CS 580N r/IAmA Comment and Twitter Stream Crawler Report

Cheng Cui
ccui7@binghamton.edu
Binghamton University
Binghamton, NY

Tianze Liu
tliu76@binghamton.edu
Binghamton University
Binghamton, NY

Mingjie Yan
myan28@binghamton.edu
Binghamton University
Binghamton, NY

ABSTRACT

In this project, a crawler is designed to continuously scrape data from Twitter stream and r/IAmA comments in real time. The system is implemented using Python and MongoDb database and is deployed in the given virtual machine.

1 INTRODUCTION

MongoDB[1] is the nosql database system that we used to store the data we collected. The data of each data source is stored in their respective database. We applied for the authorization of using official API from the reddit and Twitter and used http request methods to collect data from the response of the endpoint url. For each tweet, we will collect the text, the time stamp, the hashtags, urls of media attachments and public metrics including count of likes and retweets. For the hashtags, we stored them as hashtag and tweet id pair in an independent collection so that we can handle them more conveniently in the future. We expect to analyse the most popular topics and and sentiments of tweets and IAMA and compare the results of two data sources.

2 DATASET

2.1 preliminary exploration of the data

We firstly learn the tweet data dictionary in the website [2] and dig into these json structures to check which parts we needed. for example, in tweet json structure, id text author_id created_at features and so on are needed. also, considering the fact many users use images/videos to more intuitively and conveniently express their feeling, we therefore decide to extract media objects. to get the information of posted media, in our request, attachments.media_keys are required according to the introduction of expansion[3]. for subreddit IAmA, we collected the thread_title user_id flag author and score time stamp and so on. we take reddit official API as a reference[5].

2.2 data collection

As of the afternoon, Nov.8th, the twitter collection system [1] has been deployed for over half a day.and the reddit collection [2] system has been deployed for several hours. the size of collected data and the updated projection on how much data is we are likely to collect is as followed Table 1 and Table 2.

Table 1: Twitter stream Database

Table Name	Collected Data	Projection
hashtag_collection	0.71M	10M
twitter_collection	2.30M	100M

Table 2: Reddit r/IAmA Database

Table Name	Collected Data	Projection
comments	10.8K	75K
CrossPosts	23	40
IAmA	64	240

3 ARCHITECTURE

3.1 Twitter Stream Data Collection

The process to collect the twitter stream data is that a request with authorized token is made to the twitter stream data endpoint and a response with twitter is returned. If the tweet has hashtags, the hashtags and tweet id will be stored in hashtag collection. Then the tweet with all other needed fields of data will be stored in twitter collection. We separate the storage to make the handling of tweets data with hashtags more convenient in our future analysis.

3.2 Reddit r/IAmA Comments Data Collection

When collecting Reddit r/IAmA comments data, we designed a centralized “job issuer”, as detailed in Appendix Figure 1. First establishes a link to database. Here we use MongoDB[1] database, python can use pymongo[4] library to facilitate database operation. After that, we call the “login” module (see Figure 2 in the attachment) to set up the Reddit api[5] Oauth2 head with the username and password and other necessary web api information.

The Oauth2[6] head will be one of the necessary data for us to use the Reddit api. We also need the subreddit prefix and postid, since all the comments we get are from r/IAmA so we just need the postid to start crawling comments.

Then we call the “subreddit” module, see Figure 3 in the attachment. This module will capture the newest 25 posts in r/IAmA. If there are crossposts among them, we will process them separately later. If there are no crossposts, we will update the "IAmA" table in the database directly. The postid of the post will be saved and ready to be passed to the next module. The reason for crosspost is that the discussion itself does not take place in the r/IAmA, the post in r/IAmA is just for redirection. The only data we have are the AMA introduction and a link to a Reddit post. So we use regular expression to extract the subreddit prefix and postid from the link.

Now we have the Oauth2 head, subreddit prefix name and postid, the conditions for capturing a particular comment are complete. Calling "comments" module, see Appendix Figure 4. The "comments" module takes a list from "subreddit" module, if the list is NOT from a crosspost it means the post is from r/IAmA itself. Create an http request to get the comment details via the Reddit api. However, sometimes the number of comments can be very large

(say 5,000) that Reddit cannnot return all the comments, only the first 100, and the subsequent comments only the come with comment_id. So we need a special function to retrieve all the comments. However, crosspost is quite common (about 20 out of 100 posts in r/IAmA), so we built a separate table in the database for crosspost to distinguish it. The "CrossPosts" table has the same list of attributes as the "IAmA" table. Since the number of comments on a crosspost is often large, we hope this separate design can improve the search performance of the database. When the "CrossPosts" table is updated, the comments that need to be retrieved will be passed to the comments module to retrieve the content.

After updating all the posts in the list the crawler goes to sleep, here we use a finite state machine to achieve dynamic hibernation. If there are a lot of posts to be updated in a short period of time, the hibernation interval will be shortened, and if there are no posts to be updated in a long period of time, the hibernation interval will be lengthened. The maximum hibernation period is set to 8 hours, after 8 hours the hibernation time will return to the minimum value of 5 minutes. Once the hibernation is over, restart the crawling process, get the Ouath2 head, check for new posts, get comments, and hibernate again.

At this point, the crawler works as described above. Ideally it would go on in an infinite loop, but in reality there are various problems that can interrupt the process. If any intermediate process error occurs (e.g. disk I/O error, network error, etc.) the process is forced to end and we need another process daemon to revive the crawler. We wrote some scripts using the shell to implement this process daemon, checking the status of the process every 30 seconds using "ps", reawakening the process if it can't be found, and collecting logs of the process as it runs.

4 CHALLENGES

(1) when we decided to collect the tweet media object, expansion is required with original object. we look into this and revise original request and then get the media keys and corresponding media urls.
(2) At first we didn't notice that some twitter users will type the same hashtag for several times in one tweet, so there a few repeated data. To handle this we further made an judgement to eliminate duplicate hashtags and save them more efficiently.
(3) Since the information we capture comes from user comments in r/IAmA, the number of new comments will not be large in a short period of time. To reduce unnecessary network requests and disk I/O, we designed a finite state machine to dynamically set the crawler's dormancy time. If there are a large number of posts published in a short period of time, the update frequency will increase as well.
(4) When a post has a large number of comments, the JSON returned from Reddit can not contain all the replies at once, after 100 comments, there will be only comment_id, so we need to make a special function for the content after 100 replies. Every request only get one comment. At the same time, due to Reddit API limitation, we can't have too many requests in a short period of time, so the update of some posts can't even be completed within the valid period of one https token.
(5) Since the data we collect contains posts that cross subreddit (crosspost), and thus post_id cannot be used as the primary key to update each post. So we use the subreddit prefix along with post_id as the primary key. While this ensured uniqueness, it also makes update slower. In particular, the comments database uses the subreddit prefix, post_id, and comment_id these three attributes as primary keys, making updates very slow when encountering posts with a large number of replies.

REFERENCES

[1] https://docs.mongodb.com/manual/
[2] https://developer.twitter.com/en/docs/twitter-api/data-dictionary/object-model
[3] https://developer.twitter.com/en/docs/twitter-api/expansions
[4] https://api.mongodb.com/python/current/tutorial.html
[5] https://www.reddit.com/dev/api/
[6] https://oauth.net/2/

5 ATTACHMENT

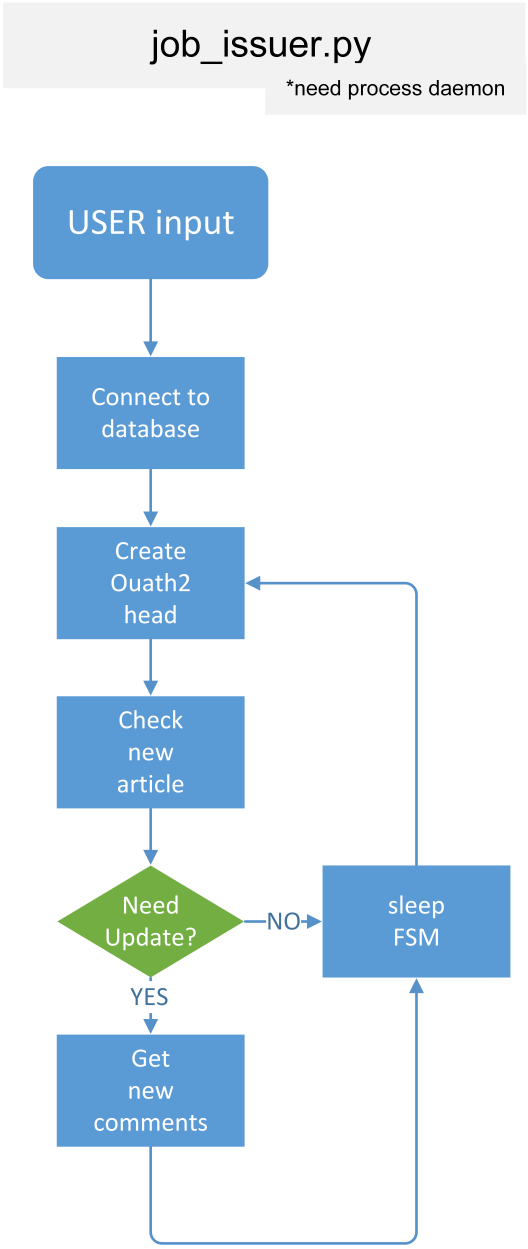

Figure 1: Flowchart of job issuer

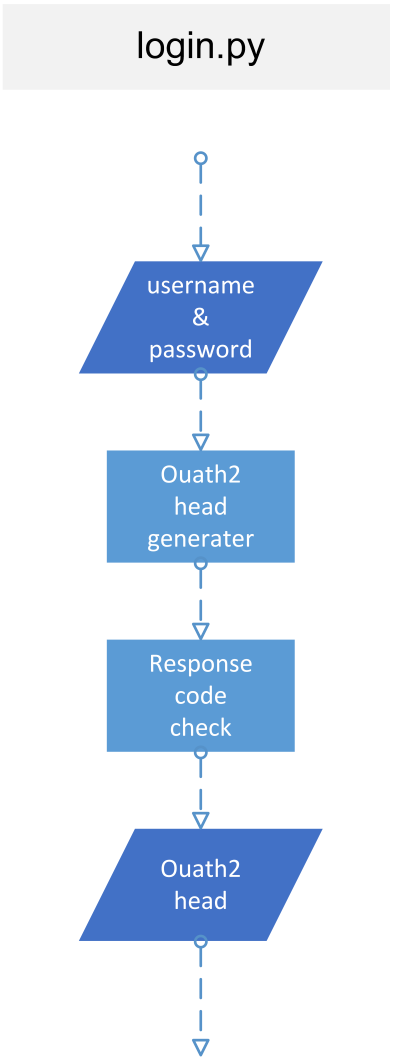

Figure 2: Flowchart of logining for reddit API

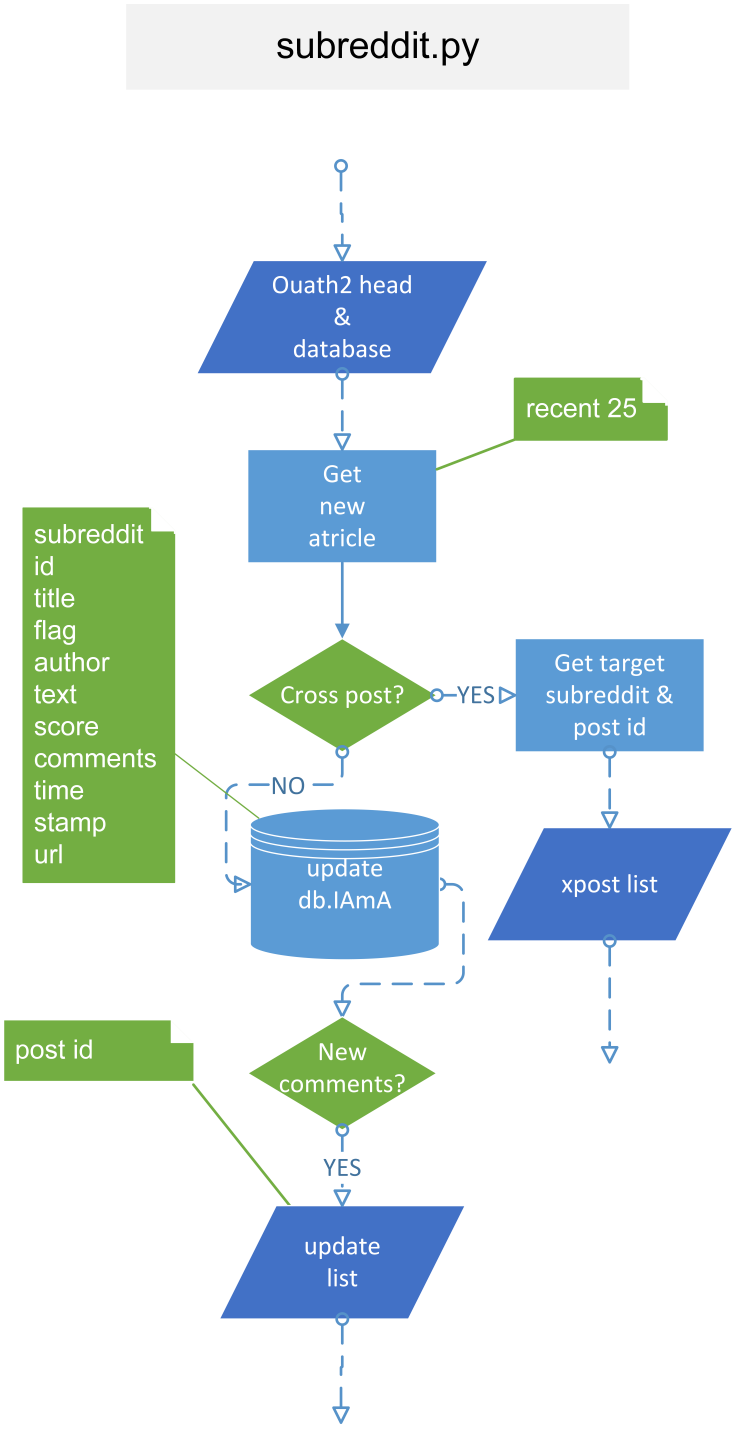

Figure 3: Flowchart of collecting subreddit Data

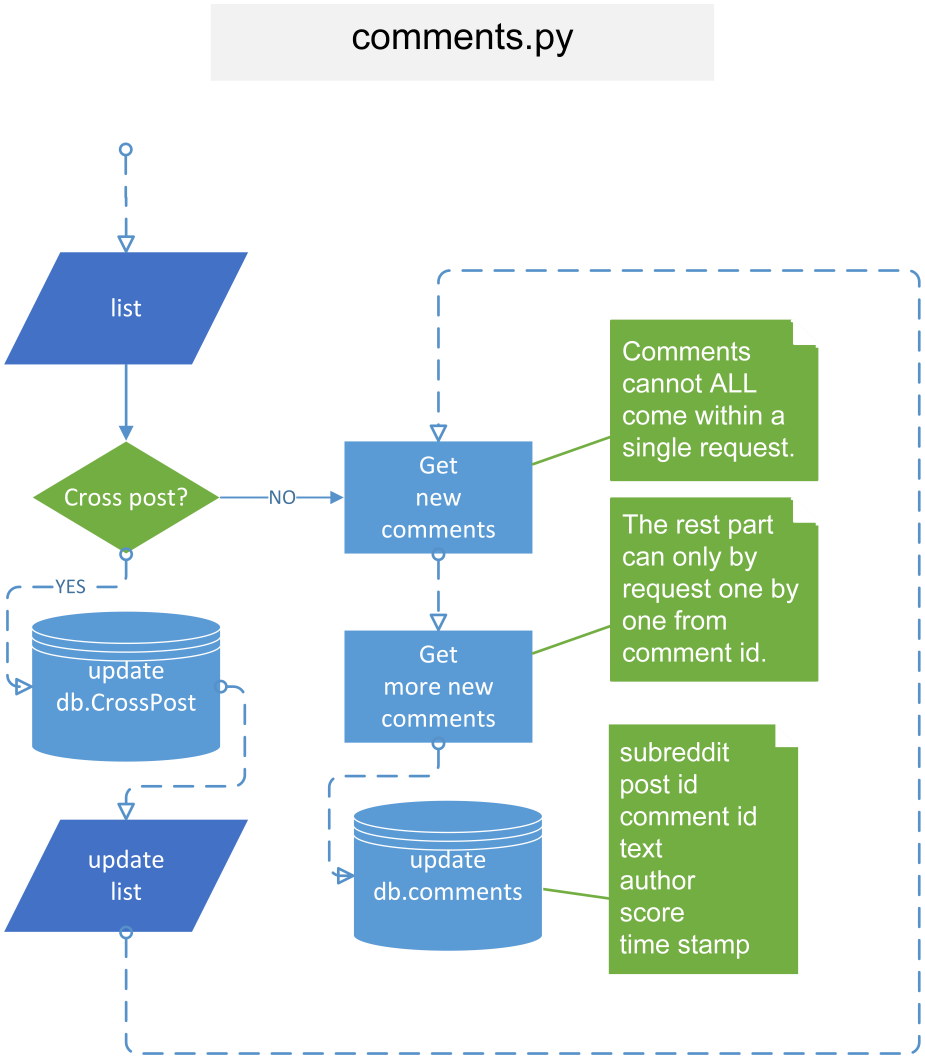

Figure 4: Flowchart of collecting r/IAmA Comments Data