

Week 9 • 소셜네트워크 데이터마이닝과 분석

Social Data Mining 02

Joonhwan Lee

human-computer interaction + design lab.

오늘 다룰 내용

- Twitter Crawling

1. Twitter Crawling Basics

Twitter API 소개

- ◆ 트위터에서 데이터를 수집하기 위해 제공되는 API
 - ◆ REST API
<https://dev.twitter.com/docs/api>
 - ◆ Streaming API
<https://dev.twitter.com/docs/streaming-apis>
등이 제공된다.
- ◆ 사용자가 너무 많은 데이터 수집하는 것을 막기 위해 여러 제한 장치를 두고 있음.
- ◆ 개발자는 먼저 트위터 개발자로 등록하여 제작할 어플리케이션에 인증도구로 사용할 consumer_key 등을 받아야 함.
 - ◆ <https://dev.twitter.com/docs>

Twitter 개발자 등록

- ✦ <https://dev.twitter.com/apps/new>

Create an application

Application Details

Name: *

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description: *

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

Website: *

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens.
(If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL:

Where should we return after successfully authenticating? For [@Anywhere applications](#), only the domain specified in the callback will be used. [OAuth 1.0a](#) applications should explicitly specify their `oauth_callback` URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank.

Twitter 개발자 등록

- ✦ <https://dev.twitter.com/apps/new>



Create an application

Application Details

Name: *

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description: *

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

Website: *

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens. (If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL:

Where should we return after successfully authenticating? For [@Anywhere applications](#), only the domain specified in the callback will be used. [OAuth 1.0a](#) applications should explicitly specify their `oauth_callback` URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank.

Twitter 개발자 등록

- ✦ Customer Key and Access Token

The screenshot shows the 'hcid-dev-test-app' page in the Twitter developer console. At the top, there are navigation tabs: 'Details', 'Settings', 'OAuth tool', '@Anywhere domains', 'Reset keys', and 'Delete'. The 'OAuth tool' tab is selected, displaying the 'OAuth Settings' section. This section contains four input fields, each with a red asterisk indicating a required field. The fields are: 'Consumer key:', 'Consumer secret:', 'Access token:', and 'Access token secret:'. Each field contains a blurred, masked value. Below the 'Consumer secret:' and 'Access token secret:' fields, there is a small text note: 'Remember this should not be shared.'

Twitter 개발자 등록

- ✦ Customer Key and Access Token

hcid-dev-test-app

Details Settings OAuth tool @Anywhere domains Reset keys Delete

OAuth Settings

Consumer key: *

Consumer secret: *

Remember this should not be shared.

Access token: *

Access token secret: *

Remember this should not be shared.

Twitter Crawling Gems

✦ **twitter gem**

- ✦ Twitter REST API를 이용하여 트위터 사용자의 정보 수집
- ✦ `gem install twitter`
- ✦ <http://rdoc.info/gems/twitter>

✦ **tweetstream gem**

- ✦ Twitter Streaming API를 이용하여 트윗 수집
- ✦ `gem install tweetstream`
- ✦ <https://github.com/intridea/tweetstream>

Twitter 사용자 정보의 수집

- ◆ 소프트웨어 설정

- ◆ 앞서 부여받은 개발자 token 을 사용하여 트위터로의 접근을 승인받을 수 있도록 설정 작업을 한다.

```
require "twitter"
```

```
client = Twitter::REST::Client.new do |config|  
  config.consumer_key      = "YOUR_CONSUMER_KEY"  
  config.consumer_secret  = "YOUR_CONSUMER_SECRET"  
  config.access_token      = "YOUR_ACCESS_TOKEN"  
  config.access_token_secret = "YOUR_ACCESS_SECRET"  
end
```

Lab 1: Twitter 사용자 정보 가져오기

- ◆ 사용자의 타임라인에서 20개의 최근 트윗을 가져오자.

```
client.user_timeline("oisoo").each do | t |  
  puts t.text  
  puts "-----"  
end
```

twitter gem으로 수집 가능한 정보

- ✦ twitter gem 은 twitter 의 REST API 에 대응.
 - ✦ <https://dev.twitter.com/docs/api/1.1>
- ✦ 해당 API가 제공하는 데이터를 가져올 수 있다.
 - ✦ 사용설명서: http://rdoc.info/gems/twitter#Usage_Examples
 - ✦ `client.user("oisoo")`
 - ✦ `client.user(48661131)`
 - ✦ `client.user_timeline("oisoo")`
 - ✦ `client.user_timeline("oisoo", :count => 100)`

Lab 1: Twitter 사용자 정보 가져오기

- ◆ 수집 가능한 사용자 속성
 - ◆ `p client.user("oisoo")`
 - ◆ `user`의 속성을 hash table 로 보여준다 (수집 가능 속성)
 - ◆ `user_profile.rb` 참고
- ◆ 다음을 입력하여 사용자의 속성을 가져와보자.
 - ◆ `location`
 - ◆ `created_at`
 - ◆ `followers_count`
 - ◆ `friends_count`
 - ◆ `status.text`
 - ◆ `status.retweet_count`

Lab 2: Twitter 사용자 정보 가져오기 2

```
Username      : oisoo
Name          : 이외수
Id            : 48661131
Location      : Hwacheon 38.166458,127.516781
User since    : 2009-06-19 18:35:25 +0900
Bio           : 화천군 감성마을 소설가 Korean Novelist 'Lee Oisoo'...
Followers     : 1518973
Friends       : 21289
Listed Cnt    : 44207
Tweet Cnt     : 9902
Geocoded      : false
Language      : en
URL           : http://twtkr.com/oisoo
Time Zone     : Seoul
Verified      : false

Tweet time    : 2012-11-22 13:18:18 +0900
Tweet ID      : 271467616776904705
Tweet text    : [18대 대선 부재자투표 신고] 11/21(수)~25(일)...
Retweet Cnt   : 203
```

Lab 3: Twitter 사용자 정보 가져오기 3

- ◆ 특정 트위터 사용자가 팔로잉하는 친구의 정보 가져오기.
- ◆ `friends = client.friend_ids(name)`
 - ◆ `name` 이 팔로잉하는 친구들의 `id`를 포함한 twitter 객체 수집하여 `friends` 에 저장
- ◆ `friends.attrs[:ids]`
 - ◆ 친구들의 `id` 출력
- ◆

```
friends.attrs[:ids].each do |uid|
  f = client.user(uid)
  puts f.followers_count
end
```

Lab 3: Twitter 사용자 정보 가져오기 3

```
name = "Yunaaaa"
user = Hash.new

friends = client.friend_ids(name)
friends.ids.each do |fid|
  f = client.user(fid)
  # Only iterate if we can see their followers
  if (f.protected? != "true")
    user[f.screen_name.to_s] = f.followers_count
  end
end

user.sort_by {|k,v| -v}.each { |user, count|
  puts "#{user}, #{count}" }
```

Rate Limiting!

- ◆ 트위터에서 데이터를 가져가려는 앱이 너무 많기 때문에 request 숫자를 제한하고 있음.
 - ◆ <https://dev.twitter.com/docs/rate-limiting>
 - ◆ Rate limit window duration is currently 15 minutes long.
 - ◆ <https://dev.twitter.com/docs/rate-limiting/1.1/limits>

Lab 3.1: Rate Limit Control

- ✦ Rate limit 에 도달하면 서버는 에러메시지를 보내준다.
- ✦ 해당 에러메시지가 발생하면 잠깐 쉬었다가 다시 시작.

```
begin
  f = client.user(uid)
  puts "processing '#{f.screen_name}'..."
rescue Twitter::Error::TooManyRequests =>
  rate_limit_error
  puts "you have reached the rate_limits."
  sleep rate_limit_error.rate_limit.reset_in
  retry
end
```

Twitter Streaming APIs

- ✦ <https://dev.twitter.com/docs/streaming-apis>
- ✦ 트위터 메시지를 실시간으로 전송하는 API
 - ✦ REST API: request 한 메시지만 가져올 수 있다.
- ✦ Public Streaming API
- ✦ User Streaming API
- ✦ Site Streaming API

Twitter Streaming APIs

- ◆ Public Streaming API
 - ◆ 전체 데이터 중 1%를 랜덤으로 실시간 전송
 - ◆ 하루 400만건 정도 수집 가능
 - ◆ Global Trends 분석 등에 사용
- ◆ User Streaming API
 - ◆ 인증된 사용자에게 한 사용자의 모든 정보를 실시간 전송
- ◆ Site Streaming API
 - ◆ 여러 사용자의 user stream 데이터를 실시간 전송

Tweetstream Gem

- ◆ Streaming API 를 사용하여 실시간으로 데이터를 수집하기 위한 루비 소프트웨어

- ◆ 설정

```
require 'tweetstream'
```

```
TweetStream.configure do |config|  
  config.consumer_key = "CONSUMER_KEY"  
  config.consumer_secret = "CONSUMER_SECRET"  
  config.oauth_token = "OAUTH_TOKEN"  
  config.oauth_token_secret = "OAUTH_TOKEN_SECRET"  
  config.auth_method = :oauth  
  
end
```

Lab 4: 키워드 스트리밍

- ✦ `TweetStream::Client.new.track(keyword1, keyword2, keyword3)`
 - ✦ 제시된 검색어를 계속 트래킹
- ✦

```
TweetStream::Client.new.track( 'apple' ,  
  'samsung' ) do |status|  
  puts "[#{status.user.screen_name}]  
#{status.text}"  
end
```

Lab 5: 사용자 타임라인 스트리밍

- ✦ `TweetStream::Client.new.follow(uid1, uid2, uid3)`
 - ✦ 제시된 검색어를 계속 트래킹
- ✦

```
TweetStream::Client.new.follow(48661131,
163171634, 76295962, 128154362) do |
  status |
    puts "[#{status.user.screen_name}]
#{status.text}"
end
```

Assignment 3

- Crawling data from Twitter
- 코드 첫 줄에 본인의 학번, 이름을 comment 로 입력
- assignment3_본인영문이름.rb로 저장하고 압축하여 etl 로 제출
- 배점: 10

Assignment 3: Crawling data from Twitter

- ◆ 자유주제

- ◆ 트위터에서 수집하고 싶은 데이터를 수집하고 간략한 결과보고서 작성.
 - ◆ 트위터 데이터를 저장하고 간단한 통계 계산.
 - ◆ 예: ooo는 하루 평균 몇개의 포스팅. (RT는 몇 개 등등)
 - ◆ 제출해야할 자료: 소스코드, 수집된 트위터 데이터, 간단한 보고서

Questions?
