number of researchers have presented data-driven analysis and measured patterns of information spreading across OSM services. Gruhl et al. [2004] studied the diffusion of information in the blogosphere based on the use of keywords in blog posts. They presented a pattern of information propagation within blogs using the theory of infectious diseases to model the flow. Adar and Adamic [2005] further extended the idea of applying epidemiological models to describe the information flow and relied on the explicit use of URLs between blogs to track the flow of information. Cha et al. [2009] analyzed the blogging network structure and information diffusion patterns within the network. Broxton et al. [2010] analyzed the diffusion of viral video popularity in OSM, but focused only on how the popularity of a video varies with its introduction in OSM. They concluded that viral videos gain popularity faster on OSM than through any other referring source or itself (e.g., search engines, etc), and that viral video popularity on Twitter is at a higher rate than in any other OSM website, but without analyzing the underlying network structure affecting the higher rate.

Recently, Archambault and Grudin [2012] conducted annual surveys of social networking at Microsoft between 2008 and 2011 to understand how these sites were used and whether they were considered to be useful for organizational communication and information-gathering. Growth in use and acceptance was not uniform, with differences based on some aspects like gender and age. Tang et al. [2012] developed a framework for classifying the type of social relationships by learning across heterogeneous networks.

Most of these studies of information diffusion on OSM are focused in one specific aspect of the diffusion and in only one OSM service. In contrast to these works, our study unveils different aspects of word-of-mouth information, such as not only of the shape of cascades, but also the impact of publishers and subscribers of content. This dissertation also explores some aspects related with the exchange of information between different OSM services, which was mostly explored before on the blogosphere.

3.3 Tools and Applications for OSM Services

Some tools and applications have been developed recently by some group of researchers. Leskovec et al. [2009] developed a framework for tracking short, distinctive phrases that travel relatively intact through online media. They observed a typical lag of around 2.5 hours between the peaks of attention to a phrase in the news media compared to blogs. Truthy is a system to analyze and visualize the diffusion of information on Twitter, developed by Indiana University Center for Complex Networks & Systems Research [Truthy, 2012]. Marcus et al. [2012] presented two systems for querying and extracting structure from Twitter-embedded data. In contrast to these tools, the focus of MIOSphere is to collect and analyze the diffusion of information in multiple OSM services.

3.4 Information Retrieval

During the development of MIOSphere architecture, a set of work from Information Retrieval were studied. Modern Web search engines have billions of pages indexed, and they have to keep information fresh, especially for the most clicked pages. In order to crawl efficiently, search engines have to schedule the most important URLs to visit at a given moment, in a problem similar to our case. There are several proposals of algorithms in the literature for scheduling URLs for search engines. Cho et al. [1998] defined several importance metrics, ordering schemes, and performance evaluation measures for this problem, and showed that a crawler with a good ordering scheme can obtain important pages significantly faster than one without. Sharma and Dixit [2008] proposed an algorithm for building an effective incremental Web crawler. Gomes and Silva [2006] modeled the persistence of Web data through the measurement of URL and content persistence across several snapshots of a national community Web, collected for 3 years. They found that the lifetimes of URLs and contents are modeled by logarithmic functions. Olston and Pandey [2008] characterized the longevity of information found on the Web, via both empirical measurements and a generative model that coincides with these measurements. They also developed new re-crawl scheduling policies that take longevity into account.

Chapter 4

On Word-of-Mouth Based Discovery of the Web

In this chapter we present an analysis about how URLs diffuse internally among Twitter users by word-of-mouth. The set of analysis discussed here were also published in Rodrigues et al. [2011].

4.1 Methodology

Twitter is a prime example of an OSM service where users discover Web content through word-of-mouth. In this part of the characterization, we used the Twitter dataset gathered in Cha et al. [2010] and studied the properties of word-of-mouth based Web discovery.

Twitter is an ideal medium for these studies for several reasons. First, the core functionality provided by Twitter, *tweeting*, is centered around the idea of spreading information. Second, Twitter provides additional mechanisms like *retweet* (act of forwarding other people's tweet), which enable users to propagate information across multiple hops in the network. Third, thanks to URL shortening services, sharing URLs has become a common practice in Twitter.

4.1.1 The Twitter Dataset

Data collection utilized the official Application Programming Interface (API) of Twitter and took over a month using 58 servers in Germany [Cha et al., 2010]. These servers were white listed by Twitter so that they can send API requests rapidly. The data comprises the following three types of information: profiles of 54,981,152 users, 1,963,263,821 directed follower links among these users, and all 1,755,925,520 public tweets that were ever posted by the collected users. The oldest tweet in this dataset is from March 2006, when the Twitter service was publicly launched. The dataset does not include any tweet information about a user who had set his account private (8% of all users). Moreover, this dataset is near-complete because user IDs were sequentially queried from all possible ranges (0–80 million) at the time of data collection in September 2009. Therefore, it provides a unique opportunity to examine the largest word-of-mouth based URL propagation event in Twitter.

A Twitter user might follow another user to receive his tweets, forming a social network of interest. The node in-degree and out-degree distributions measured on this network are heavy-tailed, and the network topology is similar to those of other OSM services like Facebook. They can be fit well with a Power-Law distribution with exponents 2.19 for in-degree and 2.57 for out-degree (R^2=0.05–0.09%). While a very small fraction of users have an extremely large number of neighbors, the majority of users have only a few neighbors; 99% of users have no more than 20 in- or out-degree neighbors. The most popular users include public figures like Barack Obama, celebrities like Oprah Winfrey, as well as media sources like BBC. A social link in Twitter is directional. Unlike other OSM services, the Twitter network exhibits extremely low reciprocity; only 23% of all links are bidirectional, which means that high in-degree nodes are not necessarily high out-degree nodes.

4.1.2 URLs in Tweets

A URL is treated as a clean piece of information that spreads in Twitter. The number of tweets containing URLs has increased rapidly over the years, as shown in Figure 4.1. Since 2009, on average 22.5% of tweets contain URLs, and as of September of 2009 more than 30% of tweets contain URLs. This is equivalent to sharing 1.3 million distinct URLs per day in 2009. The URL usage is even higher in retweeets: 47%. Interestingly, the number of retweets grew abruptly after July of 2008. This is because retweeting became a convention between users around this time [Boyd et al., 2010].

In analyzing the Web links within tweets, it was found that the majority of the URLs (nearly 75%) were from URL shortening services (e.g., *bit.ly*), which substantially shorten the length of any Web link. Hence we had to take into account several possible confounding factors such as when multiple short URLs refer to the same long URL or when the short URLs are recycled after not being used for some time (e.g., when there is no human visitor for 120 days, some URL shortening services allow a short URL to point to a new content). Therefore, a large pool of URLs over several short time

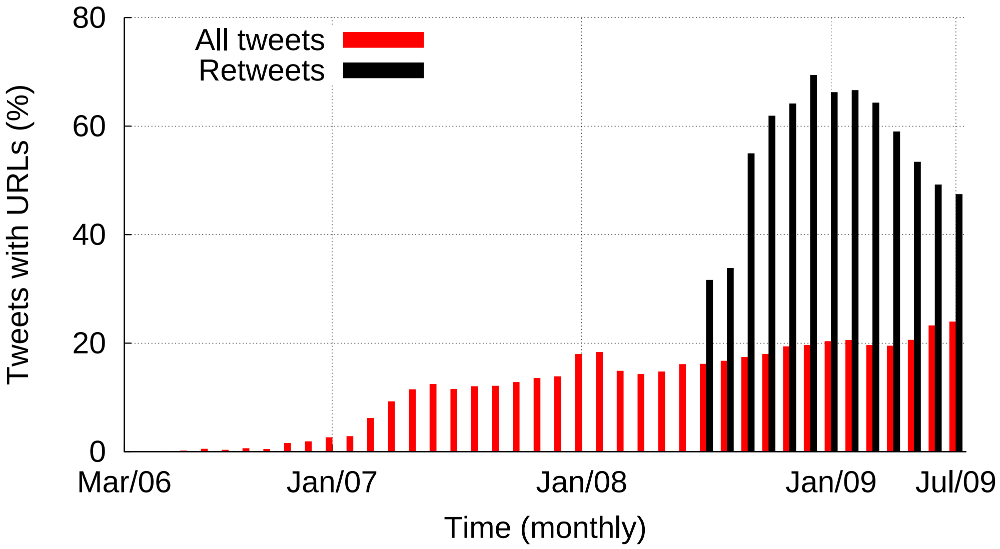


Figure 4.1. Usage of URLs on tweets over time

periods (so that these URLs refer to identical content) was picked, and then resolved all the short URLs to the long URLs for data analysis in this paper.

Table 4.1 displays the summary of datasets we analyzed. Each week period contains several million distinct URLs. Because the samples are from a one week period, certain URLs were already in the process of word-of-mouth propagation. Hence, the entire tweet dataset was scanned to find all tweets that contain the URLs in Table 4.1 and additionally considered them in our analysis. We made sure that none of the added URLs were recycled. For a better readability, only results for Dataset 2 in Table 4.1 is presented. All the conclusions hold for Dataset 1 as well.

	Period	Distinct URLs	Tweets	Retweets	Users
Dataset 1	Jan 1, 2009 – Jan 7, 2009	1,239,445	6,028,030	295,665	995,311
Dataset 2	Apr 1, 2009 – Apr 7,2009	4,628,095	17,381,969	1,178,244	2,040,932

Table 4.1. Statistics of the two Twitter datasets analyzed

4.2 Which Content is Popular?

In this section is examined which URL shortening services are widely used in Twitter, which Web domains these short URLs point to, and what kinds of content is popular in word-of-mouth discovery of the Web.

4.2.1 URL Shortening Services

URL shortening services make it easy for Internet users to share Web addresses by providing a short equivalent [Antoniades et al., 2011]. For example, a Web link `http://topics.nytimes.com/top/news/business/companies/twitter/` can be shortened to `http://nyti.ms/1VKbrC` by a commercial service Bit.ly [Bit.ly, 2012a], which will redirect any request access to the original NYTimes website. URL shortening services allow otherwise long Web addresses to be referred to in various OSM services like Twitter that often impose character limit in tweets and comments. There are hundreds of commercial URL shortening services. Hence, the same Web address can have several short alternatives in services like tinyURL [tinyURL, 2012] and Ow.ly [Ow.ly, 2012].

In order to identify whether a given Web address is a short or long URL, a heuristic approach was taken. A Python script was written to resolve a URL in a tweet by sending a Web access request to that URL and comparing the domain of the original URL with that of the resolved URL. If the two domain names were different, the URL in the tweet is considered to be a short URL, otherwise, it is considered it a long URL.

A total of 30 URL shortening services were in use from 2006 to summer 2009 in Twitter. Table 4.2 displays the top 10 services and their share of tweets. Usage of the top two services (*tinyurl.com* and *bit.ly*) make up more than 90% of the total usage. Between January to April of 2009, we find that *bit.ly* doubled its presence. The 3rd ranked service (*is.gd*) also continued to gain presence in Twitter.

Rank	**Web Domain**	**Dataset 2**	**Dataset 1**
1	tinyurl.com	4,398,940 (68.2%)	1,883,032 (81.4%)
2	bit.ly	1,530,613 (23.7%)	262,171 (11.3%)
3	is.gd	493,124 (7.6%)	142,497 (6.2%)
4	snipurl.com	27,146 (0.4%)	24,606 (1.1%)
5	hugeurl.com	1,578 (0.0%)	841 (0.0%)
6	ur1.ca	1,116 (0.0%)	451 (0.0%)
7	xrl.in	361 (0.0%)	250 (0.0%)
8	u.nu	282 (0.0%)	139 (0.0%)
9	simurl.com	216 (0.0%)	6 (0.0%)
10	doiop.com	101 (0.0%)	90 (0.0%)

Table 4.2. Top 10 URL shortening services in 2009

4.2.2 Popularly Linked Web Domains

Next, we checked whether URLs popularly shared on Twitter come from major Web domains in the Internet (such as *nytimes.com* or *google.com*). The motivation of this analysis is to verify a widely held belief that word-of-mouth can help popularize niche or esoteric information from domains that are not otherwise very popular. The translated long URLs were used for this analysis and the rest of this work, and the URLs were grouped based on their domain names.

In total, there were 4,638,095 long URLs that came from 429,551 distinct Web domains. The top 20% of the Web domains accounted for 95% of these URLs. We ranked the domains based on the number of distinct URLs that belong to the domain as well as the total size of the audience reached by URLs belonging to the domain. Experiments using both ranking methods had similar results. We compared the list of top domains in the resulting rankings with the list of top ranked domains in the general Web published by Alexa [Alexa.com, 2012]. Table 4.3 displays the top 5 domains based on the number of URLs, their description, the fraction of all URLs that belong to the domain, and their rank from *alexa.com*.

Rank	Top list	Description	URLs	Alexa rank
1	twitpic.com	photo sharing	8.5%	103
2	blip.fm	music sharing	3.0%	6,736
3	youtube.com	video sharing	2.1%	3
4	plurk.com	social journal	2.1%	1,146
5	tumblr.com	blog	1.4%	100

Table 4.3. Top 5 domains in Twitter (April, 2009)

The most popular domain, *twitpic.com*, accounted for 8.5% of all URLs in the tweet data. The coverage of the other top domains quickly drops with decreasing rank. The Alexa ranking shows that the top 5 domains are quite different from the top list in the Web. Only *youtube.com* is within the top 10 sites from *alexa.com*. The top 5 list in Alexa includes major search engines (Google) and portals (Yahoo, Live). We also found that Twitter users often share user-generated content, as seen in the table. Twitter users share photos (*twitpic.com*), videos (*youtube.com*), blog articles (*techcrunch.com*), as well as participate and promote social events (*abolishslavery.com* and *earthday.net*).

Figure 4.2 shows the fraction of top K domains that also appear in the top K domain list from *alexa.com* for various values of K. The bar plot also shows a comparison to the top URLs identified by the size of the audience reached within

Twitter (including initiators, spreaders, and receivers). The overlap is minimal; fewer than 30% of the top K domains in Twitter overlap with that of the general Web, for all ranges of K=100,···,1000. This finding suggests that as word-of-mouth becomes a dominant source of discovering information, a different set of domains might become popular in the Web in the near future.

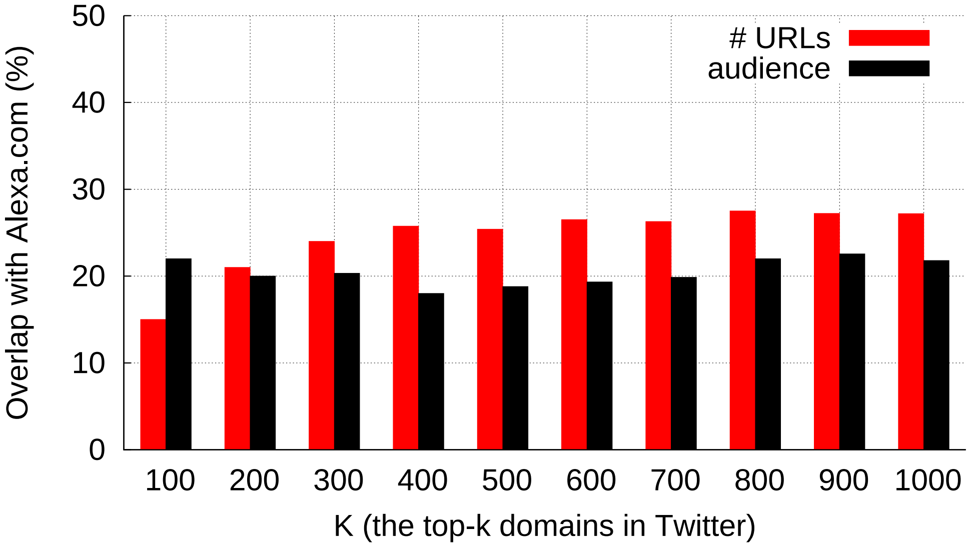


Figure 4.2. Popular domains in Twitter and Alexa rank

A recent work characterizing the usage of short URLs on Twitter also presented an analysis comparing the popularity of domains on Twitter and on the general Web [Antoniades et al., 2011]. Although the authors of that work considered only the domains pointed by short URLs, which differs from our analysis, they also found that the most popular domains shared on Twitter differs significantly from the general Web case.

4.2.3 Popular Individual URLs

Next we focus on popularity of individual URLs within domains. Of particular interest to us is the hypothesis that word-of-mouth gives all URLs and content a chance to become popular, independent of popularity of the domain it comes from. The hypothesis is rooted in the observation that anyone could identify an interesting URL and start a viral propagation of the URL, independent of the reputation or popularity of the domain where it is published.

To verify this hypothesis, the size of the audience reached by individual URLs within each domain was computed. Figure 4.3 plots the minimum, the average, and the maximum size of the audience for individual URLs within each domain. Given the large number of domains (over 400,000 of them) we ranked the domains based on number of URLs, grouped them into bins of 5,000 consecutively ranked domains and plotted one data point for each bin. It is striking to observe that URLs from some unpopular domains beyond the rank of 300,000 reached an audience that is comparable to the size of the audience reached by URLs from the most popular domains. On average, URLs in the top 5,000 domains reached 49,053 users, while URLs from the bottom 5,000 domains reached 1,107 users. Although URLs from top domains reached a 44 times larger audience than those from bottom domains, there do exist individual URLs from bottom domains that reach as large as audience as the most popular URL from the top domains.

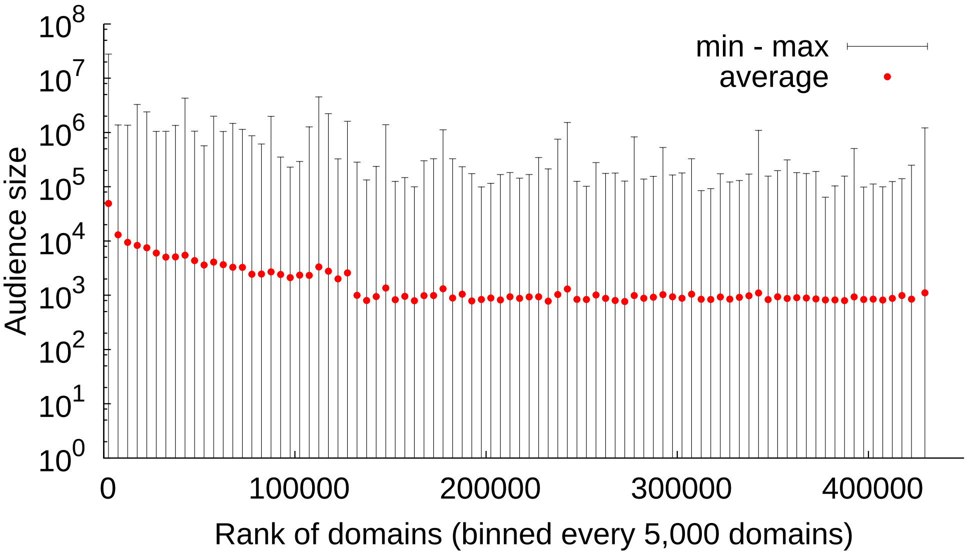


Figure 4.3. Audience size across the domain ranks

Thus, word-of-mouth does offer a chance for all content to become popular, independent of the domain it is published in. Previous work on book and DVD recommendations showed that viral marketing is effective for niche products compared to mass marketing [Leskovec et al., 2007a]. Our analysis suggests a similar trend.

4.2.4 Content Types

With millions of URLs published per day on Twitter, several different types of contents are shared. A natural question that arises from this observation is whether the type of content affects the word-of-mouth propagation dynamics.

In order to identify the content type, several interest categories based on Open Directory Project (DMOZ) were selected, which is a human-edited directory of the Web [Project, 2012]. DMOZ's content classification relies on the fact that a domain name has a hierarchical structure. So, a domain name of a Web address could potentially be used to identify the specific content category, given the list of predefined classifications for many Web domains as in the DMOZ service. For example, *nytimes.com* is classified in the news category; *last.fm* is classified as the music category.

Among various categories DMOZ supports, 5 categories of interest were picked: photo sharing, videos, music, news, and applications, and downloaded the list of Web domains in each of these categories. In total, the DMOZ listing contained 343 domains across all five categories. For the application category, the list of applications that are widely used within Twitter were selected, such as *tweetdeck.com* and *wefollow.com.*[1]

Table 4.4 displays the share of each topical category in Dataset 2. Matching the domains of the URLs in Dataset 2 with the domains listed in these five categories in DMOZ, we were able to successfully categorize 17.6% of websites, although a much larger fraction of users (43.7%) were covered. The most popular category is photo sharing, which has 458,662 URLs, posted by 271,138 users on 735,137 tweets. The average audience reached by each photo's URL is 436 users. The second most popular category is music, followed by videos, news and applications.

Category	**URLs**	**Users**	**Tweets**	**Audience**
Photos	458,662 (9.7%)	271,138 (13.3%)	735,137 (4.0%)	436
Music	181,676 (3.8%)	46,483 (2.3%)	338,654 (1.8%)	316
Videos	123,412 (2.6%)	261,081 (12.8%)	509,975 (2.8%)	877
News	58,467 (1.2%)	113,667 (5.6%)	305,911 (1.7%)	2,492
Applications	15,223 (0.3%)	198,499 (9.7%)	370,047 (2.0%)	3,285

Table 4.4. Summary Information of Categories of URLs

While photo sharing seems a dominant activity in Twitter as opposed to news or application sharing, the set of most popular individual URLs are from a diverse set of topical categories as shown in Table 4.5. The most popular URL was the social application, *wefollow.com*, which reached an audience of 28 million (i.e., nearly half of the entire Twitter network).

Although this analysis about the different content types shared on Twitter is an interesting aspect of our study, we note that our methodology has some limitations. First, we were able to categorize only 17.6% of the URLs, which might not

[1] `http://www.dmoz.org/Computers/Internet/On_the_Web/Online_Communities/Social_Networking/Twitter/`

Rank	URL domain	Audience	Description
1	wefollow.com	28M	Social application that suggests list of users to follow
2	facebook.com	14M	Social network (warning page)
3	abolishslavery.org	4.5M	Social organization dedicated to combating human traffic (initial page)
4	twitpic.com	4.5M	Photo sharing (photo published by famous actor Ashton Kutcher)
5	youtube.com	4.5M	Video sharing (popular comedy video with title "David After Dentist")
6	tweetvalue.com	4.3M	Application that measures the value of a Twitter account (initial page)
7	techcrunch.com	3.6M	Blog (article with rumors about Google in talks to buy Twitter)
8	earthday.net	3.3M	Social organization dedicated to the Earth's natural environment (initial page)
9	twibes.com	3M	Application to find people with similar interests on Twitter (initial page)
10	latenightwithjimmyfallon.com	2.4M	TV Show from NBC (initial page)

Table 4.5. Top 10 URLs domains in terms of the audience size reached by the most popular URL

be representative. Second, DMOZ is often criticized for its lacks of representativeness and transparency [Goodman, 2012], but it is not easy to categorize content to begin with, and we manually checked the list of Web domains in each category we used from DMOZ. Moreover, our main interest in the URL categorization is at understanding the similarities and differences between the propagation of different content types on Twitter. We are not trying to say that Twitter users share more photos than videos or news, for example.

4.3 The Shape of Word-of-mouth

This section presents an analysis of the size and shape of word-of-mouth based URL propagation patterns in Twitter.

4.3.1 How Large is the Largest Word-of-mouth?

The skew in popularity across different URLs is examined here. Figure 4.4 shows the size distribution of spreaders and audience for URL propagations in Twitter. An average URL was spread by three users and gained an audience of 843 users through word-of-mouth. In contrast, the most popular URL engaged 426,820 spreaders and reached an audience of 28 million users, which is more than half of the entire Twitter network at that time. The power of word-of-mouth extends beyond the few most popular URLs. Each of the 100 most popular URLs reached an audience of more than

1 million users and 15% of the URLs reached an audience of over 1000 users.

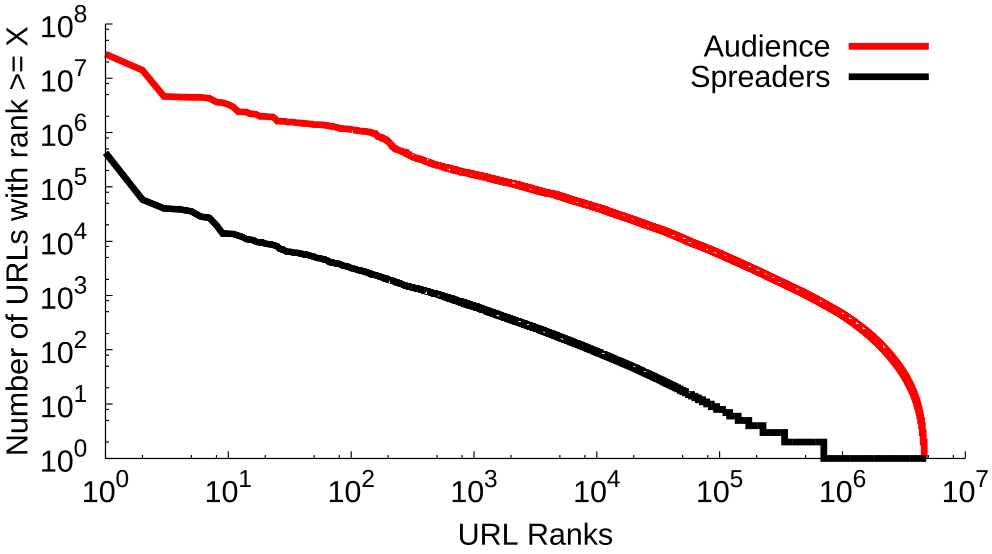


Figure 4.4. The size of word-of-mouth

The difference between the number of spreaders and the size of the audience is nearly two orders of magnitude, for popular URLs as well as niche URLs that have only a few spreaders. This demonstrates the potential of word-of-mouth in reaching a large audience. As opposed to a typical Web page that is viewed by individual visitors, content shared in word-of-mouth fashion is *collaboratively* shared by other visitors who liked it and can reach a much larger audience.

Both distributions for spreaders and audience exhibit power-law behavior (a straight line waist in a log-log plot). The best fit power-law exponents of these distributions $y = cx^{-\alpha}$ were $\alpha = 1.71$ for spreaders, and $\alpha = 1.98$ for audience, indicating that the skew in popularity among the most popular and the least popular URL became slightly more severe due to audience.

4.3.2 The Role of Initiators

Users were classified into three types based on their position within a cascade: initiators, spreaders, and receivers (Figure 2.2). Initiators are at the root of each cascade tree and share URLs to others independently in Twitter. The role of initiators in particular was investigated, and the following questions are asked: To what extent can initiators alone reach a large audience (without the help of spreaders)? How many initiators share the same URL? Is having multiple initiators essential for yielding a large cascade?

In Twitter, nearly 90% of all URLs are introduced only by initiators without involving any spreaders. URLs that were propagated further by spreaders went multiple hops in the Twitter social graph and gained a 3.5 times larger audience than those spread by initiators. This implies that while initiators' role is dominant and that most URLs propagate only 1-hop in the network, multi-hop propagation by spreaders can extend the readership of URLs by a significant amount.

Next, the relationship between the number of initiators and cascade size is investigated. In order to focus on URLs with multiple initiators, only URLs with at least two participants (i.e., initiators or spreaders) were selected for this analysis. URL cascades were grouped into three types: small (1,10), medium [100,1000), and large [1000,∞). Figure 4.5 shows the number of initiators per URL for the three types of cascades. The plot shows the 5th percentile, median, average, and 95th percentile values. The number of initiators is orders of magnitude larger over the cascade size. The median number of initiators is 2, 114, and 792 for small, medium, and large cascades, respectively. Furthermore, very few URLs (0.2%) had more than 100 initiators who independently shared the same URL. While certain large cascades only involved a single initiator, the plot indicates that the number of initiators does affect the size of the cascade for most URLs — larger cascades likely involve more users who independently share the URL.

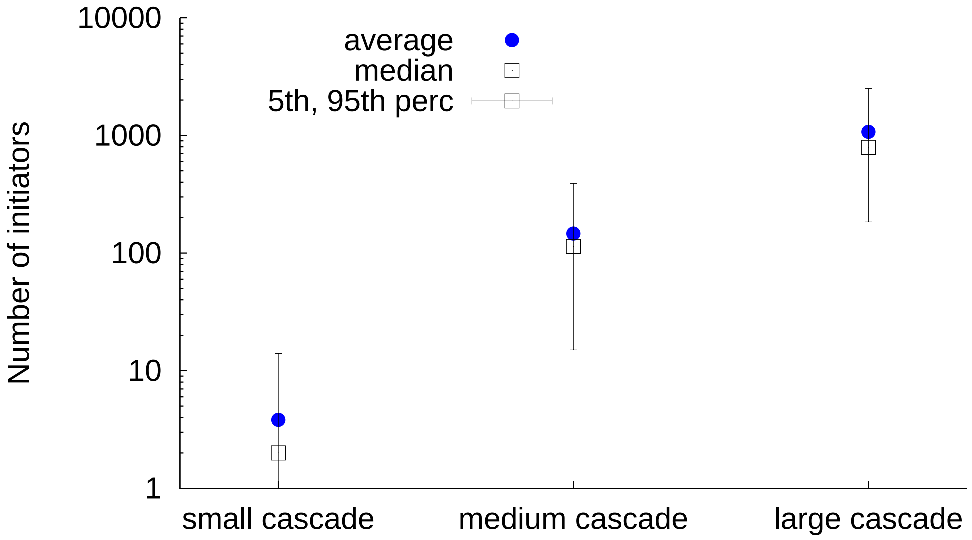


Figure 4.5. The number of initiators per URL

While there seems to be a relationship between the number of initiators and cascade size in general, it is not clear whether all of the initiators contributed equally in obtaining audience, or whether a few major initiators played a significant role. Figure 4.6 investigates the fraction of the largest sub-tree for each URL over cascade size. The plot shows much variability, shown by the wider range of the 5th and 95th percentiles. Nonetheless, the median and the average data points show a clear trend.

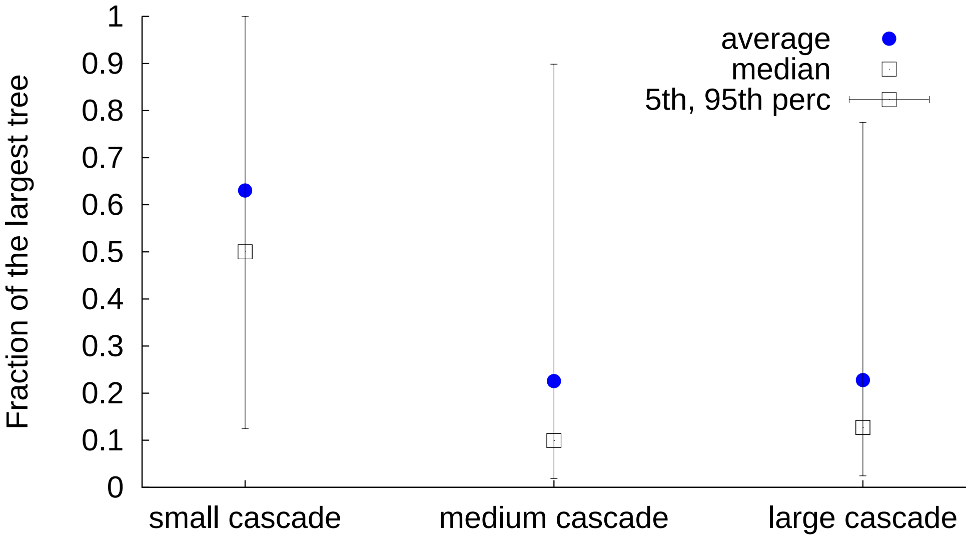


Figure 4.6. Roles of initiators by the largest subtree

For small cascades, the fraction of total audience reached in the largest sub-tree (i.e., by a single major initiator) is nearly 50%. For larger cascades, however, the fraction of the largest sub-tree is marginal (10-20%). This implies that a single initiator's role is likely to be limited in these cases and that popular content usually propagates through several different propagation trees. In fact, there exists a strong positive correlation between the number of initiators and the total size of the audience (Pearson's correlation ρ=0.7171 [Wikipedia, 2012]).

On Digg, a social news aggregator that allows users to submit links to and vote on news stories, a story of general interest usually spreads from many independent seed sites, while a story that is interesting to a narrow community usually spreads within that community only [Lerman and Galstyan, 2008]. This observation might also be true in the case of URLs propagated in Twitter, but we left the investigation for future work.

4.3.3 The Shape of Word-of-mouth

Having investigated the size of URL cascades, we next examine the overall shape of URL propagations. The maximum hop count from root to any of the leaf nodes is referred as the *height* of the tree. The *width* of the tree is defined as the maximum number of nodes that are located at any given height level. For instance, a two-node cascade graph has height of 2. Because a single URL propagation may have multiple tree structures, we consider only the largest propagation tree for each URL and examine its width and height.

Figure 4.7 shows the distributions of height and width for all URLs. Nearly 0.1% of the trees had width larger than 20, while only 0.005% of the URLs had height larger than 20. This suggests that cascade trees in Twitter are wider than they are deep. In fact, the maximum observed width of any propagation tree was 38,418, while the maximum observed height was 147 — a difference of two orders of magnitude.

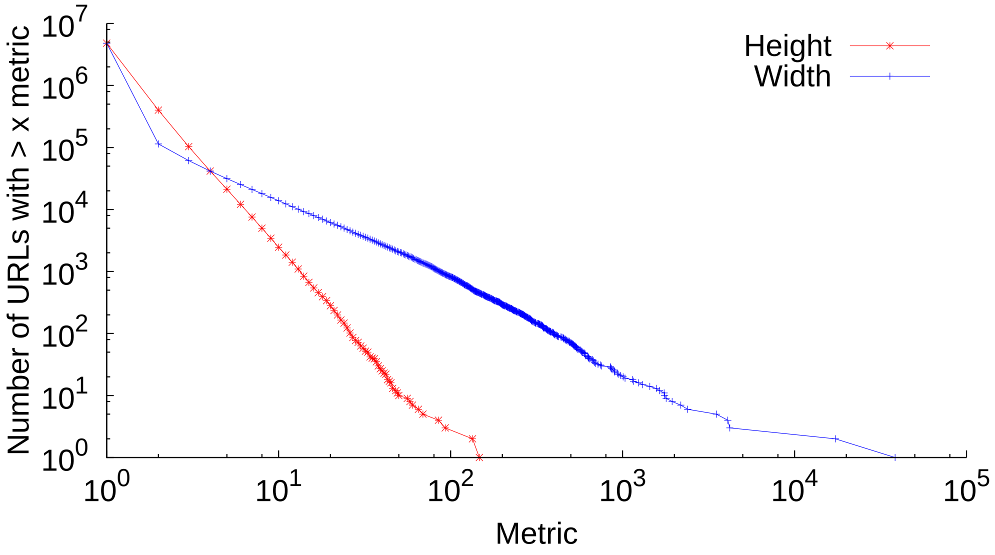


Figure 4.7. Height and width distributions

Figure 4.8 shows the relationship between the width and height level, as the 10th, 50th (median), and 90th percentile width over every height level. Trees were grouped based on their size (according to the number of spreaders). Small trees with fewer than 100 spreaders tend to have a very narrow shape of width 1 or 2 throughout the height level. Larger trees with more than 100 spreaders were widest at low heights and the width decreased slowly towards the leaf nodes. Interestingly, the median width remained near 10 even at heights above 80. Visualization revealed that these large trees occasionally included bursts at all height levels, i.e., the branch out factor is suddenly large at particular spreaders. The visualization also revealed not one but multiple such bursts for every popular URL.

Finally, Figure 4.9 shows the size of a typical cascade for the five different types

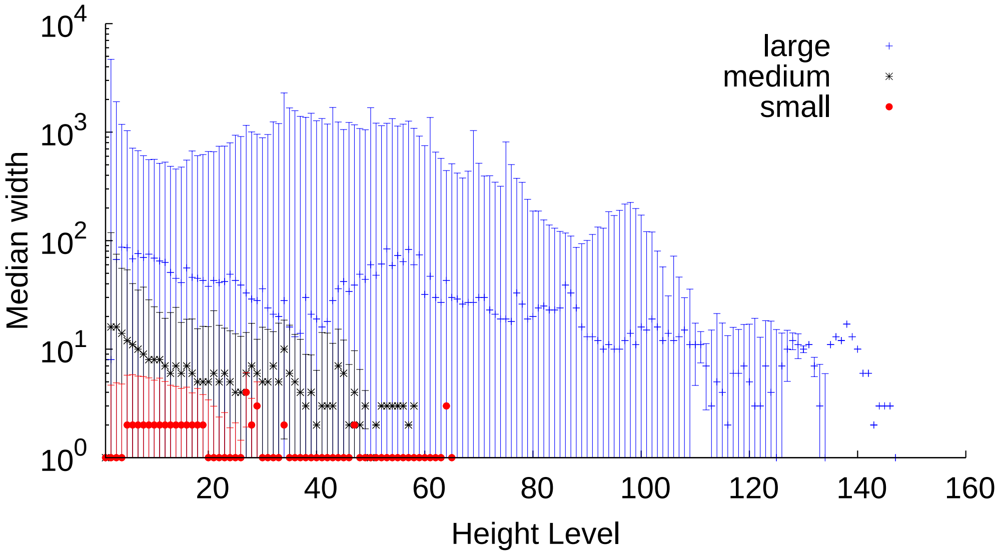


Figure 4.8. Cascade size distributions

of URLs: photos, music, videos, news and applications. Several interesting differences were observed across content types. Videos propagations likely involve a larger cascade tree; more than 30% of videos URLs involved at least two participating users who shared the URL. News and applications propagation also involved multiple users (28% and 23%, respectively), while photos and music were mostly shared by a single initiator (90% and 97%, respectively). This observation indicates that the type of content affects the potential of the eventual cascade size. The probability of involving 10 or more users in spreading is around 10% for news, applications and videos, while it is only 1% for photos and music.

4.4 Content Distribution

So far, several key characteristics of word-of-mouth based URL propagation in Twitter were investigated. The observed propagation patterns have direct implications on systems, especially on content distribution and caching strategies. In this section, some of these observations are revisited. Geo-location information of users was used to examine how far word-of-mouth content travels around the globe.

For this analysis, the location of users who post and receive tweets needs to be known. The location information written in user profiles of Twitter is in free text form and often contains invalid location like “Mars" making it difficult to automate the process. Invalid locations were filtered out and plausible locations of users were

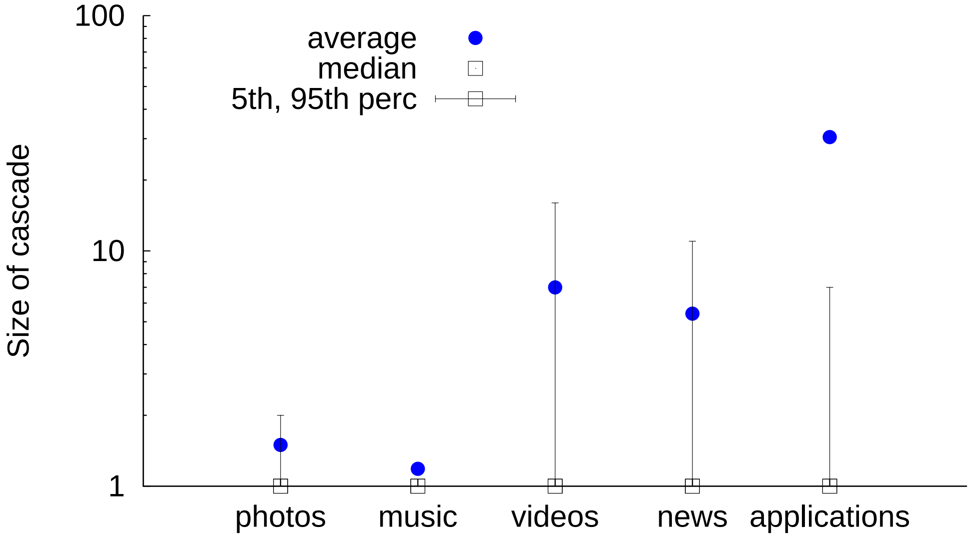


Figure 4.9. Size per content type

inferred by using the Google Geocoding API [API, 2012], which converts addresses or city names written in free text form into geographic coordinates of latitude and longitude. In total, the location of 1,096,804 users was identified. In the remainder of this section, only the network and the URL propagation patterns among these one million users were considered.

4.4.1 Content Producers and Consumers

Physical proximity between content producers and consumers in Twitter is first investigated. Here, a content producer represents a user who posts a URL independently of others (i.e., root nodes in any cascade tree) and a content consumer represents all other nodes in a cascade tree. Figure 4.10 shows the distribution of physical distance between any two users in the word-of-mouth URL propagation. Physical distance is computed based on the latitude and longitude information of two users. The distance is in units of 10km, representing a local community. The graph shows the probability distribution function for each distance d, which is the physical distance among all user pairs (u, v), where either (1) user u explicitly retweeted the URL that another user v shared or (2) user u follows another user v and shared the same URL after v posted it on Twitter. If either of these two conditions holds, we say that there is a *propagation* link from user u to user v.

For comparison purposes, Figure 4.10 also shows the distance distribution for two users who have a (unidirectional) follow link between them. We call this a *friendship*

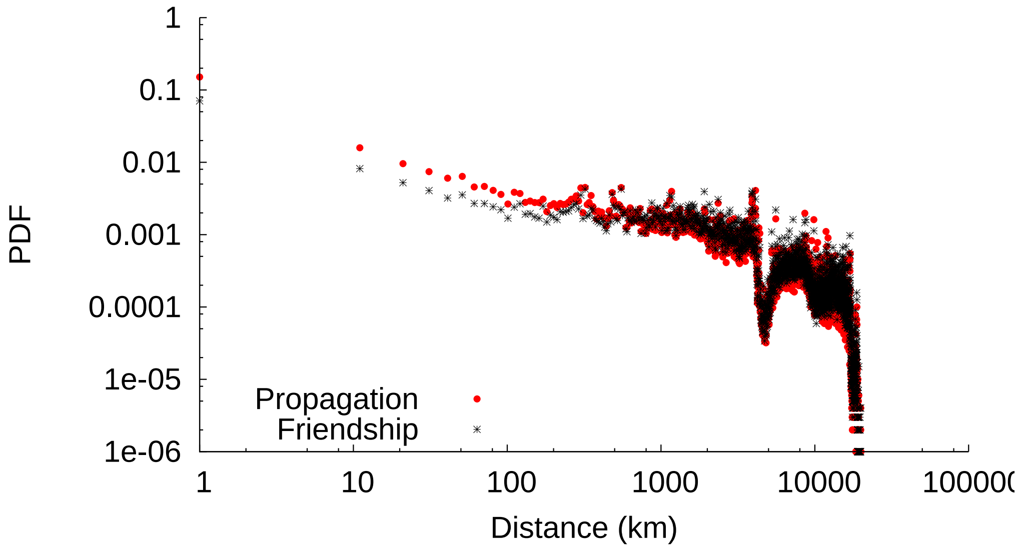


Figure 4.10. Physical distance of Twitter friendship links and URL propagation links

link. The friendship links represent the full potential of content distribution through word-of-mouth (i.e., when every follower actively reads or consumes the URL she receives from others).

A significant correlation between the content propagation probability and physical proximity of users is observed. That is, users within a short geographical distance (e.g., 10 km) have a higher probability of posting the same URL than those users who are physically located farther apart. The current OSM services infrastructures could exploit this physical proximity between content producers and consumers. Moreover, a significant correlation between the friendship and physical proximity is also observed. This is expected as users tend to interact more with other users who are physically nearby. Liben-Nowell et al. [2005] previously found a strong correlation between friendship and geographic location among LiveJournal users.

Interestingly, the correlation between the content propagation probability and physical proximity of users is slightly higher than that observed for having a friendship link. This might be explained by the fact that Twitter users follow not only their friends, but also media companies and celebrities, as well as distant users that post content that is valuable to them. However, when it comes to retweeting other users' messages, Twitter users chose tweets posted by those geographically nearby.

We tested if the level of locality changes according to the popularity of content producers and different content types. The first set of bars in Figure 4.11 shows the probability that content producers and consumers are located within a distance of

50 km, which roughly represents a large metropolitan area. In order to separate out the impact of producer popularity, content producers were grouped into three groups based on their in-degree. Also, in order to see the representativeness of the results, the location between all users was randomized by shuffling the location tags of users and computed the distance between them (shown as 'random' in the figure).

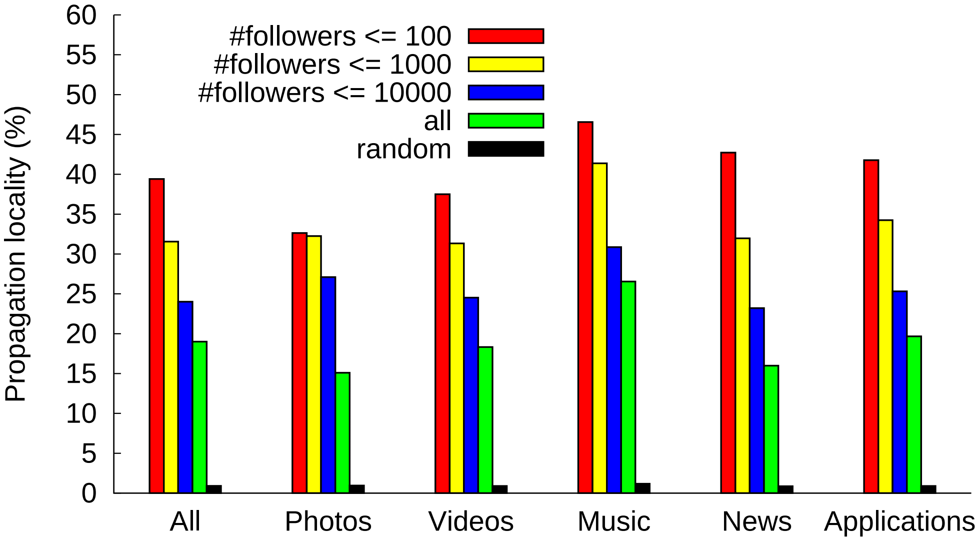


Figure 4.11. Distribution locality across content types

First, we focus on the impact of content producer's popularity on distribution locality and examine the first set of bars labeled 'All' content type in the x-axis. Overall, about 24% of the users who propagated content are physically close to very popular content producers who had more than 1,000 followers, 32% are close to content producers with between 100 and 1000 followers, and 39% are close to content producers with less than 100 followers. This result indicates that producers with a small number of followers tend to incur content propagations to geographically nearby locations. On the other hand, content producers with a large number of followers tend to be celebrities or well-known people and incur content propagations across wider areas.

Next, we focus on the impact of content types on distribution locality. Figure 4.11 also shows the result for different types of content (photos, videos, music, news and applications). Interestingly, the locality for photos is the lowest, while the locality for music is the highest. News and applications had much stronger local appeal than photos and videos. Overall, a non-negligible fraction (15-25%) of content propagated locally. The reasons for why certain content type has more local appeal than the others are not clear. We leave investigating these reasons as exciting future work.

Motivated by the overall high content locality at 1-hop users, we look at the distance a URL travels as it is further propagated by users within 2- and 3-hops away from the content creator. Figure 4.12 shows the geographical distance related to the content creator (initiator) as a function of the height level of spreaders on the hierarchical trees. Clearly, content tends to spread short distances on the first hops. As soon as friends-of-friends join the cascade and share the same URL, it reaches users located in different regions and, consequently, reaches distant locations.

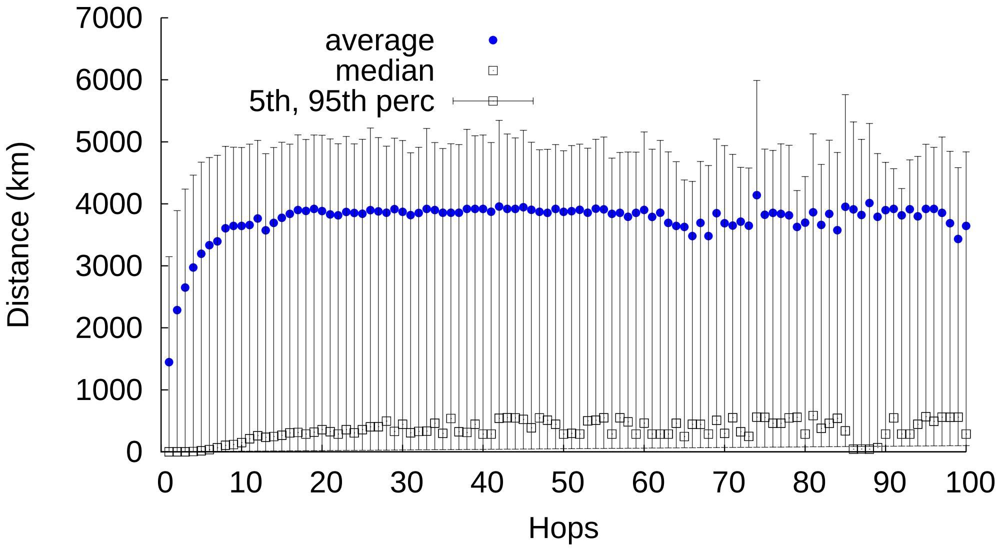


Figure 4.12. Distance between initiators and spreaders

The observations that social content produced by users with a small number of followers is usually consumed by users that are located within a small physical distance of the content creator could be exploited for caching design and content delivery networks.

Chapter 5

Cross-Pollination of Information in Online Social Media: a Case Study on Popular Social Networks

In this chapter we discuss how URLs generated in external OSM services diffuse on Twitter. The set of analysis discussed here were also published in Jain et al. [2011].

5.1 Methodology

Next, a description of the data collection framework used in this part of the characterization and some characteristics of the datasets collected are given.

5.1.1 Data Collection

The data collection framework is composed of two phases (see Figure 5.1). In the first phase, Twitter Streaming Application Program Interface (API) [Twitter, 2012b] is used to collect all tweets periodically, using a set of keywords. This step was part of a research project, developed by a Brazilian Research Institute, [1] which tracks information about important events in several social and traditional media sources, like newspapers, blogs, and online social networks. [2] After this step, all URLs that appear on the content of the tweets are filtered. Due to the usage of URL shorteners like *bit.ly* [Twitter, 2012a], all shortened URLs are expanded and all tweets with YouTube videos URLs, Flickr photos URLs, and Foursquare location URLs are filtered. Then, all tweets that contain

[1] *Instituto Nacional de Ciência e Tecnologia para a Web*, `http://www.inweb.org.br/`

[2] The *Observatório da Web* Project, `http://observatorio.inweb.org.br/`

these types of URLs are inserted into *Foreign meme Database* (FMDb). In the second phase, YouTube [YouTube, 2012], Flickr [Flickr, 2012], and Foursquare [Foursquare, 2012] APIs are used to collect information about the foreign memes and their uploaders, storing the same in *Objects Database* (ODb).

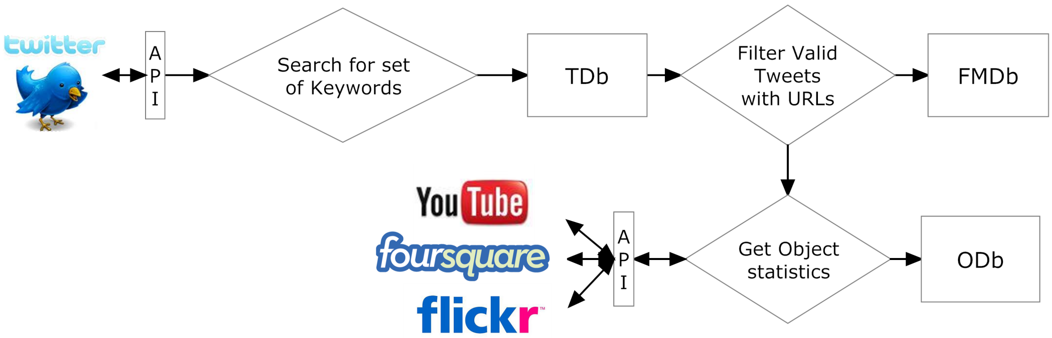


Figure 5.1. Data collection framework

Out of the most discussed topics on Twitter in 2010 [Twitter, 2010], a dataset for FIFA World Cup (FWC), a global event, was created. The FWC is an international football competition contested by the senior men's national teams of the members of Fédération Internationale de Football Association (FIFA), the sport's global governing body. The event happens every 4 years and in 2010 it took place in South Africa, from June 11th to July 11th. The FWC event was monitored from June 10th to July 12th, using 112 keywords (e.g. worldcup, FIFA and southafrica) in 7 different languages (like Portuguese, English and Spanish). To ensure no data loss, several redundant machines were used to collect the same data.

5.1.2 Datasets

Table 5.1 presents the descriptive statistics of the datasets used in this part of the characterization. A total of 34,306 unique videos URLs were shared on Twitter during the FWC, in a total of 141,118 tweets, posted by 88,231 users. The videos were uploaded by 26,026 YouTube users. Table 5.1 also presents statistics about Foursquare and Flickr datasets, which are less popular than YouTube on Twitter, but still have a representative number of URLs to study. A *baseline* dataset, which was created containing only local memes, is used in several analyses to contrast the characteristics of Cross-Pollinated networks with the characteristics of Twitter itself. In total, the baseline dataset has more than 29 million tweets, created by 3.5 million users. This comparison helps in understanding how the introduction of foreign memes affects the diffusion OSM.

Source OSM (SOSM)	URLs	Tweets	Twitter Users	SOSM Users
YouTube	34,306	141,118	88,231	26,026
Foursquare	14,896	23,252	14,401	-
Flickr	1,719	2,560	1,419	711
Baseline	-	29,038,497	3,511,044	-

Table 5.1. Descriptive statistics of the datasets

In order to verify the representativeness of the datasets, all the analysis were repeated using keywords related to another popular event in 2010 on Twitter — the Brazilian Presidential Election. [3] This event was especially monitored during the candidate's campaign, which started on July 6^{th} and ended on October 31^{st}, the final election day. A set of 30 keywords (e.g. dilma, serra and marinasilva) related to the candidates and their political parties was used to monitor this event. For a better readability, only the results for FWC datasets are presented, but most of the conclusions hold for the Brazilian Presidential Election datasets as well.

5.2 Cross-Pollination Characteristics

In this section a key question about Cross-Pollination is investigated: What are the characteristics of Cross-Pollination? We start with some temporal characteristics, and then we analyze some topological characteristics.

5.2.1 Sharing Activity

An important temporal characteristic of Cross-Pollination is the volume of tweets generated by foreign memes on a given day during a certain period of time. Figure 5.2 shows the total number of tweets with foreign memes created on each day during the FWC event. For comparison purposes, the figure also shows the total number of tweets with local memes created per day (using baseline dataset). During the whole period, a similar trend is observed, for all datasets analyzed. The trend of volume of tweets created due to meme (both foreign and local) sharing is relatively uniform and similar during the whole period, with small peaks occurring on the same days. Hence, foreign meme sharing activity follows local meme sharing activity, although absolute numbers

[3] Dilma Rouseff, elected president of Brazil, was the second most cited person on Twitter in 2010.

differ significantly (around 10^3 YouTube foreign memes on Twitter and 10^6 local memes on baseline dataset).

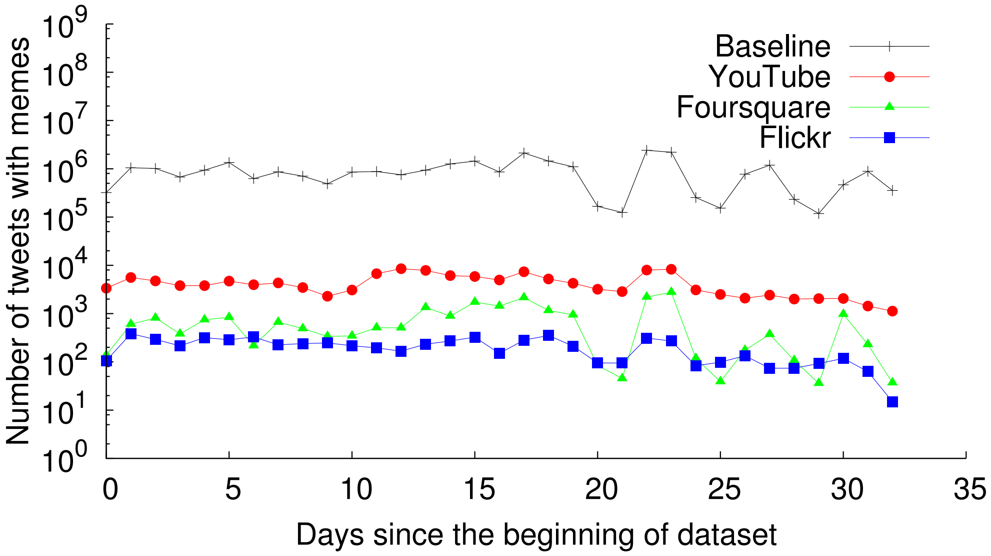


Figure 5.2. Foreign and local meme creation over time

5.2.2 User Participation

In order to verify whether users contribute equally in the traffic generated by Cross-Pollination on Twitter, *User Participation* (UP) was defined as the average number of tweets with memes created per day, for each user. Users were divided into bins according to their UP, and then the percentage of users in each bin is calculated (see Figure 5.3). Users contribute equally for the traffic generated by Cross-Pollinated networks; vast majority of users (more than 90%) are in the same bin, with less than 2 tweets with foreign memes created per day. Furthermore, Cross-Pollinated networks follow the diffusion OSM in this aspect, as the vast majority (more than 70%) of users from the baseline dataset is on the same bin. The same observation can be done for Flickr photos and Foursquare locations.

5.2.3 Diffusion Delay

Diffusion delay is defined as the time between the creation of a tweet and its retweet. Figure 5.4 shows the complementary cumulative distribution function (CCDF) for the diffusion delay of the three Cross-Pollinated networks studied and the baseline dataset. On average, 75% of the memes is retweeted in less than 1 hour, and 97% is retweeted

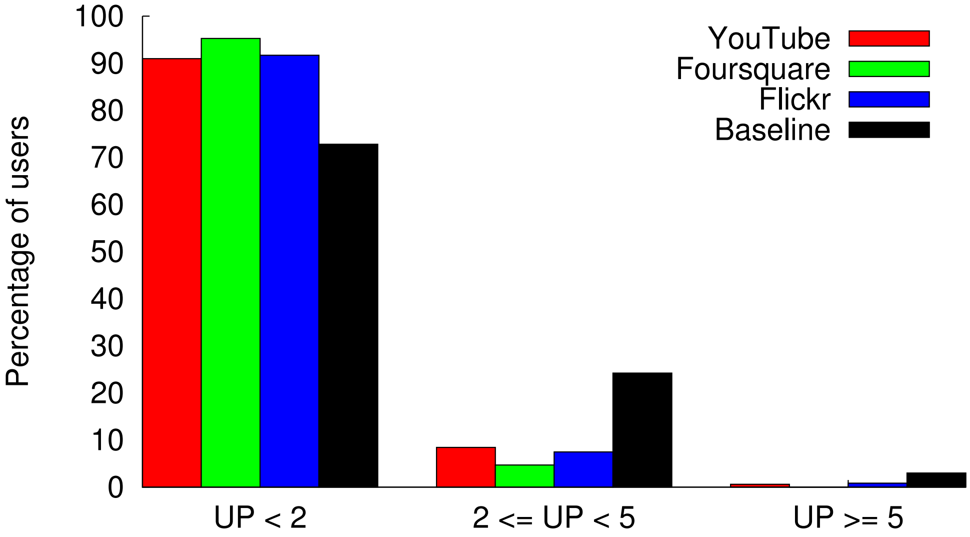


Figure 5.3. User participation (UP) in meme creation

within a day. Note that YouTube and Flickr memes tend to be retweeted with a slightly higher delay than Foursquare and local memes. For example, around 50% of tweets with YouTube and Flickr memes have a delay larger than 1,000 seconds (around 16 minutes), while 30% of retweets from Foursquare and local memes have a delay larger than 1,000 seconds. Nature of the content is a reasonable explanation for this difference. A user can easily read and quickly respond a direct message (local meme), while a foreign meme becomes an indirect message as the user is expected to view the content of the URL before forwarding it. In this case, Foursquare memes are more similar to local memes because they are usually automatically posted messages which contain the name of the place from where the user "checked in" together with the URL of the location. "I am at DCC, UFMG `http://4sq.com/XyZw`" is an example of this kind of tweet.

We now turn our focus to analyze topological characteristics of Cross-Pollinated networks.

5.2.4 Information Cascades

Figure 5.5 shows distribution of number of cascades with cascade size. Out of the cascades formed by foreign memes diffused, most cascades are composed of only one initiator and one spreader (i.e., of cascade size 2), which is termed as one level of diffusion. There are only few cascades which have large cascade size, reaching many users and users' followers. For foreign memes that were diffused, the level of diffusion

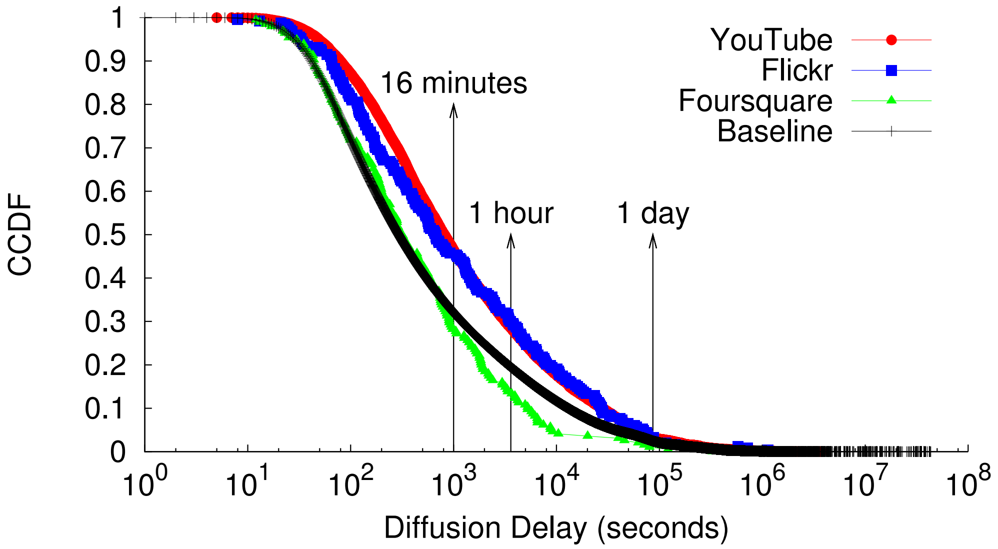


Figure 5.4. Diffusion delay of memes

remains to only one user. Local memes have a similar distribution. Number of cascades follow 90-10 Pareto distribution with 90% cascades with size ≤ 3 and 10% cascades with size ≥ 3, for both Cross-Pollinated network and baseline.

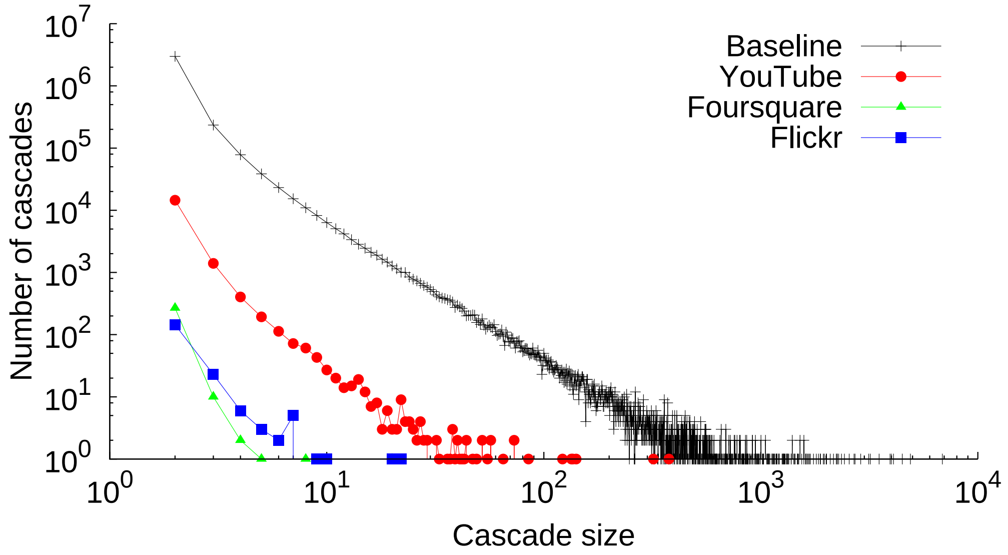


Figure 5.5. Distribution of cascades with cascade size.

Table 5.2 shows a comparison of some graph metrics for information cascades of the three Cross-Pollinated networks in study and the baseline dataset (all numbers showed are averages). Twitter users are most attracted towards posting and forwarding

YouTube videos than Flickr photos, Foursquare locations and local memes (highest # spreaders / meme). Even then, note that average cascade size for YouTube remains close to the other datasets. A large number of small cascades neutralize larger cascade sizes. Average in- and out-degree for Cross-Pollinated networks are higher than baseline and close to each other. Hence, Cross-Pollinated networks behave similarly, irrespective of the type of foreign meme diffused.

Source OSM	D	IN	OUT	PL	NCM	CS	NSM
YouTube	1.06	1.17	1.12	0.37	2.81	2.53	7.08
Flickr	1.11	1.06	1.48	0.43	1.11	2.69	2.97
Foursquare	1.03	1.09	1.06	0.48	1.02	2.13	2.18
Baseline	1.07	0.53	0.53	-	1.00	2.78	2.78

Table 5.2. Information cascade statistics for three Cross-Pollinated networks and baseline. Legend: D = degree, IN = in-degree, OUT = out-degree, PL = path length, NCM = # cascades / meme, CS = cascade size, NSM = # spreaders / meme

5.3 Relationship Between OSM Services in a Cross-Pollinated Network

Two important questions related with the relationship between the OSM services involved in a Cross-Pollinated network are answered in this section: (1) Does Cross-Pollination across OSM services help to increase the audience reached by the information diffused? (2) Is popularity of foreign meme on source/diffusion OSM a factor affecting its popularity on diffusion/source OSM?

The popularity of a video on YouTube is measured by its view count. On Flickr, the popularity of a photo is also measured by its view count. The popularity of a location on Foursquare is measured by the number of "check-ins." On Twitter, the popularity of a foreign meme is given by the number of tweets with it. We obtained the popularity ranking of foreign memes in both the source OSM and the diffusion OSM. In order to compare two rankings, we used the *Kendall's Tau coefficient* [Kendall, 1938], which is a measure of the rank correlation, denoted by τ. A τ of -0.0001 is observed on the Cross-Pollinated network between YouTube and Twitter, which demonstrates that the ranking of popularity of videos in both OSM services are independent. In other words, if a video is popular on YouTube does not mean that it will also be popular on Twitter, and vice-versa. Interestingly, the same observation was found for Flickr and

Foursquare datasets, as the τ is -0.0013 and -0.0001, respectively. Hence, popularity of foreign meme in source OSM does not influence its popularity in diffusion OSM.

Another interesting aspect to analyze is whether the diffusion of foreign memes in the diffusion OSM helps in increasing the traffic (popularity of memes) in the source OSM. Although our datasets do not have information about how many clicks each URL received on Twitter, some URL shorteners provide APIs with statistics of access to their links. One of the most popular services is *bit.ly*, which provides an API [Bit.ly, 2012b]. This API was used to analyze how many clicks each meme shortened in a *bit.ly* URL received from Twitter. In order to do this analysis, data for all *bit.ly* URLs of our dataset was collected, and then checked how many clicks they received from the referrer *twitter.com*. In total, there are 13,158 videos from YouTube dataset (38.4% of total) and 1,719 photos from Flickr dataset (38.2% of total) shortened with *bit.ly*. Foursquare dataset was not considered as we had only 37 *bit.ly* URLs, which is not representative. [4]

We then analyze the fraction of views [5] that each foreign meme (videos and photos) received from Twitter. We observe low fractions of views from Twitter, for many foreign memes tweeted. About 97% of the videos received no more than 1% of their views from Twitter, and almost 59% of the photos received no more than 1% of their views from Twitter. Twitter does not seem to be effective in increasing the popularity of foreign meme on the source OSM. By including only clicks from *bit.ly* URLs, we have a lower bound of the fraction of views that came from Twitter. There can be various other sources contributing to number of views for a foreign meme which we did not analyze here.

From the above analysis, it is observed that popularity of information on one OSM does not imply / affect its popularity on other OSM, and vice versa.

[4] Foursquare has its own URL shortener *4sq.com*, which might be the reason for a small number of *bit.ly* URLs in our dataset.

[5] We consider that each click on the URL represents one view in the source OSM.

Chapter 6

Developing a System to Monitor Information in Online Social Media

In this chapter we explain the architecture proposed for MIOSphere, a system to monitor information in the Online Social Media Sphere. MIOSphere was developed to monitor the diffusion of URLs in OSM services, but it can be easily extended to monitor any meme.

6.1 Lessons Learned

In the characterization work (Chapters 4 and 5) several findings were crucial for the development of MIOSphere. The most important lessons learned are described below:

1. All contents have a chance to become popular.

   - At some point in time, all memes should be scheduled to be searched. It is not easy to predict which URL will become popular, as it is not uncommon to see extremely popular URLs from unpopular domains, or posted by unpopular initiators.

2. Large cascades are rare, but they are extremely large when they occur.

   - Large cascades should receive priority when scheduled to be searched. Although the volume of URLs posted per day is extremely large, the fraction of popular URLs is small, so it is important to detect and give priority to the most popular URLs in order to avoid losing information about them.

3. A single initiator's role is likely limited, and popular content usually propagates through several different propagation trees.

    - There is a strong correlation between the total size of the audience and the number of initiators, which means that the most popular URLs are posted by several independent users in the social graph. Moreover, the relative size of the largest propagation tree is marginal. So, a priori, it does not seem to be helpful to always prioritize URLs initially posted by popular users.

4. The vast majority of tweets with URLs are retweeted in less than 1 hour.

    - On average, 75% of the tweets with URLs are retweeted in less than 1 hour, and 97% are retweeted within a day. As most part of the URLs is unpopular, and most part of the retweets occurs within a day, the system can detect when a URL stopped to being posted.

5. The relationship between OSM services is weak.

    - The crawler can run separated for each OSM service monitored. The diffusion follows the characteristics of the diffusion OSM only, so it is better to run a separated crawler for each OSM service, setting up parameters according to the characteristics of the OSM service.

Another set of lessons were not used in MIOSphere but might be useful to investigate as future work:

1. Cascade trees on Twitter are likely shallow and wide.

    - A visualization revealed that the large cascade trees occasionally included bursts at all height levels, i.e., the branch out factor is suddenly large at particular spreaders. Identifying the important spreaders in the cascade, and the height level might be useful to predict when a given URL is about to explode in popularity.

2. The type of content affects the potential of the eventual cascade size.

    - News, applications, and videos have a higher probability of involving 10 or more users in spreading than photos and music. Identifying the content of a URL might help to predict and adopt different policies while scheduling.

3. Content is diffused locally, specially at initial hops.

- There is a higher probability of users within a short geographical distance (e.g., 10 km) posting the same URL than those users who are physically located farther apart. Moreover, content tends to spread short distances only on the first hops. As soon as friends-of-friends join the cascade and share the same URL, it reaches users located in different regions and, consequently, reaches distant locations. Exploring this content locality might be useful for a distributed crawler.

4. Content published by celebrities or well-known people have a higher chance of propagating across wider areas.

    - Initiators with a small number of followers tend to incur content propagations to geographically nearby locations. On the other hand, initiators with a large number of followers tend to be celebrities or well-known people and incur content propagations across wider areas. Identifying and categorizing users might help in scheduling in a distributed crawler.

Next, we propose and explain the architecture used in MIOSphere, where the influence of these lessons will be clear.

6.2 Architecture

A typical Web crawler is composed by three main components: a *scheduler* to maintain a queue of URLs to be visited, a *downloader* to download the Web pages visited, and a *storage* to index and return information about the Web pages to the scheduler [Baeza-Yates and Ribeiro-Neto, 2011]. MIOSphere is a focused Web crawler, so this architecture was chosen for it (see Figure 6.1). The scheduler is responsible for choosing which URLs should be collected at a given time, based on a importance metric calculated for each URL. The downloader is responsible for downloading information about the monitored URLs. The storage is responsible for storing the collected data.

MIOSphere has a distributed architecture composed by one *master* and several *slave* machines. The master is responsible for controlling the whole process, while the slaves just receive URLs from the master and search for new posts with them in the OSM services monitored. The workflow of the system is explained below (steps of Figure 6.1):

1. The scheduler requests URLs available to collect from the storage

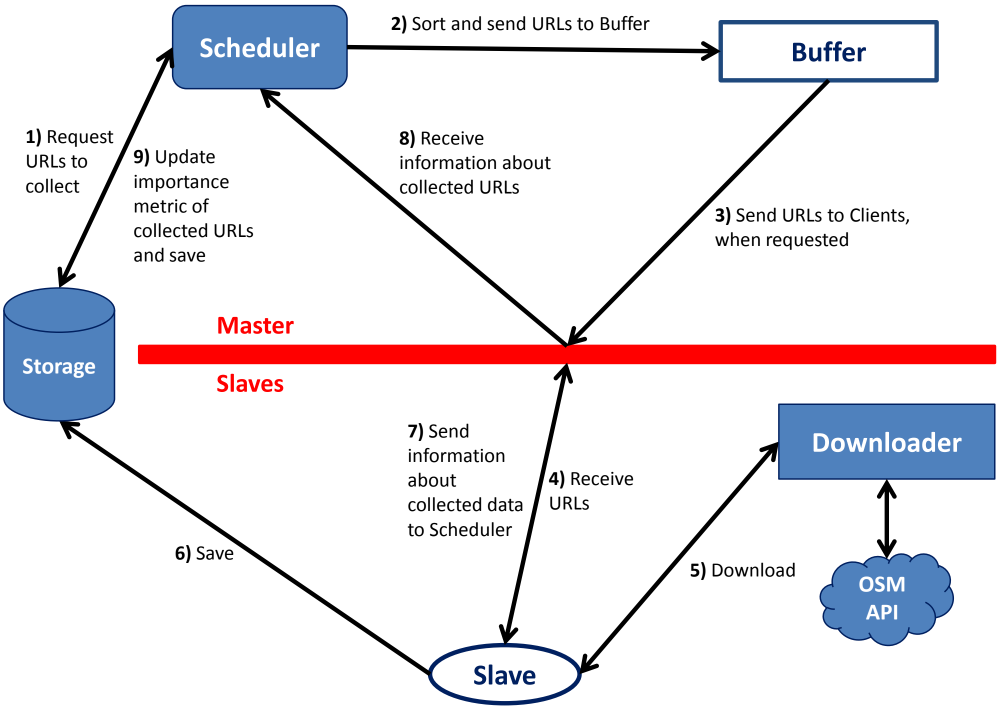

Scheduler
2) Sort and send URLs to Buffer
Buffer
1) Request URLs to collect
9) Update importance metric of collected URLs and save
8) Receive information about collected URLs
3) Send URLs to Clients, when requested
Storage
Master
Slaves
7) Send information about collected data to Scheduler
4) Receive URLs
Downloader
6) Save
5) Download
OSM API
Slave

Figure 6.1. MIOSphere's architecture and workflow

2. The scheduler order the URLs received according to their importance metric, adding them into a buffer
3. When requested by a slave, URLs from the buffer are sent to be collected
4. The slave receives the URLs to collect
5. The downloader collects information about the URLs using the OSM service Application Programming Interface (API)
6. The slave saves the information collected in the storage
7. The slave sends back to the scheduler statistics about the collected data
8. The scheduler receives information about the collected URLs
9. The scheduler updates the importance metric (see Section 6.2.1.2) of each URL collected and save it on the storage

The master is constantly checking if there is some free space in the buffer, requesting more URLs from the storage if necessary, and then the cycle starts again on step 1.

In Chapter 5 it was learned that the relationship between the OSM services is weak. Based on this observation, MIOSphere runs separately for each OSM service monitored, but sharing the storage component.

6.2.1 Scheduler

The scheduler is the most important component of MIOSphere's architecture, for several reasons. First, millions of URLs are shared everyday by OSM users, and most of them are unpopular, i.e., the popular ones should receive a priority to avoid losing information about them. Second, there are several limitations in any crawling process, like the bandwidth available and the politeness to avoid overloading Web servers. For instance, most OSM APIs impose limits the number of requests made in a given time window. Third, the system must keep updated information about the URLs monitored, i.e., it must have a good scheduling policy to optimize the usage of resources. A bad scheduling policy, for example, can frequently lose requests by always selecting unpopular URLs or URLs which already have updated information. In summary, the scheduler is like the brain of MIOSphere, and has a strong impact on the system's performance.

6.2.1.1 Scheduling Policies

A scheduling policy determines the rules and internal organization of the scheduler to select which URLs should be collected at a given point in time. MIOsphere's scheduler is organized into *Multiple Priority Queues* (MPQ). In this organization, the URLs being monitored are divided into multiple priority queues according to their importance at a given time. MIOSphere has a priority queue for new URLs, which just entered in the system, and a set of queues dividing the URLs by their popularity. When getting URLs from the storage to insert into the buffer, the scheduler will get a number of URLs from each priority queue.

Using queues is a good approach to the problem as it is assured that all URLs will be selected at some point in time. Dividing the URLs according to their popularity is also a good policy because popular URLs will be selected more frequently, as their priority queue will be smaller. On the other hand, popular URLs might be quickly selected due to the smaller priority queue and bring a problem. MIOSphere may lose a request if it quickly selects a given URL, as the interval between two consecutive requests might not be enough to find new posts with that URL. A parameter with a minimum amount of time to wait in a priority queue was set up to avoid this problem.

The scheduler also has a policy to reduce the number of URLs being monitored. When there are no new posts with a given URL for a given period of time, the scheduler considers that URL as dead, removing it from the priority queues. In Chapter 4 we found that most URLs are unpopular (90% have only 1 tweet) and quickly retweeted (75% of the retweets occur in less than 1 hour), so keeping these URLs in the priority queues would just result in loss of resources, i.e., requests without any new result, space in the queue, and bandwidth. A parameter to set up the period of time without new posts to consider a URL as dead was created for this purpose.

6.2.1.2 Importance Metric

In order to divide the URLs into the multiple priority queues, the scheduler calculates an importance metric for each monitored URL. The importance metric $IM_{u,t}$ of a URL u is the number of results found for u in a sliding time window with duration t. In other words, the importance metric of a URL is the number of new posts collected with it in the last hour, for example.

The size of the sliding time window is a parameter of MIOSphere, which enables the system to handle with the dynamics of each OSM service. Moreover, most URLs call attention of users at a limited period of time, so using a sliding time window to consider the popularity of the URLs avoids giving priority to URLs that were popular in the past but are already forgotten by users.

6.2.2 Algorithm

MIOSphere's algorithm is explained here. Algorithm 1 is the algorithm of the master, and Algorithm 2 is the algorithm of the slaves. Both are easy to understand, as they follow the workflow presented on Figure 6.1.

6.2.2.1 Master

The master stays in a loop waiting for requests from the slaves and scheduling the URLs to send for each slave to collect. The buffer is initialized in the beginning of Algorithm 1. The master checks if the buffer has free space, and then reads m URLs from all priority queues, sort then in decreasing order based on their importance metric and insert into the buffer (lines 4-9). If the master receives a request from a slave, it will send n URLs from the buffer to the slave collect (lines 11-12). When the slave sends back to the master some statistics about the collected data, the scheduler will

Algorithm 1 MIOSphere Algorithm - Master

```
1:  buffer ⇐ empty
2:  while True do
3:      {Copy new URLs from storage, sort and add into the buffer}
4:      if buffer.size() < $k$∗buffer.maxsize() then
5:          {threshold k is a fraction where $0 < k \leq 1$}
6:          list_urls = read($m$,all_queues) {Read $m$ URLs, from all queues}
7:          sorted_list = reorder_queue(list_urls) {Sort URLs based on their importance
            metric}
8:          enqueue(buffer,list_urls) {Insert URLs on buffer}
9:      end if
10:     {Slave is requesting URLs to collect}
11:     if request(slave) then
12:         send(dequeue(buffer,$n$)) {Send the next $n$ URLs in buffer to slave collect}
13:     end if
14:     {Slave is sending statistics about collected URLs}
15:     if receive(slave,stats) then
16:         for all URL $u$ collected do
17:             $i$ = update_importance_metric($u$,stats)
18:             save($u$,$i$,appropriated_queue) {Insert $u$ in appropriated queue based on $u$'s
                updated importance metric $i$}
19:         end for
20:     end if
21: end while
```

receive and update the importance metric for each URL collected, inserting the URLs back in the appropriated priority queue (lines 15-20).

6.2.2.2 Slaves

Algorithm 2 MIOSphere Algorithm - Slaves

```
while True do
    info ⇐ empty
    list_urls = request(master) {Request URLs to collect from the master}
    for all URL $u$ in list_urls do
        buffer = download($u$) {Collect information about $u$}
        stats = calculate_stats(buffer) {Calculate some popularity statistics of $u$}
        enqueue(info,($u$,stats))
        save($u$,info) {Save collected information on storage}
    end for
    {Send information about collected URLs to master}
    send(master,info)
end while
```

The slave stays in a loop requesting URLs to collect. A queue (info) is initialized to save information about the collected data, on line 2 of Algorithm 2. The slave sends a request for the master and waits for URLs to collect (line 3). Then, the slave users the OSM service API to search for new posts with each URL received, calculates some popularity statistics about the data collected and save in the storage (lines 4-9). After collecting all URLs, the slave sends to the master the statistics about the collected data (line 11).

6.3 Simulations

As a tool to help in the development of MIOSphere, a trace-driven simulator was implemented to test several hypotheses and scheduling policies. A trace-driven simulator users a time-ordered record of events from a real system, i.e., a trace, as its input [Jain, 1991]. In this section we explain details about the simulator and analyze some results to show how the performance of the system varies with several configurations of its parameters.

The simulator was written in Python [Language, 2012] and uses the same architecture proposed and explained in Section 6.2, except that it does not run in parallel. The simulator works in cycles, simulating the action of the master and each slave. Several architectures, scheduling policies and parameters were experienced during the development until reach the final version proposed for MIOSphere.

6.3.1 Traces

Traces from different periods were created from the Twitter dataset gathered in Cha et al. [2010], the same dataset used in the characterization presented in Chapter 4. The traces are composed of all tweets posted with URLs in a period of time. Table 6.1 shows some statistics about the traces.

	Period	**URLs**	**Tweets**
Trace A	Feb 15, 2009 — Feb 21, 2009	2,489,153	3,289,856
Trace B	Mar 1, 2009 — Mar 7, 2009	3,096,624	4,187,365
Trace C	Mar 16, 2009 — Mar 30, 2009	8,254,440	11,658,592
Trace D	Apr 1, 2009 — Apr 15, 2009	10,005,374	14,082,999

Table 6.1. Statistics of the traces utilized in the simulations

Note that for a better evaluation of the system, we created traces from different periods with different durations.

6.3.2 Assumptions

In any simulation process is necessary to do some assumptions. In this work, the duration of each cycle is considered as 1 second. All tweets are collected correctly, i.e., the API and the network will not fail. In other words, all tweets the simulator fails to collect is due to not scheduling the URLs appropriately. The time taken to collect each URL is assumed to be 1 cycle.

Before starting the simulations, the first URL in the sample is set to post time 0. The post time of the other tweets with URLs is the number of seconds after time 0. A URL will start to being tracked when the current cycle in the simulation is greater than or equal to the post time of the first tweet with it. For example, if a URL u was first tweeted on time 10, only after cycle 10 u will be available to be collected.

6.3.3 Evaluation Metrics

Some metrics were defined to evaluate the performance of the simulated system:

- **Tweet Coverage (TC):** the percentage of all tweets with the monitored URLs which were collected

- **URL Coverage (UC):** the percentage of URLs that have at least one tweet collected

- **URL Precision (UP):** the percentage of URLs that have all tweets collected

- **Correlation ($CORR$):** the Pearson's correlation coefficient between the total number of tweets and the total number of tweets collected for each URL

Note that most of the proposed metrics are similar to popular evaluation metrics for Information Retrieval algorithms — precision and coverage [Baeza-Yates and Ribeiro-Neto, 2011]. TC captures what fraction of the total available data the system was able to collect. UC and UP capture what fraction of the URLs the system was able to collect partially and completely, respectively. $CORR$ is used to evaluate whether MIOSphere was able to collect more data from the most popular URLs. The higher $CORR$ is, the best the performance is, but $CORR$ must be analyzed as a complement of the other evaluation metrics. For instance, consider a scenario A where $TC = 10\%$ and $CORR = 1.0$, and a scenario B where $TC = 90\%$ and $CORR = 0.8$. In this case, the performance of B is much better than A, as it was able to collect much more tweets with a high $CORR$.

6.3.4 Parameters

The simulator has several parameters to adapt the system for the dynamics and characteristics of each OSM service monitored. Most parameters are part of MIOSphere's architecture, and a small part is specific of the simulator.

The parameters are divided into two groups: *fixed* and *variable*. The fixed set of parameters is composed by those parameters which do not have their valued changed during the simulations. Most fixed parameters are dependent of the resources available and cannot be changed in a real implementation. Other fixed parameters are assumptions we had to make and some parameters not related with the scheduler. On the other hand, the variable set of parameters is composed by those parameters which have their value changed during the simulations. All of them are related with the scheduler policies which we want to evaluate. A description of the fixed and variable parameters is provided below:

- ***Fixed:***
  - **Number of URLs to each slave collect ($NUSC$):** the number of URLs to send for each slave collect
  - **Maximum number of results (MNR):** the maximum number of tweets returned when requesting data
  - **Maximum number of requests ($MNRQ$):** the maximum number of requests each slave can make during a given period of time
  - **Period $MNRQ$ ($PMNRQ$):** the period of time to consider $MNRQ$
  - **Time to collect data (TCD):** the amount of time that each slave takes to collect data, given in number of cycles of the simulator
  - **Buffer size (BS):** the size of the buffer component, given in number of URLs to store
  - **Minimum free space buffer ($MFSB$):** the minimum amount of free space in the buffer in order to fill it with new URLs from the database
  - **Number of slaves (NS):** the number of slaves to collect data
- ***Variable:***
  - **Importance sliding window size ($ISWS$):** the period of time to consider in the calculus of the importance metric
  - **Dead threshold (DT):** the amount of time to consider a URL as dead

- **Waiting time queue (WTQ):** the minimum period of time each URL has to wait in the priority queue before being sent to the buffer
- **Number of priority queues (NPQ):** the number of priority queues used
- **Priority queues thresholds (PQT):** thresholds to divide URLs in the priority queues

All time periods are in seconds. $MNR = 2,000$, $MNRQ = 350$, and $PMNRQ = 3,600$ were adjusted according to the limits imposed by the official Twitter API. TCD is the number of cycles taken to collect, fixed at 1 cycle in the simulations. $MFSB$ was fixed at 4,000 and BS was fixed at 5,000. NS was fixed at 100, and $NUSC$ was fixed at 25.

6.3.5 Analysis

A *simple design* is the experimental design chosen to perform and analyze the simulations. In this experimental design, we start with a typical configuration of parameters and vary one parameter at a time to evaluate how that parameter affects the performance [Jain, 1991].

We are aware that, this simple design does not make the best use of the effort spent and it is not statistically efficient, but the goal of the simulator is just to help in the development of MIOSphere, as we do not need to download the data from the Web for each scheduler policy we want to test. Moreover, the simple design enables us to analyze changes in the performance due to the variation of one independent parameter.

6.3.5.1 Dead Threshold (DT)

The DT parameter is the time to consider a URL as dead, i.e., when there are no new tweets with that URL for a determined period of time. DT is varied from 6 hours (21,600 seconds) to 7 days (604,800 seconds), while the other variable parameters are fixed at the following values: $WTQ = 3,600$, $ISWS = 3,600$, $NPQ = 2$, $PQT = [100]$. Table 6.2 shows the results, which are averages from all the traces.

Small variations were observed for different DT values (the same is true for each trace separately). Note that the best results were achieved when $DT = 86,400$, i.e., if we remove from the priority queues URLs which were not tweeted for 24 hours. This result is not surprising as the vast majority of the URLs have only a few tweets (see Chapter 4). Furthermore, a recent blog post showed that, on Twitter, a URL receives half of its clicks in only 2.8 hours, on average [Bit.ly, 2011]. Note that using a higher

DT	TC	UC	UP	CORR
21,600	78.4	84.2	82.1	0.7893
43,200	79.5	84.0	82.4	0.8112
86,400	79.7	83.5	82.2	0.8154
259,200	78.0	82.3	81.0	0.8097
604,800	77.6	82.1	80.7	0.8055

Table 6.2. Dead Threshold results (average of all datasets)

DT value is not good as the queues will have more URLs possibly already forgotten by users (the performance starts to decrease after $DT = 86,400$).

For comparison purposes, a *baseline* approach where the scheduler simply stores all URLs monitored in a single queue was implemented. Table 6.3 shows the results.

DT	TC	UC	UP	CORR
21,600	79.1	84.7	82.8	0.5590
43,200	78.7	83.0	81.5	0.5558
86,400	75.3	78.4	77.5	0.5514
259,200	66.8	69.1	68.5	0.4469
604,800	61.2	62.7	62.2	0.4421

Table 6.3. Dead Threshold results (average of all datasets) for the baseline scheduler policy

In the baseline approach, we observe a decreasing tendency in the performance as *DT* values are increased. Using high *DT* values, a large number of already dead URLs will be kept on the queue, which is unnecessary and waists requests, space, bandwidth, and time.

Comparing *TC*, *UC*, and *UP* values in tables 6.2 and 6.3, note that, surprisingly, the baseline approach achieves a good performance with $DT = 21,600$. However, when comparing *CORR* values, we note that the performance of MPQ is much better. Figure 6.2 shows a scatter plot between the number of tweets collected and the total number of tweets, for all URLs taken from Trace D with $DT = 86,400$ for the MPQ Scheduler (left), and $DT = 21,600$ for the Baseline approach (right). The same pattern is observed for traces A, B and C as well.

In some scenarios the baseline approach could collect more URLs than MPQ scheduling policy (*UC* slightly larger), but losing more tweets from the most popular URLs (see Figure 6.2). As the goal of MIOSphere is to track the diffusion of URLs, losing more information from the most diffused URLs is not a good scenario.

Another interesting point to note is that the performance of the MPQ scheduling policy is much better than the baseline with higher *DT* values. Waiting more time

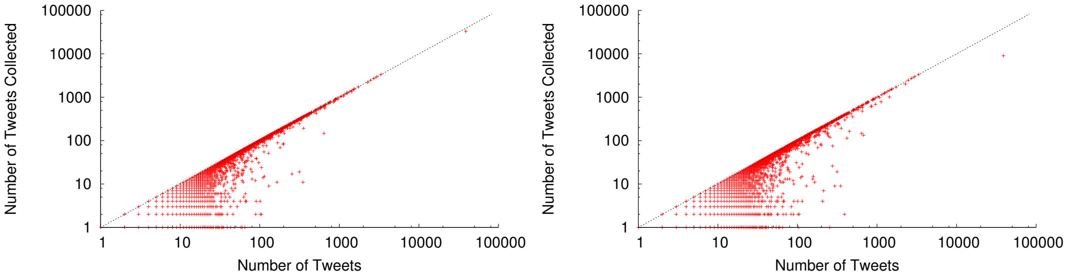


Figure 6.2. Number of tweets collected versus total number of tweets, for each URL of Trace D, using MPQ (left) and the baseline scheduling (right).

before considering a URL as dead may be good to avoid killing URLs that have a "cold start", i.e., take more time to start being tweeted after the first tweet, or those URLs that diffuse slowly.

6.3.5.2 Waiting Time Queue (WTQ)

Waiting some time before scheduling a URL to search for new posts might be interesting for the system's performance because users might take some time to diffuse a URL and the OSM service API might also take some time to process it. The purpose of WTQ parameter is to avoid scheduling the most popular URLs quickly, as the high priority queues are usually small. In order to verify this, WTQ was varied from 1 minute (60 seconds) to 12 hours (43,200 seconds). The other variable parameters were fixed at the following values: $DT = 259,200$, $ISWS = 3,600$, $NPQ = 2$, $PQT = [100]$. Table 6.4 shows the results, which are averages from all traces. All the conclusions hold for all traces individually.

WTQ	TC	UC	UP	CORR
60	76.9	81.9	80.6	0.5979
1,800	77.7	82.2	80.8	0.7936
3,600	78.0	82.3	81.0	0.8097
21,600	76.7	80.5	79.4	0.6987
43,200	75.6	79.0	78.0	0.6600

Table 6.4. Waiting Time Queue results (average of all datasets)

The best performance was achieved with $WTQ = 3,600$ seconds (1 hour). Small WTQ values are not efficient because a URL might be quickly selected by the scheduler, without having time to be tweeted by more users. Large WTQ values are also not efficient because the URLs might take a long time to be selected by the scheduler, which may result in losing posts with them (Twitter API only returns a maximum of 2,000 results of tweets which are less than 7 days old).

Note that if the volume of data is huge the WTQ will not affect the performance, as the priority queues will be larger and a URL will take more time to reach the first positions in the queue. WTQ is effective in the beginning of the simulations, when the queues are empty.

6.3.5.3 Importance Sliding Window Size ($ISWS$)

The purpose of the $ISWS$ parameter is to capture the dynamics of the OSM service monitored. In Section 6.2.1.2, the importance metric ($IM_{u,t}$) of a URL u was defined as the number of results found for u in a sliding time window with duration t. $ISWS$ is the duration t, which was varied from 1 hour (3,600 seconds) to 7 days (604,800 seconds). The other variable parameters were fixed at the following values: $DT = 259,200$, $WTQ = 3,600$, $NPQ = 2$, $PQT = [100]$. Table 6.5 shows the results, which are averages from all traces. All the conclusions hold for all traces individually.

ISWS	TC	UC	UP	CORR
3,600	78.0	82.3	81.0	0.8097
43,200	77.9	82.3	81.0	0.6153
86,400	78.0	82.3	81.0	0.8040
259,200	78.0	82.2	80.9	0.8087
604,800	78.0	82.1	80.8	0.8012

Table 6.5. Importance Sliding Window Size results (average of all datasets)

The performance is not significantly affected when $ISWS$ is varied. The reason is that most part of the URLs are unpopular (90% have only 1 tweet) and are quickly retweeted (75% in less than 1 hour). Based on this observation, a real implementation could calculate the importance metric of a URL by simply considering the number of results returned in the last attempt to collect posts with it. This is good for a real-time system because the calculus is straightforward, without any additional memory or CPU usage.

We believe $ISWS$ may be useful to handle with the dynamics of other OSM services with different diffusion patterns, but testing this hypotheses was left for future work.

6.3.5.4 Number of Priority Queues (NPQ) and Priority Queues Thresholds (PQT)

Finally, we analyzed the performance of the scheduler varying the number of priority queues and the thresholds to separate the URLs. The other variable parameters were fixed at the following values: $DT = 259,200$, $WTQ = 3,600$, $ISWS = 3,600$. Table 6.6 shows the results, which are averages from all traces. All the conclusions hold for all traces individually.

NPQ [PQT]	TC	UC	UP	CORR
3 [1]	74.5	79.1	77.9	0.6073
3 [10]	77.9	82.2	80.9	0.8031
3 [100]	78.0	82.3	81.0	0.8097
3 [1,000]	78.0	82.3	81.0	0.8094
4 [1;10]	74.4	79.2	77.9	0.6034
4 [10;100]	77.4	82.2	80.9	0.6226
4 [100;1,000]	78.0	82.3	81.0	0.8041
5 [1;10;100]	74.4	79.2	77.9	0.6042
5 [1;100;1,000]	74.5	79.1	77.9	0.6078
5 [10;100;1,000]	77.8	82.3	80.9	0.6226

Table 6.6. Number of Priority Queues and Priority Queues Thresholds results (average of all datasets)

The performance is significantly affected when different thresholds are used to divide URLs into the priority queues, specially separating the most unpopular URLs. We learned from the characterization part of the work that large cascades are rare but extremely large when they occur (see Chapters 4 and 5). Separating the unpopular URLs from those which have been shared a few times makes a significant difference in the performance. Creating several priority queues is just dividing the popular URLs. However, we note that this result might be different for different OSM services or memes with a different popularity distribution. Testing this hypothesis is left for future work.

6.3.6 Discussion

Huge variations were not observed in the performance of the system, which may be due to the skewed popularity distribution of the Twitter traces utilized. However, we believe that MIOSphere is efficient and robust, for several reasons. Scalability is reached by using a distributed architecture with several machines collecting data in parallel. Robustness is reached by using a rich set of parameters to adapt the system for the characteristics of each OSM service. Efficiency is reached by dynamically separating the most discussed URLS from the unpopular ones, as well as by cleaning the dead URLs from the scheduler queue.

MIOSphere's architecture needs to be tested under different conditions, i.e., for different memes with different popularity distributions and propagation patterns. In general, we believe the performance achieved by MIOSphere on Twitter was satisfactory for a first step (average of almost 80% of all possible tweets collected using only 100 clients).

6.4 Prototype

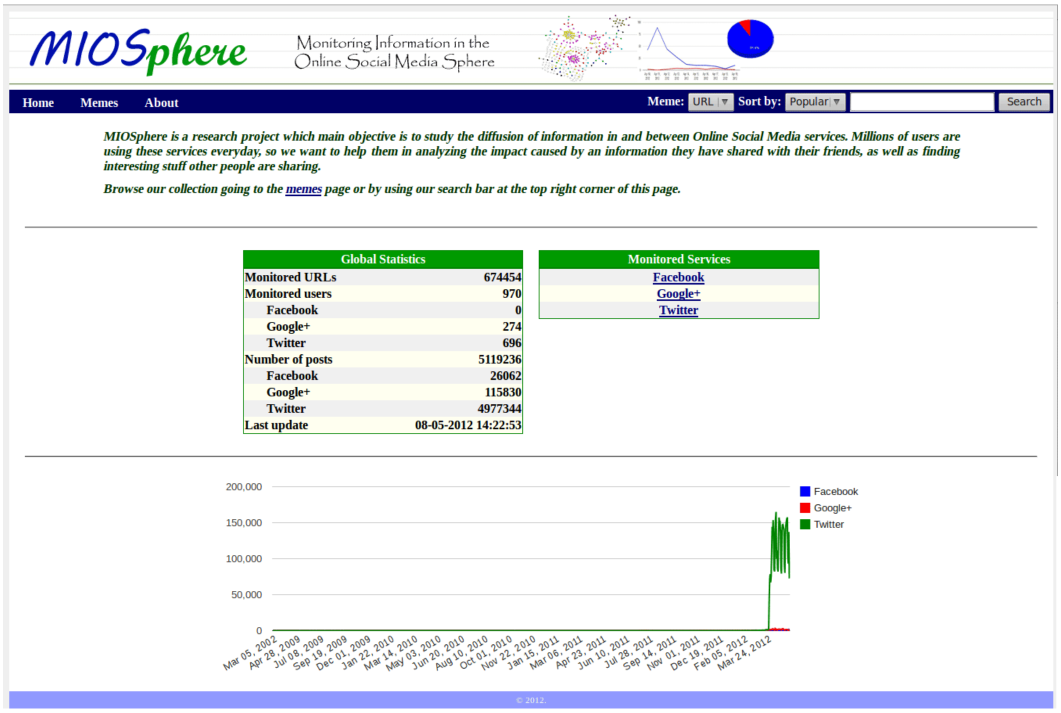
In order to validate and test the proposed architecture, we implemented a prototype to track the diffusion of URLs on popular OSM services (Twitter, Facebook, and Google+). A set of seed user accounts, which usually post several URLs per day, was created to feed MIOSphere with URLs to track. This set of seed user accounts is composed by the top 100 Twitter accounts (based on number of followers) from the top 10 categories (Celebrity, Music, Socialmedia, Entrepreneur, News, Blogger, Tech, Tv, Actor, Comedy) of the Web application *wefollow.com* [WeFollow, 2012], which suggest Twitter user accounts to follow. In total, 696 unique users were selected.

We also created a heuristic to find the correspondent accounts on Google+. In this heuristic, the first result returned by Google+ search engine using the Twitter username as the query was selected. Then, we manually checked if the accounts were from the same real person or company. 274 Google+ users were identified.

MIOSphere's prototype is written in Python, uses MySQL [MySQL, 2012] server for the database, and Django [Django, 2012] with Apache [Apache, 2012] for the Web application. The prototype uses the proposed architecture and has a centralized master coordinating several slaves to collect data. MIOSphere is available at `http://miosphere.speed.dcc.ufmg.br/`.

Figures 6.3, 6.4, 6.5, and 6.6 presents some snapshots of the prototype.

MIOSphere's prototype has to deal with many challenges to work properly. For instance, (1) MIOSphere deals with the presence of short URLs, resolving them; (2) MIOSphere detects when a client has been blocked by the OSM service API; (3) when the network is unavailable; (4) and when a client is not working; (5) MIOSphere also recovers from a previous run which has been killed (loss of energy, for example); and (6) a communication protocol between the server and the clients was developed. Most of these things were not considered in the simulations.



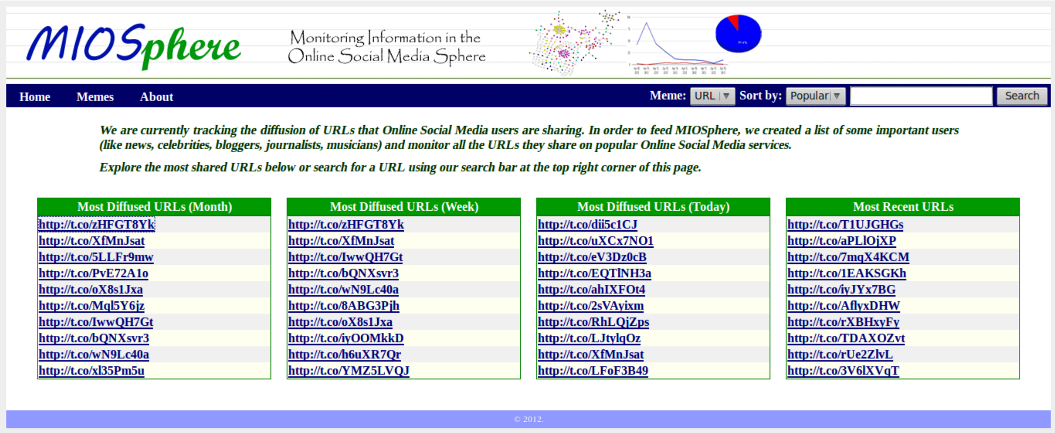
MIOSphere is a research project which main objective is to study the diffusion of information in and between Online Social Media services. Millions of users are using these services everyday, so we want to help them in analyzing the impact caused by an information they have shared with their friends, as well as finding interesting stuff other people are sharing.

Browse our collection going to the memes page or by using our search bar at the top right corner of this page.

Global Statistics	
Monitored URLs	674454
Monitored users	970
Facebook	0
Google+	274
Twitter	696
Number of posts	5119236
Facebook	26062
Google+	115830
Twitter	4977344
Last update	08-05-2012 14:22:53

Monitored Services
Facebook
Google+
Twitter

Figure 6.3. Snapshot of MIOSphere's interface: Home page

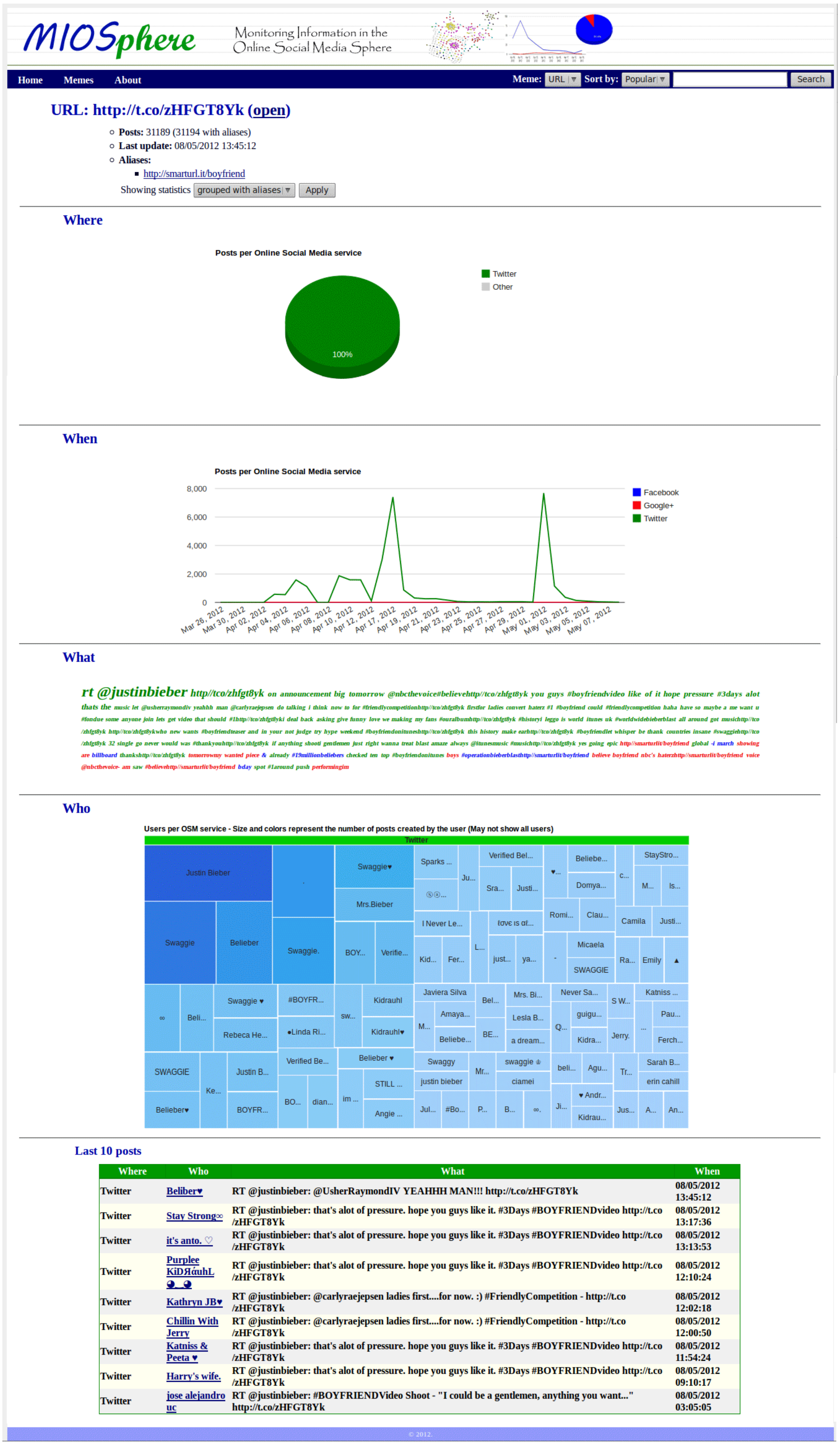


We are currently tracking the diffusion of URLs that Online Social Media users are sharing. In order to feed MIOSphere, we created a list of some important users (like news, celebrities, bloggers, journalists, musicians) and monitor all the URLs they share on popular Online Social Media services.

Explore the most shared URLs below or search for a URL using our search bar at the top right corner of this page.

Most Diffused URLs (Month)	Most Diffused URLs (Week)	Most Diffused URLs (Today)	Most Recent URLs
http://t.co/zHFGT8Yk	http://t.co/zHFGT8Yk	http://t.co/dii5c1CJ	http://t.co/T1UJGHGs
http://t.co/XfMnJsat	http://t.co/XfMnJsat	http://t.co/uXCx7NO1	http://t.co/aPLlOjXP
http://t.co/5LLFr9mw	http://t.co/IwwQH7Gt	http://t.co/eV3Dz0cB	http://t.co/7mqX4KCM
http://t.co/PvE72A1o	http://t.co/bQNXsvr3	http://t.co/EQTlNH3a	http://t.co/1EAKSGKh
http://t.co/oX8s1Jxa	http://t.co/wN9Lc40a	http://t.co/ahIXFOt4	http://t.co/iyJYx7BG
http://t.co/Mql5Y6jz	http://t.co/8ABG3Pjh	http://t.co/2sVAyixm	http://t.co/AflyxDHW
http://t.co/IwwQH7Gt	http://t.co/oX8s1Jxa	http://t.co/RhLQjZps	http://t.co/rXBHxyFy
http://t.co/bQNXsvr3	http://t.co/iyOOMkkD	http://t.co/LJtylqOz	http://t.co/TDAXOZvt
http://t.co/wN9Lc40a	http://t.co/h6uXR7Qr	http://t.co/XfMnJsat	http://t.co/rUe2ZlvL
http://t.co/xl35Pm5u	http://t.co/YMZ5LVQJ	http://t.co/LFoF3B49	http://t.co/3V6lXVqT

Figure 6.4. Snapshot of MIOSphere's interface: most popular URLs monitored



URL: http://t.co/zHFGT8Yk (open)

- Posts: 31189 (31194 with aliases)
- Last update: 08/05/2012 13:45:12
- Aliases:
  - http://smarturl.it/boyfriend

Showing statistics grouped with aliases Apply

Where

When

What

rt @justinbieber http//tco/zhfgt8yk on announcement big tomorrow @nbcthevoice#believehttp//tco/zhfgt8yk you guys #boyfriendvideo like of it hope pressure #3days alot thats the music let @usherraymondiv yeahhh man @carlyraejepsen do talking i think now to for #friendlycompetitionhttp//tco/zhfgt8yk firstfor ladies convert haterz #1 #boyfriend could #friendlycompetition haha have so maybe a me want u #fondue some anyone join lets get video that should #1http//tco/zhfgt8yki deal back asking give funny love we making my fans #ouralbumhttp//tco/zhfgt8yk #historyi leggo is world itunes uk #worldwidebieberblast all around got musichttp//tco/zhfgt8yk http//tco/zhfgt8ykwho new wants #boyfriendteaser and in your not judge try hype weekend #boyfriendonituneshttp//tco/zhfgt8yk this history make earhttp//tco/zhfgt8yk #boyfriendlet whisper be thank countries insane #swaggiehttp//tco/zhfgt8yk 32 single go never would was #thankyouhttp//tco/zhfgt8yk if anything shooti gentlemen just right wanna treat blast amaze always @itunesmusic #musichttp//tco/zhfgt8yk yes going epic http//smarturlit/boyfriend global -i march showing are billboard thankshttp//tco/zhfgt8yk tomorrowmy wanted piece & already #19millionbeliebers checked ten top #boyfriendonitunes boys #operationbieberblasthttp//smarturlit/boyfriend believe boyfriend nbc's haterzhttp//smarturlit/boyfriend voice @nbcthevoice- am saw #believehttp//smarturlit/boyfriend bday spot #1around push performingim

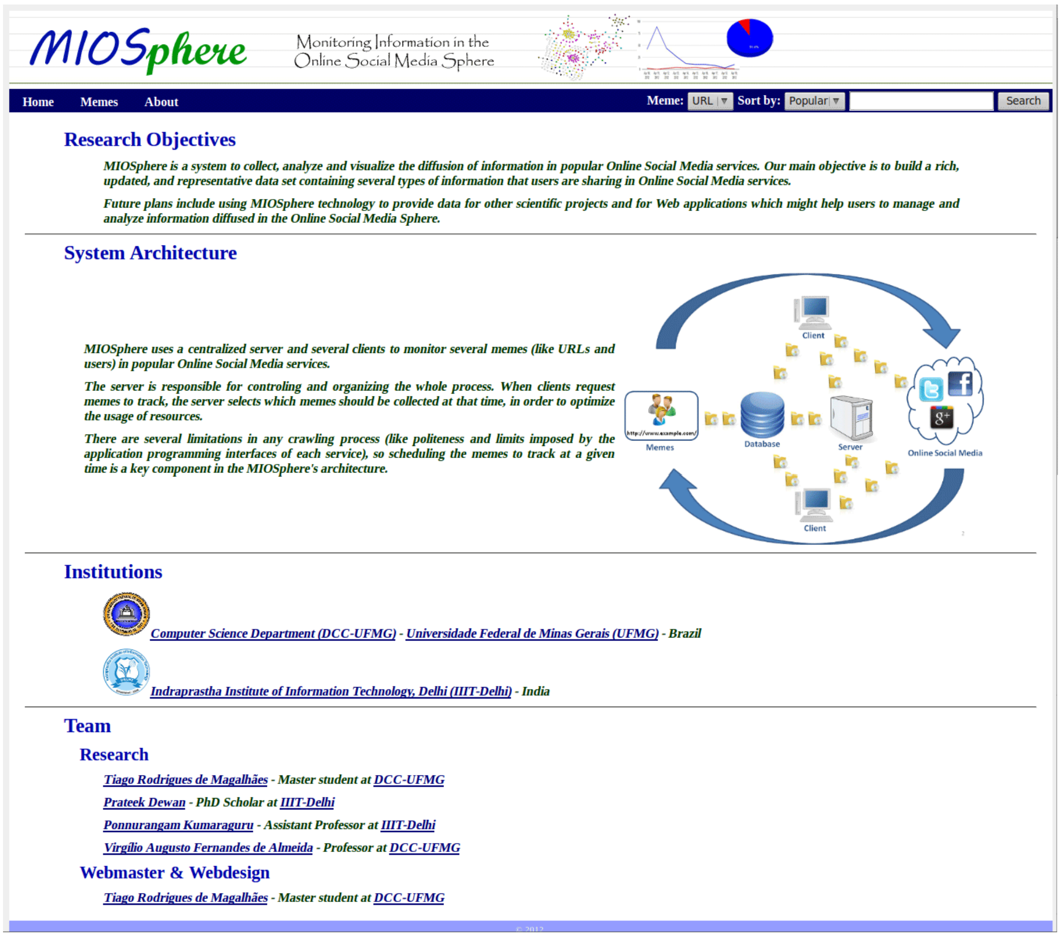
Who

Last 10 posts

Where	Who	What	When
Twitter	Beliber♥	RT @justinbieber: @UsherRaymondIV YEAHHH MAN!!! http://t.co/zHFGT8Yk	08/05/2012 13:45:12
Twitter	Stay Strong∞	RT @justinbieber: that's alot of pressure. hope you guys like it. #3Days #BOYFRIENDvideo http://t.co/zHFGT8Yk	08/05/2012 13:17:36
Twitter	it's anto. ♡	RT @justinbieber: that's alot of pressure. hope you guys like it. #3Days #BOYFRIENDvideo http://t.co/zHFGT8Yk	08/05/2012 13:13:53
Twitter	Purplee KiDЯάuhL ◕‿◕	RT @justinbieber: that's alot of pressure. hope you guys like it. #3Days #BOYFRIENDvideo http://t.co/zHFGT8Yk	08/05/2012 12:10:24
Twitter	Kathryn JB♥	RT @justinbieber: @carlyraejepsen ladies first....for now. :) #FriendlyCompetition - http://t.co/zHFGT8Yk	08/05/2012 12:02:18
Twitter	Chillin With Jerry	RT @justinbieber: @carlyraejepsen ladies first....for now. :) #FriendlyCompetition - http://t.co/zHFGT8Yk	08/05/2012 12:00:50
Twitter	Katniss & Peeta ♥	RT @justinbieber: that's alot of pressure. hope you guys like it. #3Days #BOYFRIENDvideo http://t.co/zHFGT8Yk	08/05/2012 11:54:24
Twitter	Harry's wife.	RT @justinbieber: that's alot of pressure. hope you guys like it. #3Days #BOYFRIENDvideo http://t.co/zHFGT8Yk	08/05/2012 09:10:17
Twitter	jose alejandro uc	RT @justinbieber: #BOYFRIENDVideo Shoot - "I could be a gentlemen, anything you want..." http://t.co/zHFGT8Yk	08/05/2012 03:05:05

© 2012.

Figure 6.5. Snapshot of MIOSphere's interface: detail about the diffusion of a URL



Research Objectives

MIOSphere is a system to collect, analyze and visualize the diffusion of information in popular Online Social Media services. Our main objective is to build a rich, updated, and representative data set containing several types of information that users are sharing in Online Social Media services.

Future plans include using MIOSphere technology to provide data for other scientific projects and for Web applications which might help users to manage and analyze information diffused in the Online Social Media Sphere.

System Architecture

MIOSphere uses a centralized server and several clients to monitor several memes (like URLs and users) in popular Online Social Media services.

The server is responsible for controling and organizing the whole process. When clients request memes to track, the server selects which memes should be collected at that time, in order to optimize the usage of resources.

There are several limitations in any crawling process (like politeness and limits imposed by the application programming interfaces of each service), so scheduling the memes to track at a given time is a key component in the MIOSphere's architecture.

Institutions

Computer Science Department (DCC-UFMG) - Universidade Federal de Minas Gerais (UFMG) - Brazil

Indraprastha Institute of Information Technology, Delhi (IIIT-Delhi) - India

Team

Research

Tiago Rodrigues de Magalhães - Master student at DCC-UFMG

Prateek Dewan - PhD Scholar at IIIT-Delhi

Ponnurangam Kumaraguru - Assistant Professor at IIIT-Delhi

Virgílio Augusto Fernandes de Almeida - Professor at DCC-UFMG

Webmaster & Webdesign

Tiago Rodrigues de Magalhães - Master student at DCC-UFMG

Figure 6.6. Snapshot of MIOSphere's interface: about the project page

6.4.1 Collected Data

The prototype is running since March 26, 2012 using 15 machines at Universidade Federal de Minas Gerais. After 7 weeks running (May 7, 2012), the 970 seed accounts (see Section 6.4) posted 673,463 URLs (original URLs and their aliases, i.e., the short and resolved links). In total, 5,108,239 posts with these URLs were collected in all the 3 OSM services monitored. A total of 4,966,557 posts are from Twitter, 115,712 are from Google,+ and 25,970 posts are from Facebook. Manual inspection on the most shared URLs revealed that most of them as posted in more than one OSM service, but usually most part of the posts are from a single service. All the top 10 URLs are more popular on Twitter, and usually most part of the posts are retweets. Moreover, different propagation patterns were observed. Some URLs have a huge peak of posts in one day and then quickly decreases, while others are periodic with a similar number of posts in several days. Figure 6.7 shows the number of posts collected per day for the top 5 URLs monitored by MIOSphere.

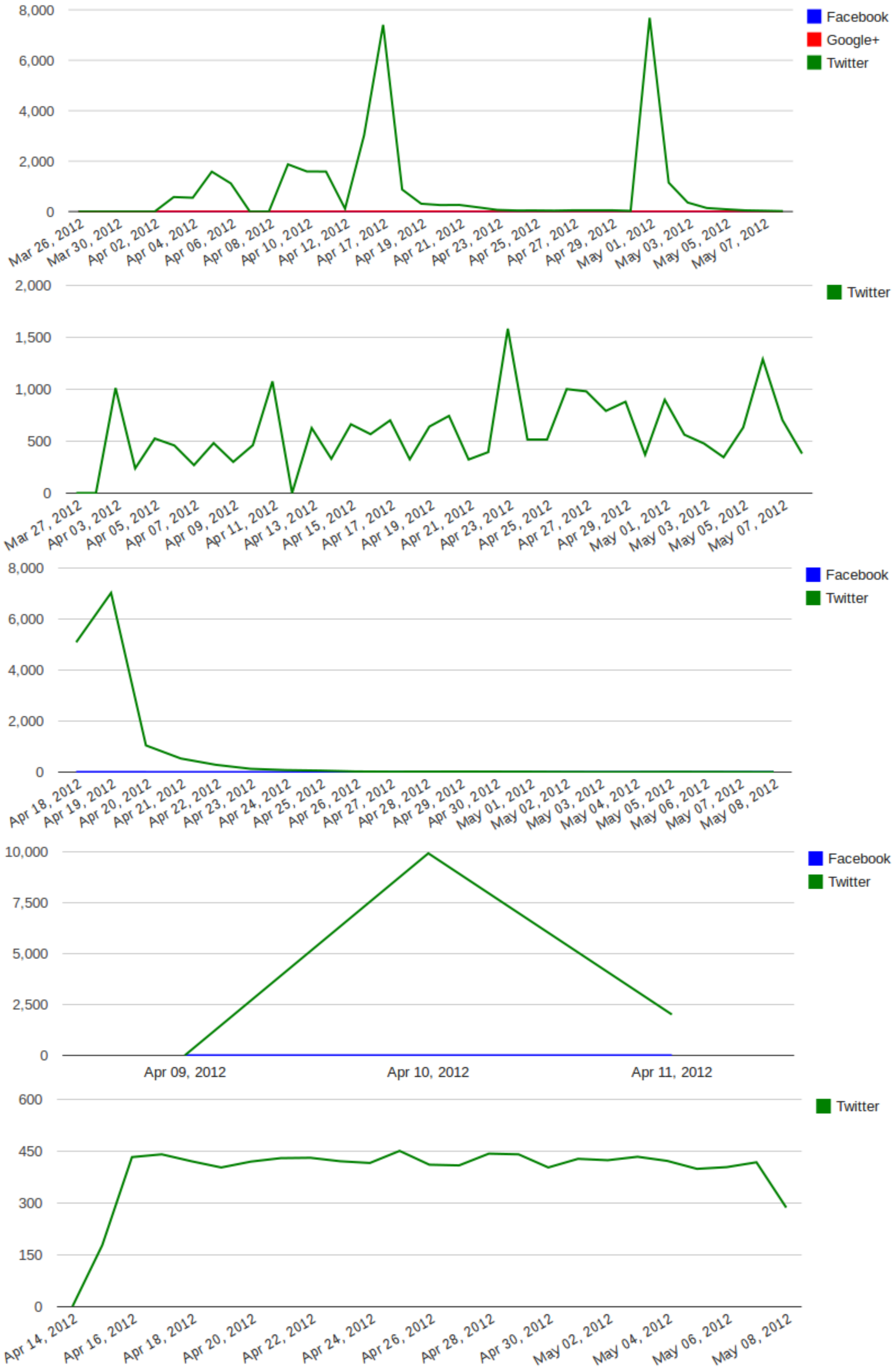


Figure 6.7. Top 5 shared URLs collected by MIOSphere

The most shared URL monitored in the period is `http://t.co/zHFGT8Yk`, which was posted 31,194 times (31,189 on Twitter, 1 on Facebook, and 4 on Google+). This URL corresponds to the iTunes page which is selling the song "Boyfriend", from the pop singer "Justin Bieber". The second most shared URL is `http://t.co/XfMnJsat`, which points to a Japanese Web page and looks like a news portal. `http://t.co/5LLFr9mw` is the third most shared URL and links to a YouTube video advertising a DVD from the band "One Direction". The fourth and fifth most popular URLs, `http://t.co/XfMnJsat` and `http://t.co/5LLFr9mw`, respectively, are also related to music. The fourth points to a video clip of a new song of the pop singer "Justin Bieber", the same song of the most shared URL monitored. The fifth URL links to a live streaming of a radio station, which might explain the pattern of around 400 tweets every day.

This manual inspection give us an evidence to generalize the finding that the relationship between OSM services is weak (see Section 6.1), as we are tracking URLs created in 2 source OSMs and diffused into 3 diffusion OSMs, but we let a better analysis on this topic as a suggestion for future work.

Chapter 7

Conclusions and Future Work

In this dissertation we presented MIOSphere, a system to monitor the diffusion of information through Online Social Media services. A real prototype was implemented to show that the proposed system works not only in theory. Moreover, several aspects of the diffusion of URLs through popular OSM services were presented, a crucial step for the development of MIOSphere.

7.1 Main Contributions

There are two main contributions of this dissertation: (1) several interesting findings on the wide characterization about the diffusion of URLs, and (2) a scalable and efficient solution for monitoring information in OSM services.

Some of the main findings presented in this dissertation are: (a) Word-of-mouth can be used to spread a single URL to a large audience, in some cases several million users; (b) Popular URLs spread through multiple disjoint propagation trees; (c) Domains of URLs popularly spread by word-of-mouth tend to be different from those popularly accessed in the general Web. Independently of the domain, all URLs have a chance to become popular; (d) There is a significant correlation between propagation and physical proximity; (e) Cross-Pollinated networks follow temporal and topological characteristics of the diffusion OSM; (f) Popularity of a meme on source OSM does not imply its popularity on diffusion OSM.

The most important contribution of this work is the architecture of a scalable system to track, in real-time, the diffusion of memes in several OSM services. One crucial aspect for the performance of such kind of a system is to choose which memes should be downloaded at each time, especially if the resources are limited and the number of memes to track is huge. With a good scheduling scheme, the amount of data collected can be significantly increased.

7.2 Future Work

There are several directions towards this work can evolve. Many suggestions were already given all over this dissertation. As expected for a first approach to the problem, there are several possible improvements to be done in the architecture. In Section 6.1 we discussed some lessons learned from the characterizations that could be explored. Combining techniques like RSS and streaming with the proposed may worth studying. Several aspects of the diffusion of information in OSM services can be exploited, like determining the topological structure of initiators that will speed up the propagation of content; modeling the propagation of content; and improving the ranking of actual search engines for real-time content based on the spread patterns of word-of-mouth propagation of URLs. Developing such kind of tools would have an important impact in the commercial world, such as advertising and political campaigns, as well as for Web users in general.

7.3 Limitations

Although the work done in this dissertation was wide and near-complete, it has some limitations. First of all, most of the studies are based on Twitter datasets. In the literature exists other characterization works on different OSM services which show similar characteristics, but there are some differences. MIOSphere is supposed to run separated for each OSM service, with a different configuration of parameters to handle well with the dynamics and characteristics of each service. Another limitation is the lack of similar works to compare the performance achieved. As a consequence, we do not know whether we can improve our performance or not, and we might still be far from the optimal solution for the problem.

Bibliography

Adar, E. and Adamic, L. A. (2005). Tracking information epidemics in blogspace. In *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, WI '05, pages 207--214, Washington, DC, USA. IEEE Computer Society.

Alexa.com (2012). Alexa.com. http://www.alexa.com.

Antoniades, D., Polakis, I., Kontaxis, G., Athanasopoulos, E., Ioannidis, S., Markatos, E. P., and Karagiannis, T. (2011). we.b: the web of short urls. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 715--724, New York, NY, USA. ACM.

Apache (2012). Apache. http://www.apache.org/.

API, G. G. (2012). Google geocoding api. http://code.google.com/intl/en/apis/maps/documentation/geocoding.

Archambault, A. and Grudin, J. (2012). A longitudinal study of facebook, linkedin, &#38; twitter use. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, CHI '12, pages 2741--2750, New York, NY, USA. ACM.

Baeza-Yates, R. A. and Ribeiro-Neto, B. A. (2011). *Modern Information Retrieval - the concepts and technology behind search, Second edition.* Pearson Education Ltd., Harlow, England.

Bakshy, E., Karrer, B., and Adamic, L. A. (2009). Social influence and the diffusion of user-created content. In *Proceedings of the 10th ACM conference on Electronic commerce*, EC '09, pages 325--334, New York, NY, USA. ACM.

Bit.ly (2011). You just shared a link. how long will people pay attention? http://blog.bitly.com/post/9887686919/you-just-shared-a-link-how-long-will-people-pay.

Bit.ly (2012a). Bit.ly. http://www.bit.ly.

Bit.ly (2012b). Bit.ly application program interface. http://code.google.com/p/bitly-api/wiki/ApiDocumentation.

Boyd, D., Golder, S., and Lotan, G. (2010). Tweet, tweet, retweet: Conversational aspects of retweeting on twitter. In *Proceedings of the 2010 43rd Hawaii International Conference on System Sciences*, HICSS '10, pages 1--10, Washington, DC, USA. IEEE Computer Society.

Broxton, T., Interian, Y., Vaver, J., and Wattenhofer, M. (2010). Catching a viral video. In *Proceedings of the 2010 IEEE International Conference on Data Mining Workshops*, ICDMW '10, pages 296--304, Washington, DC, USA. IEEE Computer Society.

Campbell, A. (2009). Social activity becomes significant source of website traffic. http://smallbiztrends.com/2009/03/social-activity-significant-source-website-traffic.html.

Cha, M., Benevenuto, F., Ahn, Y.-Y., and Gummadi, K. P. (2012). Delayed information cascades in flickr: Measurement, analysis, and modeling. *Computer Networks*, 56(3):1066--1076.

Cha, M., Haddadi, H., Benevenuto, F., and Gummadi, K. P. (2010). Measuring User Influence in Twitter: The Million Follower Fallacy. In *In Proceedings of the 4th International AAAI Conference on Weblogs and Social Media (ICWSM)*, Washington DC, USA.

Cha, M., Kwak, H., Rodriguez, P., Ahn, Y.-Y., and Moon, S. (2007). I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, IMC '07, pages 1--14, New York, NY, USA. ACM.

Cha, M., Perez, J. A. N., and Haddadi, H. (2009). Flash floods and ripples: The spread of media content through the blogosphere. In *AAAI Int'l Conference on Weblogs and Social Media*, ICWSM.

Cho, J., Garcia-Molina, H., and Page, L. (1998). Efficient crawling through url ordering. *Comput. Netw. ISDN Syst.*, 30(1-7):161--172.

Darwin, C. (1900). *The effects of cross and self fertilisation in the vegetable kingdom.* J. Murray, London.

De Choudhury, M., Sundaram, H., John, A., Seligmann, D. D., and Kelliher, A. (2010). "birds of a feather": Does user homophily impact information diffusion in social media? *Arxiv preprint arXiv*, 1006(4):31.

Django (2012). Django. https://www.djangoproject.com/.

Dodds, P. S. and Watts, D. J. (2005). A generalized model of social and biological contagion. *Journal of Theoretical Biology*, 232(4):587--604.

Econsultancy (2012). What is social media? here are 34 definitions... http://econsultancy.com/us/blog/3527-what-is-social-media-here-are-34-definitions.

Flickr (2012). Flickr application program interface. http://www.flickr.com/help/api.

Foursquare (2012). Foursquare application program interface. https://developer.foursquare.com.

Galuba, W., Aberer, K., Chakraborty, D., Despotovic, Z., and Kellerer, W. (2010). Outtweeting the twitterers - predicting information cascades in microblogs. In *Proceedings of the 3rd conference on Online social networks*, WOSN'10, pages 3--3, Berkeley, CA, USA. USENIX Association.

Ghosh, R. and Lerman, K. (2010). Predicting influential users in online social networks. volume abs/1005.4882.

Gomes, D. and Silva, M. J. (2006). Modelling information persistence on the web. In *Proceedings of the 6th international conference on Web engineering*, ICWE '06, pages 193--200, New York, NY, USA. ACM.

Gomez Rodriguez, M., Leskovec, J., and Krause, A. (2010). Inferring networks of diffusion and influence. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, pages 1019--1028, New York, NY, USA. ACM.

Goodman, A. (2012). Why the open directory (dmoz) is not so open? http://www.dmozsucks.org/why-the-open-directory-is-not-so-open.php.

Granovetter, M. (1978). Threshold models of collective behavior. *American Journal of Sociology*, 83(6):1420--1443.

Gruhl, D., Guha, R., Liben-Nowell, D., and Tomkins, A. (2004). Information diffusion through blogspace. In *Proceedings of the 13th international conference on World Wide Web*, WWW '04, pages 491--501, New York, NY, USA. ACM.

Guide, T. S. M. (2012). 50 definitions of social media. http://thesocialmediaguide.com/social_media/50-definitions-of-social-media.

Hanneman, R. and Riddle, M. (2005). *Introduction to Social Network Methods*. Digital form at http://faculty.ucr.edu/~hanneman/.

Jain, P., Rodrigues, T., Magno, G., Kumaraguru, P., and Almeida, V. (2011). Cross-pollination of information in online social media: A case study on popular social networks. In *SocialCom/PASSAT*, pages 477–482. IEEE.

Jain, R. (1991). *Art of Computer Systems Performance Analysis Techniques for Experimental Design: Measurements Simulation and Modeling*. Wiley Computer Publishing.

Karsai, M., Kivelä, M., Pan, R. K., Kaski, K., Kertész, J., Barabási, A.-L., and Saramäki, J. (2010). Small but slow world: How network topology and burstiness slow down spreading. *CoRR*, abs/1006.2125.

Kendall, M. G. (1938). A new measure of rank correlation. *Biometrika*, 30(1/2):81--93.

Krishnamurthy, B., Gill, P., and Arlitt, M. (2008). A few chirps about twitter. In *Proceedings of the first workshop on Online social networks*, WOSN '08, pages 19--24, New York, NY, USA. ACM.

Kwak, H., Lee, C., Park, H., and Moon, S. (2010). What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 591--600, New York, NY, USA. ACM.

Language, P. P. (2012). Python programming language. http://www.python.org/.

Lerman, K. and Galstyan, A. (2008). Analysis of social voting patterns on digg. In *Proceedings of the first workshop on Online social networks*, WOSN '08, pages 7--12, New York, NY, USA. ACM.

Leskovec, J., Adamic, L. A., and Huberman, B. A. (2007a). The dynamics of viral marketing. *ACM Trans. Web*, 1(1).

Leskovec, J., Backstrom, L., and Kleinberg, J. (2009). Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 497--506, New York, NY, USA. ACM.

Leskovec, J., McGlohon, M., Faloutsos, C., Glance, N., and Hurst, M. (2007b). Cascading behavior in large blog graphs: Patterns and a model. In *Society of Applied and Industrial Mathematics: Data Mining (SDM07)*.

Liben-Nowell, D. and Kleinberg, J. (2008). Tracing information flow on a global scale using internet chain-letter data. *Proceedings of the National Academy of Sciences*, 105(12):4633--4638.

Liben-Nowell, D., Novak, J., Kumar, R., Raghavan, P., and Tomkins, A. (2005). Geographic routing in social networks. *Proceedings of the National Academy of Sciences*, 102(33):11623--11628.

Marcus, A., Bernstein, M. S., Badar, O., Karger, D. R., Madden, S., and Miller, R. C. (2012). Processing and visualizing the data in tweets. *SIGMOD Rec.*, 40(4):21--27.

MySQL (2012). Mysql. http://www.mysql.com/.

Nazir, A., Raza, S., and Chuah, C.-N. (2008). Unveiling facebook: a measurement study of social network based applications. In *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, IMC '08, pages 43--56, New York, NY, USA. ACM.

Olston, C. and Pandey, S. (2008). Recrawl scheduling based on information longevity. In *Proceedings of the 17th international conference on World Wide Web*, WWW '08, pages 437--446, New York, NY, USA. ACM.

Ow.ly (2012). Ow.ly. http://www.ow.ly.

Project, O. D. (2012). Open directory project. http://www.dmoz.org.

Rao, L. (2010). Twitter seeing 90 million tweets per day, 25 percent contain links. http://techcrunch.com/2010/09/14/twitter-seeing-90-million-tweets-per-day.

Rodrigues, T., Benevenuto, F., Cha, M., Gummadi, K., and Almeida, V. (2011). On word-of-mouth based discovery of the web. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, IMC '11, pages 381--396, New York, NY, USA. ACM.

Scellato, S., Mascolo, C., Musolesi, M., and Crowcroft, J. (2011). Track globally, deliver locally: improving content delivery networks by tracking geographic social cascades. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 457--466, New York, NY, USA. ACM.

Schonfeld, E. (2010). Facebook drives 44 percent of social sharing on the web. http://techcrunch.com/2010/02/16/facebook-44-percent-social-sharing.

Sharma, A. K. and Dixit, A. (2008). Self adjusting refresh time based architecture for incremental web crawler. *Journal of Computer Science*, 8(12):349--354.

Steeg, G. V., Ghosh, R., and Lerman, K. (2011). What stops social epidemics? In Adamic, L. A., Baeza-Yates, R. A., and Counts, S., editors, *ICWSM*. The AAAI Press.

Sun, E., Rosenn, I., Marlow, C., and Lento, T. M. (2009). Gesundheit! modeling contagion through facebook news feed. In Adar, E., Hurst, M., Finin, T., Glance, N. S., Nicolov, N., and Tseng, B. L., editors, *ICWSM*. The AAAI Press.

Szabo, G. and Huberman, B. A. (2010). Predicting the popularity of online content. volume 53, pages 80--88, New York, NY, USA. ACM.

Tang, J., Lou, T., and Kleinberg, J. (2012). Inferring social ties across heterogenous networks. In *Proceedings of the fifth ACM international conference on Web search and data mining*, WSDM '12, pages 743--752, New York, NY, USA. ACM.

tinyURL (2012). tinyurl. http://www.tinyurl.com.

Truthy (2012). Truthy. http://truthy.indiana.edu.

Twitter (2010). Top twitter trends in 2010. http://yearinreview.twitter.com/trends.

Twitter (2012a). How to shorten links urls. http://support.twitter.com/articles/78124-how-to-shorten-links-urls.

Twitter (2012b). Twitter application program interface. http://dev.twitter.com/pages/streaming_api.

Wang, D., Wen, Z., Tong, H., Lin, C.-Y., Song, C., and Barabási, A.-L. (2011). Information spreading in context. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 735--744, New York, NY, USA. ACM.

Watts, D. J. (2002). A simple model of global cascades on random networks. *Proceedings of the National Academy of Sciences*, 99(9):5766--5771.

WeFollow (2012). Wefollow. http://wefollow.com.

Wikipedia (2012). Pearson product-moment correlation coefficient. http://en.wikipedia.org/wiki/Pearson_product-moment_correlation_coefficient.

Yang, Z., Guo, J., Cai, K., Tang, J., Li, J., Zhang, L., and Su, Z. (2010). Understanding retweeting behaviors in social networks. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 1633--1636, New York, NY, USA. ACM.

YouTube (2012). Youtube application program interface. http://code.google.com/apis/youtube/overview.html.