

1.0 LEARNING ANALYTICS DASHBOARD

1.1 System Overview

This document describes the installation and use of the Learning Analytics (LA) Dashboard. It is an instructor-facing, online tool for visualizing Twitter activity collected from students who post as part of a course.

The dashboard system consists of **two** main components

1. A **web interface** (front-end). This is the web-based interface the instructor sees when using the dashboard. It consists of a set of visualization panels that are each created from various queries that use collected Twitter data. It is housed in a single html file that uses Javascript and a pre-defined set of library functions to create queries and visualizations. It uses a third party repository to query data and set of library functions for the visualizations. An example of the dashboard can be seen here:

<https://dashboard.socialmediadata.org/educhat/>

2. A **data processing system** (back-end). The back-end consists of a single Python script that uses the Twitter streaming API to collect tweets based on a particular query. It processes this data and sends it to the third-party repository which is subsequently used by the front-end interface

1.2 Purpose

This describes how to use and set up the dashboard environment as well as details of the system itself.

1.3 Audience

The audience for the document is for researchers who wish to use and deploy the system, and for developers who wish to modify and customize the system as necessary. Some knowledge of the Linux operating system is required, and preferably knowledge of the Python programming language.

1.4 Points of Contact

1.4.1 Information

Dr. Anatoliy Gruzd, Associate Professor
Nadia Conroy, Postdoctoral Researcher

1.4.2 Coordination

This tool is managed, authored and owned by Social Media Lab @ Ryerson TRSM

2.0 BACKGROUND

The affordances of Twitter in teaching and learning are many. The use of social media such as Twitter may be helpful in enriching learning by relying on networks of social ties, improving engagement, and allowing autonomy to create tailored learning environments.

This software is a Learning Analytics (LA) dashboard to support Twitter use in teaching. It is an instructor-facing, web tool that collects and monitors Twitter data. It uses data visualizations to summarize and present information on real-time Twitter use by students. The Learning Analytics (LA) Dashboard was developed by the Social Media Lab at Ryerson University.

The dashboard analyzes tweets produced under a specific course hashtag, and displays the resulting visualizations for an instructor to use for their own exploration. The goal is to allow instructors to see the ways in which students engage with each and respond to course material using Twitter.

This document describes in detail the steps to install the Learning Analytics Dashboard. Instructions on how to use the dashboard once it has been installed is provided in this external document:

[Learning Analytics Dashboard User Guide](#)

2.1 Functionality Overview (Querying Course Hashtags)

The system is driven by the data collected through the Twitter Streaming API. The system listens for the presence of incoming tweets that are matched against a specific course hashtag. For example, if the dashboard were being used in the course, *Information Literacy and Instruction LIS2001*, the hashtag #LIS2001 could be used to designate tweets relating specifically to content for that course.

Each tweet with a hashtag is collected. The data is processed, and certain metadata from the tweet are extracted and then stored in an internal database (MongoDB). The dashboard system queries this database to generate visualizations that compose the dashboard.

3.0 SETTING UP DASHBOARD INSTANCE

The section describes how to instantiate a version of the dashboard to work with a particular course, using a unique hashtag to query the Twitter stream for incoming tweets.

It consists of a back-end script to collect, analyze and store Tweets, a node webserver to collect and send data, and a front-end to use a data source of tweets to query and present the content.

The following steps are based on using the Ubuntu Linux-based operating system, version 14.04

3.1 Download Dashboard Files from GitHub Repository

The system consists of the modules and external packages listed in Table 1. The file package is located on the GitHub repository `RUSocialMediaLab/TwitterDashboard`. All application files are installed from the repository to the directory tree:

File Type	Description
<code>TwitterDashboard/app.js</code>	Javascript file which starts the collection script and acts as a server to localhost
<code>TwitterDashboard/lib/dashboard_routes.js</code>	Javascript which queries the database and defines GET and POST requests
<code>TwitterDashboard/scripts/</code>	Directory containing all files used by the data collection script
<code>TwitterDashboard/scripts/create_stream.py</code>	Python script which connects to the Twitter API and stores data with MongoDB
<code>TwitterDashboard/scripts/config.json</code>	JSON file read by collection script to store configuration options, including caching frequency
<code>TwitterDashboard/scripts/keys.json</code>	JSON file used to store Twitter API Keys
<code>TwitterDashboard/views/index.js</code>	Embedded Javascript (EJS) template defining the overall layout of the site
<code>TwitterDashboard/views/partials</code>	Directory containing EJS templates for each chart or section of the site
<code>TwitterDashboard/public</code>	Contains all of the site's public static files, including CSS and images
Table 1: File types comprising dashboard system	

Complete the following steps to use the repository to install the dashboard.

- Navigate to the directory where you would like to install it:

```
$ cd ~/Dashboard
```

- Clone the repository. This will copy all the content files listed in Table 1 to your current directory

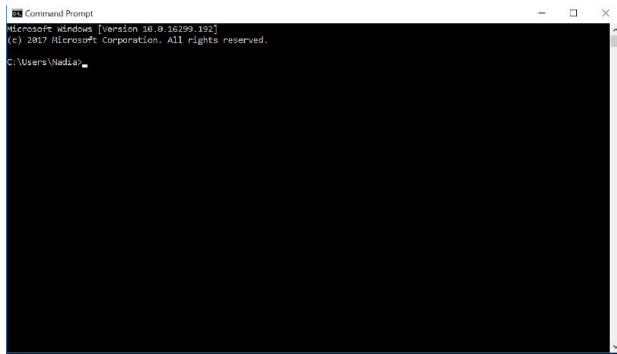
```
git clone https://github.com/RUSocialMediaLab/TwitterDashboard.git
```

- Go to the new dashboard directory

```
$ cd LA_dashboard_gen
```

3.3 Install Python & Library Dependencies

- a. First, open a Terminal console



- b. On Ubuntu 14.04 Python 2.7 and 3.4 is installed by default. If you are on a different platform follow steps b and c to install a Python environment. Otherwise proceed to step d.

```
$ sudo apt-get install build-essential checkinstall
$ sudo apt-get install libreadline-gplv2-dev libncursesw5-dev libssl-dev
    libsqlite3-dev tk-dev libgdbm-dev libc6-dev libbz2-dev
```

- c. Then install a version of Python 2.7

```
$ sudo apt-get update
$ sudo apt dist-upgrade
$ sudo apt-get install python2.7 python-pip
```

- d. Install the various third-party library dependencies:

NLTK interface:

```
$ sudo pip install -U nltk
$ sudo pip install -U textblob
$ python -m textblob.download_corpora
```

Twitter API:

```
$ sudo pip install tweepy
```

Java:

```
$ sudo apt-get update
$ java -version
```

MongoDB:

```
$ python -m pip install pymongo==3.8.0
```

If you don't see a version of java installed, run the command to get the Java runtime environment

```
$ sudo apt-get install default-jre
```

e. Download Stanford Named Entity Recognizer

```
$ wget http://nlp.stanford.edu/software/stanford-ner-2014-06-16.zip
```

Alternatively, download the stanford-ner.jar and classifier model located here: [Named Entity Recognizer version 3.4](#)

f. Unzip the contents to the destination folder. Install the unzip tool if necessary. The backend script will look for the entity recognizer in the /usr/local directory

```
$ sudo apt-get install unzip
$ unzip stanford-ner-2014-06-16.zip
$ sudo mv stanford-ner-2014-06-16 /usr/local/
$ sudo mv /usr/local/stanford-ner-2014-06-16 /usr/local/stanford-ner
```

3.4 Install MongoDB

To install the correct version of the database software, use the command:

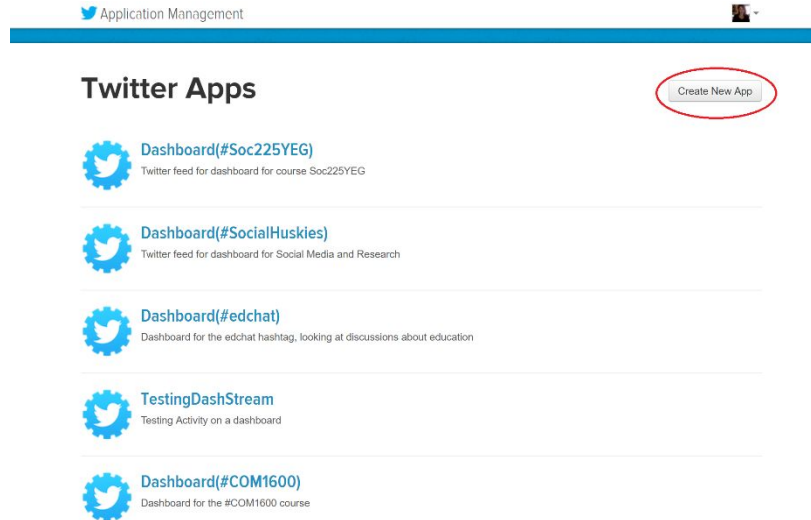
```
sudo apt-get install -y mongodb-org=3.6.8 mongodb-org-server=3.6.8
mongodb-org-shell=3.6.8 mongodb-org-mongos=3.6.8 mongodb-org-tools=3.6.8

sudo service mongod start
```

3.5 Register the Twitter App

To restrict the traffic on Twitter to a single API key, a new Application must be registered that specifies the version of the dashboard you wish to instantiate.

- If you have a Twitter user account, go to the Application Management page (<https://apps.twitter.com/>). Here you will see a list of your existing applications you have registered with Twitter.
- If you don't have a Twitter account you must [create one first](#) and then proceed with the remaining steps.
- Click on the Create New App button on the top right to create a new app for the dashboard.



- This will take you to the App registration page. Fill in the required fields, including the name of the app should follow the naming convention:

Dashboard (#COURSECODE)

The URL for the dashboard should be the actual web address of the dashboard. The URL follows the naming convention, <https://dashboard.socialmediadata.org/<hashtag>/>

- When you are done filling out the fields, click ‘Create Your Twitter Application’.

- After filling out the required fields, click the ‘Create your Twitter application’ on the bottom of the page. Upon doing so you will be taken to the main page for the application.

- Select the Keys and Access Tokens tab.
- Then click the Generate Access Token button on the bottom, as shown,

Twitter Application Management

LIS 2001 Test OAuth

Details Settings **Keys and Access Tokens** Permissions

Application Settings

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

Consumer Key (API Key)	VICsI78IRU5mOssgLQlomXX9
Consumer Secret (API Secret)	kw9X7IT11yAv0mwuiyw4mubSeguWFB2p3omuNCpeR7wFcus3mK

Access Level	Read and write (modify app permissions)
Owner	nadia_conroy
Owner ID	859796321057861634

Application Actions

[Regenerate Consumer Key and Secret](#) [Change App Permissions](#)

Your Access Token

This access token can be used to make API requests on your own account's behalf. Do not share your access token secret with anyone.

Access Token	859796321057861634-7zAvBEC667xN3z51xBsXRjHvzMrCmAm
Access Token Secret	CA6q1p98H1bEPCOBVYy6yhy73DIUd4gBHPTYhdiJmMckM

Access Level	Read and write
Owner	nadia_conroy
Owner ID	859796321057861634

- You will need the keys and access tokens on this page, specifically Consumer Key, Consumer Secret, Access Token, and Access Token Secret. Make note of them for later.

3.6 Edit the Config file

The next step is to edit the config file so that it reads the proper Twitter Keys access keys .

- Navigate to the directory of dashboard config files.

```
$ cd (your git directory)/scripts
```

- Open the dashboard.config file in a text editor:

```
$ vi keys.json
```


- Upon opening the file, you will see blank access keys. Populate the key/value pairs in this file, placing each Twitter key within the quotes:

```
{
  /* PUT KEYS HERE */
  "access_token": "",
  "access_token_secret": "",
  "consumer_key": "",
  "consumer_secret": ""
}
```

3.7 Install Node and Node Dependencies

- Navigate to your working directory:

```
cd (your git directory)/scripts
```

- Install NodeJS:

```
sudo apt-get install nodejs
```

- Install Node Package Manager:

```
sudo apt-get install npm
```

- Install node packages:

```
npm install ejs@2.6.1
npm install express@2.5.8
npm install mongoose@5.5.11
```

3.8 Start the Dashboard

To start the server and collection script, enter the following command from your working directory, replacing ‘hashtag’ with the tag you would like to track. This will also automatically start the collection script.

```
node app.js hashtag
```

To see the dashboard, go to: <http://localhost:8999/hashtag>, replacing ‘hashtag’ with the tag you’re replacing.

4.0 TESTING THE DASHBOARD

- To first test your dashboard, send some tweets! Log into your Twitter account and create a tweet to make sure the script will add it to your stream. Make sure to include the hashtag for the course.
- If you open the dashboard before it has collected any data, an error message will be displayed notifying you that you need to run the collection script.
- To test if tweets are being added to the database, open the MongoDB interface using the command “mongo”, then type the following, replacing hashtag with your own tag:

```
use hashtag  
db.tweets.find()
```

- If data is being collected, this will show you a sample of the data.