



TWITTER METADATA

CS 6604 Final Report

Abstract

The Twitter Metadata project was a class project as part of CS 6604. In it, I created TweetID, a tool for twitter collection collaboration and I define standards for collection level and tweet level metadata.

Michael Shuffett
mshuffett@gmail.com

Table of Contents

Table of Figures	2
Executive Summary.....	3
Users' Manual	4
Developer's Manual	4
Technologies Used	5
Setup Guide.....	5
Prerequisites	5
Instalation Requirements.....	5
Before Running for the first time.....	5
Running command-line commands	5
High-level explanation of files.....	6
Current Limitations (next steps)	6
Lessons Learned	7
Proposed Timeline	7
Problems and Solutions	7
Tweet-Level Metadata	7
Collection-Level Metadata	8
TweetID	11
Acknowledgements.....	15
References	16
Apendix A: Screenshots	17

Table of Figures

Figure 1: TweetID Homepage	4
Figure 2: Proposed Timeline	7
Figure 3: Example Namespaced JSON Extracted Using TweetID	8
Figure 4: Who, What, When, Where, How of Metadata	9
Figure 5: Core Prov Objects	10
Figure 6: PROV-O Starting Point Classes	12
Figure 7: Example Collection Level Metadata in Turtle Syntax.....	13
Figure 8: YourTwapperKeepr JSON File	14
Figure 9: Tab Seperated Value Input File	14
Figure 10: Upload Screen	17
Figure 11: Collections View	18
Figure 12: Collection Details View	19
Figure 13: Merged Collection Details View	20
Figure 14: Tweets Screen	21
Figure 15: Tweet Detail Screen	22

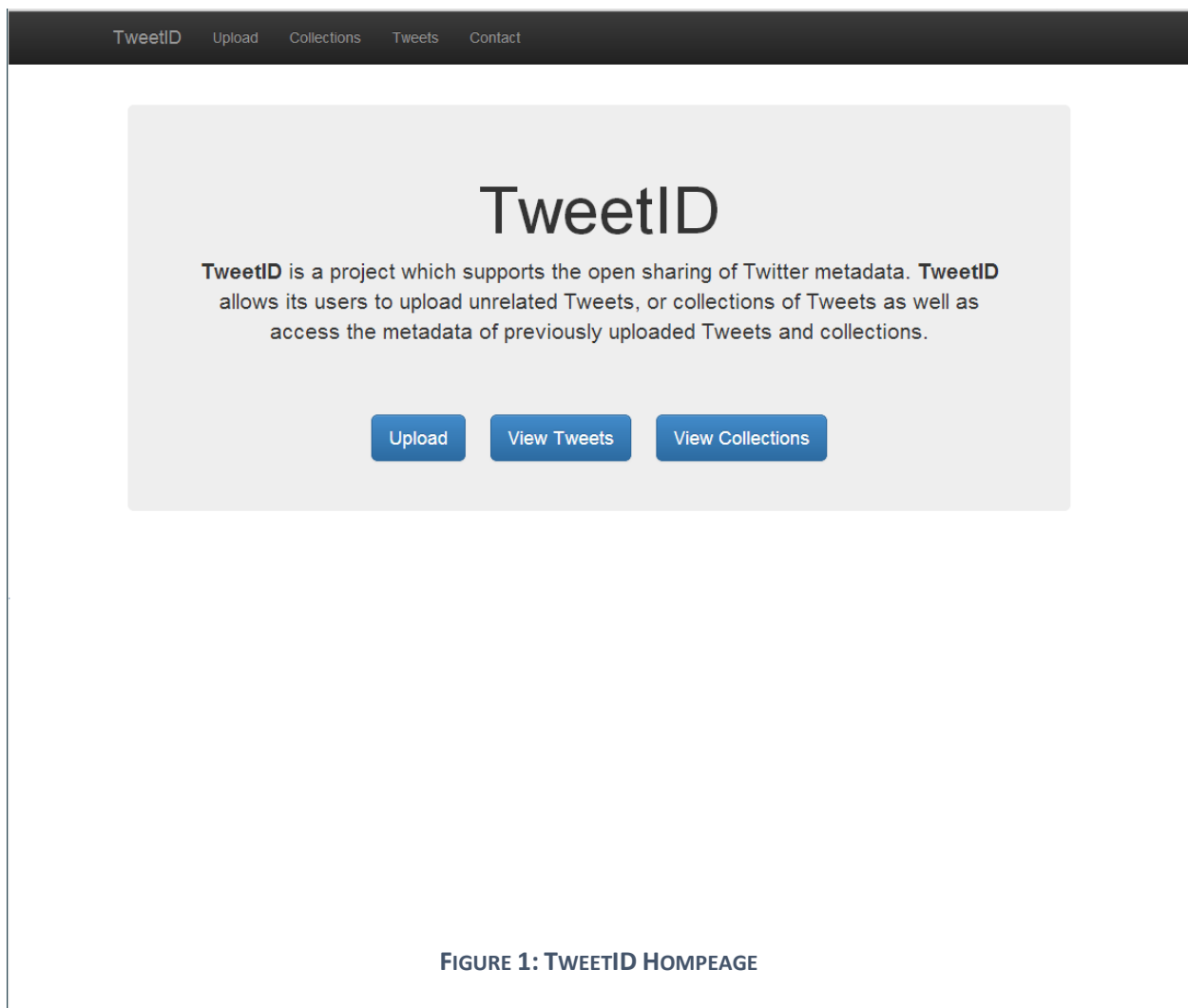
Executive Summary

A number of projects and research efforts work with collections of tweets. Of particular interest is the collection of tweets related to world events. Many organizations have their own individual tweet collections regarding specific events; however, there is currently no effective support for collaboration. Metadata standards foster collaboration by allowing groups to adhere to a unified format and seamless interoperate. In part one of the Twitter Metadata project, I define a tweet-level metadata standard that leverages the Twitter API format, as well as a collection-level metadata standard which combines Dublin Core and PROV-O. By combining two diverse existing standards (Dublin Core and PROV-O) into a single RDF document, the proposed standard is able to capture both the descriptive metadata as well as provenance of the collections. In part two of the Twitter Metadata project, I create a tool called TweetID in order to further foster collaboration with tweet collections. TweetID is a web application that allows its users to upload tweet collections. TweetID extracts, and provides an interface to, the underlying tweet-level and collection-level metadata. Furthermore, TweetID also provides the ability to merge multiple collections together, allowing researchers to compare their collections to others', as well as potentially augment their event collections for higher recall.

Users' Manual

A video demo can be found through the file *Michael Shuffett – TweetID demo.mp4* which goes step by step through each screen and piece of functionality of **TweetID**.

TweetID is a project which supports the open sharing of Twitter metadata. With **TweetID**, you can upload tweet collections, view uploaded tweets and collections, and merge collections. **TweetID** extracts collection level metadata from the uploaded collection. For user's interested in metadata, **TweetID** also provides unique uri's for collections as well as tweets. **TweetID** also extracts the standard Twitter JSON from a variety of formats (useful for converting for collaboration).



Developer's Manual

Technologies Used

git. **Python 2.7**. **Flask** for the server. **sqlite3** for the db. **SQLAlchemy** with **Flask-SQLAlchemy** for the ORM. **jinja2** for the templates. **WTForm** and **Flask-WTForm** for forms, validation, and uploading. **Bootstrap** and **Flask-Bootstrap** for the CSS. **jQuery** for the client side scripting. **Flask-Script** for easy management.

Most of the code is pretty self-explanatory and serves as the best natural documentation. The challenge is understanding the large array of technologies used.

For an introduction to Flask, I recommend that you go through the documentation on [the flask website](#).

Setup Guide

Prerequisites

- Python 2.7
- pip

Installation Requirements

The requirements have been listed in `requirements.txt`. In order to install all of the requirements, simply run `pip install -r requirements.txt`. (You might need sudo).

Before Running for the first time

You will need to edit the `tweetid/config.py` file. Change `UPLOAD_FOLDER` to the absolute path to the folder to store the uploads in and change `INSTANCE_FOLDER_PATH` to a temporary folder. On windows I recommend `C:/Users/<user name>/AppData/Local/Temp/` and on Linux I recommend `/tmp`. You will then need to initialize the database as shown below.

Running command-line commands

TweetID uses **Flask-Script** for runnable commands from the command line. You can find these commands in `manage.py`. To see the command line help just type `python manage.py`. To see the help for a specific command type `python manage.py command -h`.

To start out, cd into the same directory that `manage.py` is in.

Initializing the database

First we need to init the database (currently sqlite3 through SQLAlchemy and Flask-SQLAlchemy).

```
python manage.py init_db
```

Clearing the database

```
python manage.py drop_all
```

Starting the server

```
python manage.py run
```

By default the server runs only on the localhost. In order to specify a different ip address or port you can use the following command. Again, to see the help information for this or other commands you can pass the `-h` flag.

```
python manage.py runserver -t HOST -p PORT
```

[Load a YourTwapperKeeper JSON file from stdin](#)

Note: Some of the collection level metadata may be hard coded.

```
cat file.json | python manage.py import_json
```

[Load a tsv file of the format provided by QCRI from stdin](#)

Note: Some of the collection level metadata may be hard coded.

```
cat file.tsv | python manage.py load_tsv
```

[Downloading json files from YourTwapperKeeper](#)

Currently files are simply downloading using the json API and wget (not programmatically). You can use a call like follows. (Replace the variables between <>). Any tweet limit above ~80000 was found to cause a server error.

```
wget http://spare05.dlib.vt.edu/apiGetTweets.php?id\=<archive id>\&l\=<tweet limit> -O <output filename>
```

High-level explanation of files

The main file is `app.py`. The setup of the server and all of the routing happens there. `models.py` is where the **SQLAlchemy** models are setup. `load_json.py` and `tsv.py` are utility files for loading the respective files. All of the templates are in the `templates/` folder. They all inherit from `base.html`. The `static/` folder is where all of the static files that are loaded are stored (js, css). `tweetid.db` in the root folder is the sqlite database. It isn't in the git but I will include a copy of it in the files so the developer has some test data to work with.

Current Limitations (next steps)

- Currently upload only works on the tsv files with the format from QCRI. Loading the YourTwapperKeeper JSON files should be extremely easy since the code is already largely there.
- Extra input metadata should go into an extra field in the database and populate the organization's namespace.
- Merged collections aren't persisted.
- The name of a collection is being used as the unique id but that might not be desirable.
- After a collection has been merged it would be nice to still know where the tweets came from.
- If the server is to be used for more than research purposes, we should use an enterprise level server not the internal Flask server.

Lessons Learned

Proposed Timeline

Task	Date
Reach out to stakeholders	Feb 5
Initial meeting with Mohamed	Feb 14
Gain access to QCRI Twitter collections	Feb 16
Gain access to VT Twitter collections	Feb 26
Proposal for metadata fields	Mar 6
Have base architecture for metadata extraction implemented	Mar 20
Extracted desired features from collections	April 10
Allow for merging of collections	April 25
Turn in deliverables	May 8

FIGURE 2: PROPOSED TIMELINE

Problems and Solutions

Tweet-Level Metadata

I spent some time evaluating potential solutions to a tweet-level metadata standard. Two design goals emerged that help inform the final decision. 1) Make conversion to the new standard as straightforward as possible for all varieties of stored formats. 2) Make the standard flexible enough to support any metadata fields an organization might have.

A key insight emerged that leads to the effective support of the first goal. No matter what format the tweets end up being stored as, they were all initially ingested from the standard twitter API format (Twitter, Inc., 2013). The format behind this API is well defined and inclusive.

However, the Twitter API format is rigid. In order to meet the second goal, the approach of namespacing the JSON by organization was taken. The JSON which would have originated from the Twitter API is nested under a `{Twitter:}` object, while the fields unique to any organization are listed under the name of the organization (see Figure 3 for an example that was automatically extracted using TweetID). This approach maintains the high detail and interoperability of the Twitter API standard, while allowing organizations to specify any additional fields they may desire.

Recommended Format

The recommended metadata format for a tweet is to use JSON where under the root object there is a mandatory “twitter” object and optional objects for each organization. Under the “twitter” object, either “id” or “id_str” should always appear. All other fields are optional. The organization objects are meant to be a catchall for any items that are not representable by the Twitter API format. The idea is to define a mapping from your native format to the Twitter format within TweetID. Anything that is not

mappable can be automatically contained in the organization-level metadata. Figure 3 is an example of the desired JSON that was extracted automatically using TweetID. The tweetid object has a single entry, collections, which automatically has a list of all of the collections the tweet belongs to.

JSON

```
{
  "twitter": {
    "entities": {
      "urls": [
        {
          "expanded_url": "http://dlvr.it/2rv5PP"
        }
      ]
    },
    "created_at": "2013-01-27 08:37:34",
    "id_str": "295450465339064321",
    "coordinates": {
      "type": "Point",
      "coordinates": [
        10000.0,
        10000.0
      ]
    },
    "screen_name": "Fachrurrozy__"
  },
  "tweetid": {
    "collections": [
      "Queensland Floods"
    ]
  }
}
```

FIGURE 3: EXAMPLE NAMESPACE JSON EXTRACTED USING TWEETID

Collection-Level Metadata

In addition to the Tweet-Level Metadata standard, a collection-level standard was required. The standard needed to include metadata covering a wide variety of areas.

The following are some of the fields that were identified that the standard should support:

- start and end, time and date stamps, indicating the time coverage of the collection
- geographic coverage of the collection
 - of those tweeting
 - of what is discussed in the tweets
- names and other information on the person, group, and institution that did the collecting
- keyword lists or other query characterizations used to get the tweets

- other details on how the tweet collection was prepared
 - if there was filtering of the original raw tweets collected
 - name/URI of raw tweet collection
 - format of that collection
 - details on how this collection was obtained from the raw tweet collection e.g., features and classifiers and parameters employed
 - if additional metadata has been added after the raw collecting, e.g., labels -- and details about that process

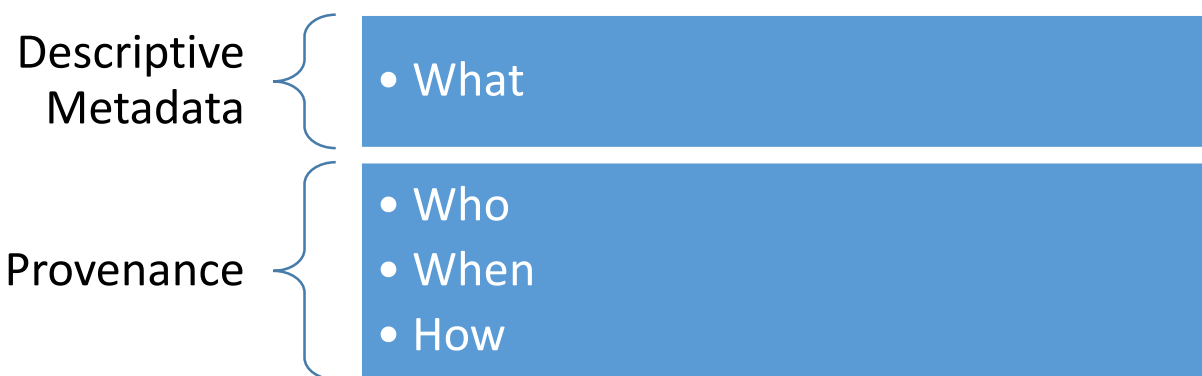


FIGURE 4: WHO, WHAT, WHEN, WHERE, HOW OF METADATA

It was clear that the typical metadata standards such as Dublin Core (DCMI Usage Board, 2012), were not sufficiently expressive to capture some of the details about the preparation of the collections. As you can see in Figure 4, the metadata was split into the who, what, when, where, and how, of the collection. “The what” of the collection could be addressed using typical metadata representations such as the Dublin Core Terms (DCMI Usage Board, 2012) but the who, when, and how of the collection would need to be expressed using a different type of metadata standard.

PROV is a set of W3 technical reports about provenance. In PROV-Overview, provenance is defined as “information about entities, activities, and people involved in producing a piece of data or thing, which can be used to form assessments about its quality, reliability or trustworthiness (Groth & Moreau, 2013).” Within the PROV family of documents, I identified PROV-O an OWL2 ontology allowing the mapping of the PROV data model to RDF (Belhajjame, et al., 2013).

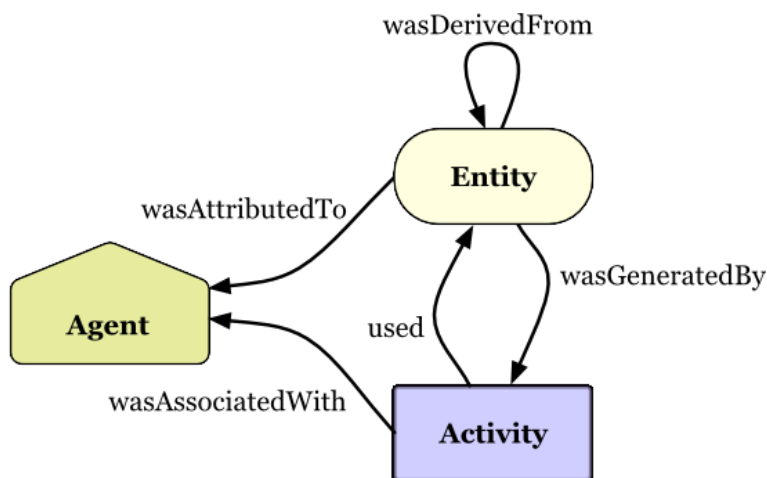


FIGURE 5: CORE PROV OBJECTS

In PROV Model Primer, the authors present a conceptual overview of PROV which includes three main types of objects: entity, agent, and activity as seen in Figure 5. PROV-O defines a set of *Starting Point Classes*, which are analogous to Dublin Core Terms—the key set of classes and predicates that can express the majority of applications, shown in Figure 6.

Recommended Format

By combining Dublin Core and PROV-O, we can achieve the desired collection-level metadata standard. The turtle representation for RDF is recommended but the format is syntax agnostic. For each collection, a `prov:Collection` should be defined that represents it. Under this collection there are five required fields:

Field	Description
dcterms:title	The title of the collection
dcterms:description	A brief description of the collection
prov:hadMember	A list of one or more tweets identifiable by a unique uri. (TweetID uri recommended)
prov:attributedTo	Minimally a single attributedTo entry mapping to a <code>prov:Organization</code> which describes the organization behind the collection. Multiple entries describing additional organizations or people (<code>prov:Person</code>) are optional.
prov:wasGeneratedBy	A <code>prov:activity</code> that describes how the collection was created. Further described below.

One or more `prov:atLocation` fields is also recommended for each location in the underlying tweets. Each location should be represented using ISO 3166-2 (International Organization for Standardization, 2012).

An activity should also be defined which precisely describes the process used to collect the tweets. It mandatorily includes `prov:startedAtTime` and `prov:endedAtTime` which are represented by the `dateTime` datatype from the W3 XMLSchema (Biron & Malhotra, 2004).

The standard requires that all of the above constraints be met; however, it does not limit the use of additional PROV-O Starting Point Classes or Dublin Core Terms. Currently, all tweet collections examined have used keyword queries as their source. A full example of a metadata representation in turtle syntax is presented in Figure 7.

TweetID

TweetID was the tool I created to support the unique identification of collections and tweets, the automatic standardization of tweet metadata, the extraction of new metadata fields from tweets and collections, the presentation of collections and tweets, and the merging of tweet collections. See figures 2-7 and the video, *Michael Shuffett – TweetID demo.mp4*, for an overview of TweetID. I made TweetID a web application to potentially support the tool being accessed by multiple organizations with zero setup required. The main challenge here was the amount of work required for implementation. Creating a polished and styled web application proved to be much more work than creating an equivalent one time script.

Different File Formats

One of the challenges presented in the implementation of TweetID was the fact that there were multiple heterogeneous input formats (see Figure 8 and Figure 9). The way I addressed this was to define a mapping between each format and the backend representation (which in turn maps to the standards I defined previously). Currently the extra metadata fields that were unexpected are being thrown away, but the design of the twitter-level standard allows for them to be stored under the organization's namespace.

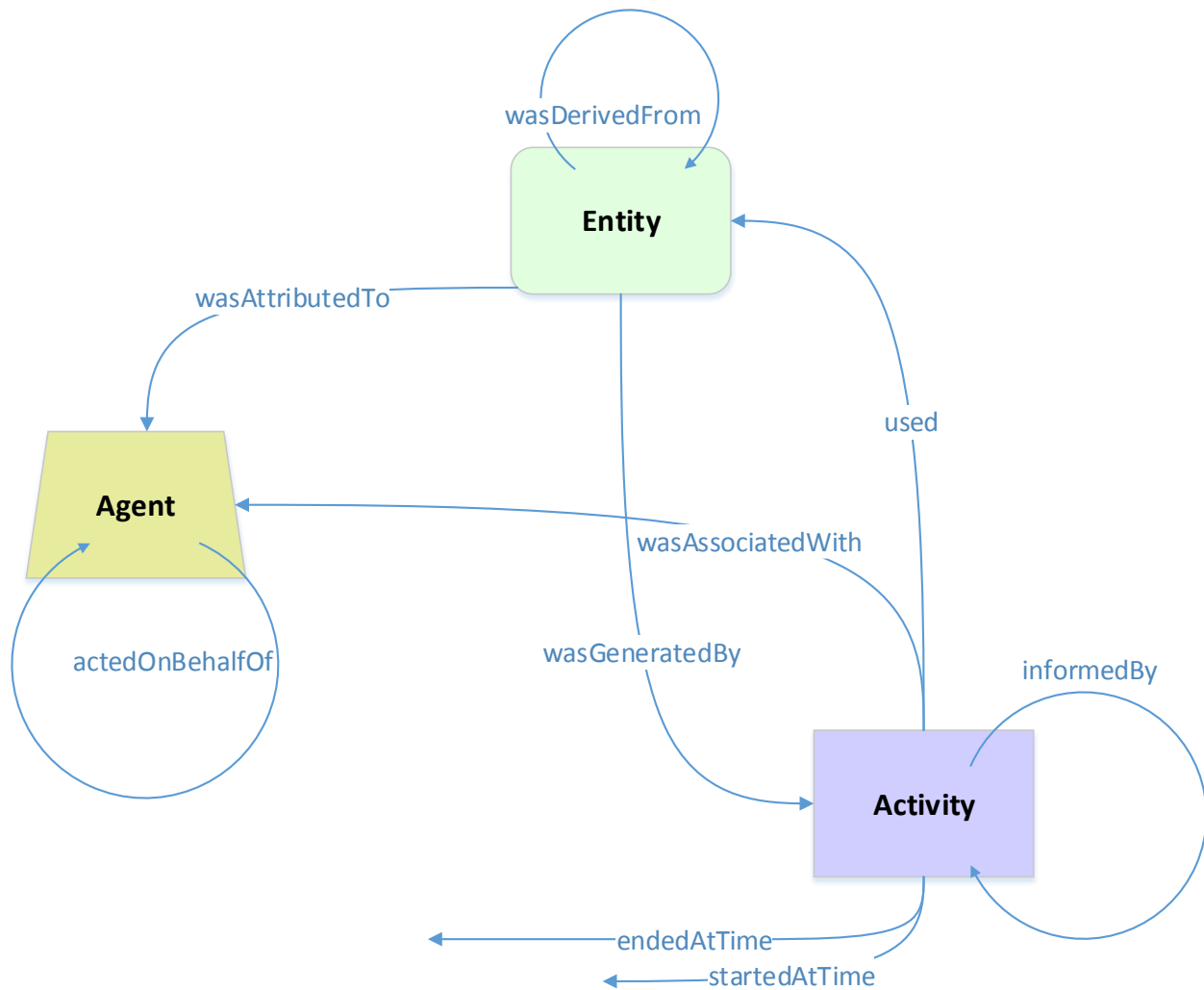


FIGURE 6: PROV-O STARTING POINT CLASSES

Collection Merging

Each field in the collection-level metadata needs to be merged when two collections are merged. However, each field might have unique aggregation logic. For example, the set of organizations needs to be unioned, the set of keywords needs to be unioned, the minimum start date should be taken, and the maximum end date needs to be taken. This is currently how it works; however, I am defining these aggregate operations in a more or less, ad-hoc manner. One of the visions I have for TweetID, should it undergo further development, is to allow for the definition of new metadata fields, including input mappings, persistence mappings, and aggregation operations in a declarative manner—through the use of meta-programming.

```

@prefix xsd:    <http://www.w3.org/2001/XMLSchema#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix prov:   <http://www.w3.org/ns/prov#> .
@prefix tid:    <http://www.tweetid.org/tweet/> .
@prefix :       <http://example.com/> .

:vt-oklahoma-tornado
  a prov:Collection;
  dcterms:title "Oklahoma Tornado"
  dcterms:description "Tweets about Oklahoma Tornado";

  prov:hadMember
    tid:295450465339064321, tid:3954504653346069327, tid:5829504653346069829, ...;

  prov:wasAttributedTo :mark;
  prov:wasAttributedTo :vt;

  prov:atLocation "US OK" # ISO 3166-2

  prov:wasGeneratedBy :keyword-query;
.

:mark a prov:Person, prov:Agent, prov:Entity .

:vt a prov:Organization, prov:Agent, prov:Entity .

:keyword-query
  a prov:Activity;
  prov:startedAtTime "2013-06-07T16:28:17Z"^^xsd:dateTime;
  prov:endedAtTime "2013-06-07T16:28:17Z"^^xsd:dateTime;
  prov:used :keyword-list;
.

:keyword-list
  a prov:Collection;
  prov:hadMember
    "oklahoma tornado", "oklahoma storm", "#okc flood" ...;
.

```

FIGURE 7: EXAMPLE COLLECTION LEVEL METADATA IN TURTLE SYNTAX

Acknowledgements

Project Client – Mohamed Magdy (mmagdy@vt.edu)

Class Instructor/Project Advisor Dr. Edward Fox (fox@vt.edu)

NSF IIS - 1319578: Integrated Digital Event Archiving and Library (IDEAL)

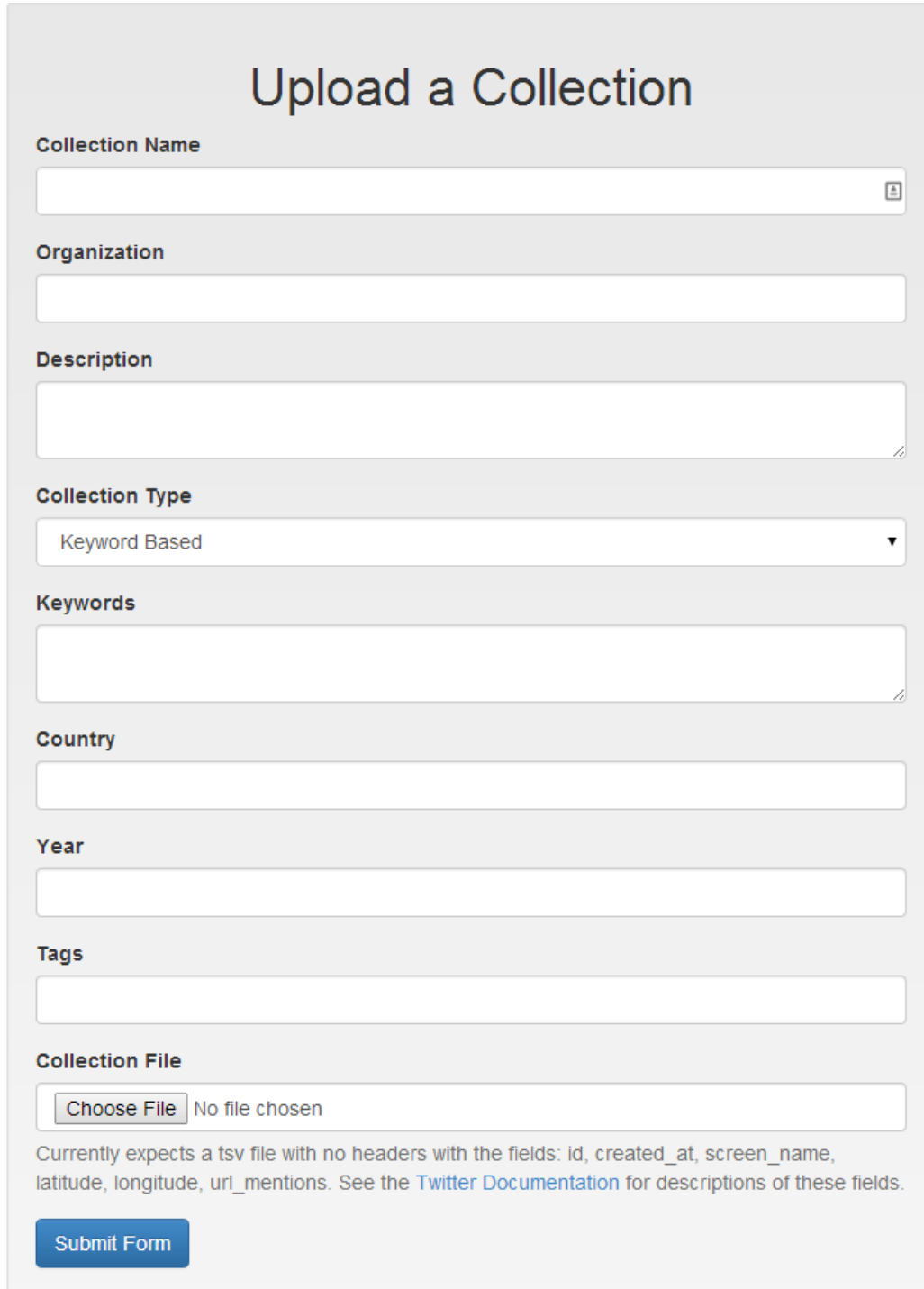
Thanks to Carlos Castillo and Muhammad Imran for providing reference metadata.

References

- Belhajjame, K., Cheney, J., Corsar, D., Garijo, D., Soiland-Reyes, S., Zednik, S., & Zhao, J. (2013, April 30). *PROV-O: The PROV Ontology*. Retrieved from World Wide Web Consortium: <http://www.w3.org/TR/2013/REC-prov-o-20130430/>
- Biron, P. V., & Malhotra, A. (2004, October 28). *XML Schema Part 2: Datatypes Second Edition*. Retrieved from World Wide Web Consortium: <http://www.w3.org/TR/xmlschema-2/#dateTime>
- DCMI Usage Board. (2012, June 6). *DCMI Metadata Terms*. Retrieved from Dublin Core Metadata Initiative: <http://dublincore.org/documents/2012/06/14/dcmi-terms/>
- Groth, P., & Moreau, L. (2013, April 30). *PROV-Overview*. Retrieved from World Wide Web Consortium: <http://www.w3.org/TR/2013/NOTE-prov-overview-20130430/>
- International Organization for Standardization. (2012). *ISO 3166 - Country codes*. Retrieved from International Organization for Standardization: http://www.iso.org/iso/home/standards/country_codes.htm#2012_iso3166-2
- Twitter, Inc. (2013, August 13). *Tweets*. Retrieved from Tweets | Twitter Developers: <https://dev.twitter.com/docs/platform-objects/tweets>

Appendix A: Screenshots

See Figure 1 for a screenshot of the homepage.



The screenshot shows a web form titled "Upload a Collection". The form is set against a light gray background and contains several input fields and a submit button. The fields are arranged vertically, each with a label above it. The "Collection Name" field has a small icon on the right. The "Collection Type" field is a dropdown menu. The "Keywords" field has a small icon on the right. The "Collection File" field includes a "Choose File" button and a "No file chosen" status. Below the file field is a paragraph of text explaining the expected TSV file format and providing a link to the Twitter Documentation. At the bottom is a blue "Submit Form" button.

Upload a Collection

Collection Name

Organization

Description

Collection Type

Keyword Based ▼

Keywords

Country

Year

Tags

Collection File

No file chosen

Currently expects a tsv file with no headers with the fields: id, created_at, screen_name, latitude, longitude, url_mentions. See the [Twitter Documentation](#) for descriptions of these fields.

FIGURE 10: UPLOAD SCREEN

295452270915645440	2013-01-27 08:44:45	AussieClaireC	-35.282	149.129	None
295452527695122432	2013-01-27 08:45:46	Switch1197	37.6873	-122.401	http://fb.me/1gmrC466R

View More Tweets

Oklahoma Tornado

Merge Don't Merge

Virginia Tech

VT Oklahoma Tornado

Merge Don't Merge

Collection Type
keyword

Keywords
oklahoma tornado

Year
2013

Country
US

First Tweet Date
2013-06-07T16:28:17

Last Tweet Date
2014-05-01T22:37:08

ID	Created	Screen Name	Latitude	Longitude	URL Mentions
461997779577888769	Thu May 01 22:37:08 +0000 2014	ChristianHollar	0	0	None
461997447955234817	Thu May 01 22:35:49 +0000 2014	KWiebs31	0	0	None
461996378734862336	Thu May 01 22:31:34 +0000 2014	OKReadiness	0	0	None
461996321788796929	Thu May 01 22:31:21 +0000 2014	IsalyWolf	0	0	None
461995840004307000	Thu May 01 22:28:00 +0000 2014	NickPender/MPO	0	0	None

FIGURE 11: COLLECTIONS VIEW

Queensland Floods

6030 total

Organization QCRI Collection Type keyword Keywords #qldflood,#bigwet,queensland flood,australia flood, Missing, #qldfloods, #Bundaberg, queensland, #floods Year 2013 Country Australia First Tweet Date 2013-01-27T08:37:34 Last Tweet Date 2013-02-07T23:55:57						
ID	Created	Screen Name	Latitude	Longitude	URL Mentions	
295450465339064321	2013-01-27 08:37:34	Fachrurrozy__	10000	10000	http://dlvr.it/2rv5PP	
295451140710084608	2013-01-27 08:40:15	JaydenLestrangle	-25.2744	133.775	None	
295451317361602560	2013-01-27 08:40:57	ahmadhafiz2010	10000	10000	None	
295451408361209856	2013-01-27 08:41:19	geosurfnews	10000	10000	http://dlvr.it/2rv7Y0	
295451574178824192	2013-01-27 08:41:59	tiegan_brooks	10000	10000	None	
295451761676783616	2013-01-27 08:42:43	RyhannahCranney	-25.2744	133.775	None	
295451812083953664	2013-01-27 08:42:55	Mellyjaneeee	-25.2744	133.775	None	
295452249256239104	2013-01-27 08:44:40	MarcusForbes	-36.8474	174.766	None	
295452270915645440	2013-01-27 08:44:45	AussieClaireC	-35.282	149.129	None	
295452527695122432	2013-01-27 08:45:46	Switch1197	37.6873	-122.401	http://fb.me/1gmrC466R	
295452848509030400	2013-01-27 08:47:02	amydegan	-37.8143	144.963	None	

FIGURE 12: COLLECTION DETAILS VIEW

Oklahoma Tornado and VT Oklahoma Tornado Merged

2822582 total

Organizations

QCRI, Virginia Tech

Collection Type

keyword

Keywords

#okc volunteer, #moore, #okhaves, #okc, pottawatomie, moore relief, #okc relief, moore disaster, shawnee, #okwx, oklahoma volunteer, #ok disaster, #ok relief, #okc disaster, #ok, #ok storm, #ok flood, norman, oklahoma relief, oklahoma storm, tornado, oklahoma disaster, oklahoma tornado, moore storm, moore flood, moore tornado, #okneeds, #okc flood, #okc tornado, #okc storm, mary fallin, #ok tornado, moore volunteer, #ok volunteer

Year

2013

Country

US

First Tweet Date

2013-05-20T23:13:40

Last Tweet Date

2014-05-01T22:37:08

ID	Created	Screen Name	Latitude	Longitude	URL Mentions
336620770472112128	2013-05-20 23:13:40	meg_hurrell	38.8381	-104.821	None
336620771017359360	2013-05-20 23:13:40	vthompson2010	38.8816	-77.091	None
336620771042553857	2013-05-20 23:13:40	angelafraser87	10000	10000	None
336620771126415361	2013-05-20 23:13:40	Mktg4theFuture	10000	10000	http://bit.ly/16GvMrA
336620771243872256	2013-05-20 23:13:40	galduric	42.2774	-83.7333	None
336620771449380864	2013-05-20 23:13:40	SassySmartSgRHO	10000	10000	None
336620771629727744	2013-05-20 23:13:40	SarahMarae95	10000	10000	None
336620771738796032	2013-05-20 23:13:40	Cooooolin11	35.4866	-96.6859	None
336620771843661824	2013-05-20 23:13:40	kellicolleen	36.1333	-95.9756	None
336620772145627137	2013-05-20 23:13:40	rose_hisayo	10000	10000	None

FIGURE 13: MERGED COLLECTION DETAILS VIEW

Tweets

2828612 total

ID	Created	Screen Name	Latitude	Longitude	URL Mentions
295450465339064321	2013-01-27 08:37:34	Fachrurrozy__	10000	10000	http://divr.it/2rv5PP
295451140710084608	2013-01-27 08:40:15	JaydenLestrange	-25.2744	133.775	None
295451317361602560	2013-01-27 08:40:57	ahmadhafiz2010	10000	10000	None
295451408361209856	2013-01-27 08:41:19	geosurfnews	10000	10000	http://divr.it/2rv7Y0
295451574178824192	2013-01-27 08:41:59	tiegan_brooks	10000	10000	None
295451761676783616	2013-01-27 08:42:43	RyhannahCranney	-25.2744	133.775	None
295451812083953664	2013-01-27 08:42:55	Mellyjaneee	-25.2744	133.775	None
295452249256239104	2013-01-27 08:44:40	MarcusForbes	-36.8474	174.766	None
295452270915645440	2013-01-27 08:44:45	AussieClaireC	-35.282	149.129	None
295452527695122432	2013-01-27 08:45:46	Switch1197	37.6873	-122.401	http://fb.me/1gmrC466R
295452848509030400	2013-01-27 08:47:02	amydegnan	-37.8143	144.963	None
295455474051055616	2013-01-27 08:57:28	Joxrox	-21.9113	-49.3653	None
295455486273269760	2013-01-27 08:57:31	WildWindTornado	10000	10000	None
295455881561247744	2013-01-27 08:59:06	CoastCath	-26.6754	153.05	None
295455909851832320	2013-01-27 08:59:12	clintost	37.6873	-122.401	None
295456814986829825	2013-01-27 09:02:48	jacross	10000	10000	None
295456835916398593	2013-01-27 09:02:53	Moyes_88	-31.9554	115.859	None
295456908238798850	2013-01-27 09:03:10	grant_mitchell	-37.8143	144.963	None
295457107845722112	2013-01-27 09:03:58	jessica_howe	-19.2581	146.818	None

FIGURE 14: TWEETS SCREEN

Tweet ID 295450465339064321

ID
295450465339064321

Created
2013-01-27 08:37:34

Screen Name
Fachrurrozy__

Latitude
10000

Longitude
10000

URL Mentions
<http://dlvr.it/2rv5PP>

Collections
[Queensland Floods](#)

JSON

```
{
  "twitter": {
    "entities": {
      "urls": [
        {
          "expanded_url": "http://dlvr.it/2rv5PP"
        }
      ]
    },
    "created_at": "2013-01-27 08:37:34",
    "id_str": "295450465339064321",
    "coordinates": {
      "type": "Point",
      "coordinates": [
        10000.0,
        10000.0
      ]
    },
    "screen_name": "Fachrurrozy__"
  },
  "tweetid": {
    "collections": [
      "Queensland Floods"
    ]
  }
}
```

FIGURE 15: TWEET DETAIL SCREEN