Final Report: Webpages (WP) Team

Jostein Barry-Straume, Cristian Vives, Wentao Fan, Peng Tan, Shuaicheng Zhang, Yang Hu, Tishauna Wilson

CS 5604: Information Storage & Retrieval
Instructed by Professor Edward Fox

Fall 2020
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061

VIRGINIA TECH™

FINAL STRETCH TIMELINE

	SUBTEAM 1	SUBTEAM 2	SUBTEAM 3	SUBTEAM 4	ALL
WEEK 8					MEET W/ INT TEAM RE: DOCKER CONTAINERS
WEEK 9				PROTOTYPE ELASTICSEARCH INDEX RESULTS	GET LARGER SAMPLE DATA
WEEK 10	VALIDATE SERVICE W/ LARGER DATA SUBSET				DEPLOY DOCKER CONTAINERS
WEEK 11	TEXT SUMMARIZATION	TEXT CLASSIFICATION	TEXT SUMMARIZATION	MVP PIPELINE	UNIT TESTING
WEEK 12				RANCHER	
WEEK 13	CONTAINER TESTING, EVALUATION, AND INTEGRATION IN CS CLOUD KUBERNETES CLUSTER				
WEEK 14					

Final Stretch Milestones

10/25	IR3: Live demo of data generation and ingestion pipelines
10/30	IR3: Expand on report to include relevant progress to date
11/20	Test and evaluate data pipelines and services
12/02	Final Project Presentation: Live demo a fully tested and deployed software for data extraction and ingestion of webpages
12/09	Final Project Report: Deliver a fully tested and deployed software for data extraction and ingestion of webpages

Service Table

service_id	service_name	service_description
1	extract_url	Extracts URLs from Twitter tweets
2	archive_webpage	Archives web pages based on their URLs
3	extract_webpage	Extracts web page content from archived web pages
4	index_data	Indexes web pages to the given endpoint with the following fields: URL, title, webpage_content, webpage_summary
5	summarize_text	Provides summarization of web page textual content
6	classify_text	A COVID-19 relevance classifier for a collection of text data

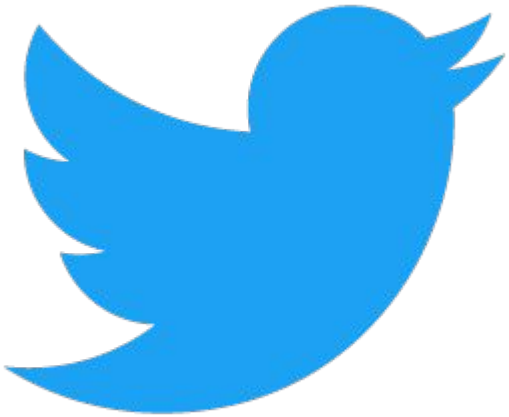
Goal Table

goal_id	goal_name	goal_description	goal_format	envrionment_variable
1	tweet_file	RAW DATA	<Text file>	TWEET
2	extract_url	EXTRACTED DATA	<Text file>	URL
3	archive_webpage	STORED DATA	<warc.gz file>	WARC
4	extract_webpage	PARSED DATA	<JSONL file>	JSONL
5	index_data	SORTED DATA	<Elasticsearch index>	JSONL
6	summarize_text	FILTERED DATA	<JSONL file>	WEBPAGE
7	classify_text	CLASSIFIED DATA	<Text file>	WEBPAGE

Reasoner Table

goal_id	service_id	input_goal_id
2	1	1
3	2	2
4	3	3
5	4	4
7	5	6
9	6	8

Input Data



- Input data in form of tweets
- Tweets represented as JSONL files
- Extract URLs mentioned in tweets
- Not all tweets contain URLs!

Elasticsearch Index Data Structure


```
1  {"index" : {}}
2  {"URL": "https://www.reuters.com/article/us-health-coronavirus-usa-race-idUSKBN21Q08O?taid=5e8d4eb89a7fcd0001c4c2e7&utm_campaign=trueAnthem:+Trending+Content&utm_medium=trueAnthem&utm_source=twitter",
3  "title": "AfricanAmericans dying of coronavirus at higher rates preliminary data shows  Reuters",
4  "collection_type": "coronavirus",
5  "Rouge_Score": {"rouge-1": {"f": 0.5053449913643314, "p": 1.0, "r": 0.3381014304291287}, "rouge-2": {"f": 0.5024342708143609, "p": 0.9961389961389961, "r": 0.3359375}, "rouge-l": {"f": 0.5530434742597203, "p": 1.0, "r": 0.3822115384615384}},
```


- One JSONL file per collection

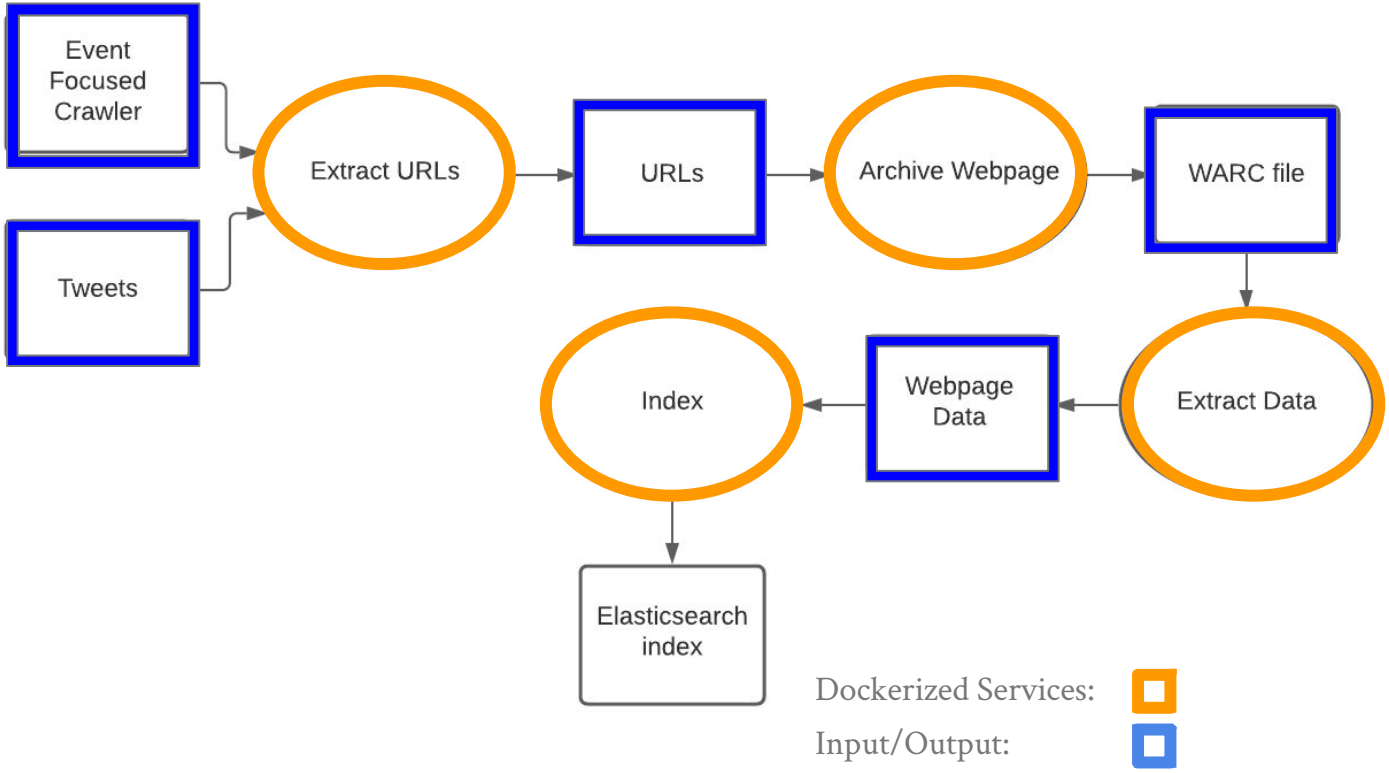
Elasticsearch Index Data Structure

"webpage_text": "African-Americans dying of coronavirus at higher rates, preliminary data shows | ReutersDiscover Thomson ReutersDirectory of sitesLoginContactSupportfor-phone-onlyfor-tablet-portrait-upfor-tablet-landscape-upfor-desktop-upfor-wide-desktop-upCoronavirus: Full CoverageUpdated African-Americans dying of coronavirus at higher rates, preliminary data showsBy Brad Brooks, Michael Martina, Catherine Koppel5 Min Read(Reuters) - The new coronavirus is killing African-Americans at a higher rate than the U.S. population at large, according to preliminary numbers from Louisiana, Michigan and Illinois that officials say point to disparities in health and healthcare access.The figures were reported by state and city leaders at briefings on the coronavirus, including Louisiana Governor John Edwards who said more than 70% of the 512 people killed by the coronavirus in Louisiana as of Monday were black, a much larger percentage than the state\u2019s population that black people represent, about 33 percent.Michigan officials also said that the coronavirus took a disproportionate toll on African-Americans with 40% of the reported deaths in the state, whose population is 14% African-American. As of Tuesday, confirmed cases in Michigan were 18,970 with 845 deaths.The data is

Elasticsearch Index Data Structure

"webpage_summary": "African-Americans dying of coronavirus at higher rates, preliminary data shows | ReutersDiscover Thomson ReutersDirectory of sitesLoginContactSupportfor-phone-onlyfor-tablet-portrait-upfor-tablet-landscape-upfor-desktop-upfor-wide-desktop-upCoronavirus: Full CoverageUpdated African-Americans dying of coronavirus at higher rates, preliminary data showsBy Brad Brooks, Michael Martina, Catherine Koppel5 Min Read(Reuters) - The new coronavirus is killing African-Americans at a higher rate than the U.S. population at large, according to preliminary numbers from Louisiana, Michigan and Illinois that officials say point to disparities in health and healthcare access.The figures were reported by state and city leaders at briefings on the coronavirus, including Louisiana Governor John Edwards who said more than 70% of the 512 people killed by the coronavirus in Louisiana as of Monday were black, a much larger percentage than the state\u2019s population that black people represent, about 33 percent.Michigan officials also said that the coronavirus took a disproportionate toll on African-Americans with 40% of the reported deaths in the state, whose population is 14% African-American.\nHowever, community leaders and public health officials said it could reflect both higher levels of underlying illnesses that make African-Americans more vulnerable as well as possibly lower levels of access to healthcare.U.S. Surgeon General Jerome Adams, acknowledging the early data, said on Tuesday that black Americans were more likely to have heart disease, diabetes and high blood pressure.Diabetes, heart disease and long-term lung problems are the most common underlying conditions among Americans hospitalized with COVID-19, the respiratory illness caused by the new coronavirus, the U.S. Centers for Disease Control and Prevention (CDC) said in a report here published on March 31.",

Data Ingestion Pipeline



Dockerized Service I/O

```
RUN pip install -r requirements.txt

ENV TWEET=/mnt/nfs1/wp/data/coronavirus0408_100.jsonl
ENV URL=/mnt/nfs1/wp/data/urls.txt
#ENV VERBOSE=true
#VOLUME /code/data
# copy the content of the local src directory to the working directory
COPY . .
#RUN mkdir data
CMD ["python","extractURL.py"]
```

extractURL Dockerfile

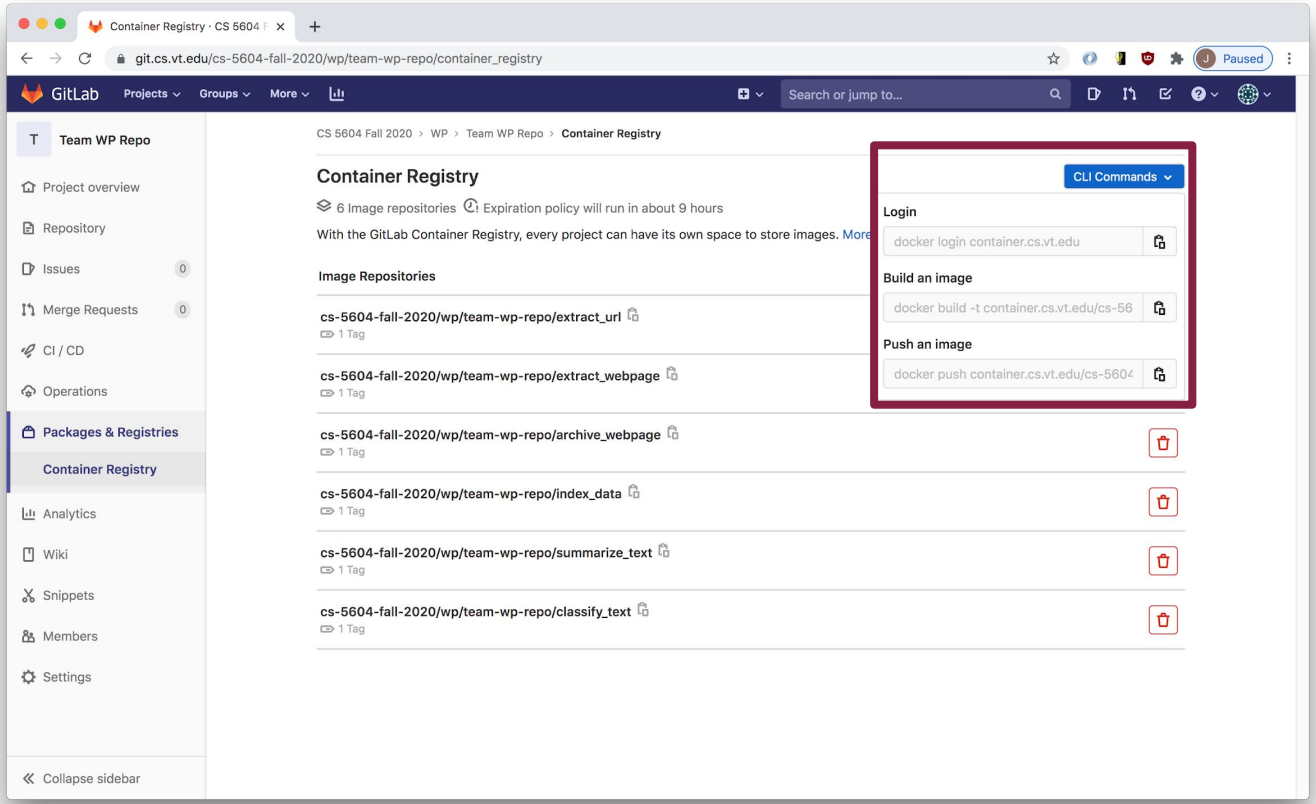
```
docker run

--env TWEET_INPUT_FILE=data/coronavirus0408_100.jsonl
--env URL_OUTPUT_FILE=data/urls.txt
--env VERBOSE=true

-v /"$(pwd)"/data:/code/data extract_url
```

Running extract_url docker container

- Use of environment variables
- Can override environment variables when running container
- Use of volumes to provide I/O for services

Gitlab Container Registry

https://git.cs.vt.edu/cs-5604-fall-2020/wp/team-wp-repo/container_registry

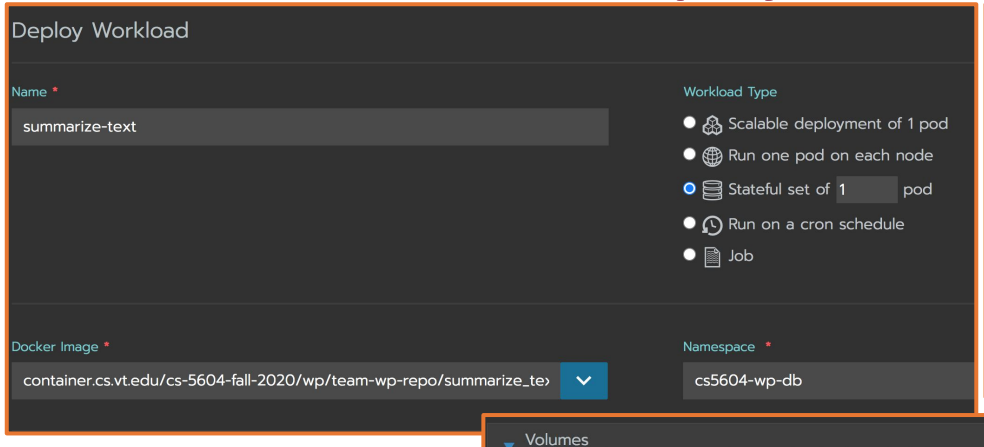


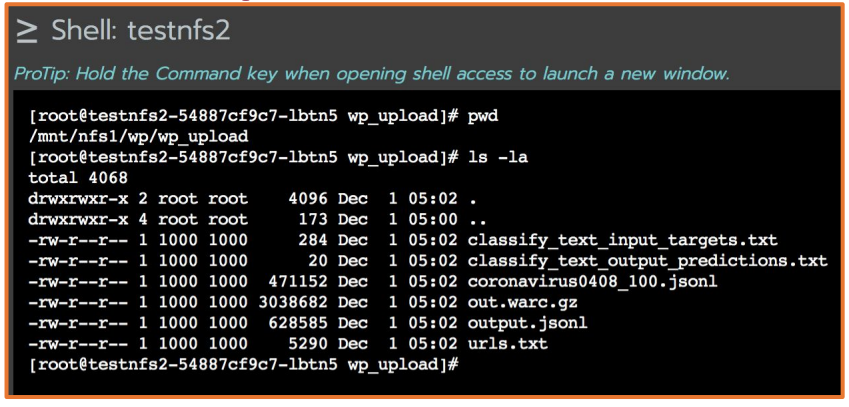
http://cloud.cs.vt.edu/

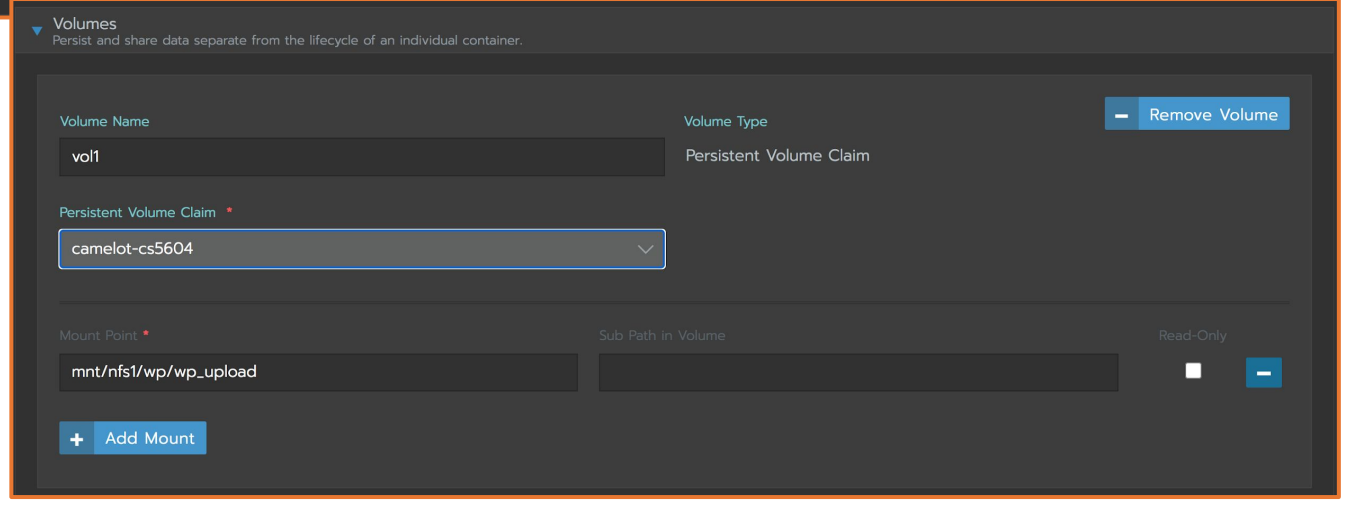
Deployed Services Status

State	Name	Image	Scale
Namespace: cs5604-wp-db			
Updating	archive-webpage ReplicaSet "archive-webpage-868496fc74" has timed out progressing.; D...	container.cs.vt.edu/cs-5604-fall-2020/wp/team-wp-repo/archive_webpa... 1 Pod / Created 2 days ago / Pod Restarts: 0	1
Active	centos	centos:latest 1 Pod / Created 2 months ago / Pod Restarts: 0	1
Active	classify-text	container.cs.vt.edu/cs-5604-fall-2020/wp/team-wp-repo/classify_text 1 Pod / Created 3 minutes ago / Pod Restarts: 0	1
Updating	extract-url ReplicaSet "extract-url-6c8c5f7c99" has timed out progressing.; Deploym...	container.cs.vt.edu/cs-5604-fall-2020/wp/team-wp-repo/extract_url 1 Pod / Created 19 hours ago / Pod Restarts: 0	1
Updating	extract-webpage ReplicaSet "extract-webpage-55c9f78d66" has timed out progressing.; D...	container.cs.vt.edu/cs-5604-fall-2020/wp/team-wp-repo/extract_webpag... 2 Pods / Created 16 days ago / Pod Restarts: 0	1
Updating	index-data ReplicaSet "index-data-7ddc545c69" has timed out progressing.; Deploy...	container.cs.vt.edu/cs-5604-fall-2020/wp/team-wp-repo/index_data:latest 2 Pods / Created 16 days ago / Pod Restarts: 0	1
Active	summarize-text	container.cs.vt.edu/cs-5604-fall-2020/wp/team-wp-repo/summarize_text 1 Pod / Created 4 minutes ago / Pod Restarts: 0	1
Active	testnfs	centos:latest 1 Pod / Created 21 days ago / Pod Restarts: 0	1
Active	testnfs2	centos:latest 1 Pod / Created 21 days ago / Pod Restarts: 0	1

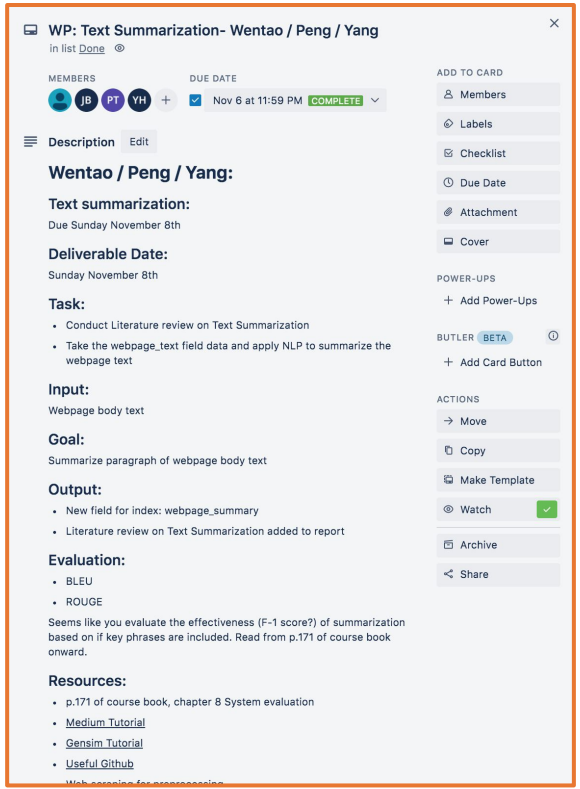
Deployed Services Setup



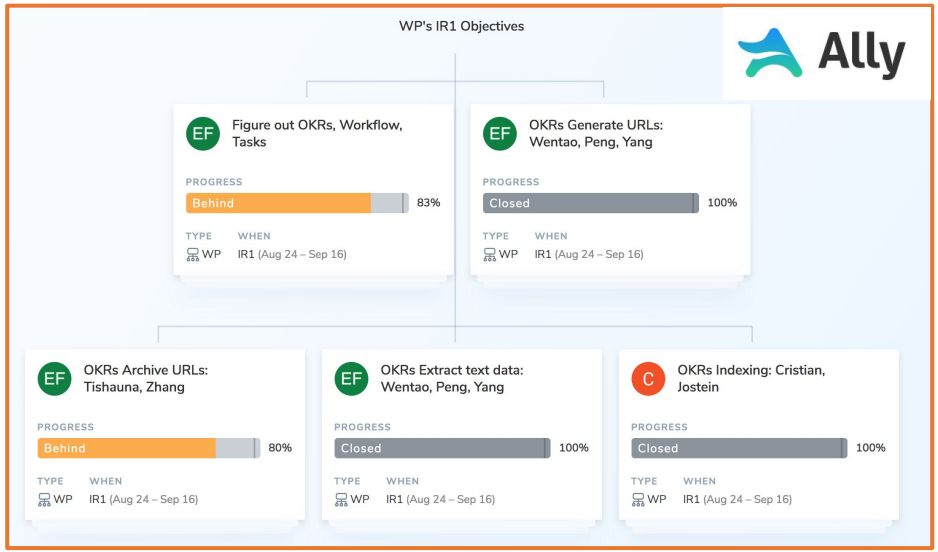




DevOps Learning Lessons



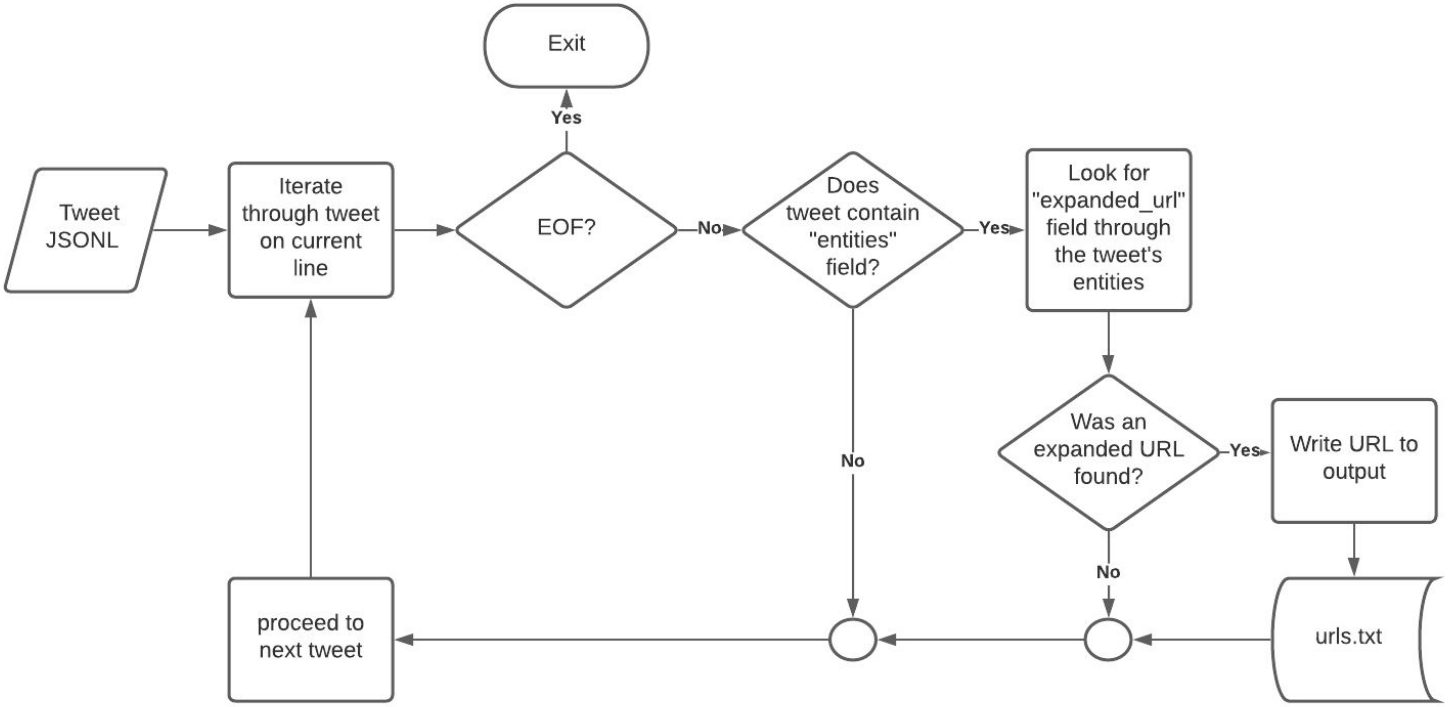
- Found greater success as a team operating with Trello
- Ally is too unfamiliar and low adoption rate



Extract URL Service:

Peng Tan, Wentao Fan, Yang Hu

Extract URL Flow Graph



```
def extract_urls(json_list,output_name,verbose):
    logging.basicConfig(filename="extractURLs.log", level=logging.INFO)
    url_list = []
    for line in json_list:
        if line.strip():
            try:
                tweet = json.loads(line)
            except Exception as e:
                # garbage in, garbage out
                logging.error(e)
                #return line
                continue
            #checks if provided tweet links directly to the webpage
            if "entities" in tweet:
                found_url = False
                for url_dict in tweet["entities"]["urls"]:
                    if'expanded_url' in url_dict:
                        url = url_dict['expanded_url']
                        if not re.match(r'^https?://twitter.com/', url):
                            if verbose:
                                try:
                                    uprint("{} : TYPE : {}".format(url,"URL found in tweet text"))
                                except UnicodeEncodeError:
                                    uprint(u"{} : TYPE : {}".format(url,"URL found in tweet text"))
                                print("--------------------")
                            url_list.append(url)
                            found_url = True
```

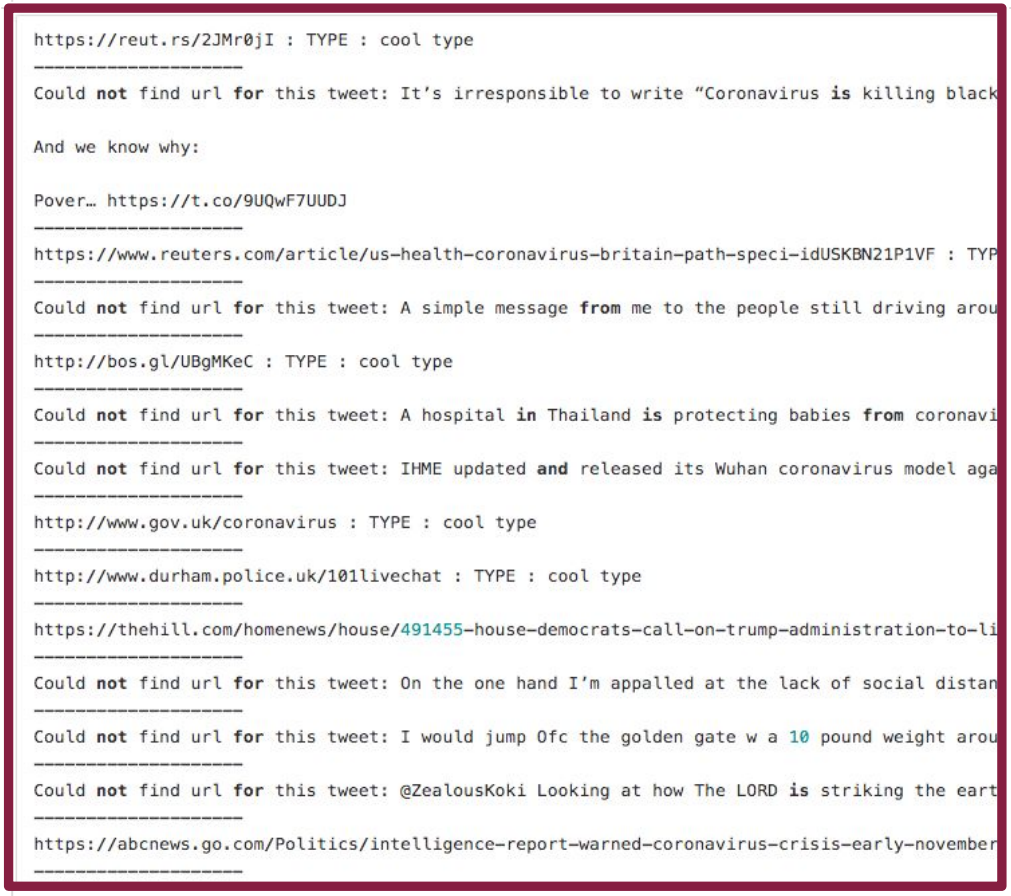
- Using two packages: JSON and RE
- JSON used to parse the JSON file
- RE used to match the specified string

Extract URL

```
        #checks that the quoted tweet is quoting an extended tweet
        if not found_url and "extended_tweet" in tweet["quoted_status"]:
            #case where tweet is quoting an extended tweet
            for url_dict in tweet["quoted_status"]["extended_tweet"]["entities"]["urls"]:
                if'expanded_url' in url_dict:
                    url = url_dict['expanded_url']
                    if not re.match(r'^https?://twitter.com/', url):
                        if verbose:
                            try:
                                uprint("{} : TYPE : {}".format(url,"quoted an extended tweet"))
                            except UnicodeEncodeError:
                                uprint(u"{} : TYPE : {}".format(url,"quoted an extended tweet"))
                            print("--------------------")
                        url_list.append(url)
                        found_url = True
    if not found_url:
        if verbose:
            try:
                uprint("Could not find url for this tweet: {}".format(tweet["text"]))
            except UnicodeEncodeError:
                uprint(u"Could not find url for this tweet: {}".format(tweet["text"]))
            print("--------------------")
#now, write to file
with open(output_name,'w') as file:
    for line in url_list:
        file.write("{}\n".format(line))
```

- Store these URLs obtained from previous data source into JSON file for the next group

Extract URL



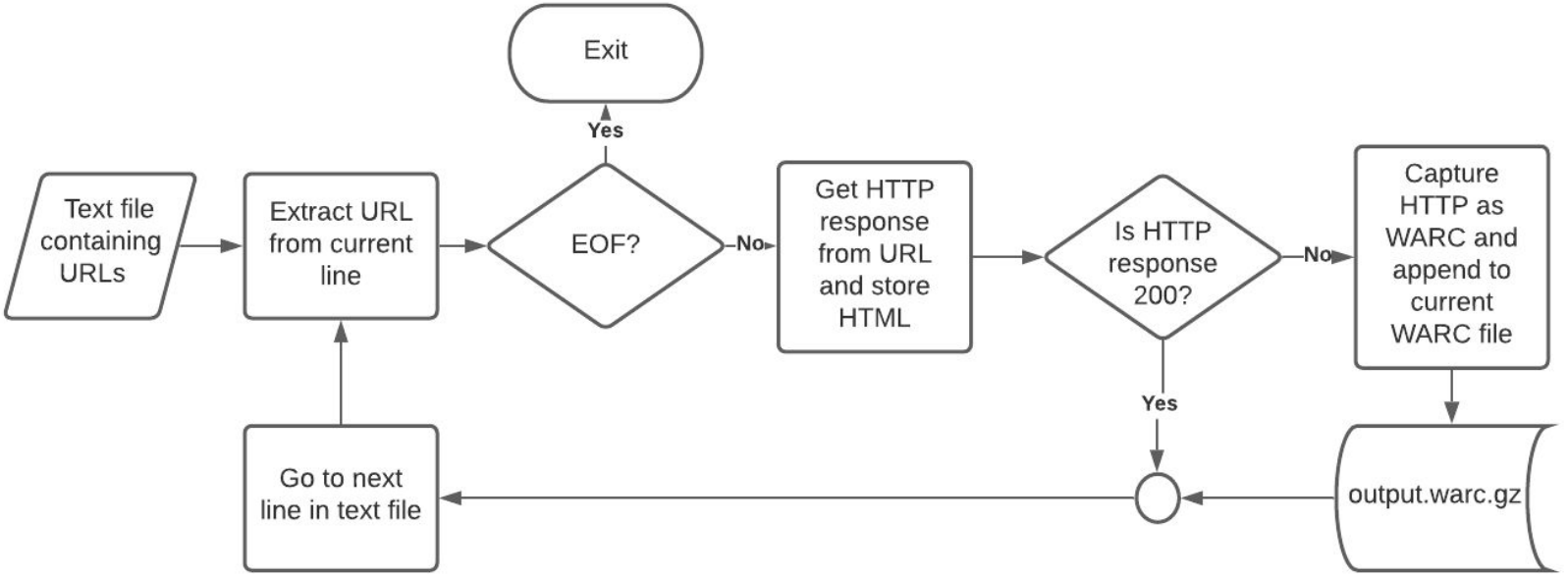
- Input: JSON file containing text with embedded URLs
- Grab URLs
- Put URLs in list
- Store them in JSON file

Extract URL

Archive Webpage Service:

Shuaicheng Zhang, Tishauna Wilson

Archive Webpage Flow Graph



```
from warcio.capture_http import capture_http
import requests  # requests must be imported after capture_http
import sys

# TODO: Point urlsFile to extracted URLs files
# urlsFile = sys.argv[1]

# TODO: remove urls first assignment
headers = {
    'User-Agent': 'Sleeper Agent Alpha Zero Charlie',
    'From': 'screwyou@tryingtoblockmyrequest.com'
}
urls = []
with open("urls.txt",'r') as file:
    urls = file.readlines()
urls = [u.strip() for u in urls]
# Determine if a particular request and response records should be written to the WARC file or skipped
def filter_records(request, response, request_recorder):
    # return None, None to indicate records should be skipped
    if response.http_headers.get_statuscode() != '200':
        return None, None

    return request, response
# TODO: Change the name of the warc file
# To read warc file from command line - warcio index test.warc.gz
with capture_http('covid.warc.gz', filter_records):
    for url in urls:
        print(url)
        response = requests.get(url,headers=headers)
        statusCode = response.status_code
        if(statusCode != 200):
            pass
```

- We take URL lists from a JSON file from the previous team as an input
- We iterate through actual COVID-19 webpages
- We check the status when accessing each of the webpages; if the status is 200 then we accept the webpage content, otherwise ignore it
- We archive all the webpages into a WARC file for indexing using WARCIO.

https://reut.rs/2JMr0jI

https://www.reuters.com/article/us-health-coronavirus-britain-path-speci-idUSKBN21P1VF

http://bos.gl/UBgMKeC

http://www.gov.uk/coronavirus

http://www.durham.police.uk/101livechat

https://thehill.com/homenews/house/491455-house-democrats-call-on-trump-administration-to-lift-restrictions-on-fetal

https://abcnews.go.com/Politics/intelligence-report-warned-coronavirus-crisis-early-november-sources/story?id=70031273

https://www.theguardian.com/science/2020/apr/07/cancer-research-uk-to-cut-funding-for-research-by-44m?CMP=share_

https://www.defensenews.com/coronavirus/2020/04/08/the-pentagons-supply-chain-faces-an-economy-under-siege/

https://cbsloc.al/2RoEGG5

https://www.latimes.com/homeless-housing/story/2020-04-07/la-fi-home-buying-coronavirus

https://www.nytimes.com/2020/04/06/upshot/coronavirus-four-benchmarks-reopening.html

https://english.elpais.com/society/2020-04-08/spain-to-test-30000-families-for-the-coronavirus.html

https://www.cnn.com/2020/04/08/politics/donald-trump-coronavirus/index.html?fbclid=IwAR089v_4M4lsFtLjz8bU1quHp79

https://bit.ly/3bSVhcS

https://www.queerty.com/fox-news-officially-sued-peddling-coronavirus-misinformation-20200406

https://thedcpatriot.com/just-in-democrats-want-illegal-immigrants-to-receive-coronavirus-stimulus-checks/

https://www.reuters.com/article/us-health-coronavirus-britain-path-speci-idUSKBN21P1VF

https://www.aljazeera.com/news/2020/04/critically-ill-covid-19-uk-patients-bme-backgrounds-200407143303604.html

http://ketv.com/article/nebraska-medicine-weighs-in-on-anti-malaria-drug-used-on-coronavirus-patients/32073113?src=

https://www.thewrap.com/broadway-will-remain-dark-until-june-at-least-as-coronavirus-shutdown-extended/

https://www.pscp.tv/w/cVyWHTI2MTAyMHwxRFh4eWVFTU5abnhNSwMk-ZLIMMgUv0PnO4TB5hFxiklwzml6tGYzcq-erM4=

- Snapshot of web pages being archived

```
import requests
import logging
import contextlib
try:
    from http.client import HTTPConnection # py3
except ImportError:
    from httplib import HTTPConnection # py2

def debug_requests_on():
    '''Switches on logging of the requests module.'''
    HTTPConnection.debuglevel = 1

    logging.basicConfig()
    logging.getLogger().setLevel(logging.DEBUG)
    requests_log = logging.getLogger("requests.packages.urllib3")
    requests_log.setLevel(logging.DEBUG)
    requests_log.propagate = True

def debug_requests_off():
    '''Switches off logging of the requests module, might be some side-effects'''
    HTTPConnection.debuglevel = 0

    root_logger = logging.getLogger()
    root_logger.setLevel(logging.WARNING)
    root_logger.handlers = []
    requests_log = logging.getLogger("requests.packages.urllib3")
    requests_log.setLevel(logging.WARNING)
    requests_log.propagate = False

@contextlib.contextmanager
def debug_requests():
    '''Use with 'with'!'''
    debug_requests_on()
    yield
    debug_requests_off()
```

- In order to test out the robustness of our code, we have set up several debug functions to test out the functionality of archiving URL functions

```
headers = {
    'User-Agent': 'Sleeper Agent Alpha Zero Charlie',
    'From': 'screwyou@tryingtoblockmyrequest.com'
}
requests.get('https://www.miamiherald.com/news/local/immigration/article241829036.html',headers=headers)
```

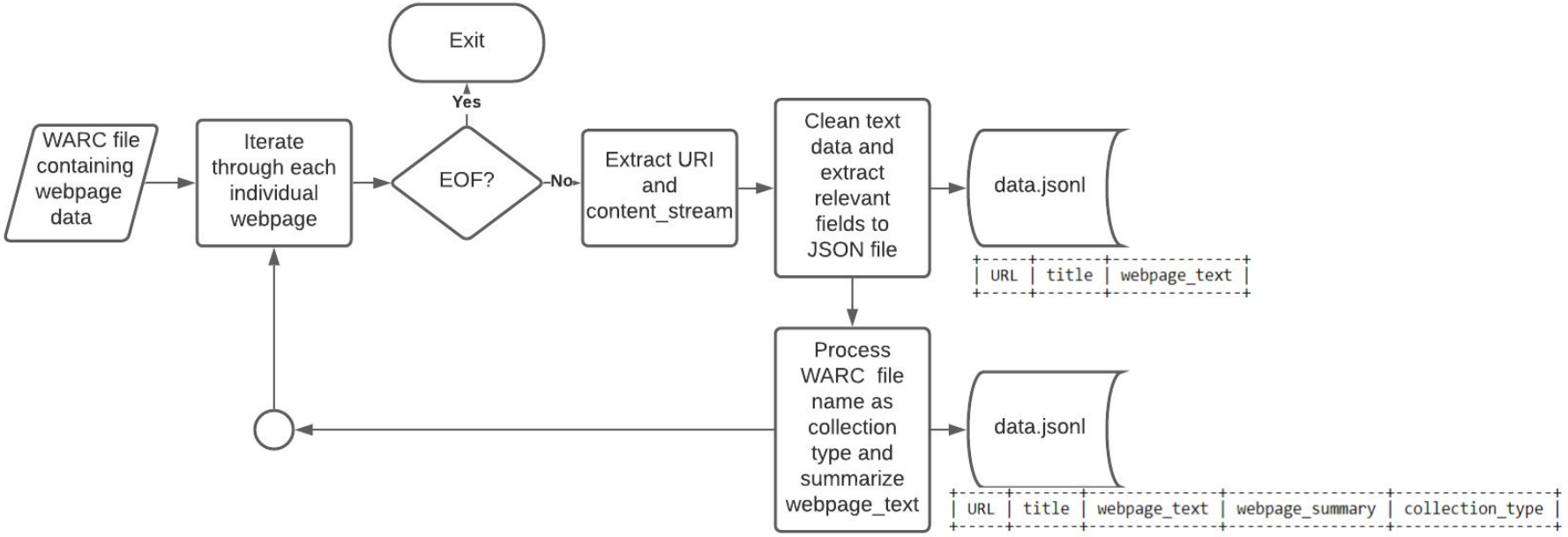
```
DEBUG:urllib3.connectionpool:Starting new HTTPS connection (1): www.miamiherald.com:443

DEBUG:urllib3.connectionpool:https://www.miamiherald.com:443 "GET /news/local/immigration/article241829036.html
HTTP/1.1" 200 33218
send: b'GET /news/local/immigration/article241829036.html HTTP/1.1\r\nHost: www.miamiherald.com\r\nUser-Agent: My
User Agent 1.0\r\nAccept-Encoding: gzip, deflate\r\nAccept: */*\r\nConnection: keep-alive\r\nFrom:
youremail@domain.com\r\n\r\n'

reply: 'HTTP/1.1 200 OK\r\n'

header: Server: MI

header: Content-Type: text/html;charset=utf-8

header: Set-Cookie:
ak_bmsc=51D21C010247C599A94924A909240FBC17C89EA1D1170000E271875F422A477B~plt2CLURtEqGJLchJnhrzsNf
 expires=Wed, 14 Oct 2020 23:47:14 GMT; max-age=7200; path=/; domain=.miamiherald.com; HttpOnly

header: X-Proxy-Forwarding-Type: WhiteList

header: X-Meter: s

header: Access-Control-Allow-Origin: *

header: Access-Control-Allow-Methods: GET,POST,OPTIONS

header: Access-Control-Allow-Headers: *

header: Access-Control-Allow-Credentials: false

header: Access-Control-Max-Age: 86400

header: Vary: Accept-Encoding

header: X-Akamai-Path-Stats: [3:37428:572]

header: MI-Cache: HIT
```

- Result from the robustness test

Extract Data Service:

Peng Tan, Wentao Fan, Yang Hu

Extract Data Flow Graph



```python
def parse_content(url, content, json_list,collection_type):
    '''
    Parses content provided by a URL and appends it to a list
    '''
    soup = BeautifulSoup(content, 'html.parser')
    content = soup.get_text()
    content = re.sub('\n\r+', '\n\r',re.sub('\n+', '\n', content))
    content = remove_html(content)
    title = soup.find('title').string
    title = re.sub(r'[^ \\t\w\.]', '', title)
    summary = summarize(content)
    summary = summary.replace('/\r?\n|\r/g', '').strip()
    if len(content) != 0 and len(summary) != 0:
        rouge_dict = rouge_score(summary, content)
        json_list.append({"URL":url, "title":title,
                          "webpage_text":content, "webpage_summary":summary,
                          "collection_type":collection_type, "Rouge_Score": rouge_dict["Rouge_Score"]})
    elif len(summary) == 0:
        json_list.append({"URL": url, "title": title,
                          "webpage_text": content,
                          "collection_type": collection_type})
```

```python
def remove_html(text):
    soup = BeautifulSoup(text, 'lxml')
    return soup.get_text().strip()

def rouge_score(summary, reference):
    new_dict = {}
    summary = [summary]
    reference = [reference]
    rouge = Rouge()
    rouge_score = rouge.get_scores(summary, reference)
    new_dict['Rouge_Score'] = {"rouge-1":rouge_score[0]["rouge-1"],
                               "rouge-2":rouge_score[0]["rouge-2"],
                               "rouge-l":rouge_score[0]["rouge-l"]}
    return new_dict
```

- Using two packages: BeautifulSoup and Gensim
- BeautifulSoup used to get text from HTML
- Gensim.summarization to summarize text from BeautifulSoup
- Use Rouge_score to evaluate summarization performance

output.jsonl — Edited

{"URL": "https://www.reuters.com/article/us-health-coronavirus-usa-race-idUSKBN21Q08O?taid=5e8d4eb89a7fcd0001c4c2e7&utm_campaign=trueAnthem:+Trending+Content&utm_medium=trueAnthem&utm_source=twitter", "webpage_text": "African-Americans dying of coronavirus at higher rates, preliminary data shows | ReutersDiscover Thomson ReutersDirectory of sitesLoginContactSupportfor-phone-onlyfor-tablet-portrait-upfor-tablet-landscape-upfor-desktop-upfor-wide-desktop-upU.S. NewsUpdated African-Americans dying of coronavirus at higher rates, preliminary data showsBy Brad Brooks, Michael Martina, Catherine Koppel5 Min Read(Reuters) - The new coronavirus is killing African-Americans at a higher rate than the U.S. population at large, according to preliminary numbers from Louisiana, Michigan and Illinois that officials say point to disparities in health and healthcare access.The figures were reported by state and city leaders at briefings on the coronavirus, including Louisiana Governor John Edwards who said more than 70% of the 512 people killed by the coronavirus in Louisiana as of Monday were black, a much larger percentage than the state\u2019s population that black people represent, about 33 percent.Michigan officials also said that the coronavirus took a disproportionate toll on African-Americans with 40% of the reported deaths in the state, whose population is 14% African-American. As of Tuesday, confirmed cases in Michigan were 18,970 with 845 deaths.The data is preliminary and not national and does not explain what is causing the disparities. However, community leaders and public health officials said it could reflect both higher levels of underlying illnesses that make African-Americans more vulnerable as well as possibly lower levels of access to healthcare.U.S. Surgeon General Jerome Adams, acknowledging the early data, said on Tuesday that black Americans were more likely to have heart disease, diabetes and high blood pressure.Diabetes, heart disease and long-term lung problems are the most common underlying conditions among Americans hospitalized with COVID-19, the respiratory illness caused by the new coronavirus, the U.S. Centers for Disease Control and Prevention (CDC) said in a report here published on March 31. One in five people requiring intensive care had no such health issues, it said.QuanTez Pressley, a 33-year-old preacher at the Third New Hope Baptist Church in Detroit, said that issues such as lack of access to fresh foods and people crowded into small living quarters \u201cmake those individuals in marginalized communities more

- After Extracting data from WARC files, we can get URL of HTML
- Then we can get text from HTML

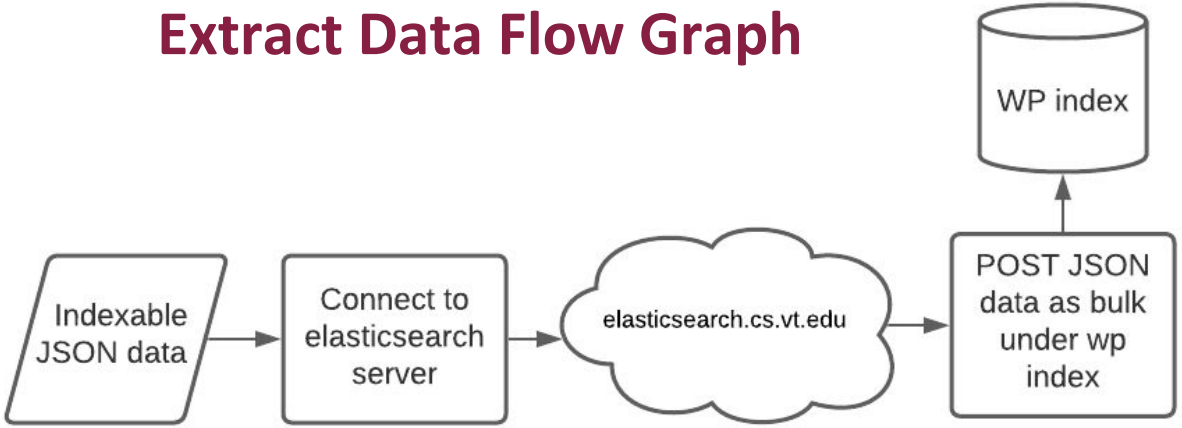
```
class Test_method(unittest.TestCase):

    @classmethod
    def setUpClass(cls):
        print("Before test case============")

    @classmethod
    def tearDownClass(cls):
        print("after test case============")

    def test_generateurls(self):
        str_ = 'https://reut.rs/2JMr0jI'
        Flag = 0

        with open('covid.warc.gz', 'rb') as file:
            extract_from_warc(file)
        with open("output.jsonl",'w') as output_file:

            for line in output_file.readlines():
                print(line)
                if str_ in line:
                    Flag=1
                    break

        self.assertEqual(Flag, 1)
```

- Add unit test for our extract data method
- Set up test method before and after test case

Index Data Service

Cristian Vives, Jostein Barry-Straume

Extract Data Flow Graph

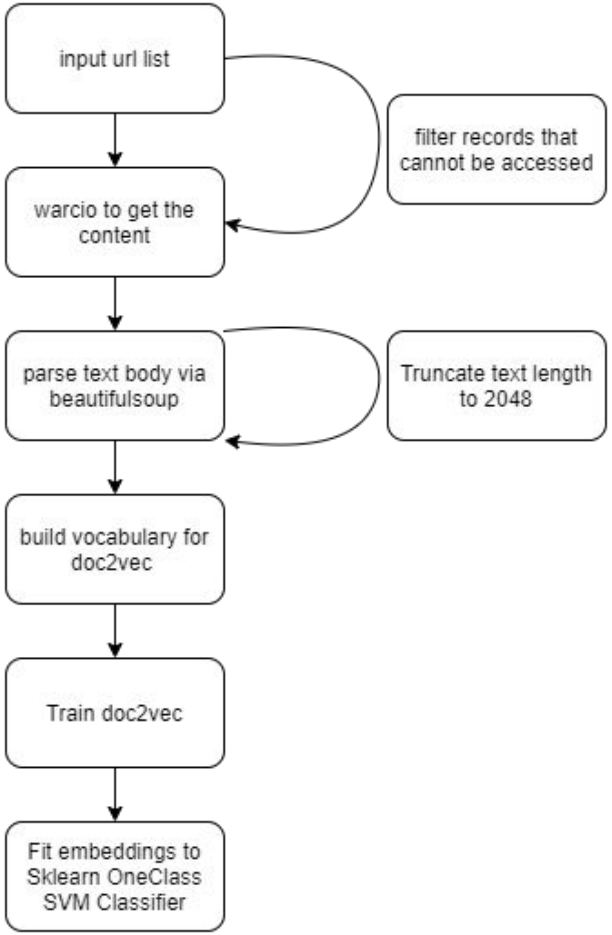


- Similar to most teams
- Simple script
- Doesn't preprocess any data

Classify Text Service:

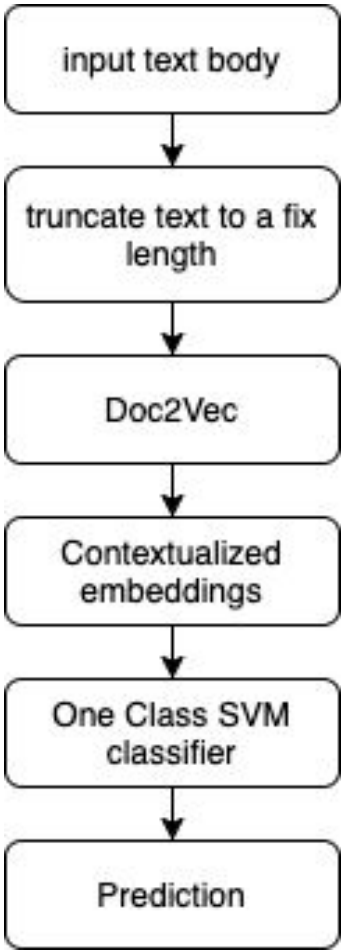
Shuaicheng Zhang

Classify text flow



Two step procedure:

- Training
- Classifying



Code preview

```
from gensim.test.utils import common_texts
from gensim.models.doc2vec import Doc2Vec, TaggedDocument
from nltk.tokenize import word_tokenize
from sklearn.svm import OneClassSVM
from joblib import dump, load
import numpy as np

class CoronaClassifier:

    def __init__(self, load_emb_model=False, load_svm=False):
        self.max_epochs = 100
        self.vec_size = 20
        self.alpha = 0.025
        if load_svm:
            self.clf = load('./svm.joblib')
        else:
            self.clf = OneClassSVM(gamma='auto')
        if load_emb_model:
            self.model = Doc2Vec.load("./d2v.model")
        else:
            self.model = Doc2Vec(size=self.vec_size,
                         alpha=self.alpha,
                         min_alpha=0.00025,
                         min_count=1,
                         dm =1)

    def train_data(self, sents):
        tagged_data = [TaggedDocument(words=word_tokenize(_d.lower()), tags=[str(i)])
                       for i, _d in enumerate(sents)]
        self.model.build_vocab(tagged_data)

        for epoch in range(self.max_epochs):
            print('iteration {0}'.format(epoch))
            self.model.train(tagged_data,
                        total_examples=self.model.corpus_count,
                        epochs=self.model.iter)
            # decrease the learning rate
            self.model.alpha -= 0.0002
            # fix the learning rate, no decay
            self.model.min_alpha = model.alpha
        vecs = list()
        for sent in sents:
            vecs.append(self.model.infer_vector(sent.split()))
#             print(self.model.infer_vector(sent.split()))

        vecs = np.array(vecs)
        print(vecs)
        self.clf.fit(vecs)
```

```
    def predict_one_sent(self, sent):
        embeding = self.model.infer_vector(word_tokenize(sent))
        print(self.clf.predict([embeding]))
        if self.clf.predict([embeding]) == 1:
            return True
        return False

    def predict_multiple(self, sents):
        acc_list = []
        for sent in sents:
            acc_list.append(self.predict_one_sent(sent))
        return acc_list

    def save_model(self, path):
        dump(self.clf, path+"svm.joblib")
        print("classifier saved")
        self.model.save("d2v.model")
        print("Model Saved")

    def load_model(self, path):
        self.clf = load(path+"svm.joblib")
```

```
from warcio.archiveiterator import ArchiveIterator
from bs4 import BeautifulSoup
import re
import pickle

sents = []
with open('train2.warc.gz', 'rb') as stream:
    for record in ArchiveIterator(stream):
        text = record.content_stream().read()
        soup = BeautifulSoup(text, 'lxml')
        simple_text = re.sub(' +', ' ', soup.text.replace("\n", " "))[:2048]
        sents.append(simple_text)
        print(simple_text)
print("finish adding texts")
with open('sents.pkl', 'wb') as f:
    pickle.dump(sents, f)
```

Report Editor:

Tishauna Wilson

Report Editing

- Fix spelling errors and capitalizations based on Professor’s edits
- Added figures and tables
- Provided example structure of each input for the four subteams
- Updated sections
- Read reviews by other teams
- Implemented changes as needed based on other team’s feedback

Questions?