BIG DATA CS-GY 6513
SPRING 2023

COMMONCRAWL INSIGHTS AND TOPIC EXPLORATION

PROJECT REPORT

SUBMITTED BY:

NAME	NET ID	N NUMBER
Arjun Naga Siddappa	as15840	N16338575
Anil Poonai	ap5254	N11988828
Rohan Patel	rp3617	N13328358

Tandon School of Engineering
New York University

TABLE OF CONTENTS

PROBLEM STATEMENT

The vastness of the internet, with over 1.13 billion websites, presents challenges in analyzing content topics, website popularity, and top-level domain distribution. Leveraging CommonCrawl data and Big Data technologies, we aim to develop a fast, distributed pipeline to address three primary objectives:

- **Topic Modeling**: Efficiently group websites based on their content topics, revealing underlying patterns and themes.
- **Top-Level Domain Distribution**: Analyze the distribution of top-level domains across the web.
- **Website Popularity**: Rank websites by popularity, as determined by the frequency of references from other sites.

DATASET

The Common Crawl corpus is a valuable resource for web-based research, containing petabytes of data gathered since 2008. This extensive dataset consists of raw web page data, extracted metadata, and text extractions, providing a comprehensive view of the internet's content.

The data in the corpus is organized monthly, with approximately 88,000 text files available for each month. These text files comprise HTTP requests for numerous websites, offering a wealth of information for analysis. For our project, we focused on processing the files from November and December 2022.

Here is a sample of the structure of WET files.

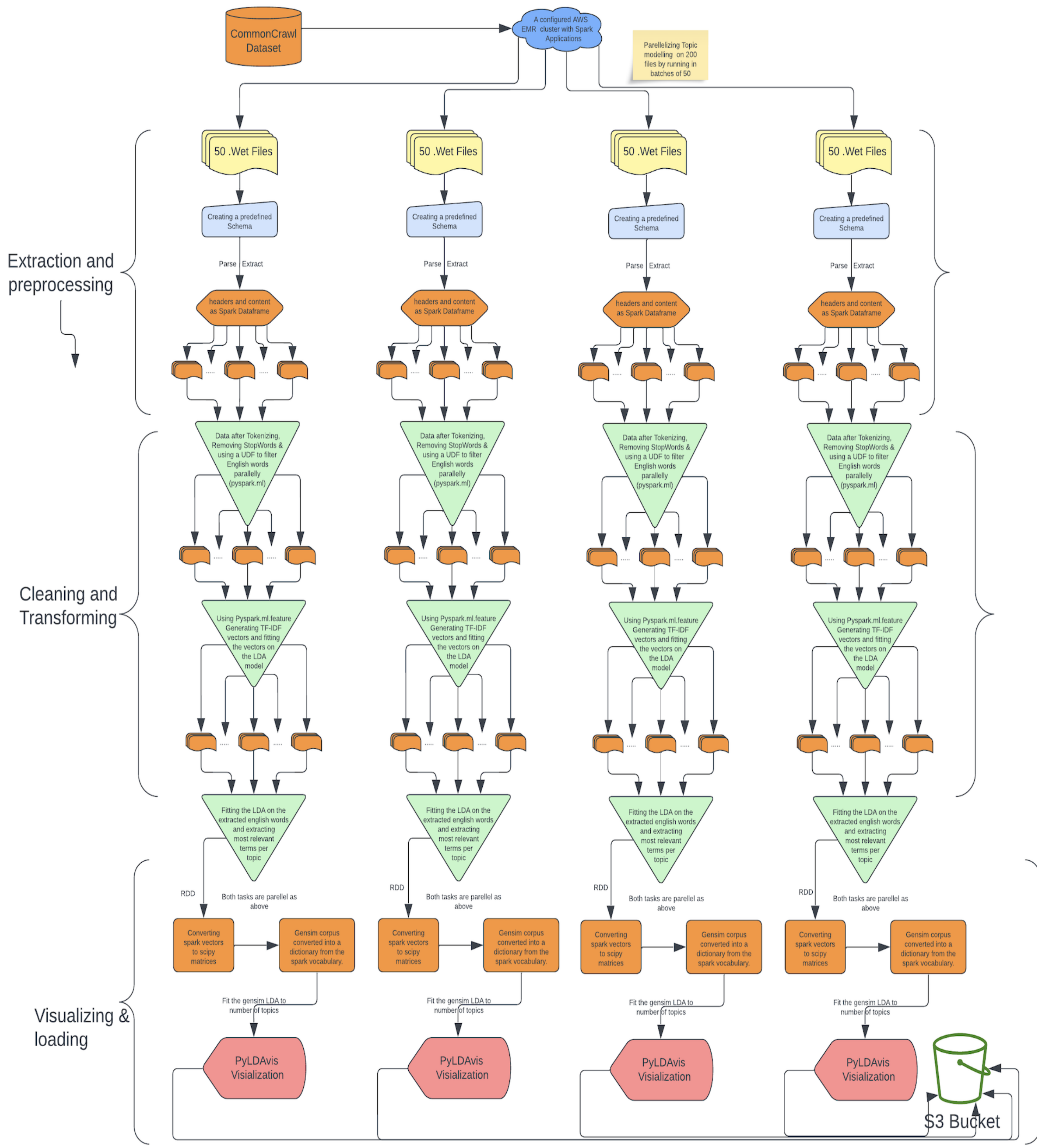
```
WARC/1.0
WARC-Type: conversion
WARC-Target-URI: http://news.bbc.co.uk/2/hi/africa/3414345.stm
WARC-Date: 2014-08-02T09:52:13Z
WARC-Record-ID:
WARC-Refers-To:
WARC-Block-Digest: sha1:JROHLCS5SKMBR6XY46WXREW7RXM64EJC
Content-Type: text/plain
Content-Length: 6724

BBC NEWS | Africa | Namibia braces for Nujoma exit
...
President Sam Nujoma works in very pleasant surroundings in the small but
beautiful old State House...
```

ARCHITECTURE

3.1 Topic Modeling

The figure below shows how we have parallelized Topic modeling on the CommonCrawl Dataset which is a large corpus of web data, utilizing pySpark along with it’s several ML libraries we achieved our results in a distributed manner and Visualized or results using PyLDAvis as explained further.



The process shown above can be broken down into the following steps:

1. **Setting up AWS Cluster:** Configured the cluster, kernel and PySpark with appropriate driver memory and retrieved web files from the S3 bucket after importing os, warcio, re, nltk, gensim and PyLDAvis.

2. **Defining Schema and Initializing PySpark:** Established a schema using StructType and StructField to define the structure of data we are working with.
3. **Parsing the WARC Data:** Read the .wet files iterating through its records using warcio.archiveiterator.Archiveiterator to extract the headers and content for further creating a Spark DF.
4. **Creating a Spark DF :** Above records were converted to Spark DF for further distributed computing.
5. **Preprocessing Spark DF:** Tokenized the data using RegexTokenizer, removed stop words with StopWordsRemover, and filtered English words using a user-defined function (UDF) with the nltk 'words' corpus. All of these tasks were performed using the pyspark.ml.feature library.
6. **Transforming the preprocessed data:** Generated TF-IDF vectors using Countvectorizer from pyspark.ml.feature library.
7. **Performing Topic modeling with LDA:** We fit an LDA model with specified number of topics and iterations. Most importantly, we extracted the most relevant terms for each topic using a UDF.
8. **Visualize LDA results:** Created a gensim corpus from the scipy matrices created from the spark vectors using an RDD map and converted the corpus to a gensim dictionary using Spark Vocabulary.
9. **Writing to S3 Bucket:** Finally, we stored the results into an S3 bucket.

3.2 Top level domain distribution

In this task as shown in the architecture below, the following steps take place:

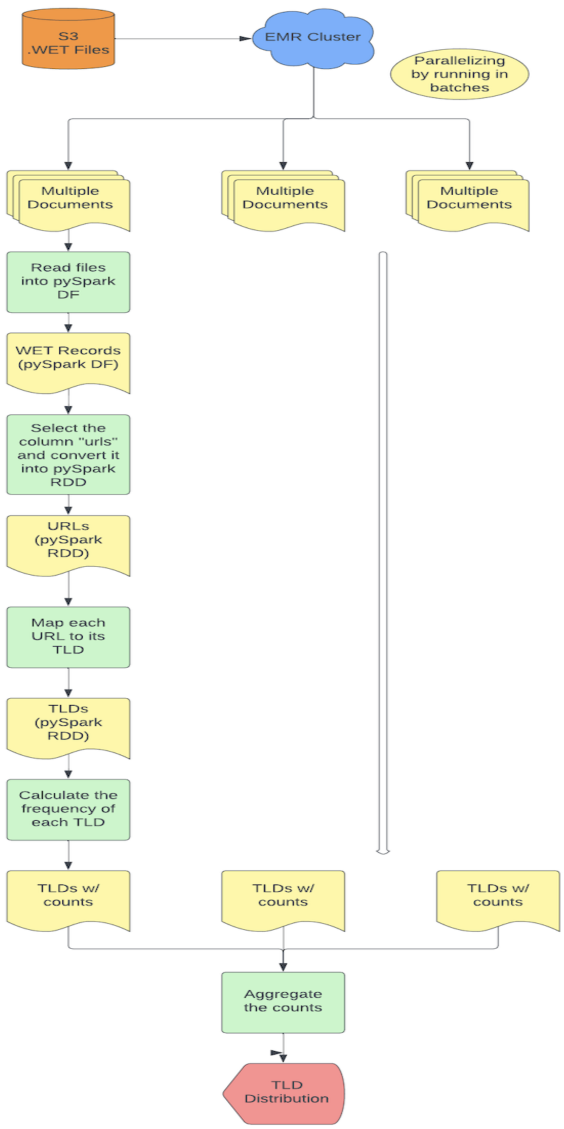
1. Data from the input files are read into a single pySpark dataframe
2. The dataframe consists of many fields. We select the ‘url’ column and remove duplicates. We then convert the column object into a pySpark RDD.
3. We map each URL to it’s top level domain
4. We then perform the countByValue operation to get the counts of each TLD.

3.3 Popularity of websites

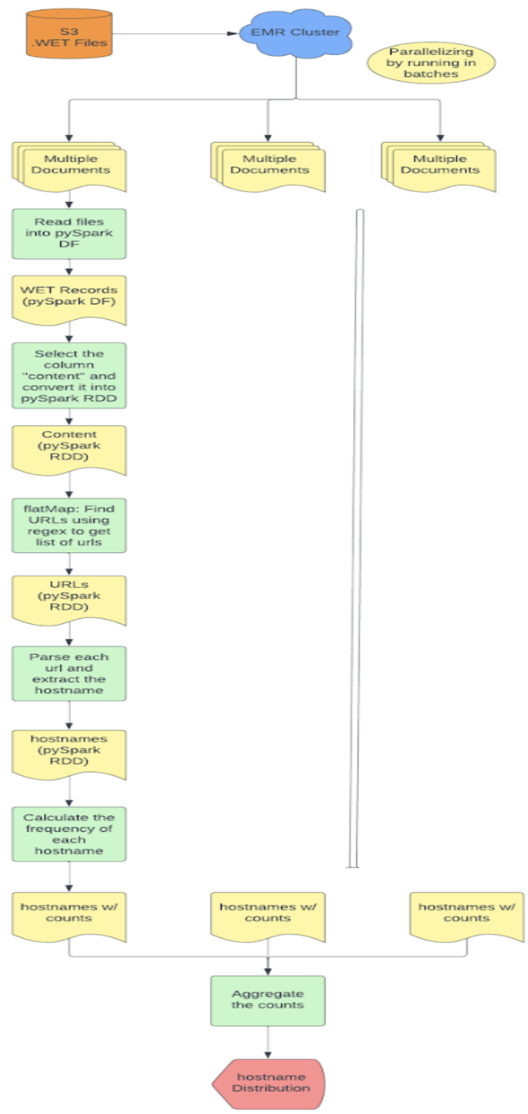
In this task as shown in the architecture below, the following steps take place:

1. Data from the input files are read into a single pyspark dataframe
2. The dataframe consists of many fields. We select the ‘content’ column and convert it into a pySpark RDD.
3. For each content entry we extract the list of urls present in the text using Regular expressions. For this, we use the flatMap function so that the list from all entries are merged into a single result list.
4. On the resultant RDD from the previous step, we map each url to the hostname.
5. We then obtain the frequency of each hostname using the CountByValue function.

Top Level Domain Distribution



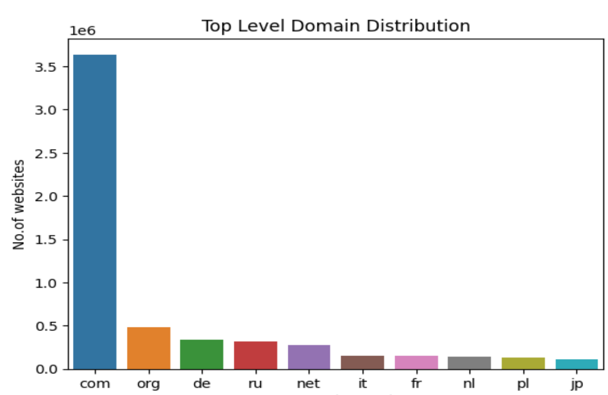
Popularity of Websites



RESULTS

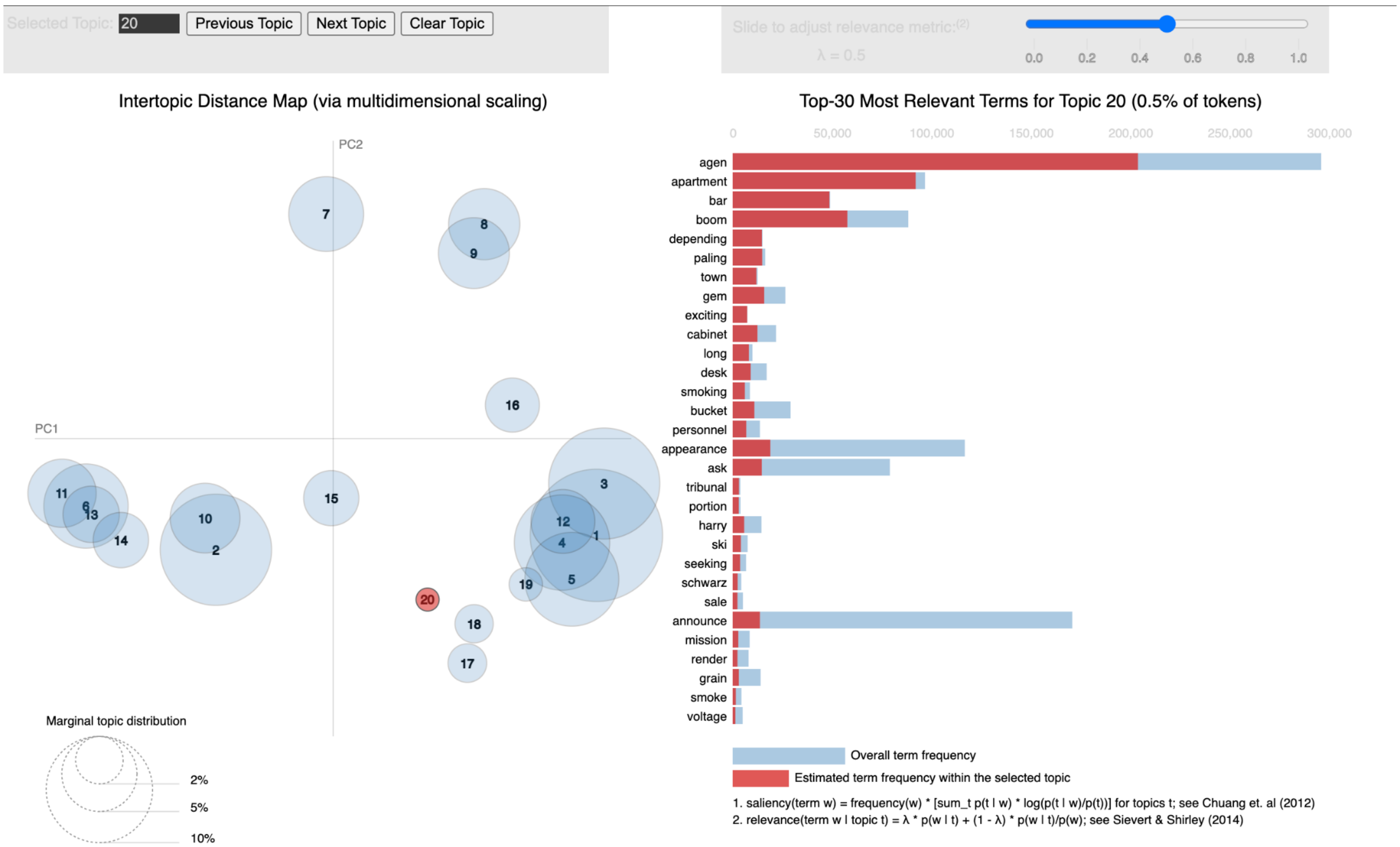
Top Level Domain Distribution:

- We can see that there are significantly more number of ".com" websites followed by ".org"
- Of the corpus processed, it appears that the websites are from Russia and European countries



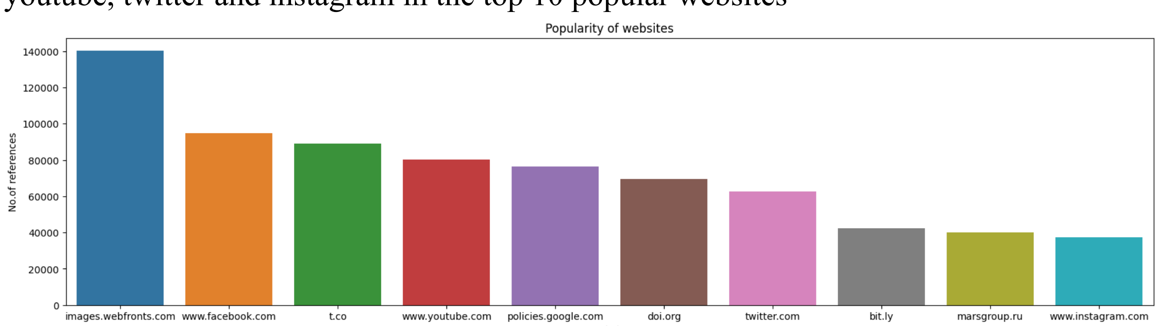
Topic Modelling(PyLDAvis):

- **Intertopic Distance Map:** 2D plot with topic circles indicating similarity.
- **Circle Size**: Topic prevalence.
- **Topic Relevance:** Top terms ranked by relevance.
- **Hovering:** Reveals related topics, term frequency/rank.
- **Tooltip Info:** Shows topic number, prevalence, top terms.
- **Adjusting λ:** Tailors term rankings, emphasizing frequent or unique terms.



Popularity of Websites :

- We define popularity as the number of times a particular website is hyperlinked in other websites
- The results show that "images.webfront.com" an image upload and view website from Russia is the most hyperlinked website
- We can also see the expected popular social network and social media sites such as Facebook, youtube, twitter and instagram in the top 10 popular websites



CONCLUSION

- We successfully built a pipeline using pySpark for processing a portion of the Common Crawl dataset and performing three analytical tasks on the data
- We determined that the more efficient way to do this is to use EMR that would distribute tasks across the cluster and leverage concurrency
- We performed topic modeling using PySpark's ML module and LDA algorithm to group websites based on content similarity
- We obtained the distribution of top-level domains and found ".com" to be the most frequent
- We aggregated website reference counts in the content of all the websites in the data."images.webfronts.com" is the most popular site in the sample
- The pipeline we have developed has the potential to analyze the (1 PiB) of common crawl data in ~1000 hours using just the lower-end processing capabilities
- We realized and demonstrated the power of distributed computing for large-scale data analysis and valuable insights.

APPENDIX

Data
Commoncrawl Website: https://commoncrawl.org/, https://index.commoncrawl.org/

Code
GitHub Repository: https://github.com/DevonARP/Common_Crawl_Big_Data_6613_Project

Algorithm
LDAvis: https://nlp.stanford.edu/events/illvi2014/papers/sievert-illvi2014.pdf
LDA Topic Modeling: https://towardsdatascience.com/latent-dirichlet-allocation-lda-9d1cd064ffa2