Team Assignment 1: Data collection and documentation from Twitter's Streaming API

Objective of this assignment

- Execute large-scale data collections from digital platforms via Application Protocol Interfaces (APIs; here: Twitter)
- Introduce you to the programming and scripting language Python, using Jupyter Notebook as an interface
- Explore JSON objects using JSONviewer.stack.hu or ATOM's pretty-json package
- Learn how to document raw data accurately (for you, your future self, and others to build upon your work)

Prerequisites

- Please install all necessary software, and follow the web clips in the tutorial published on Canvas. Make sure to install the right packages (see our detailed installation instructions).
- Obtain authentication credentials for the Twitter API (check the web clip on how to do that) – **it may take a couple of days for Twitter to send you the credentials, so apply for them soon enough.**

Introduction

The academic community increasingly uses data from digital and social media to conduct research (see, e.g., the Twitter paper used in this course, or the Spotify paper by Hannes, https://tiu.nu/spotify). In this assignment, you will collect real-time data from the Twitter Streaming API[1] related to an event (e.g., a TV show, a press conference, a concert livestream, etc.), *while* the event is occurring. Afterwards, you will document and "package" the data for the student and research community to (re)use, based on a readme template available at http://tilburgsciencehub.com/workflow/documenting-data/.

The data collection will also serve as an input for team assignment 2 (although you can choose from the data sets that other students have uploaded, including your own).

Tips for picking an event:

- Make sure your event has a sufficient number of users that tweet about it. (We'd like to avoid that you end up with a shockingly small dataset with, let's say, less than 1,000 tweets).
- Further, does the event have recognizable hashtags that you can base your tracker on?
- Also, have a look at the show's website or social media profiles. How is the company/event present online / can you expect activity on Twitter?
- In doubt? Send us a quick message on WhatsApp.

Collect your data

In gathering the data, please use the example API scraper that has been made available to you ("tweepy_student"). Adapt the code to include your chosen hashtags and then try to run it. First, make a test run, e.g., download the data for a period of, e.g., 5 minutes. If you are sure that your code runs, you can start with your actual data collection.

Run your scraper while the event is happening.

[1] The name of the API is "streaming API", because it shares Twitter data in real-time. Once connected to the API, data flows/streams to your computer.

- Start about 10 minutes before, and stop approximately 10 minutes after.
- During the collection, always check whether the file size is increasing; the scraper may likely crash, so monitor it well and restart it if necessary.
- It makes sense doing the scraping on two computers concurrently (with different login credentials) to prevent data loss (e.g., one computer crashing during the event).

Tip: The scraper automatically adds to the output file, so, if you first practice, and later on add real data for your show, then both of these data are included. To make sure you're starting "clean," you should first delete the .json output file.

Document your data

Please document your data by filling out the readme.txt template published on http://tilburgsciencehub.com/workflow/documenting-data/.

- Try to answer the questions to the best of your ability.
    - Imagine you would have to work with this data in the future – how would you write up the documentation so that you (and your future self) may understand it?
    - In your writing, be as concise as possible. We've had some recent experience using tools like Grammarly to improve our writing (there is a free version available), but there are certainly alternatives available!
    - The original paper on which the readme template is based on (references available on the site) provides a few helpful (filled in) examples that may provide some inspiration.
- We recommend you to use a proper text editor (e.g., ATOM) to work on your readme.
    - Good text editors will also indicate to you where new lines should occur (to make sure that the text stays readable).
- Next to the .txt, **please convert the readme to a PDF file**, which we can easily use to grade your readmes on Canvas.

Below, we give you specific instructions for each of the sections in that readme.txt file.

1. Name of the dataset/database
   Please pick a good name for your dataset. This name will be the first thing potential users of your data will see.

2. Section 1 (Motivation)
   In motivating your data collection, please bear in mind the data should be suitable to use in an academic research project. A few examples of (possible) motivations to collect data from Twitter are (but are certainly not limited to…):
    - How does the public comment upon live press conferences?
    - Which artists generate the most engagement during XYZ on Twitter?
    - During a game streamed on Twitch, what are drivers of user engagement?
    - What does the engagement pattern (e.g., volume, retweets) look like for TV show X?
    - To what extent does the origin of a Tweet (e.g., official Twitter account of the event versus regular users) influence the number of retweets?)

3. Section 2 (Composition)
    - To answer some of the questions in this section, you need to explore the JSON objects that you have collected. To do so, you can copy-paste one object at a time to a JSON viewer (e.g., https://jsonviewer.stack.hu, or ATOM), which helps you to visualize its tree structure and understand its content.

    - [Optional]
      Some questions can also be (more accurately) answered by writing a small parsing script. However, doing so is optional and should be attempted by those students that wish to be challenged (i.e., already engage with material that will be taught at a later moment in this course).
        - When parsing, select *which* elements of the JSON tree you would like to parse.

- Think about attributes like creation time or text to do a quick search on any inappropriate use of language. However, do not overcomplicate it (stick to the goal of your task to collect data and document it to the best of your ability).
- Pursue to hand in a high-quality code (e.g., have a clear structure, annotate it using Markdown cells, try to formulate every command well, and make sure it contributes to the actual outcome – your parsed data). Aim to make your script free of mistakes, so that it directly runs on our computers, too. Use efficient error handlings (i.e., don't wrap everything in a big try/except), and name your input and output files. We have made available coding tips on http://tilburgsciencehub.com/tips/coding/.
- We invite you to share snippets of your parsing scripts via Gists on GitHub with other teams. You can post URLs to these Gists for others to view/use/reuse on Canvas.

4. Section 3 (Collection process)
   Please answer all questions.

5. Section 4 (Preprocessing/cleaning/labeling) – please skip this section, but leave the unanswered questions in the document.

6. Section 5 (Uses)
   Please answer all questions.

7. Section 6 (Distribution) and section 7 (Maintenance) – please skip these sections, but leave them unanswered in the document.

Preparing your shareable data submission

Your submission consists of several parts, **packed into one zip file**:

1. Your collected raw data in JSON format.
   - Please let (part) of your dataset title reappear in the name of this raw data. Don't use spaces, though, and keep the name brief.
   - It's also possible to include more JSON files if you have collected more than one (e.g., on multiple computers, or at different moments in time).

2. The accompanying readme.txt file (+ corresponding .pdf file).

   While a regular "readme.txt" is sufficient, you can *optionally* choose to produce your readme in Jupyter Notebook, including some of the statistics that you may be interested in. In that case, you can submit the Jupyter Notebook (.ipynb) *and a generated PDF version* of the same notebook.

3. [Optional] Parsing scripts used to reproduce some of the metrics reported in your readme (e.g., Jupyter Notebook file(s) (.ipynb), Python (.py) files, or R scripts (.R)).

Attention: Please consider your submission "*publishable*." I.e., if you wish to remain anonymous later on, do not use your last names anywhere in the submission. Neither use any student numbers in the submission. We match your grades using your team numbers.

Submission and grading

- Submit your readme.***PDF*** on Canvas – no PDF, no grade!

- Further, pack all relevant files (JSON data, readme.txt & readme.pdf, and [optionally] parsing scripts) in one zip file. Then, email the zip file to rsm@tilburguniversity.edu **exclusively using https://filesender.surf.nl.** We do not accept submissions via Google Drive/WeTransfer, etc.

  Give it a good file name ("short name"), **including your team number**. The template is: "teamXX_your-title-goes-here." Replace XX by your team number, and replace the title (spaces replaced by dashes) with your own dataset title.

- Readmes will be graded based on the rubric (see attachment).

Feedback / Q&A

- We will use the feedback livestream on 24 April 2020 to discuss the various issues you may encounter in filling in your readme's (and potential solutions).
- Quick questions can also be answered in our WhatsApp chat.

Good luck! / Succes!

Grading rubric

You can use the evaluation rubric below to understand how you are evaluated in this group assignment, and how we judge an element to be e.g., insufficient, good, or excellent. Further, you can use it to get detailed feedback on your performance once results are known and learn about improvements you could make. Your final grade is computed as the sum of all points.

	Ratings			
	Exceeds Expectations (10)	Meets Expectations (6.5)	Below Expectations (3)	Points
1. Gather your data	Data is present in one (or multiple) JSON files, included in one submitted zip file. The data has been appropriately collected for the duration of the event and can be opened in the JSON file format.	Data is present in one (or multiple) JSON files, included in one submitted zip file. The data has been collected with gaps (i.e., the data collection stopped unintentionally in-between), but can still be opened in the JSON file format.	Data is absent from the zip file, or, if present, cannot be opened properly using the JSON file format.	3
2. Readme.txt – Name of the dataset	The dataset's name is concise and accurate, and describes the event and/or context of the data collection. Beyond that, the title entices the reader to download the data, is broad and generalizable, and conveys a sense of timelessness.	The dataset's name is concise and accurate, and describes the event and/or context of the data collection reasonably well.	The dataset's name is not concise (too lengthy) or not accurate (mismatch between title and content). As such, the title doesn't describe the event and/or context of the data collection correctly.	0.5
3. Readme.txt – Motivation	The purpose of the data collection and the choice for a particular event are motivated remarkably. For example, students point out a highly impressive (collection of) research question(s) that could be answered using this data.	The purpose of the data collection and the choice for a particular event are motivated adequately. For example, students point out a possible (collection of) research question(s) that could be answered using this data. While this/these RQ(s) are feasible given the collected data, either the description or relevance itself may not be optimally described.	The purpose of the data collection and the choice for a particular event is not motivated in a sufficient way. For example, it may not be possible to provide answers to the proposed research question(s) with the collected data, or the description of the RQ(s) is unclear.	1.5
4. Readme.txt - Composition	The description is based on a combination of manual inspection of (some of) the JSON objects (e.g., in JSON Viewer or ATOM), and also using parsing scripts that answer some of the questions (e.g., counting the number of objects, scanning the data for inappropriate words, etc.). Most questions have been answered in great detail, and the overall description is complete so that a potential user can judge the content of	The description is based on the manual inspection of (some of) the JSON objects (e.g., in JSON Viewer or ATOM). Based on that inspection, the description of the dataset is accurate and written up concisely. While not all questions are answered in greatest detail, the overall description seems reasonably complete so that a potential user can judge the content of the data.	A description of the data composition is provided but is either incomplete (does not provide a picture of the data composition that comes near to being complete) or too many statements in the description are unclear or convey incorrect information on the content of the data.	1.5

	the data and its potential applications.			
5. Readme.txt - Collection process	The data collection is described in a highly accurate way so that – if ported back in time to the event in question – other researchers could directly replicate the exact data collection. Further, students base their answers to questions 3.9 onwards (consent) on Twitter's API Terms of Use or any other relevant documents.	The data collection is described in a sufficiently accurate way so that – if ported back in time to the event in question – other researchers could reasonably well replicate the data collection.	The data collection is only described briefly, or not in a clear way. As such, many questions are remaining if researchers were trying to replicate the data collection.	1.5
6. Readme.txt – Uses	The description gives a concise and compelling account of its potential use situations, and when the use of the data would rather be inappropriate.	The description provides a good account of how the data could be used, and when the data should not be used.	The description does not give a sufficient account of how the data could be used, and when the data should not be used (e.g., arguments for either use or misuse are incorrect or not feasible given the nature of the data).	1.0
7. Readme.txt – Format	The template's format has been maintained, and the .txt file (and converted PDF document) are readable and visually appealing. Also, the readme file and corresponding statistics are generated from one Jupyter Notebook.	The template's format has been maintained, and the .txt file (and converted PDF document) are readable and visually appealing.	The template's format has not been maintained. The .txt or PDF files are hard to read (e.g., inconsistent line breaks), and hence not visually appealing.	1.0
8. Bonus – research potential	Wow. Fantastic work. Overall, we're incredibly enthusiastic about this data collection and its potential use cases! You should think about developing this into a proper paper or published repository.			Range 0 -1 point bonus (total rounding to ten!)