CS 5604: Information Storage and Retrieval TWT - Tweet Collection Management

Akash Patil, Hitesh Baadkar, Ikjot Juneja, Irith Sharma, Manisha Kusuma, Megan Hicks, Pranav Chimote, Ujjval Mehta

Instructor: Dr. Edward A. Fox
SME: Xinyue Wang
Department of Computer Science, Virginia Tech
Blacksburg, VA 24061
December 2, 2020

Agenda

1. Overview
2. Requirements
3. Design and Implementation
4. Deliverables
5. Future work

Overview

Project overview:

- Milestone 1. Tweet collections (Ingestion)
- Milestone 2. Content of tweets cleaned and metadata extracted
- Milestone 3. Tweet categorization via TwiRole
- Milestone 4. Extracted/derived information from Tweets
- Milestone 5. Store tweets in a more compact and accessible format for big data systems

Overview (Cont'd)

For each information goal:

- Draft workflow (a sequence of steps/information states to reach the information goal)
- Identify services (block of operation/code that will have its own section in the pipeline)
- Develop service
- Register service with the Integration team

Information Goals

Goal ID	Input	Information Goal
1	Raw collection of tweets	ELS field on geolocation
2	Raw collection of tweets	ELS field on hashtags
3	Raw collection of tweets	ELS field on originating username
4	Raw collection of tweets	ELS field on user mentions
5	Raw collection of tweets	Filename for user download
6	Local file of tweets	Uploaded collection of tweets
7	Raw collection of tweets	ELS field on keywords
8	Raw collection of tweets	ELS field on TwiRole
9	Raw collection of tweets	ELS field on timestamp

Preprocessing

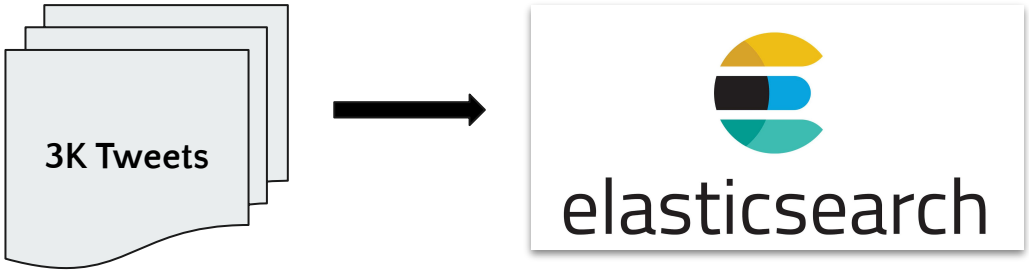
- 3 formats for tweets (yourTwapperKeeper (YTK) , Social Feed Manager (SFM), DMI-TCAT)
- Collections stored in WARC files


```
{
    "archivesource": "twitter-search",
    "text": "RT @thereaIbanksy: Never forget. \n\n#WalterScott http://t.co/EdEiU8ZwLk",
    "to_user_id": "",
    "from_user": "Jamal_Chatha",
   "id": "586220033648406528",
    "from_user_id": "2601261336",
    "iso_language_code": "en",
    "source": "<a href=\"http://twitter.com/download/android\" rel=\"nofollow\">Twitter for Android</a>",
    "profile_image_url": "http://abs.twimg.com/images/themes/theme1/bg.png",
    "geo_type": "",
    "geo_coordinates_0": 0,
    "geo_coordinates_1": 0,
    "created_at": "Thu Apr 09 17:32:02 +0000 2015",
    "time": 1428600722
}
```

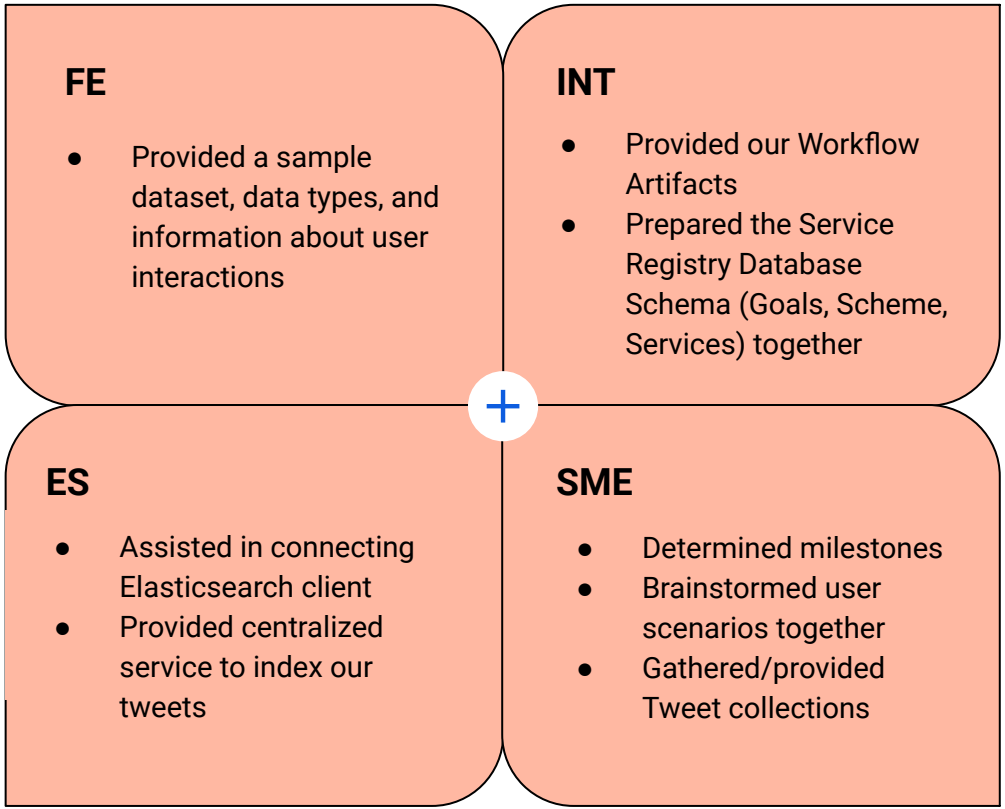

Example tweet in YTK format

Indexing

- Extract key fields, such as hashtags, username, mentions, geo-location, keywords, and timestamp
- Indexed these tweets into Elasticsearch
- Modularized our services to extract a single field



Collaboration



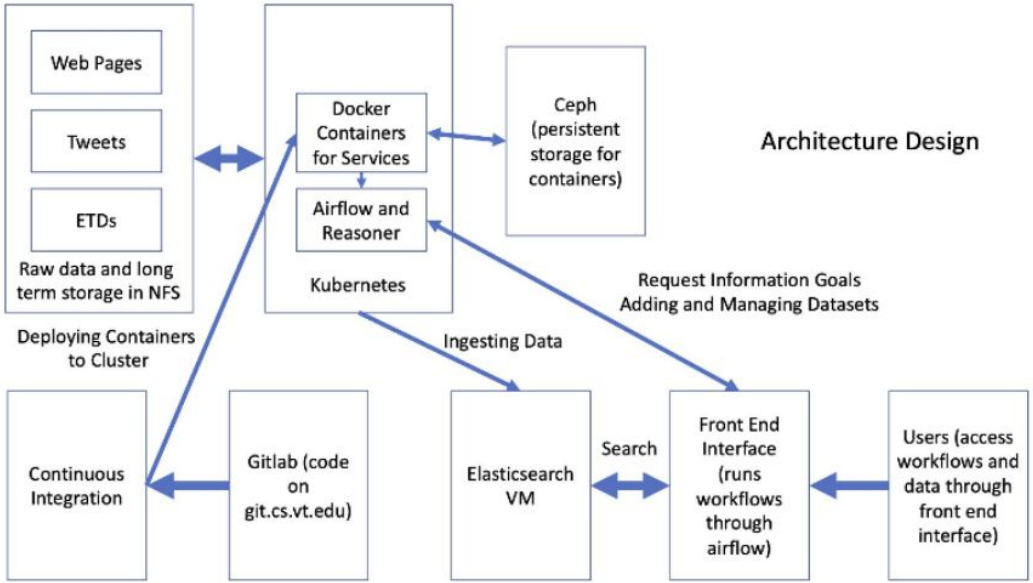
Design & Implementation

Tools

- Python: for framework
- Docker: to run services
- TwiRole: to classify user
- Gensim (Text Summarization module): to find keywords within a tweet's textual content.
- Elasticsearch: for processing and querying collections of tweets, based on the different columns of the index created for our collections.
- Apache Parquet: to backup large collections of tweets (complement existing collections in JSON format).
- YTK and SFM: to collect data
- WARCIO: to read data from .warc format (and store in .json format)

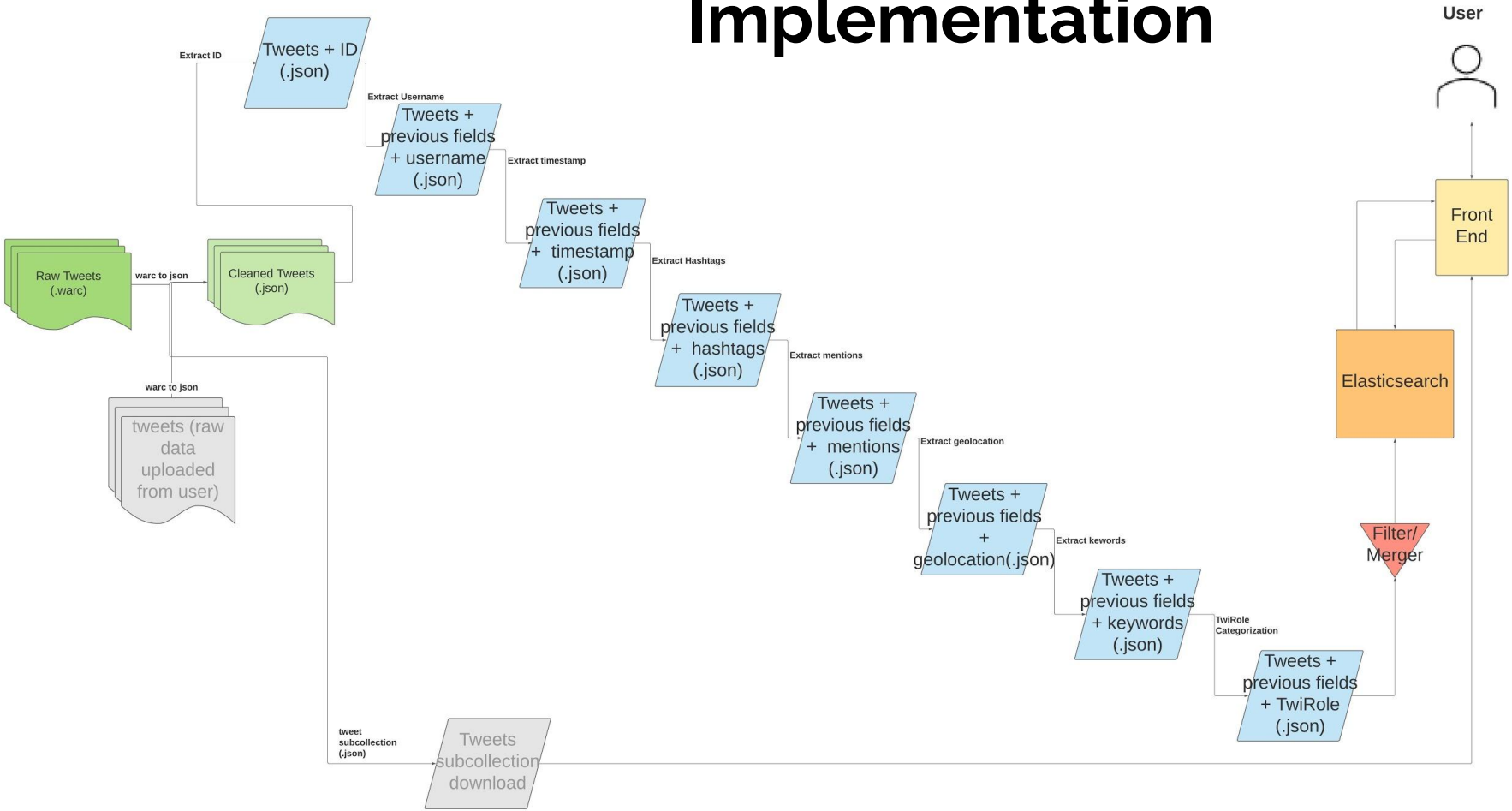
Workflow

- Workflow-based system solution
- 4 other teams: ETD, WP, FE, INT



Overview of Fall 2020 system and team interactions

Implementation



Workflow diagram, depicting states of data and services to move between states

ID	Service Name	Input(s)	Output
1	WARC to JSON converter	Name of raw tweets .warc, backup location for collection	Name of .json containing cleaned tweets, name of .parquet containing cleaned tweets
2	Add ID field	Name of .json containing cleaned data	Name of .json containing tweets with ID field added
3	Add username field	Name of .json containing cleaned data	Name of .json containing tweets with username field added
4	Add timestamp field	Name of .json containing cleaned data	Name of .json containing tweets with timestamp field added
5	Add hashtags field	Name of .json containing cleaned data	Name of .json containing tweets with hashtags field added
6	Add username mentions field	Name of .json containing cleaned data	Name of .json containing tweets with username mentions field added
7	Add geo-location field	Name of .json containing cleaned data	Name of .json containing tweets with geo-location field added
8	Add keywords field	Name of .json containing cleaned data	Name of .json containing tweets with keywords field added
9	Generate list of unique users	Name of .json containing cleaned data	Name of .json containing unique users
10	Add TwiRole classification field	Name of .json containing cleaned data, name of .json containing unique users and previous categorizations	Name of .json containing tweets with TwiRole classification field added, name of .json containing updated unique users and their categorizations
11	Filter/merge fields	Name of .json containing cleaned data + fields of interest	Name of .json containing only fields of interest
12	Generalized indexing using Elasticsearch	Name of .json containing tweets with fields to be indexed	Name of .json containing indexing results

Services to support tasks

Services

Services:

WARC to JSON:

- Original/raw tweet collections are in WARC format
- All the tweets are compiled into one WARC file along with the metadata
- Utilize WARCIO streaming library to search for tweet data in the WARC records
- The JSON file can now be processed with our other services

Services (Cont'd)

Username:

- Look for *users* field containing username data and add data to new *username* column

Hashtags:

- Look for hashtags data already extracted and add data to new *hashtags* column
- If not true, extract hashtags using regular expression operations from the tweet's text.

Mentions:

- Look for *mentions* data already extracted and add them to new *mentions* column
- If not true, extract mentions using regular expression operations from the tweet's text

Services (Cont'd)

Twirole

- TwiRole for role-related user classification on Twitter by Liuqing Li, Ziqian Song, Xuan Zhang and Edward A. Fox
- Crawl a user's profile, profile image and recent tweets, and classify a Twitter user into *Brand*, *Female* or *Male*

```
Task    1: CNN              =>  •[30mBrand  •[0m•[30m[Brand: 100.00%,
Female: 0.00%, Male: 0.00%]•[0m
Classifier_1:   [Brand: 80.00%, Female: 20.00%, Male: 0.00%]
Classifier_2:   [Brand: 100.00%, Female: 0.00%, Male: 0.00%]
Classifier_3:   [Brand: 99.99%, Female: 0.00%, Male: 0.01%]
Task    2: ArianaGrande     =>  •[31mFemale •[0m•[31m[Brand: 0.10%, Female:
99.85%, Male: 0.05%]•[0m
Classifier_1:   [Brand: 50.00%, Female: 50.00%, Male: 0.00%]
Classifier_2:   [Brand: 20.00%, Female: 70.00%, Male: 10.00%]
Classifier_3:   [Brand: 0.00%, Female: 99.99%, Male: 0.01%]
```

Sample TwiRole categorization

Services (Cont'd)

ID:

- Extract unique ID as string for index versioning.

Timestamp:

- Check various fields for tweet time and extract for time range searches.

Keywords:

- Use the gensim keywords library to determine relevant keywords from corresponding tweets' texts.

Services (Cont'd)

Geolocation

- Geolocation service on Twitter is used to track breaking news and events. Some researchers/users are also interested in knowing what topics are trending in certain locations.
- For each tweet, look for "coordinates" field which will have location information, and add them to new column 'geolocation' and write it file back to .json.

Testing

End-to-end test

Containers used in manual end-to-end test

Unit Testing

- Use of pytest framework
- Unit tests for each service (running in CI/CD)
- Test cases:
  - Null input/non-ASCII
  - Non-existing file
  - Invalid format (not YTK or SFM JSON etc.)
  - Correct output otherwise

Deliverables

- Services registered and working
  - warc-to-json conversion, ID, username, timestamp, hashtags, mentions, geolocation, keywords, filter, index
- Dockerized services
- End-to-end test through CS cluster and Airflow/reasoner

Future Work

- Parallel workflow based system (parallel configuration)
  - Run a specific service
- Work on DMI-TCAT
- Extracting implicit Geolocation information from tweets

Acknowledgements

- Thank you to the guidance and support from:
  - Dr. Edward Fox
  - Prashant Chandrasekar
  - Xinyue Wang
  - NSF (through CMMI-1638207)

Q&A