实验 5. 社交网络数据获取与分析

一．实验目的

1. 了解并利用微博 API 爬取数据
2. 对获取的数据进行初步统计分析

二．实验背景

新浪微博（Weibo/Sina Weibo，官方名称为微博），是一个由新浪网推出，提供微博客的服务网站。新浪微博是一个类似于 Twitter 和 Facebook 的混合体，用户可以通过网页、WAP 页面、外部程序和手机短信、彩信等发布 140 汉字（280 字符）以内的信息，并可上传图片和链接视频，实现即时分享。新浪微博可以直接在一条微博下面附加评论，也可以直接在一条微博里面发送图片、视频，新浪微博最先添加这两点功能。新浪微博是一个基于用户关系的信息分享、传播以及获取信息的平台，它占据中国微博用户总量的 57%，以及中国微博活动总量的 87%，是中国大陆访问量最大的网站之一。

新浪微博是现在对中文社交网络、短文本分析领域重要的分析对象。对于这些平台的数据获取，一般分为两种方式：第一种是通过正常的浏览网页获取完整的 HTML/SHTML 数据，再进行网页解析获取所需要的信息。第二种方法是通过 API（application programming interface）的方式获取数据。前者的优点是可以避免平台的权限限制。但是由于新型的前端架构，使得直接获取数据变得困难，并且解析网页也是一个复杂的工作，往往要随着网页机构变化而重新编写数据分析部分。而后者，通过 API 获取数据一般是平台方允许第三方获取没有冗余的数据的方法。当然同时也对获取的数据进行监控，并进行权限限制。本次实验就通过新浪微博提供的 API 获取数据。

数据清洗时进行数据分析的重要组成部分。由于新浪微博有大量的噪声微博（自动生成的分享类微博、垃圾账号的推销微博等）。这些数据对于后续的实验没有任何用处，而且是噪声内容需要剔除。针对这些噪声内容有很多的识别方法，有通过规则的方法或者利用统计的机器学习的方法。本次实验希望大家利用规则的方法来对数据进行降噪。具体内容可以见附录内容。

在社交网络中对用户的基本行为，以及特征进行分析是社交网络用户行为分析的基础内容。基于这些用户特征可以进行用户分类，以及用户行为预测等进一步研究。希望大家获取指定用户所有微博、关注者与粉丝并对这些数据进行简单

分析。

注意：使用 Eclipse 开发可能会出现问题，请使用 MyEclipse。

三．实验内容

首先根据新浪微博的引导（http://open.weibo.com/），创建一个属于自己的应用，如下图。

点击继续创建，然后填写一些基本信息后，点击微链接中的移动应用，即初步创建应用成功。

创建应用成功后，通过右上角我的应用进入到应用界面，在应用信息、高级信息下填写授权回调页等信息，如下图。

在 http://open.weibo.com/wiki/index.php/SDK 下载 java 的 SDK 后，按照要求修改 src/config.properties 文件中的信息。Client_ID 对应 App Key，client-SERCRET 对应 App Secret， redirect_URI 对应授权回调页地址。授权回调页是前一步填写的；App Key/Secret 可在应用基本信息找到。

完成这些基本步骤后，先需要授权获取一个 AccessToken，具体的获取步骤可以参考 SDK 首页的说明。首先执行位于 example/oauth2/文件夹中的 OAuth4Code.java 程序，运行后会弹出浏览器地址跳转到授权认证页面，然后输入你的微博帐号和密码，会调转到你的回调地址页面，url 后面会传递 code 参数。然后在 console 输入 code 就能获取到 oauth2 的 accesstoken。Accesstoken 作为一个信任该应用的凭证，获取到 AccessToken 后，就可以调用 API 获取数据了。

任务 1：利用 example/timeline/文件夹中所给的获取 public timeline 的代码，获取某新浪微博连续 1 小时以上的 public time line 微博信息。请统计所获得的微博总数，以及数目随时间的变化。

任务 2：请获取自己的微博信息，包括粉丝、关注列表和所有微博内容，并统计微博信息以及其粉丝/关注的用户基本信息。包括但不限于：用户发表微博的频率，用户的关注/粉丝用户的男女比例，地理分布情况等信息。

提示 1：SDK 的 example 文件夹中有大量可以执行的代码参考。

提示 2： 微博有比较严格的访问次数限制，超过一定频次可能被封号。

思考题

1. 使用 API 获取数据与爬取网页有何区别？优劣势如何？

附录

新浪开发者平台：

http://open.weibo.com/wiki/%E9%A6%96%E9%A1%B5

开发接口介绍：

http://open.weibo.com/wiki/%E5%BE%AE%E5%8D%9AAPI

开发应用包介绍：

http://open.weibo.com/wiki/SDK

Java SDK：

https://github.com/sunxiaowei2014/weibo4j-oauth2-beta3.1.1/

示例代码：

1. 获取 public-timeline:

```
package weibo4j.examples.timeline;
import weibo4j.Timeline;
import weibo4j.examples.oauth2.Log;
import weibo4j.model.StatusWapper;
import weibo4j.model.WeiboException;
public class GetPublicTimeline {
    public static void main(String[] args) {
        String access_token = args[0];
        Timeline tm = new Timeline(access_token);
        try {
            StatusWapper status = tm.getPublicTimeline();
            Log.logInfo(status.toString());
        } catch (WeiboException e) {
            e.printStackTrace();
        }
    }
}
```

2. 获取用户粉丝列表：

```
package weibo4j.examples.timeline;

import weibo4j.Timeline;

import weibo4j.examples.oauth2.Log;

import weibo4j.model.StatusWapper;

import weibo4j.model.WeiboException;

public class GetUserTimeline {

	public static void main(String[] args) {

		String access_token = args[0];

		Timeline tm = new Timeline(access_token);

		try {

			StatusWapper status = tm.getUserTimeline();

			Log.logInfo(status.toString());

		} catch (WeiboException e) {

			e.printStackTrace();

		}

	}

}
```

3. 获取用户微博列表：

```
package weibo4j.examples.timeline;

import weibo4j.Timeline;

import weibo4j.examples.oauth2.Log;

import weibo4j.model.StatusWapper;

import weibo4j.model.WeiboException;

public class GetUserTimeline {

	public static void main(String[] args) {

		String access_token = args[0];

		Timeline tm = new Timeline(access_token);

		try {

			StatusWapper status = tm.getUserTimeline();

			Log.logInfo(status.toString());

		} catch (WeiboException e) {

			e.printStackTrace();

		}

	}

}
```