

# CSCI 599 – spring 2016 – Assignment 1 – Group 12

This report summarizes details all of our work on first assignment. First we shall cover each point as per task list one by one along with our results and challenges faced.

## 1. Download and install Apache Tika

With the help of Tika documentation, this was easiest task of all. We downloaded entire source and built it using “mvn install” command. We were able to run tika using command line, GUI and python rest server. We also wrote small program in Java which used Tika class to detect Mime type. More information on that in Task 3.

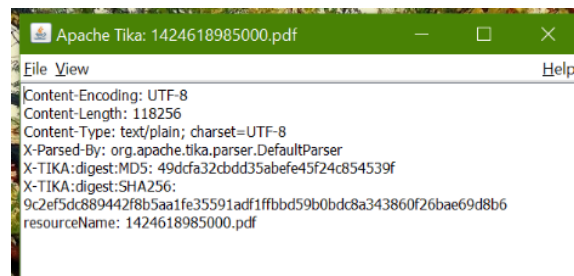
## 2. Download and install D3.js

Downloading and installing d3 was straight forward. We just downloaded d3.js and d3.min.js files and looked at rich gallery of visualization on its github page.

## 3. Download the Amazon S3-based TREC-DD-Polar data

This one ate lot of our time initially. We faced multiple issues in downloading data, cleaning data and distributing data. First of all, we downloaded all diversity Json files from “latest-commoncrawl” bucket’s main directory. Based on number of different file types and size, we chose to download “572-team1” folder to try out before we go for full download. Challenge here was that we did not know which folder contains which file types as folder hierarchy was based on crawling and had no ordering with respect to MIME types.

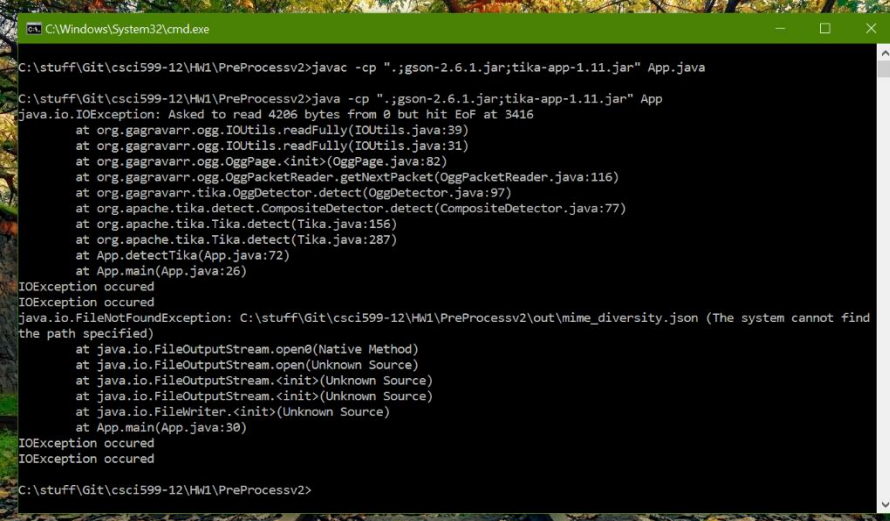
After downloading, we wrote small Java program to recursively scan through folders, run tika mime detection over each file and move files based on mime types. Before running it over entire folder we downloaded, we manually ran it on few folders and found out that there was something wrong in the data and Tika was not able to detect them correctly. E.g. PDF file at S3 url `crawl/572-team1/201/175/47/81/9abb360bb295e7b0e256e2ecf74d64aca7f13fca/1424618985000.pdf` was being detected “text/plain” and it wasn’t even opening in pdf viewer whereas most of the html



files were detected correctly. After careful review of different file types, we found that most of the non-text type were detected incorrectly. In subsequent classes, we came to know that “latest-

commoncrawl" contained data from Apache Nutch which encodes every file in cbor format which sadly Tika can't decode. We spent some time understanding cbor format and modifying our program to sort files to decode cbor before sending it to Tika. After a while we were able to decode cbor format. However, we downloaded few folders from "polar-fulldump" in parallel and found that they did not contain any cbor encoding around its data. Given the overhead of cbor format and large amount of data that we had to process, we ditched "latest-commoncrawl" and decided to download entire "polar-fulldump" data. Among three of us, we distributed 60 folders each as there were 180 folders in total.

We started batch download in the night but download went on the whole next day. Also as we did not know size distribution, one of us ended up downloading 50 GB+ data and two of us got ~7 GB and ~10 GB respectively. Two of us with lesser amount of data were quickly able to run Tika over entire data to sort the files. However, we observed that Tika was throwing IOException in determining certain files which we did not handle gracefully. We spent some time trying to



```
C:\stuff\Git\csci599-12\HW1\PreProcessv2>javac -cp ".;json-2.6.1.jar;tika-app-1.11.jar" App.java
C:\stuff\Git\csci599-12\HW1\PreProcessv2>java -cp ".;json-2.6.1.jar;tika-app-1.11.jar" App
java.io.IOException: Asked to read 4286 bytes from 0 but hit EOF at 3416
    at org.gagravarr.ogg.IOUtils.readFully(IOUtils.java:39)
    at org.gagravarr.ogg.IOUtils.readFully(IOUtils.java:31)
    at org.gagravarr.ogg.OggPage.<init>(OggPage.java:82)
    at org.gagravarr.ogg.OggPacketReader.getNextPacket(OggPacketReader.java:116)
    at org.gagravarr.tika.OggDetector.detect(OggDetector.java:97)
    at org.apache.tika.detect.CompositeDetector.detect(CompositeDetector.java:77)
    at org.apache.tika.Tika.detect(Tika.java:156)
    at org.apache.tika.Tika.detect(Tika.java:287)
    at App.detectTika(App.java:72)
    at App.main(App.java:26)
IOException occurred
IOException occurred
java.io.FileNotFoundException: C:\stuff\Git\csci599-12\HW1\PreProcessv2\out\mime_diversity.json (The system cannot find the path specified)
    at java.io.FileOutputStream.open0(Native Method)
    at java.io.FileOutputStream.open(Unknown Source)
    at java.io.FileOutputStream.<init>(Unknown Source)
    at java.io.FileOutputStream.<init>(Unknown Source)
    at java.io.FileWriter.<init>(Unknown Source)
    at App.main(App.java:30)
IOException occurred
IOException occurred
C:\stuff\Git\csci599-12\HW1\PreProcessv2>
```

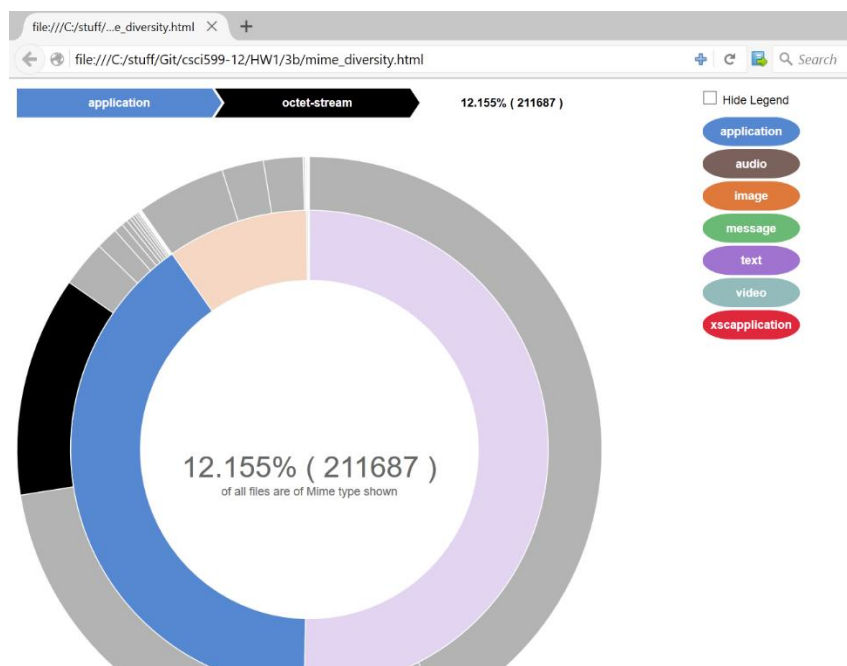
figure out why this was happening and tried bunch of different Tika version. This wasted our one day as we could not run the program over night over 50 GB data chunk which had halted due to same error. Next day we updated program and re-executed over remaining data. We faced some unexpected delay while running our program on OS X due to automatically created .DS\_STORE files. After ignoring those files we were finally able to sort files. Although, our laptops were hanging so we had to run it over multiple iteration manually over small chunks of data.

Once, we had all the data sorted, we found that mime diversity reported on <http://github.com/chris mattmann/trec-dd-polar/> was not entirely matching with mime diversity that we observed. E.g. before looking at the data, we had decided to include "xscapplication/zip" mime type for our analysis. However, we did not get even a single file detected as that type from entire "polar-fulldump" even though there were supposed to be 85 files of that type. This is probably due to evolution of Tika over time and its ability to classify correctly or incorrectly. Finally we chose 15 MIME types based on data that we had and variety of MIME type. Below screenshot shows all 15 mime type that we selected along with number of files and total size we used.

| Name                               | Subtree Percent... | Perce... | > Size    | Items   | Files   | Subdirs | Last Change       | Attri... |
|------------------------------------|--------------------|----------|-----------|---------|---------|---------|-------------------|----------|
| C:\stuff\Git\polar_data_all_mime15 |                    | [1:58 s] | 32.0 GB   | 546,012 | 545,997 | 15      | 02/28/16 04:09:00 |          |
| application_xhtml+xml              | 29.7%              |          | 9.5 GB    | 321,989 | 321,989 | 0       | 02/28/16 04:09:00 | A        |
| application_pdf                    | 27.9%              |          | 8.9 GB    | 45,968  | 45,968  | 0       | 02/28/16 01:44:43 | A        |
| video_mp4                          | 13.2%              |          | 4.2 GB    | 771     | 771     | 0       | 02/28/16 01:55:18 | A        |
| image_jpeg                         | 12.4%              |          | 4.0 GB    | 93,559  | 93,559  | 0       | 02/28/16 01:54:17 | A        |
| video_quicktime                    | 6.7%               |          | 2.1 GB    | 920     | 920     | 0       | 02/28/16 01:55:19 | A        |
| image_png                          | 5.6%               |          | 1.8 GB    | 39,625  | 39,625  | 0       | 02/28/16 01:55:17 | A        |
| image_gif                          | 1.4%               |          | 451.2 MB  | 36,043  | 36,043  | 0       | 02/28/16 04:00:42 | A        |
| application_zip                    | 1.3%               |          | 433.6 MB  | 1,611   | 1,611   | 0       | 02/28/16 01:52:25 | A        |
| audio_mpeg                         | 1.3%               |          | 431.8 MB  | 670     | 670     | 0       | 02/28/16 01:52:25 | A        |
| application_octet-stream           | 0.4%               |          | 134.9 MB  | 327     | 327     | 0       | 02/28/16 03:59:19 | A        |
| text_x-matlab                      | 0.0%               |          | 12.4 MB   | 1,284   | 1,284   | 0       | 02/28/16 01:55:18 | A        |
| application_dif+xml                | 0.0%               |          | 8.8 MB    | 2,906   | 2,906   | 0       | 02/28/16 01:28:11 | A        |
| audio_x-ms-wma                     | 0.0%               |          | 7.1 MB    | 56      | 56      | 0       | 02/27/16 17:19:31 | A        |
| audio_x-wav                        | 0.0%               |          | 3.7 MB    | 59      | 59      | 0       | 02/28/16 01:35:16 | A        |
| message_rfc822                     | 0.0%               |          | 2.3 MB    | 208     | 208     | 0       | 02/28/16 01:36:21 | A        |
| mime_types.txt                     | 0.0%               |          | 237 Bytes |         |         |         | 02/28/16 01:51:30 | A        |

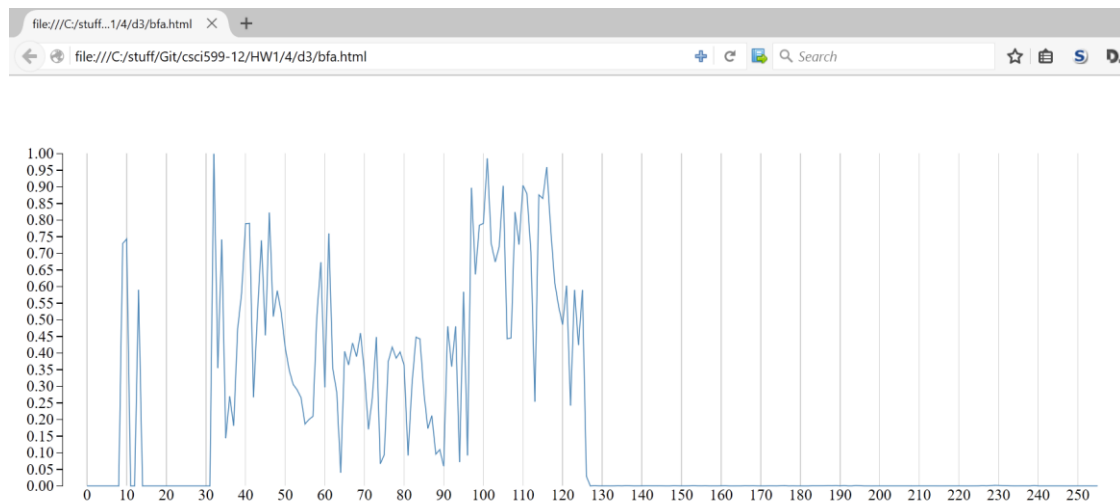
As shown in screenshot, we have chosen four types from application, three from audio, three from image, one message, one text and two video and octet stream amounting to total of 32 GB and over 500K files. Another interesting fact to point out here is that all of the files are having no extension at all which makes it harder to verify the accuracy of mime types.

Below is the screenshot of interactive Sunburst D3 chart for existing MIME diversity of the TRECDD-Polar dataset using the existing JSON breakdown from Github. We modified existing code from <https://gist.github.com/kerryrodden/7090426> to match as per our requirement. It made sense to use this kind of layered pie chart instead of using single pie chart which becomes messier as there are more than 90 mime types. Even here, we can hardly see audio/message/video/xscapplication because of small number of files. We have included this chart in our submission. Screenshot is created with mouse hover over octet stream data which one can see on the top left along with its relative percentage and number of files. On the left we are showing legend for top level mime types. As we can see from the chart, majority of files belong to text/html mime type which makes sense as this is web crawl data.



#### 4. BFA

Below is a sample BFA fingerprint plotted as a line chart applied on 75% text/x-matlab mime types. This is just one of the example. We have submitted bfa fingerprint for all fifteen mime types over 75% of our data.



#### 5. BFC and BFCC

#### 6. FHT

#### 7. Final analysis

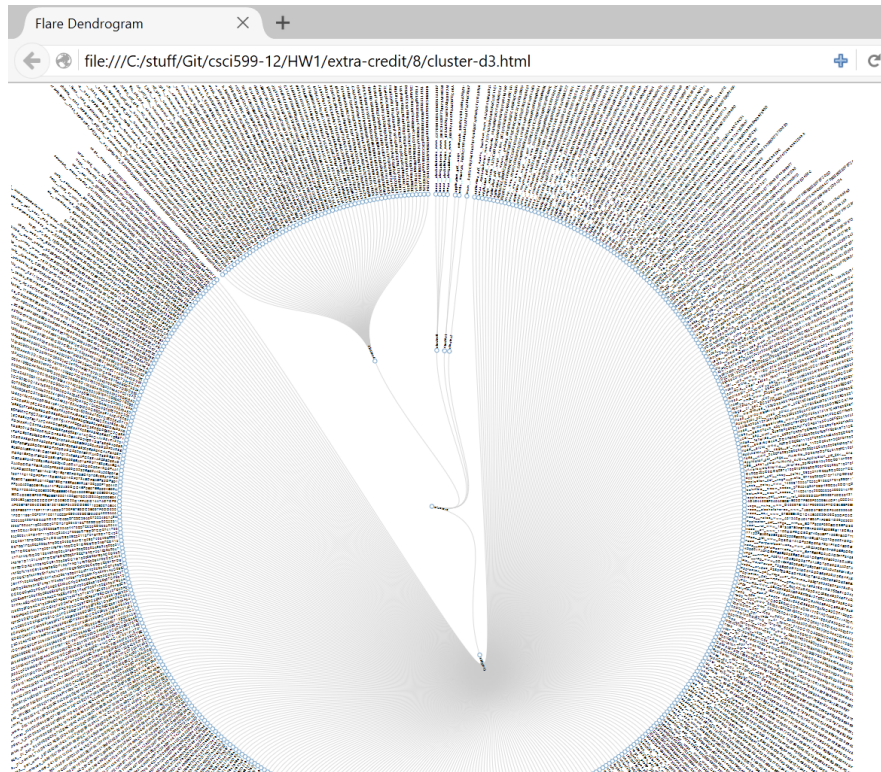
#### 8. Tika Similarity

First of all tika-similarity is really inefficient and full of bugs. Below are few of the issues we faced. First of all edit-distance repo was not maintained and we had issues installing it. We raised issue at <https://github.com/aflc/editdistance/issues/6>.

Once, we got fix, we were able to run edit distance script. However, now we got new problem where tika server was not able to handle requests from edit and cosine distance scripts and it was throwing connectError exception for large amount of mime diversity data on which we tried to run. Issue was reported here. <https://github.com/chrismattmann/tika-similarity/issues/58>

After this issue was resolved, we were able to run edit distance and cosine distance for large set of files but it was taking forever. Simply because initially we ran it on 36000 files of image/gif mime data that we had. After running for one full day, we realized that it will have to compute 36000 choose 2 scores and it is simply too much computation for the time we had. So we decided to run it over random 1000 PDF files from over mime dataset downloaded from polar. We have submitted all three distances (Jaccard, Edit distance and Cosine distance) related text, csv and html files for visualization.

In Jaccard, total of five clusters were created whereas cosine distance created 498 clusters and so did edit distance:



We have submitted all html, json and csvs for visualization of all running all three distances using tika-similarity.

## 9. Content Based MIME detector

For content based mime detection, first we created a script to generate the data (test, train and validation csv files) in format as used in git repo USCDatascience/filetypeDetection. We have submitted our code that generate these csv files. After that, we ran "main.r" file in our R console with our generated csv files. We used 300 files from "application\_diff+xml", 100 files from "application\_xhtml+xml" for each of test, train and validation test. We choose these two mime type because of possible content similarity among these two and check accuracy of our model. We have submitted tika.model file that was generated with our code.

Below is a screen shot of a run on some sample data of type "application\_diff+xml" and "application\_xhtml+xml". We can see that, content detector based detector is correctly differentiating between xhtml+xml and diff+xml types. In fact, it simply returns octet stream for xhtml+xml as we did not train for that mime type. However, we can see that content based mime detector is wrongly classifying csv file as application/diff+xml.

C:\Windows\System32\cmd.exe

```
C:\stuff\Git\csci599-12\HW1\extra-credit\9>java -cp ".;tika-app-1.11.jar" contentBasedDetection
```

```
Tika detected with default detector: application/xhtml+xml
```

```
Tika detected with content based detector: application/octet-stream
Tika detected with default detector: application/xhtml+xml
```

```
Tika detected with default detector: application/xhtml+xml
Tika detected with content based detector: application/octet-stream
```

```
Tika detected with content based detector: application/octet-stream
Tika detected with default detector: application/xhtml+xml
```

```
Tika detected with content based detector: application/octet-stream
```

```
Tika detected with default detector: application/xhtml+xml
Tika detected with content based detector: application/octet-stream
```

```
Tika detected with content based detector: application/octet-stream
Tika detected with default detector: application/xhtml+xml
```

```
Tika detected with default detector: application/xhtml+xml
Tika detected with content based detector: application/octet-stream
```

```
Tika detected with content based detector: application/octet-stream
Tika detected with default detector: application/xhtml+xml
```

```
Tika detected with content based detector: application/octet-stream
```

```
Tika detected with default detector: application/xhtml+xml
Tika detected with content based detector: application/octet-stream
```

```
Tika detected with content based detector: application/octet-stream
Tika detected with default detector: application/xhtml+xml
```

```
Tika detected with default detector: application/xhtml+xml
Tika detected with content based detector: application/octet-stream
```

```
Tika detected with default detector: application/xhtml+xml
```

```
Tika detected with content based detector: application/octet-stream
Tika detected with default detectors: application/octet-stream
```

```
Tika detected with default detector: application/xhtml+xml
Tika detected with content based detector: application/octet-stream
```

```
Tika detected with content based detector: application/octet-stream
Tika detected with default detector: application/dif+xml
```

```
Tika detected with default detector: application/diff+xml
Tika detected with content based detector: application/diff+xml
```

```
Tika detected with default detector: application/dif+xml
```

```
Tika detected with content based detector: application/diff+xml
Tika detected with default detector: application/diff+xml
```

```
Tika detected with default detector: application/dif+xml
Tika detected with content based detector: application/dif+xml
```

```
Tika detected with content based detector: application/dif+xml
Tika detected with default detector: application/dif+xml
```

```
Tika detected with content based detector: application/diff+xml
```

```
Tika detected with default detector: application/dif+xml
```

```
Tika detected with content based detector: application/diff+xml
Tika detected with default detector: application/dif+xml
```

```
Tika detected with default detector: application/diff+xml
Tika detected with content based detector: application/diff+xml
```

```
Tika detected with content based detector: application/dif+xml
Tika detected with default detector: application/dif+xml
```

```
Tika detected with content based detector: application/diff+xml
```

```
Tika detected with default detector: text/csv
Tika detected with content based detector: application/diff+xml
```

```
Tika detected with content based detector: application/diff+xml
```

```
C:\stuff\Git\csci599-12\HW1\extra-credit\9>
```

0: 0001 1 000 000000 00 000 0000 0 0000 00