Migrating database servers:

1. Postgres
   Edit .conf files to allow remote connections/change localhost ip
   Restart
2. MongoDB
   - Follow the instructions given in https://docs.mongodb.com/manual/tutorial/install-mongodb-on-ubuntu/ to install latest MongoDB community edition.
   - Start the new server. Do not stop the old server.
   - Dumping old data: mongodump --out /home/dell/Documents/mongo_backup/ --db media-db
   - Restoring on the new server: mongorestore --host <hostname/ip> --port <port> --username <user> --password <'pass'> <path to backup>
3. Elasticsearch
   - Download the latest version of Elasticsearch
   - Open /etc/security/limits.conf and add the following lines:
     - <username> soft memlock unlimited
     - <username> hard memlock unlimited
     - <username> - nofile 65536
   - Open /etc/sysctl.conf and add the following lines:
     - vm.max_map_count=262144

     NOTE: error: max virtual memory areas vm.max_map_count [65530] is too l low
     sudo sysctl -w vm.max_map_count=262144

   - Open /etc/pam.d/su and uncomment:
     - session   required   pam_limits.so
   - Open elasticsearch-version/config/elasticsearch.yml
     - Set bootstrap.memory_lock: true
     - Set network.host (for production mode). If in development mode and still want to access elasticsearch through an ip:port, set http.bind_host, http.publish_host, http.port, http.publish_port
   - Re- login after the above changes.
   - Run elasticsearch: ES_JAVA_OPTS="-Xms12g -Xmx12g" ./bin/elasticsearch (This means that we are allocating 12GB RAM to elasticsearch node. Usually we allocate half of the available system RAM. If we want to allocate 2 GB, it will be "-Xms2g -Xmx2g")
   - Create index media-db:

```
curl -XPUT 'localhost:9200/media-db?pretty' -H 'Content-Type:
application/json' -d'
{
        "settings" : {
        "index" : {
        "number_of_shards" : 5,
        "number_of_replicas" : 1
```

```
            }
        }
    }
    '
```

This will create media-db index with five primary shards and five replica shards.

- Create mapping entities_resolved

```
curl -XPUT 'localhost:9200/media-db/_mapping/entities_resolved?pretty' -H 'Content-Type: application/json' -d '{"properties": {}}'
```

This will create an empty collection entities_resolved

Migration of scripts:

1. Clone the repository
    a. https://stackoverflow.com/questions/783811/getting-git-to-work-with-a-proxy-server : Git proxy issues
2. On terminal:
    a. sudo vi ~/.bashrc
    b. Add the following lines in bashrc:
        i. export PYTHONPATH=$PYTHONPATH: <path to repository>
        ii. export http_proxy=https://act4d.iitd.ernet.in:3128 (Make sure that
        iii. access to this proxy is given to your IP)
        iv. export https_proxy=https://act4d.iitd.ernet.in:3128
        v. export ftp_proxy=https://act4d.iitd.ernet.in:3128
    c. Save and quit the file.
    d. source ~/.bashrc

    ISSUES: setting proxy for sudo; shows export bash error
    sudo pip --proxy=act4d.iitd.ac.in:3128 install

3. Start the mail server:
    a. $ cd <path to media_filter>/emails
    b. $ python ptunnel.py -d -p act4d.iitd.ac.in:3128 5587:smtp.gmail.com:587

4. Archived URLs

Configu

erations:

1. Fill in start date of all news sources for which you want to crawl URLs in media_filter/scrapy_crawlers/startprocess.py
2. Fill in end date of news sources for which start date is present. (Optional)

Steps:

a. $ sudo pip install virtualenv (Python2 virtual environment)
b. $ virtualenv ENV (ENV is the directory where you want to place virtual environment)
c. $ cd ENV
d. Path to ENV$ source bin/activate (Activating the virtual environment. All the below steps will be taken inside virtual environment)
e. virtual_env$ sudo apt-get install python-dev python-pip libxml2-dev libxslt1-dev zlib1g-dev libffi-dev libssl-dev
f. virtual_env$ sudo pip install scrapy
g. virtual_env$ cd <path to media_filter>
h. [virtual_env] <Path to media_filter>$ python scrapy_crawlers/startprocess.py

To deactivate virtual environment:

1. virtual_env$ deactivate

5. Current URLs

Steps:

1. $ cd <path to media project>
2. <path to media project>$ python rssparsers/startprocess.py

6. Article Text

Configurations:

1. Mongo *collName* (declared initially) in articletext/fetchText.py

Steps:

Installing newspaper3.py (recommended)

1. $ sudo apt-get install python3-pip
2. $ sudo apt-get install python-dev
3. $ sudo apt-get install libxml2-dev libxslt-dev
4. $ sudo apt-get install libjpeg-dev zlib1g-dev libpng12-dev
5. $ curl https://raw.githubusercontent.com/codelucas/newspaper/master/download_corpora.py | python3
6. $ sudo pip3 install newspaper3k

Running the script:

1. $ cd <path to media project>

2. <path to media project>$ python3 articletext/fetchText.py

Checks:

1. Check whether article text of Telegraph is being fetched, as in some cases open-ssl creates a problem in crawling from https sites.

7. Extracting Entities

Configurations:

1. MongoDB *collName* in opencalais/ner_new<n>.py (declared initially)
2. *publishedDate* range of articles in fetchEn() for which entities have to be extracted.

Steps:

1. $ cd <path to media project>
2. <path to media project>$ python opencalais/ner_new1.py
3. Sim, run ner_new2.py and ner_new3.py

8. Entity Resolution

Making a separate unresolved entity collection:

Configurations (entity_resolution/extract_entities_oc.py):

1. MongoDB article *collection* in extract()
2. MongoDB unresolved entities *collection* in save()

Steps:

1. $ cd <path to media project>
2. <path to media project>$ python entity_resolution/extract_entities_oc.py

Resolving entities in unresolved collection:

Configurations (entity_resolution/elasticsrch_oc_mod<n>):

1. Unresolved MongoDB entity *collection* declared during initialization of script.
2. Resolved MongoDB entity collection *mongo_coll* declared initially.
3. Resolved Elasticsearch entity collection *es_mapping* declared initially.

Steps:

1. $ cd <path to media project>

2. <path to media project>$ python entity_resolution/elasticsrch_oc_mod1.py (To resolve Person)
3. Sim, run elasticsrch_oc_mod2.py (Country, Continent), elasticsrch_oc_mod3.py (City, ProvinceOrState), elasticsrch_oc_mod4.py (Company, Organization)

9. Keyword Extraction

Configurations (keyword_extraction/RAKE/extractkeyword.py):

1. MongoDB article *collection*.

Steps:

1. $ cd <path to media project>
4. <path to media project>$ python keyword_extraction/RAKE/extractkeyword.py

10. Sentiment Analysis

1. AlchemyAPI

Configurations (alchemy-fetch/datafetch.py)

a. Mongo collection name *mongoColl* in datafetch.py

Steps:

1. $ cd <path to media project>
2. <path to media project>$ python alchemy-fetch/datafetch.py

2. SentiStrength

Configurations (sentistrength/sentiment_cal.py)

1. Mongo *collection* in sentiment_cal.py

Steps:

1. $ cd <path to media project>
2. <path to media project>$ python sentistrength/sentiment_cal.py

11. Opinion Category

Individual scripts for news sources are present in opinion_category. These scripts label opinions into EDITORIAL, COLUMN, LETTER (Letter to editor)

Configuration:

1. Mongo collection *collName* in every script within opinion_category folder

12. Author Description (for Columns)

This script extracts description of columnists published on the news website.

Configuration:

1. Mongo collection *coll* in author_extraction/author_info.py

Steps:

1. $ cd <path to media project>
2. <path to media project>$ python author_extraction/startprocess.py

Common Configurations:

- config.py- Mongo DB configurations
- storage/storemeta.connect()- Postgres configurations
- emails/sendemail.sendEmail()- Email Ids of receivers