Project Report

Large Scale Data Collection and preprocessing in Spark

Hongyong Zhang: hxz174130

Ruinong Tian: rxt172530

Shouzhe Zhang: sxz180014

Wei Luo: wxl180009

Yiwen Wang: yxw171630

**Programming Environment:**

We implemented this project under linux and mac, using python language, and store result data into MongoDB.

**Tentative method:**

1. First we will design a crawler which can avoid duplication according to the headline, author’s name and date published of different articles.

2. Then we will deal with the content duplication during the spark process.

Tentative Schedule:

Implement one step in the guideline per week.

**Step by step result:**

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

The different website has different ways to jump to the next page. Some websites create the next page’s URL using its own function. Some websites create the next page’s URL by adding parameters. Some websites even use its own function to create the parameters with its magic number to create a new next page’s URL. Some use buttons to get more news’ links. It’s difficult to analyze all the websites’ pattern to crawl the data.

In that situation, we analyze all websites’ jumping method and create the corresponding URL pattern generator program. For example, some website will be like this.



Some will be like this.

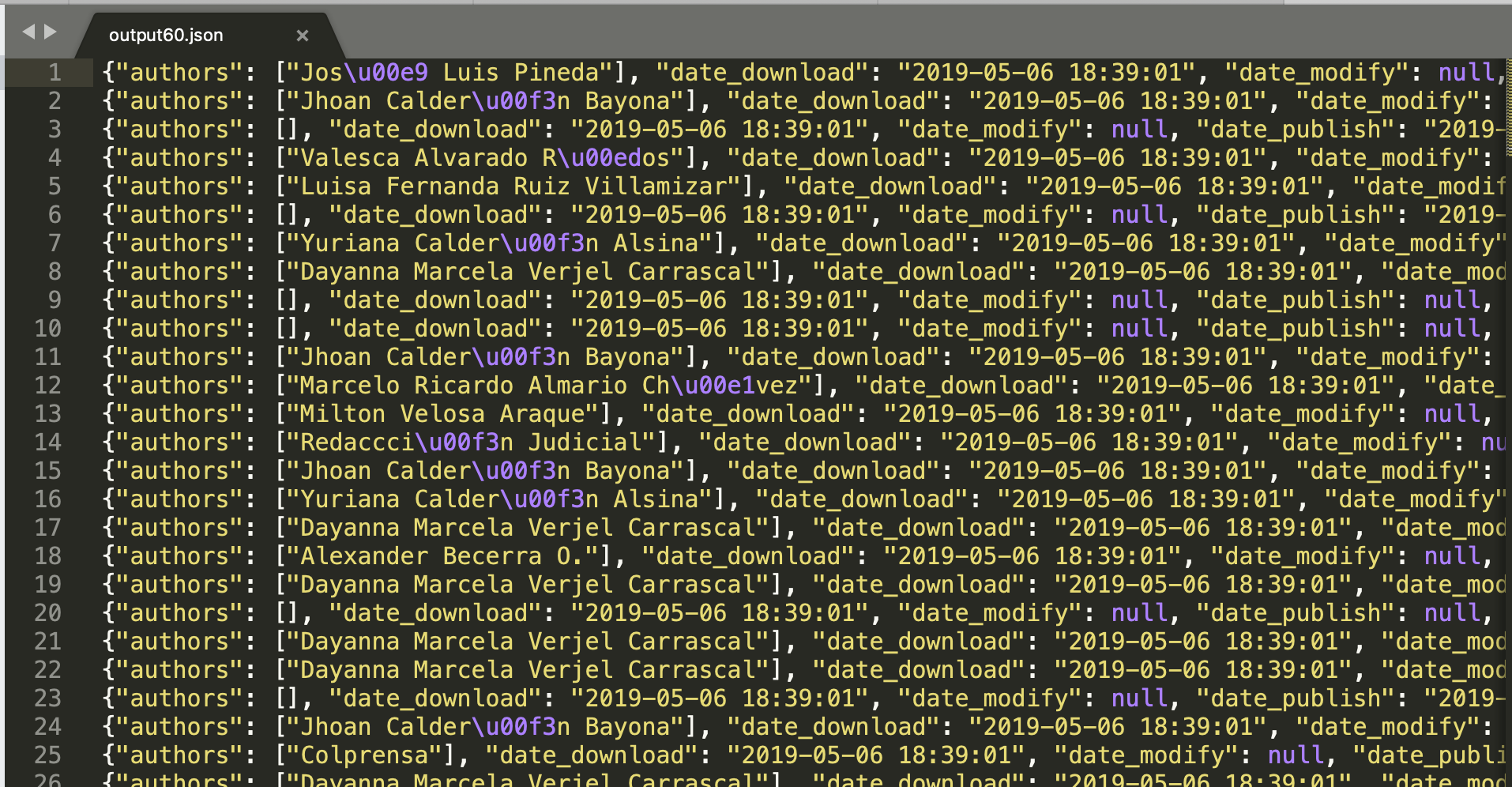


After that, we get a large website dataset to crawl our news’ URL. However, we found out there are too many useless URL. We crawl 250,000+ URL from the dataset. But there are too many duplicates and HTTP unformatted URLs. So we also apply some filters to filter the unqualified URLs. We use a regular expression to filter some strings which are not URL. And rule out URL which contains “/” less than 3. That means we don’t want the URL which might be front page. These front pages usually don’t contain news’ link.

At last, we retrieve URLs 150,000+ URLs for a day.

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

Based on 150,000 urls generated from step 1, we filtered and crawled 11,000 valid spanish news articles with their metadata using newsplease. Results are show as following:



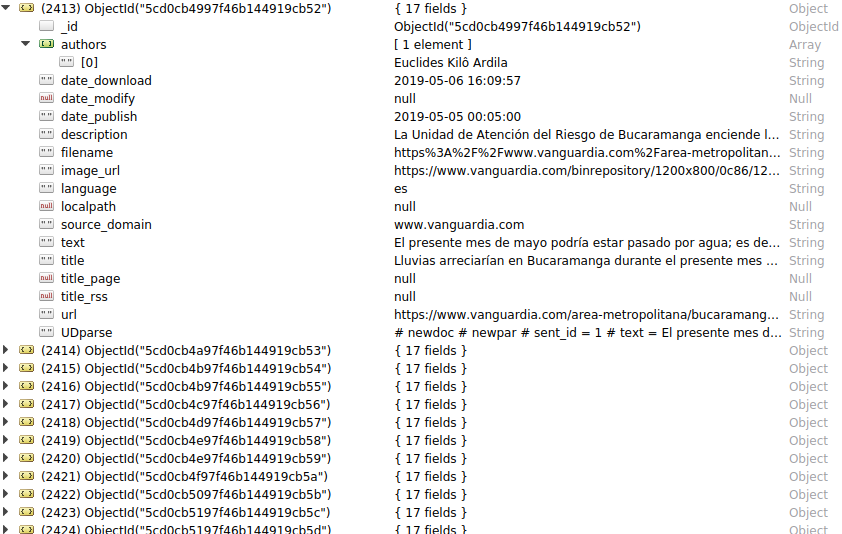
**3. Process the extracted content with udpipe and generate universal dependency parse for each sentences within the content (use Apache Spark here)**

Read json format extracted news content, push them using kafka producer. Then at the consumer side, using udpipe to generate universal dependency of each article.

****

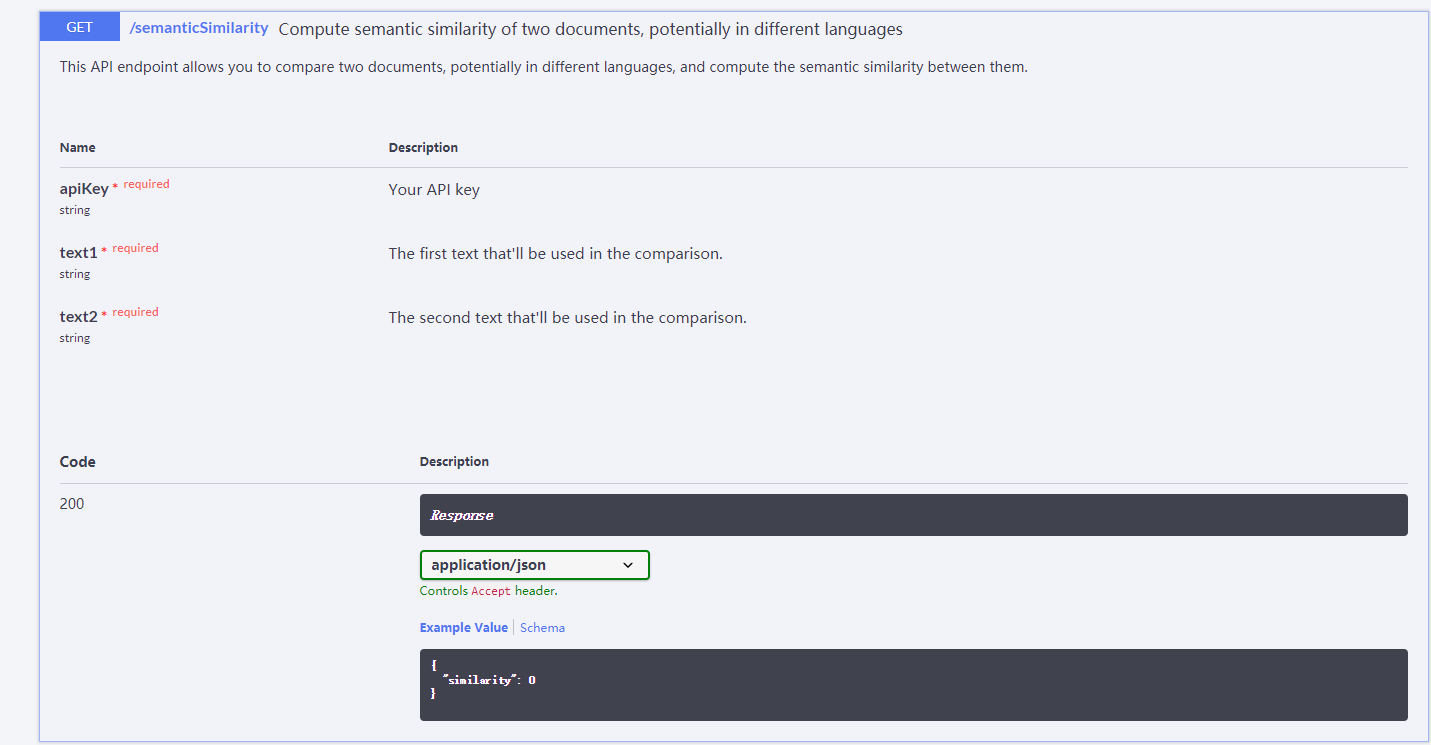
**4. Store the collected data and processed data in a way compatible with event coder’s input format in MongoDB**

Store the extracted content and processed content of each news article into mongodb. Also done at the consumer side.

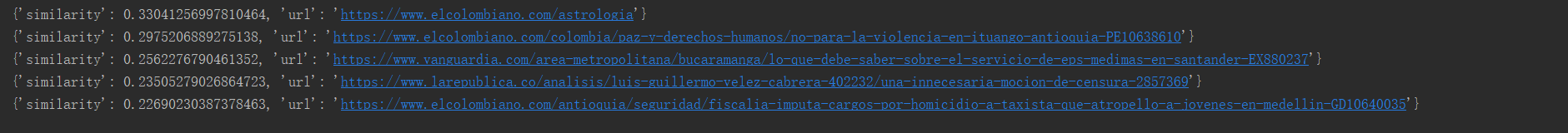


**5. Running some deduplication algorithm at content level. (Comparing two articles from different urls and find out whether they cover the same story)**

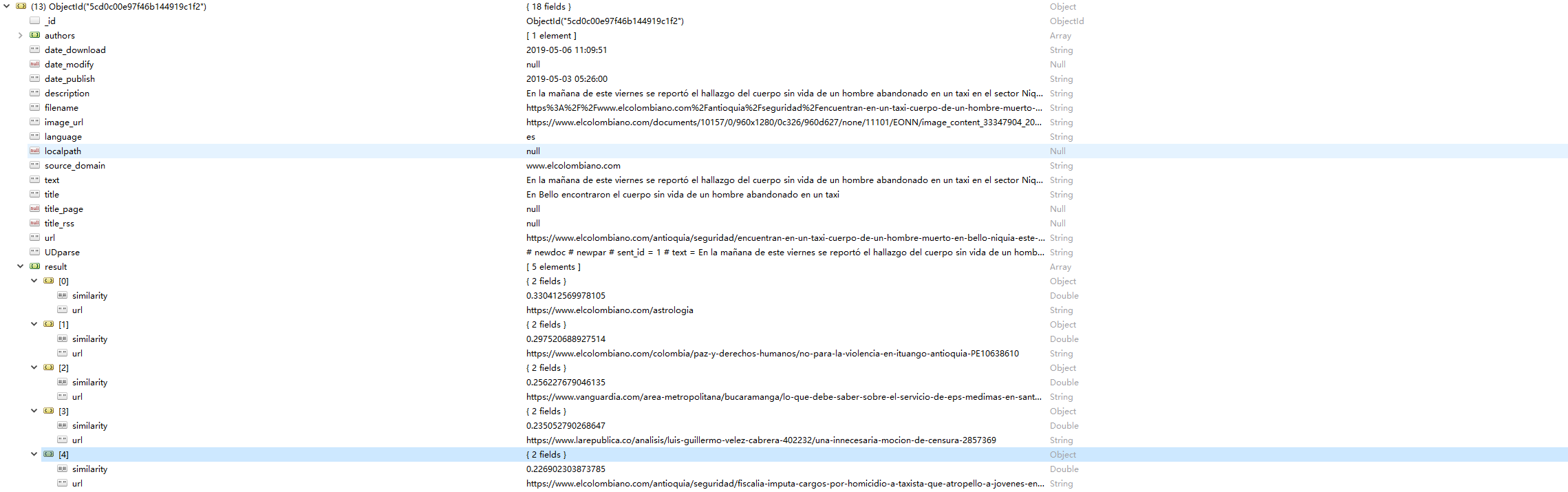
For this step, we got all the datas from mongodb and determined the similarity of different contents by posting this request to the API from website called eventregistry.org.



The outputs were in the following pictures. We have the similarity(from API) and the url of news article. Those were the urls which have the top 5 similarity, which realized by the python program we wrote. We use the 13th data in the database to compare with all the rest data to put the top 5 similarity in order.



And it also will be stored in mongodb.



**Related Literatures:**

1. News-please ​https://github.com/fhamborg/news-please

2. Scrapy ​https://scrapy.org/

3. Apache Kafka ​https://kafka.apache.org/

4. SPEC paper ​https://ieeexplore.ieee.org/document/7474330

5. Universal Dependency ​https://universaldependencies.org/

6. ufal-udpipe python package.

7. Deduplication Papers/Resources

a. <https://www.hindawi.com/journals/mpe/2016/3919043/>

b. <https://www.aclweb.org/anthology/P16-4019>

c. Dissertation of Dr. Ahmad Mustafa, <https://search.proquest.com/pqdtlocal1006281/docview/2086379093/EDE7D1E6>

ED9843E5PQ/1?accountid=7120

d. https://www.eventregistry.org/documentation?tab=semanticSimilarity

8. List of news sources https://docs.google.com/spreadsheets/d/13DmJ140wW8pCp6nyRSAk911S7AoF-6zJOJ- F77qoMuM/edit?usp=sharing