Social Media Mining For Wheater Data Documentation

Release 0.1

Dominic Looser

December 17, 2015

CONTENTS

Contents:

CHAPTER

ONE

FLICKR

- Created by Ludicorp in 2004
- Acquired by Yahoo in 2005
- 6 billion images in 2011 (we)
- 87 million registred users in 2013 (we)
- 3.5 million new images daily in 2013 (we)
- Written in PHP

1.1 API

- REST endpoint: https://api.flickr.com/services/rest/
- Return formats: XML, JSON, ...
- Parameters: method, api_key, format

1.1.1 flickr.photos.search

Parameters:

- woe_id: A 32-bit identifier that uniquely represents spatial entities
- place_id: A Flickr place id

Response structure:

photos > photo

photos: page, pages, perpage, total
photo: id, latitude, longitude, place_id, title, woeid

1.1.2 flickr.places.getInfo

Get informations about a place. Parameters:

- woe_id

- place_id

response structure:

rsp > place
place > country
country > shapedata
shapedate > polylines, urls
polylines > polyline
urls > shapefile

rsp: stat
place: place_id, woeid, latitutude, longitude, place_url, place_type, place_type_id, timezone, name, woe_name, has_shapedata
country: place_id, woeid, latitutde, longitude, place_url
shapedata: created, alpha, count_points, count_edges, has_donuthole, is_donuthole

1.1.3 flickr.places.find

Returns a list of place objects for a given query string.

Parameter: query

Response: | rsp > places | places > place*

rsp: stat
places: query, total
place: place_id, woeid, latitude, longitude, place_url, place_type

1.1.4 woe id vs place id

WOE = where on earth

1.2 Python Library

We use the library called flickrapi. Documentation: http://stuvel.eu/media/flickrapi-docs/documentation/

CHAPTER

TWO

TWITTER

2.1 Basics

- 140 Characters per tweet
- 1.9 million tweets January 2009 (twitter api: up and running, p.4)
- 340 milion tweets each day (2012)
- launched July 2006
- Twitter Inc in San Francisco

2.2 API

Schema Tweet:

```
id
lang
text
created_at
coordinates
   [coordinates]
place
entities
   [hashtags]
      text
   [urls]
```

Schema Place:

```
bounding_box
   [coordinates]
      [float]
   type
contained_within
country
country_code
full_name
name
place_type
geometry
```

2.2.1 REST-api

https://api.twitter.com/{version}

2.2.2 Search

The Search API is not complete index of all Tweets, but instead an index of recent Tweets. At the moment that index includes between 6-9 days of Tweets. (https://dev.twitter.com/rest/public/search)

2.3 Tweepy

Python library used to connect to Twitter API through python.

Schema Place full_name

Schema Status streaming-api: contributors truncated text in_reply_to_status_id id favorite_count author

User follow_request_sent profile_use_background_image

_json follow_request_sent profile_use_background_image default_profile_image id verified profile_image_url_https profile_sidebar_fill_color

2.3.1 API

- API.rate_limit_status (http://docs.tweepy.org/en/v3.2.0/api.html#API.rate_limit_status)

Response Schema:

```
{
    rate_limit_context
        access_token
    resources
        *resource_type*
            *resource_name*
                limit
                remaining
                reset
}
```

Geolocation

- tweet is geotagged by user
- in germany 1% of tweets are geotagged
- Approximately 3-5% of all tweets are geo-enabled (https://github.com/Ccantey/GeoSearch-Tweepy)
- induce location from user profile
- induce location from tweet text

CHAPTER THREE

HELPER APIS

3.1 World Weather Online

3.1.1 Historical Weather API

docs

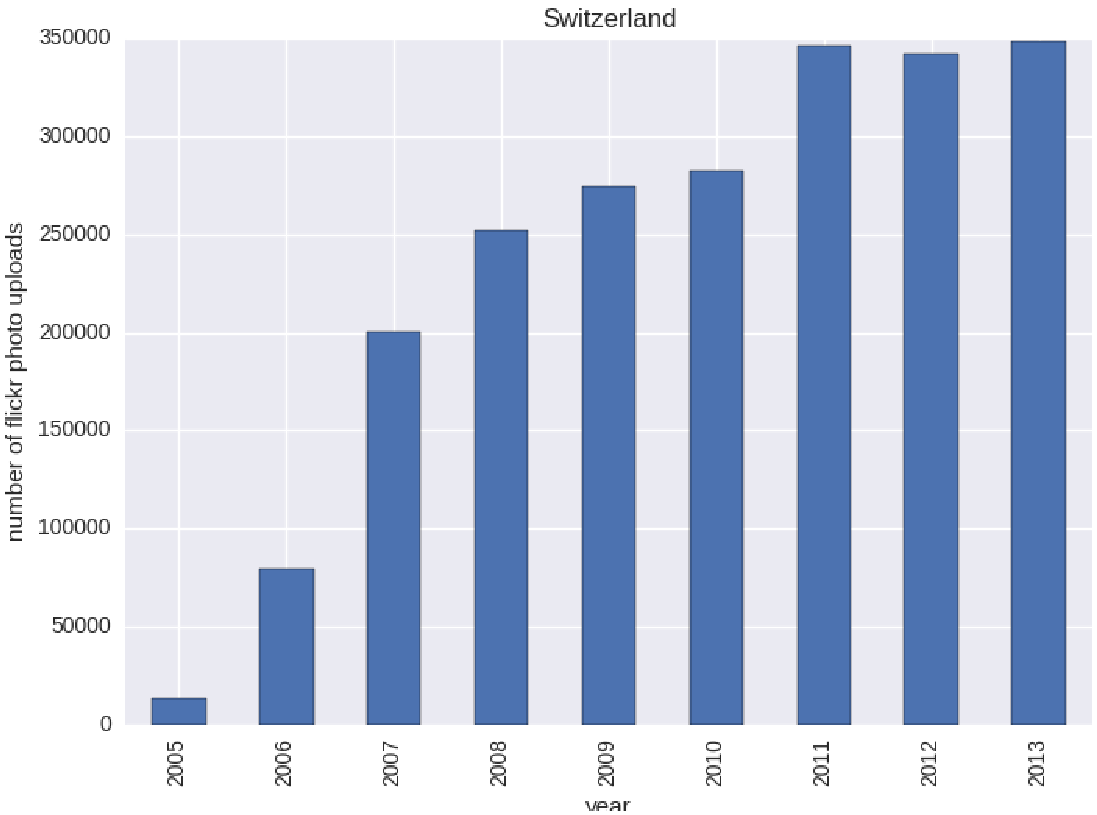
response format:

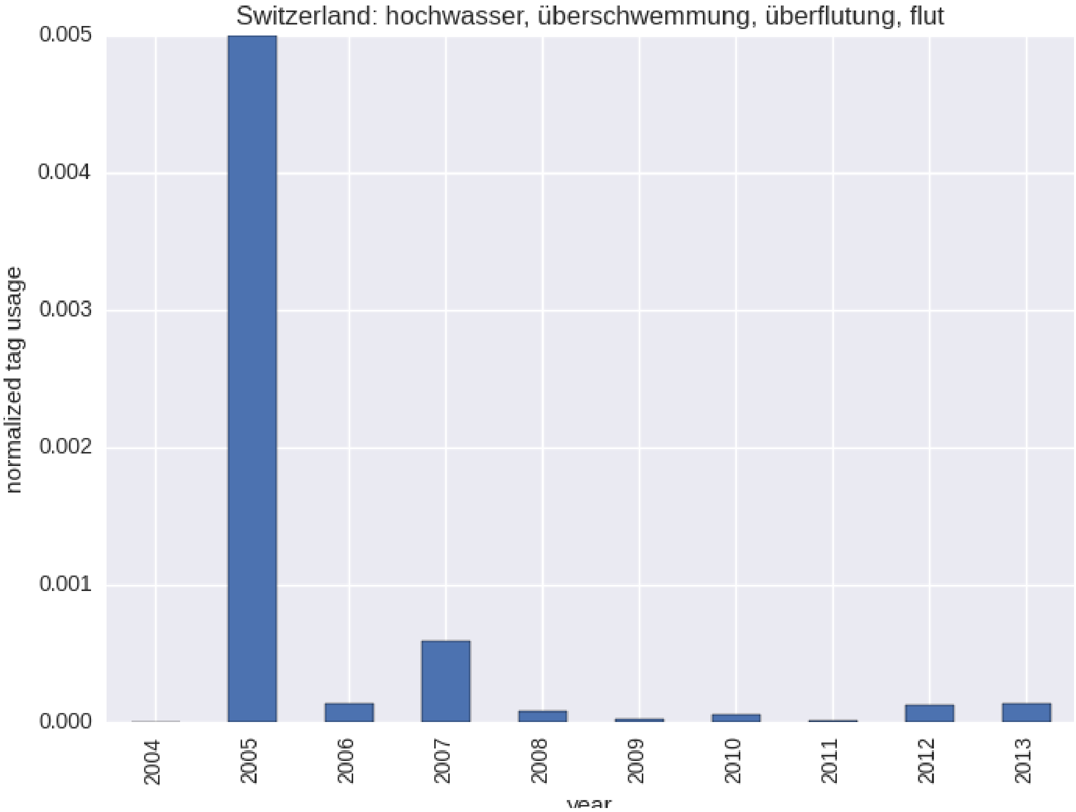
```
data
    request
        type
        query
    [weather]
        date
        [hourly]
            precipMM
```

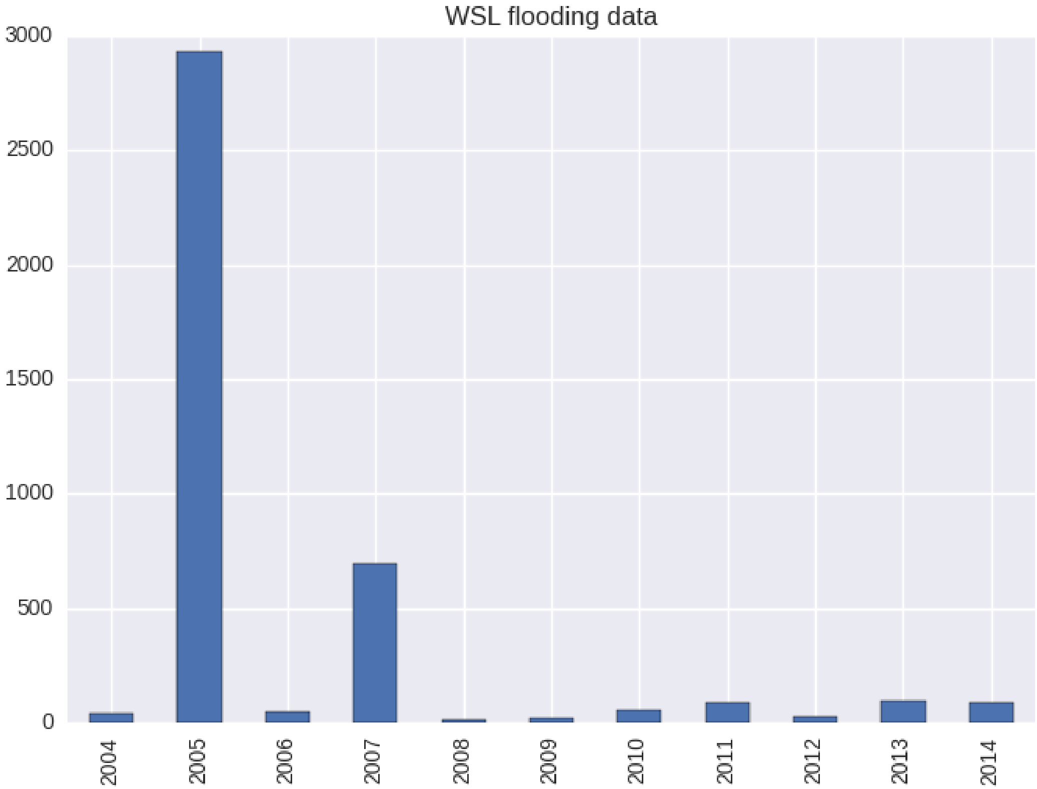
CHAPTER

FOUR

RESULTS




Switzerland: hochwasser, überschwemmung, überflutung, flut
normalized tag usage
0.005
0.004
0.003
0.002
0.001
0.000
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
year


WSL flooding data
3000
2500
2000
1500
1000
500
0
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014

CHAPTER

FIVE

CODE

All code is based on Python 3

5.1 Install

- Install anaconda (https://www.continuum.io/downloads)
- make a new conda environment called social-media-mining with:

```
conda env create -f .conda_requirements.yml
```

- activate this new environment with:

```
source activate social-media-mining
```

- download source code from git ...
- create new file called local_config.py in folder main with the line

  ROOT_DIR = path to you the root folder of the project

5.2 Important Libraries

- Pandas (data analysis)
- Matplotlib/Seaborn (plotting)
- flickrapi
- tweepy
- nltk (natural language processing)

CHAPTER SIX

.

6.1 main package

6.1.1 Submodules

6.1.2 main.flickr_analysis module

6.1.3 main.geo module

class `main.geo.`**`BoundingBox`**(*twitter_bounding_box=None*)

Bases: `object`

class `main.geo.`**`Map`**(*bounding_box*, *map_resolution=<MapResolution.INTERMEDIATE: 1>*)

Bases: `object`

`draw_densities`(*points*, *n_bins*, *color_map='Blues'*)

`draw_points`(*points*)

`save`(*path*, *format='png'*)

`show`()

class `main.geo.`**`MapResolution`**

Bases: `enum.Enum`

An enumeration.

`FULL` **= <MapResolution.FULL: 2>**

`INTERMEDIATE` **= <MapResolution.INTERMEDIATE: 1>**

class `main.geo.`**`Place`**(*twitter_place*, *wunderground_id: str*)

Bases: `object`

class `main.geo.`**`Point`**

Bases: `object`

`main.geo.`**`draw_map`**(*place*)

6.1.4 main.local_config module

6.1.5 main.store module

6.1.6 main.twitter_analysis module

6.1.7 main.utils module

class `main.utils.`**`Stopwatch`**

Bases: `object`

`start`()

`main.utils.`**`save_plot`**(*plot_id*, *directory*)

6.1.8 Module contents

PYTHON MODULE INDEX

INDEX