
Table of Contents

Foreword by John Musser	1
--------------------------------------	----------

Part I: Getting Started

Introduction.....	5
--------------------------	----------

High quantities of high quality followers	5
Use leader accounts to find quality followers	5
Engage with the most active users	6
Building an engagement account	6
Foodie example set	7
Benefits of searching for leader account screen names in tweets	7
Automating Twitter engagement for consistency	7
Twitter automation rules	8
Who should read this book?	8
Recommended programming background	8
Form follows function	9
Example code is open source	9
Version and server requirements	10
SQL is the heart of this system	10
JSON and Ajax can be added easily	10
Complete system with easy extensibility	11

Report System Preview	12
------------------------------------	-----------

Installing the Engagement Code.....	18
--	-----------

Use accounts that match your interests	18
Download the source code	18
Create a code directory on your server	18
Create a MySQL engagement database	19
User ids are permanent account identifiers	19
140dev API console	20

Insert the Twitter engagement account	21
Insert leader accounts	21
Create a Twitter app with OAuth tokens	21
Fill in config.php values	22
config.php	22
Run scripts through an SSH or Telnet client	23
show_code.php	23
Make the report directory web accessible	24
Run the engagement report system	24

Code Libraries25

Db_lib.php database library	25
db_lib.php	25
Oauth_lib.php OAuth library	27
oauth_lib.php	28
Test the engagement system	28
test_install.php	28
Sample report	30
/report/engagement_account.php	30

Coding Style32

Task scheduling	32
Print vs. echo	32
Single vs. double quotes	33
Require vs. include	33
Variable names and capitalization	33
Script naming conventions	33
Code segments as separate scripts	33
Functions in their own file	34
get_all_tweets.php	34
Controlling scripts with database flags and log files	34
Dealing with rate limits	35
Dealing with error codes	35
Commenting the code	36

Part II: Leader Account Programming

Collecting Leader Account Tweets39

Creating a relational database of tweet data	39
Recording tweet collection times and search values	42
Collecting leader account timelines	43
collect_old_leader_timeline.php	45

Collecting new leader tweets	49
collect_new_leader_timeline.php	49
Searching for tweets about leader accounts	52
collect_old_leader_search.php	53
collect_new_leader_search.php	57
Summary: Running the scripts in this chapter	60

Collecting Account Profiles.....62

Collecting leader account profiles	62
collect_leader_account_profiles.php	62
Missing user data	64
Collecting mentioned and retweeted account profiles	65
collect_mentioned_account_profiles.php	66
collect_retweeted_account_profiles.php	66
collect_account_profiles.php	67
Dealing with API failure to return users	69
Updating old account profiles	70
update_all_account_profiles.php	70
Removing deleted user accounts	70
purge_user_accounts.php	71
Collecting tags in account descriptions	72
update_user_tags.php	73
SQL queries for user tags	74
Summary: Running the scripts in this chapter	75

Report System76

Server and URL paths	76
Home page	77
/report/index.php	77
/report/page_top.html	77
/report/page_bottom.html	78
/report/menu.html	78
/report/menu.css	80
Sample engagement report	81
/report/engagement_account.php	81
Displaying users	82
get_user_profile.php	82
db_lib.php - select_array()	82
/report/display_users.php	83
display_lib.php	84
/report/user_template.html	86
/report/user.css	86
Displaying tweets	88

get_last_tweet.php	88
/report/display_tweets.php	89
/report/tweet_template.html	90
/report/tweet.css	90
Benefit of separate SQL query scripts	93
Tweet Reports.....	95
View tweets by leader accounts	95
/report/leader_tweets.php	96
/report/select_date.php	96
get_leader_tweets.php	97
View tweets by all accounts	98
/report/all_tweets.php	98
get_all_tweets.php	99
Popular leader tweets based on retweet count	99
/report/popular_leader_tweets.php	100
get_popular_leader_tweets.php	100
Popular tweets by non-leaders based on retweet count	101
/report/popular_nonleader_tweets.php	101
get_popular_nonleader_tweets.php	101
Most active times for tweeting	102
report/tweet_timing.php	
103	
get_tweet_days.php	105
get_tweet_hours.php	105
Search all tweets	106
/report/search_tweets.php	106
User Reports.....	108
Searching user descriptions	108
/report/search_users.php	108
get_search_users.php	110
Drilling down to user details	110
/report/user_profile.php	112
/report/user_profile.css	116
Domain specific results	117
Mention Reports.....	118
SQL queries for mentions	118
Most frequently mentioned by leader accounts	120
/report/leader_mentions.php	120
get_leader_mentions.php	121

Most frequently mentioned by all accounts	121
/report/all_mentions.php	121
get_all_mentions.php	122
Most frequent mentioners	122
/report/mentioners.php	123
get_mentioners.php	123
Mention timing	124
/report/mention_timing.php	125
get_mention_days.php	127
get_mention_hours.php	127

Retweet Reports..... 128

SQL queries for retweets	128
Most frequently retweeted by leader accounts	129
/report/leader_retweets.php	129
get_leader_retweets.php	130
Most frequently retweeted by all accounts	130
/report/all_retweets.php	131
get_all_retweets.php	131
Most frequent retweeters	131
/report/retweeters.php	132
get_retweeters.php	132
Retweet timing	133
/report/retweet_timing.php	134
get_retweet_days.php	136
get_retweet_hours.php	136

Tag Reports..... 137

Most used tags in leader tweets	137
/report/leader_tags.php	138
get_leader_tags.php	139
Most used tags in all tweets	139
/report/all_tags.php	139
get_all_tags.php	140
Most used tags in account descriptions	141
/report/user_tags.php	141
select_next_prev.php	142
get_user_tags.php	143

URL Reports 144

Most used URLs in leader tweets	144
---------------------------------	-----

/report/leader_urls.php	145
get_leader_urls.php	145
Most used URLs in all tweets	146
/report/all_urls.php	146
get_all_urls.php	147

Part III: Engagement Account Programming

Collecting Engagement Tweets151

Engagement account table	151
Collecting engagement account timeline	152
collect_old_engagement_timeline.php	152
collect_new_engagement_timeline.php	155
Collecting engagement account mentions and retweets	157
collect_old_engagement_search.php	157
collect_new_engagement_search.php	161
View engagement tweets	164
/report/engagement_tweets.php	164
get_engagement_tweets.php	165
View most effective engagement tweets based on retweet count	166
/report/popular_engagement_tweets.php	166
get_popular_engagement_tweets.php	167
View most active engagement account mentioners	167
/report/engagement_mentioners.php	168
get_engagement_mentioners.php	168
View most active engagement retweeters	168
/report/engagement_retweeters.php	169
get_engagement_retweeters.php	169
Engagement tweet history with specific accounts	170
/report/tweet_conversation.php	171
get_tweet_conversation.php	172
Summary: Running the scripts in this chapter	173

Collecting Engagement DMs.....174

Direct message access with OAuth	174
Sent and received direct messages	175
Direct message database storage	175
Collecting direct messages sent by the engagement account	176
collect_old_dms_sent.php	176
collect_new_dms_sent.php	179
Collecting direct messages received by the engagement account	181
collect_old_dms_received.php	181

collect_new_dms_received.php	183
View the complete direct message history	185
/report/all_dms.php	185
SQL query for direct messages	186
get_all_dms.php	186
/report/display_dms.php	188
/report/dm_template.html	188
dm.css	189
Searching engagement account direct messages	189
/report/search_dms.php	190
Direct message conversations with a specific account	191
/report/dm_conversation.php	192
get_dm_conversation.php	193
Summary: Running the scripts in this chapter	195

Automated Tweets and Retweets196

Automated tweeting overview	196
Autotweeting library	197
autotweet_lib.php	197
Tweet logging	201
Scheduled autotweets	201
post_scheduled_tweets.php	202
Recurring autotweets	203
update_recurring_tweets.php	204
post_recurring_tweets.php	204
Tweets based on RSS feeds	206
collect_rss_feeds.php	207
autotweet_lib.php - post_rss_tweet()	208
Follow Friday tweets	209
autotweet_lib.php - post_ff_tweet()	209
Auto retweeting the most retweeted leader tweets	211
autotweet_lib.php - post_leader_retweet()	212
autotweet_lib.php - post_retweet()	212
Summary: Running the scripts in this chapter	213

Collecting Engagement Friends.....214

Skipping screen shots for reports	214
Database schema	215
Paging through friends with a cursor	216
Logging new friends	216
collect_engagement_friends.php	217
Collecting friends account data	219
collect_engagement_friends_profiles.php	219

collect_user_profiles.php	220
Cleaning your engagement account friends list	222
View new friends	222
/report/new_friends.php	222
get_new_friends.php	223
View new unfriends	223
/report/new_unfriends.php	223
View friends who are not tweeting	224
/report/friends_not_tweeting.php	224
get_friends_not_tweeting.php	225
View friends who are protected	225
/report/protected_friends.php	225
get_protected_friends.php	226
Search friends	226
/report/search_friends.php	226
get_search_friends.php	228
Finding new people to follow	228
View suggested friends who mention and retweet others	228
/report/suggested_friends_tweeting_about_others.php	228
get_suggested_friends_tweeting_about_others.php	229
View suggested friends who mention and retweet engagement account	230
/report/suggested_friends_tweeting_about_engagement_account.php	230
get_suggested_friends_tweeting_about_engagement_account.php	230
View suggested friends mentioned and retweeted by engagement account	231
/report/suggested_friends_engagement_account_tweets_about.php	231
get_suggested_friends_engagement_account_tweets_about.php	232
Summary: Running the scripts in this chapter	232

Collecting Engagement Followers.....233

Database schema	233
Follower collection	234
collect_engagement_followers.php	234
collect_engagement_followers_profiles.php	236
View new followers	236
/report/new_followers.php	237
get_new_followers.php	237
View new unfollowers	238
/report/new_unfollowers.php	238
get_new_unfollowers.php	238
Search followers	239
/report/search_followers.php	239
get_search_followers.php	240
Automated tweets to new followers	241

post_new_follow_tweets.php	241
Automated DMs to new followers	242
post_new_follow_dms.php	242
Summary: Running the scripts in this chapter	243

Mutual Follow Relationships244

View mutual follows	244
/report/mutual_follows.php	244
get_mutual_follows.php	245
View friends who are not followers	245
/report/friends_not_followers.php	245
get_friends_not_followers.php	246
View followers who are not friends	246
/report/followers_not_friends.php	246
get_followers_not_friends.php	247

Part IV: Appendices

Scheduling API Code251

Rate limits control scheduling	251
Rate limits are calculated for each API call	251
One-time execution scripts	252
Scheduling crontabs	253
Scripts that run every minute	253
Scripts that run a few times a day	253
Hourly scripts	254
Account profile collection needs to be spread over time	255
Daily scripts	256
Final cron job list	256
cronjob.txt	256

Engagement Database Schema.....258

Automated tweets	258
Direct messages	260
Following	260
Key accounts	261
Tweet collection	262
Twitter user collection	263

Open Source Code License.....264

license.txt	264
-------------	-----

Acknowledgments.....	265
About the Author	266
Index	267

Foreword by John Musser

When I founded ProgrammableWeb.com in 2005, I believed that web APIs were the foundation of the next wave of computing. That prediction proved correct as our site grew along with the increasing importance of APIs as the web transitioned from web sites to web services. ProgrammableWeb now tracks almost 10,000 APIs and the Twitter API has long been one of the most popular.

Our users have long told us that the Twitter API and the masses of free data it offers is a great platform for application development. The only thing missing has been a clear set of documentation for taking advantage of it. That is why I am so pleased to see this new book by Adam Green.

I've been following Adam Green's work since way before anyone ever thought of Twitter, APIs, or even the mass market use of the Internet. I often relied upon his terrific books on the dBASE language in the 1980s as part of my work on early PC-based systems. And his educational seminars in those days played an important role in the developer ecosystem of dBASE and its clones like FoxPro.

Adam and I became personal friends at the first Mashup Camp conference in early 2006 and he has been a trusted advisor and resource on APIs ever since. When he switched to full-time Twitter API development in 2008, I knew he was on the right track. He has contributed articles to ProgrammableWeb on best practices in Twitter development, and his tutorials and open source code on his 140dev.com site have all been very useful. Unfortunately, Adam didn't have time to write as much as I would have liked. Most of his time was spent doing client work, and producing a series of websites based on the 2012 presidential election. However, this work paid off in a deep understanding of the Twitter API's strengths and weaknesses.

With this first book on Twitter API programming, Adam has finally returned to what he does best: breaking down a development platform with clear examples and the right level of detail for developers who want to build real-world applications. Adam tells me that this is just the first in a series of Twitter API programming books based on the years of Twitter consulting work he has done. If the rest of the series is as useful as this book, there is no doubt it will play an important role in the future of Twitter application development.

Reading this book, it is clear that it is written by a coder who has spent his time in the trenches. It isn't just a rehash of the docs. As technology readers we are often disappointed by books that are either too high-level to be practical or too low-level to be useful. *Twitter API Engagement Programming* is full of coding advice and scripts that form a useful whole but also stand on their own. It's like having a seasoned expert sitting next to you as you code.

I especially appreciate Adam's tips about undocumented edges, which any developer can tell you, are often the most valuable bits (e.g. rate limits and strategy for dealing with them, API idiosyncrasies, and subtle differences among seemingly similar calls). His down to earth, no-BS style reflects the difference between a writer who has been assigned to write about the Twitter API and a developer who's used Twitter's API for years, warts and all, and has put in writing that hard earned knowledge.

In addition to the text of this book, Adam's long-term contribution to the Twitter development community is his decision to release the book's example scripts as open source, guaranteeing that the code will form the foundation of many apps.

I look forward to the release of the rest of his planned series of Twitter API books, and encourage you to dive into this first book as a chance to discover just how much great functionality and data Twitter provides.

John Musser
Founder, ProgrammableWeb.com

Introduction

High quantities of high quality followers

Everyone wants more followers. I've been building Twitter apps and doing Twitter consulting since 2008, and that request always comes early in a conversation with new clients. It seems like a simple equation: more followers equals more readers of your tweets, and therefore more fame and power on Twitter. But it isn't that simple. There are plenty of Twitter accounts with tens of thousands of followers that receive almost no mentions or retweets. Does that sound like an influential account?

Gaining sheer numbers of followers is easy. You can pay many services to bulk up your followers with bot and fake spam accounts. Unfortunately, this is obvious when anyone looks at your follower list. More importantly, these fake followers contribute nothing. They just spew spam. They don't help you learn anything or share any of your ideas with others.

If you are using Twitter to promote your cause, product, or career, your goal should be to have real engagement with active users. You want people who reply to you, who include you in their conversations, and who spread your messages with retweets. Once you become part of an engaged community on Twitter, quality followers will find you and help you grow an account that offers value to everyone involved. Those people are out there for any area in which you may be interested. All you have to do is find them, and build a relationship.

This book will present an engagement strategy and source code for a collection of tools that will gain high quantities of high quality followers for any of your Twitter accounts. If you use the system presented here, you can grow your followers by several hundred or even a few thousand a month, and they will be the type of people who help spread your message and introduce you to their followers who are passionate about your area of interest.

Use leader accounts to find quality followers

The starting point in building a quality Twitter account is to play follow the leader.

There are usually a couple of dozen leader accounts in any area of interest. All you have to do is study them, find out who they interact with, and learn how to join their crowd. The leaders may have hundreds of thousands or even millions of followers, so getting leaders to follow you is not your first goal. What you want to do is find the core people who hang out with the leaders, and join their conversations. Enter their circle of friends and they will notice you.

My first career was in the personal software business during the early Eighties. We didn't have the Internet or even email yet, but we did have computer conventions, also called trade shows. The best PR advice I ever got in those days was to find the most influential people at the shows, and stand next to them during the after-show parties. I picked Dan Bricklin, the designer of Visicalc, as one of my key targets, and the strategy worked. I got to know Dan and the circle that always formed around him at public events. I spent time talking to that group of people every chance I got. In time I was a recognized member of the cool software crowd. The same approach consistently works on Twitter as well.

Engage with the most active users

One difference between a bot account and an actively engaged user is their use of mentions and retweets. Bots are running a script that delivers the same tweets with no regard to who's listening. It is difficult to build a reliable script that can include @mentions of others in an intelligent way. So when you come across accounts that include a high percentage of @mentions, and who are @mentioned by others, you know there are probably engaged humans behind those accounts.

The best way to find these accounts is to collect the @mentioners and retweeters of the leaders. This reveals the "cool kids" in any group. Following and engaging with these cool accounts will put you into the center of the conversations. This book's code will teach you how to find these engaged users and provide tools to make conversation with them an efficient process.

Building an engagement account

To really engage with a community of interest you should create an engagement account that focuses on the issues of that group. This account should have a screen name and description that clearly states your goals and position. The research we will perform in the chapters on leader accounts will show you how to find the best words and hashtags to include in your account description to appeal to the target audience. While this engagement account may be different from your personal Twitter account, there is no need to make the connection between these accounts a secret. You can have each of your accounts retweet and @mention each other to help clarify the relationship. The key is having the engagement account clearly positioned for a specific interest and point of view.

Foodie example set

The code presented here can be applied to any community of interest, such as sports, politics, or entertainment. The example set I'll be using for this book is based on the foodie community. These are people who tweet largely about food, restaurants, recipes, and food related TV shows. I've chosen two leader accounts: **@FoodNetwork**, and **@WholeFoods**. The code can support any number of leader accounts, but two or three top accounts are a good way to get started. I also created a new engagement account, **@FoodieCentral**, specifically for this project.

Benefits of searching for leader account screen names in tweets

The method we will use to identify a community of influential tweeters is to use the search API to find accounts that tweet with @mentions or retweets of the leader accounts. In this case, that means anyone who has @FoodNetwork or @WholeFoods in their tweets. This may seem too restrictive. Why not just search for words like 'food' or 'recipe'? I have found that broad searches like this have too much spam and low quality tweets. Just because someone mentions food, that doesn't mean they are seriously involved with foodie culture. People who directly mention leader accounts tend to be more engaged and have a specific goal of interacting with the most influential users in that area.

Spam accounts rarely @mention leaders in their tweets, because excessive amounts of tweets that @mention the same account can be detected by Twitter and result in suspension. So the simple rule of thumb is that spammers use general words for the broadest audience and don't @mention frequently, while sincerely engaged accounts talk to other people directly with @mentions.

This is something you can test for yourself. Compare the results of searching for 'food' versus '@foodnetwork':

- <https://twitter.com/search/realtime?q=food>
- <https://twitter.com/search/realtime?q=@foodnetwork>

You will see that the tweets mentioning a leader account are more substantive, have more @mentions, URLs, and retweets of others in that community. Tweets that use the general term 'food' are basically anyone who is hungry or has recently eaten. Much of it is spam about diets and weight loss schemes. Their quality as potential conversational partners is limited.

Automating Twitter engagement for consistency

I started by criticizing bot accounts, but that doesn't mean every tweet, mention, and retweet has to be done completely by hand. Some level of automation helps you main-

tain a steady presence in a Twitter community, even when the demands of work and life interfere with regular tweeting. The level of automation you will learn here enables you to quickly view the most actively engaged accounts, join the most important conversations, and retweet the top buzz in a timely manner. You'll also learn how to send out automated tweets at a level that establishes a base of engagement with almost no work. Then you can include handwritten tweets throughout the day as part of your overall Twitter flow. The result is a highly engaged account that steadily builds a presence with the key influencers.

Twitter automation rules

The Twitter API gives you the technical ability to perform many automated account actions, but Twitter has published guidelines for controlling the types of automation you should perform. You can find these at: <https://support.twitter.com/articles/76915-automation-rules-and-best-practices>. They can be summarized as:

- Automated tweeting based on keywords is generally frowned upon unless they are kept to a reasonable level.
- Automated following based on keywords is forbidden, but you are allowed to find follow candidates through automation and then follow them by hand. This approach is used heavily in this book.
- Automated following back in response to follows is forbidden, but you can use automation to identify new followers, and select the best of these to follow back.
- Automated DMs in response to a follow are allowed, but should not be overdone.

This can all be summarized as don't be a spammy pain in the ass. That should be a general rule for Twitter even when not using any automated tools. I've been helping clients do the type of work described in this book for years, and have never been suspended personally or had a client suspended. If you are cool, Twitter will be cool.

Who should read this book?

While there is a clear engagement strategy in this book that can benefit anyone using Twitter, this is basically a book about Twitter API programming with a heavy emphasis on database collection and mining of data. As such, it is written for experienced PHP and MySQL programmers. If you don't do any programming, you can still learn a lot about engagement strategies and effective automation techniques by reading this book. To actually run the example scripts in this book, however, you will need to have a PHP coder work with you.

Recommended programming background

If you have programmed in PHP at any level, or are familiar with other Web based languages, like Perl, Python, or Ruby, you should have no problem understanding the

code in this book. I program with PHP in a standardized format that is meant to be readable. I avoid using class libraries, except in a few cases of common code that is used repeatedly. Where possible, I choose to write code that makes it clear what I am trying to achieve, even if that takes a few extra lines.

You should also be experienced with HTML and CSS to control the formatting of web pages. The example code here uses a minimal user interface, so anyone who has built pages with HTML and CSS will be fine.

At its heart, the programming system described here is a relational database run with MySQL. If you are comfortable with the relational model, and performing SQL queries to modify or query a database, you will find this code to be straightforward.

Form follows function

There are a number of approaches to documenting something as large and complex as the Twitter API. The official Twitter documents use the divide and conquer technique. They split up the various API calls by groups of functionality, and then divide each API call by the available parameters. This method is great when you already know the API, and want to look up a specific detail. Unfortunately, it is a poor model for learning the API from scratch. There is no obvious starting point and finding a path through all the steps you need to perform a real-world task is a mystery.

This book takes the approach of laying out a practical plan for building an engaged Twitter account, and covers the programming techniques as they are needed. First we will learn how to collect tweets by and about leader accounts. Then we will extract the most influential users from these tweets. With the target accounts for engagement identified, we will cover the coding needed to manage and grow friend and follower relationships. This gather, analyze, and engage approach will lead you through the most important aspects of the Twitter API in an order that makes it clear why you need to perform each task and how to do it effectively.

Example code is open source

The example scripts in this book have been released using the MIT open source license. A copy of this license is included with the downloadable files available on the <http://140dev.com> website. You are free to use these scripts to build applications for your personal use or as the foundation of commercial websites and applications. You may modify the code as you wish and freely distribute it. The complete MIT license is in Appendix C at the back of the book.

This book is the first of a series on the many features of the Twitter API, all of which will include open source code. My goal is to have the code in this series of books become a widely used foundation for Twitter API programming.

Version and server requirements

The code in this book is written to conform to the requirements of Twitter API version 1.1, which means it requires OAuth requests for all API calls. The installation chapter will explain the basics of setting up OAuth, if you haven't used it before. The PHP scripts assume that you are using PHP 5 and MySQL 5. Both of these are the standard on all Linux/Unix servers, and are available for use with Windows and Mac OS X. You will have no problems running this code on any of these operating systems. I also recommend running a copy of phpMyAdmin to help manage and visualize the data collected.

SQL is the heart of this system

While this code appears to be dependent on PHP, the real foundation is SQL queries. All of the reports are based on SQL queries, and the movement of data into and out of the database is also done with SQL. This makes it easy to build variations of this code in other programming languages. All web based languages have easy access to MySQL. You can write extensions in any language, and they can all interact with the same database.

JSON and Ajax can be added easily

The example code uses PHP to display an HTML based user interface, but you can also add a UI based on Javascript and Ajax. The Twitter API doesn't allow Javascript to connect to the Twitter server directly, but the framework used in this code makes it easy to make database requests through Ajax. The functions that are coded to deliver HTML can be easily changed to deliver JSON data for use by Javascript.

The same approach can be taken to build mobile apps on this data. The apps can draw on data using Ajax calls, so creating a mobile app is largely a matter of a new user interface. The rest of the data collection and delivery interface is already in place.

Code and data model

There is a lot of code in this book, over 150 separate scripts. Before we dive into the details of all this programming, I'd like to offer an overview of what this code is trying to accomplish. There are 4 sets of users who are tracked with this system:

1. Leader accounts who are the most influential.
2. Accounts that engage with the leader accounts by @mentioning and retweeting them.
3. An engagement account with the goal of growing its list of engaged and influential followers.

4. Accounts that @mention and retweet the engagement account.

There are 5 basic sets of scripts:

1. Collection code makes a connection to the API and gathers tweets by leader accounts, about leader accounts, by an engagement account, and about the engagement account.

2. Data contained in the tweets is split apart by parsing and stored in a relational database as separate tweets, users, tags, URLs, mentions, retweets, friends, and followers. DMs to and from the engagement account are also recorded.

3. The account profiles of each tweet author or subject of an @mention or retweet are collected.

4. The complete friends and followers list for the engagement account are collected to make follow and unfollow suggestions.

5. Reporting scripts get related data from the database through SQL queries and display it as lists of tweets or users.

Complete system with easy extensibility

This book gives you a complete set of tools and a solid engagement strategy. It is also written to be easily extensible. You can add more data collection, more automation tools, and more reports. The goal of gaining high quality followers is within your reach, and customization to meet the needs of any specific engagement campaign is easily achievable.