Read Me

Large Scale Data Collection and preprocessing in Spark

Hongyong Zhang: hxz174130

Ruinong Tian: rxt172530

Shouzhe Zhang: sxz180014

Wei Luo: wxl180009

Yiwen Wang: yxw171630

**Step by step result:**

**1. Collects data by crawling a set of Spanish news websites on a daily basis**

Environment: Linux

First, preinstall scrapy in your machine.

1. Go to the ./newslinks folder.

2. Run cmd: scrapy crawl elcolombiano

3. Find URL.txt file which contains all crawled daily URLs. (If you want to generate a new URL file, make sure you have deleted the original one, otherwise, URLs will merge together.)

It takes 10 to 20 minutes to run the program.

Sample output can be found as follows.

<https://utdallas.box.com/s/ruiy9sgdsau157vhczm8jlzf627y6gvb>

**2. Extracts the main content of the article and related metadata (i.e headline, author, date published)**

For this step, **parseURL.py** and **getNewsMulti.py** are needed.

**parseURL.py** will take a txt file as input, and parse it into multiple txt files with 500 ulrs each.

To run, use the command as following:

python parseURL.py

Sample output can be found in:

<https://utdallas.box.com/s/y5usp2q6ik7urjfsoszyupxyg8qghq1t>

**getNewsMulti.py** will take all txt files above, fetch content and metatdata of Spanish news and dump them in multiple json file.

To run, us the the command as following:

python getNewsMulti.py

Sample output can be found in:

<https://utdallas.box.com/s/9wzl3aisqlqwjf096lds3c9sctq78sbp>

For each txt file(500 urls), it will take about 4 minutes

**3. Process the extracted content with udpipe and generate universal dependency parse for each sentences within the content (use Apache Spark here)**

First run the zookeeper, in the kafka dic, run at the command line:

bin/zookeeper-server-start.sh config/zookeeper.properties

Then run the kafka:

bin/kafka-server-start.sh config/server.properties

Read the json file line by line, each line is one json object containing one article’s information. Then like the stream\_producer.py example code from homework3, push one line per time.

Run the command at the python file dictionary to start producer:

python3 ParseJson.py

**4. Store the collected data and processed data in a way compatible with event coder’s input format in MongoDB**

Run the mongodb server at the command line:

sudo service mongod start

Run the command at the python file dictionary to start consumer:

python3 run\_udpipe.py tokenize conllu spanish.udpipe

the spanish.udpipe is a model from online source.

Json file exported from the mongodb (about 1GB):

<https://utdallas.box.com/s/t2558huq9n1vgrxbwqeeaa6nnl0t1ydg>

**5. Running some deduplication algorithm at content level. (Comparing two articles from different urls and find out whether they cover the same story)**

For this step, we got all the datas from mongodb(the last step) and determined the similarity of different contents by posting this request to the API from website called eventregistry.org. Select the target object and post request to the website to get the similarity with all the others.

python3 similarity.py

