# University of Southern California

Viterbi School of Engineering

CSCI 599: Content Detection and Analysis for Big Data

**Instructor**: Dr. Chris Mattmann

Assignment 1: MIME Diversity in the Text Retrieval Conference (TREC) Polar Dynamic Domain Dataset

**TEAM 22**

**GitHub repository**: *http://www.github.com/harshfatepuria/data-analysis-test*

**Github.io website**: *http://harshfatepuria.github.io*

*(All the visualizations with interactive capabilities available on this website)*

Date: 03/03/2016

Report submitted by:

Harsh Fatepuria, fatepuri@usc.edu
Warut Roadrungwasinkul, roadrung@usc.edu
Rahul Agrawal, rahulagr@usc.edu
(Graduate Students, Department of Computer Science)

# I. File preparation

1. We created a D3 Pie Chart using existing JSON breakdown from Github.

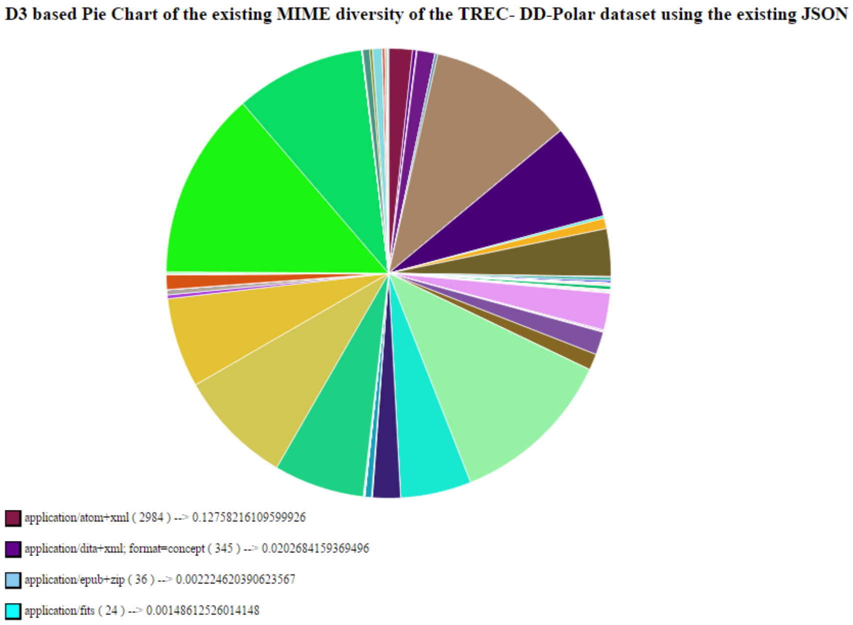
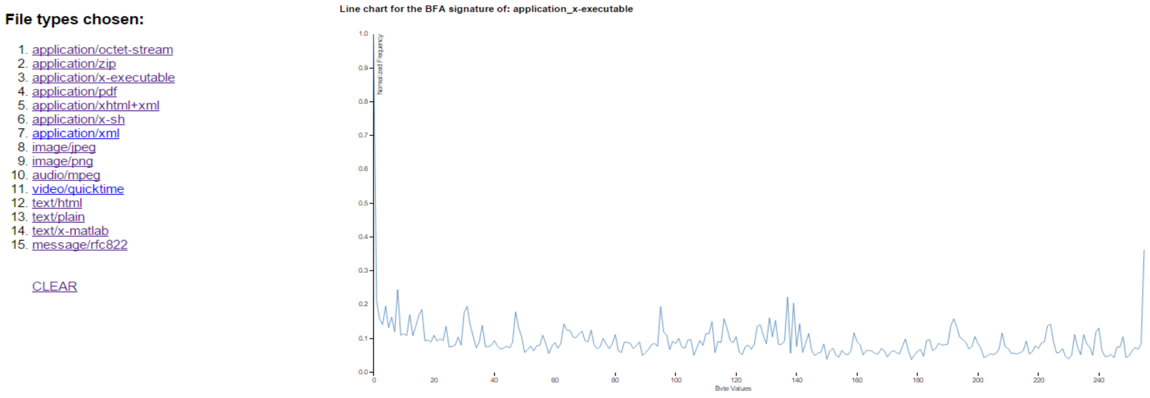


Fig 1: Pie Chart of MIME diversity of TREC-DD-Polar dataset (JSON from Github)

2. We did some file preparation to work with the dataset files easily.
   a. First, we used Apache Tika to detect file type, and then indexed file path by its MIME type in JSON format, by running a Java class **typedetect.runner.TypeDetectRunner**. This would help us when we would work on a specific file type.
   b. To do the analysis, we separated files from each type to be training samples and test samples by the ratio of 75% to 25% except for the type *application/octet-stream* which uses 50,000 files each as training and test set by using Java class **typedetect.runner.SeparateTestTrainDataRunner**.

# II. Byte Frequency Analysis on files of the selected 15 MIME types

1. We performed Byte Frequency Analysis on the chosen 15 MIME types (14 types + octet-stream). An automated script was written for this analysis.
2. The script generates 15 JSON files, one for each MIME type. The JSON files were used to generate D3 visualizations of the signatures. An example visualization is shown below.



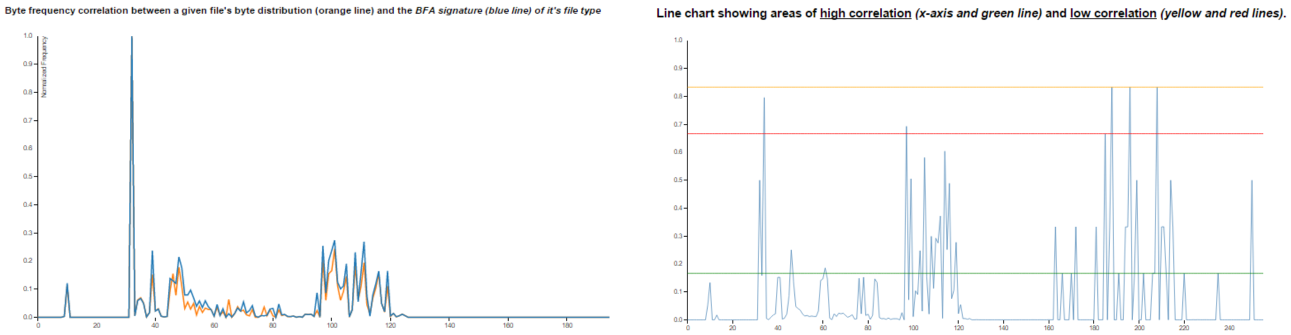
**File types chosen:**

1. application/octet-stream
2. application/zip
3. application/x-executable
4. application/pdf
5. application/xhtml+xml
6. application/x-sh
7. application/xml
8. image/jpeg
9. image/png
10. audio/mpeg
11. video/quicktime
12. text/html
13. text/plain
14. text/x-matlab
15. message/rfc822

Fig 2: Companded (μ-Law) BFC Signature of application/x-executable type

## III. Byte Frequency Correlation and Byte Frequency Cross Correlation

1. We developed a program which computes the Byte Frequency correlation between an input file and its MIME type's BFA Signature. The input to this program is a file type (as described in Tika MIME taxonomy) and a file path (of the file to be analyzed).
2. [Extra Work] For all the files present in the test data, we calculated a correlation coefficient (Pearson Correlation). We stored the results for all the file types in separate text files. For each of the files, Pearson Coefficient gives a value in range [-1, 1], 1 meaning that the files are highly correlated and -1 meaning that they are not correlated at all.
3. We then developed two D3 visualizations. A multi-line chart depicting BFA signature for the MIME type, and Byte Frequency Distribution of the input file. Another visualization depicts the difference between the two (signature and BFD of a file) and shows areas of high and low correlation.


Byte frequency correlation between a given file's byte distribution (orange line) and the *BFA signature (blue line) of it's file type*

Line chart showing areas of high correlation *(x-axis and green line)* and low correlation *(yellow and red lines).*

Fig 3: (Left) Multi line chart b/w BFA Signature (Blue Line) and BFD of File (Orange Line); (Right) Graph shows BFC (difference between BFA Signature and BFD of file). Also shows areas of high and Low correlation (Low Correlation= high difference, high Correlation= low difference)

4. We then performed BFC cross correlation on the test data to generate cross correlation matrix for the 15 MIME types. Also, we generated a D3 heat-map visualizing the same. Link to visualization: http://harshfatepuria.github.io/D3_Cross_Correlation_Signature_HeatMap.html

## IV. File Header trailer Analysis (First 4,8 and 16 bytes)

1. We developed a program which computes FHT analysis on the first 4, 8 and 16 bytes of all the files in the training set for all the 15 chosen MIME types using the training data.
2. We then generated a FHT heat map of size 256 x 16 which depict the overall data distribution of first 16 bytes of all the 15 chosen MIME types.

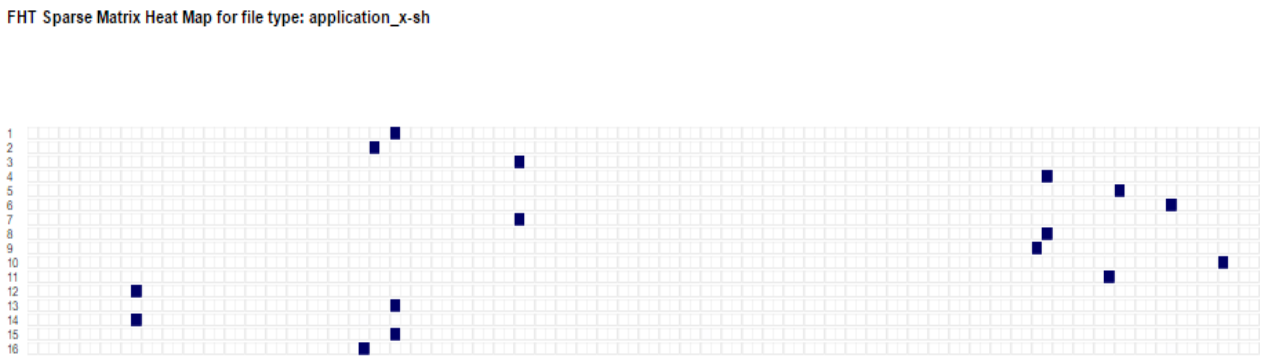

FHT Sparse Matrix Heat Map for file type: application_x-sh

Fig 4: FHT heat map of application/x-sh type

3. [Extra Work] We calculated the Correlation Coefficient for the first 4, 8 and 16 header bytes for each file in the test data compared to FHT Signature procured in the previous step. The Analysis shows some interesting observations. For example, the first 16 bytes in all the files (in test set) of application/xml , and the first 8 bytes of application/x-sh match their respective FHT Signatures 100%, i.e. they all have the same exact Magic Bytes. Visualization: http://harshfatepuria.github.io/D3_FHT_Signature_heatMap.html

## V. Important Observations and Updation of Tika MIME repository based on above analyses

1. We chose a set of 15 MIME types for our analysis because of the following reasons:
   a. These form the most diverse set from the given MIME types. Overall, our chosen types cover Messages, Applications, Audio, Video and Image files apart from scripts etc.
   b. These types contain the large amount of data such that our analysis covers almost 800,000 files.

2. We found out the first 16 bytes of all the non-empty files (total files=323) classified as application/octet-stream. This information can be found in ‘octet_stream_first_16_Bytes.txt’. We did some research on this data, and as per our BFA and FHT analysis, we can modify the tika-mimetypes.xml to detect more file types. **We found new MIME magic byte fingerprints for the following**:

   *a. application/vnd.ms-cab-compressed*

```
<magic priority=“40">
        <match value="MSCF" type="string" offset="0"/>
</magic>
```

   *b. text/html*

```
<match value="&lt;script" type="string" offset="0"/>
<match value="&lt;SCRIPT" type="string" offset="0"/>
```

   c. We found a new MIME type *‘application/x-git-index’*, which is a Git Index file. We believe this file can have a classification in Apache Tika

```
<mime-type type="application/x-git-index">
        <_comment>Git index file</_comment>
        <magic priority=“40">
                <match value="DIRC" type="string" offset="0"/>
        </magic>
</mime-type>
```

**Key Observations**:

1. Some files did not have the entire content (truncated unusually), and this might be the reason for Tika not detecting these files correctly.

2. Some Magic Bytes were missing from text/html which led to these files being classified as text/plain or application/octet_stream. We tried to fix this issue by adding new magic bytes.

3. We also found some new MIME types, results for which have been indicated above (V.2). These new MIME types were added to the tika-mimetypes.xml and dataset was analyzed again. The relevant files were now correctly detected as CAB, HTML or GIT Index files. Visualizations for the previous run of Tika (On unchanged MIME type file) v/s the new run on Tika (on changed MIME types file) are available at: http://harshfatepuria.github.io/completeDataPieChart_downloadedFiles.html and http://harshfatepuria.github.io/completeDataPieChart_usingModifiedMIME.html

4. While adding the new magic bytes for text/html we tried it for offsets “0” and “0:64”. It turns out that with offset “0”, we get a better classification of files, whereas for offset “0:64”, some non html files were getting classified as text/html.

5. D3 visualizations helped us a lot to identify byte patterns from files for various forms of analyses, especially finding the magic bytes using FHT heat map.

6. Overall, we found that Tika is easy to use, and classifies most of the files correctly. Some new MIME types can be added to classify the undetected/ wrongly-detected types.

7. [Tika v/s Mac] We analyzed the non-empty files in application/octet-stream classification, and checked their classification in Apple Mac OS X 10.11.3 using the command: *file --mime-type -b <file_name>*

   It turns out that some files were classified differently than Apache Tika:

| **Files detected as application/octet-stream by Tika** | **MIME type detected by Apple Mac OS X 10.11.3** | **If one of our 15 chosen types, the Pearson correlation coefficient for BFA** |
|---|---|---|
| cn/sh/library/dlpwd/E1EA0A0DD0B36651C58A0E3C8E14F30B42638D7D110533E8018DDC06A71F9A87 | text/plain | 0.68 |
| org/w3/www/273EA7C3C3E22540167271B06B9A86EC8F07AF55728E9A8FF33C2FCAB4353CFF | text/x-po | |
| org/translationproject/FC7CCC7E7DCA55833B393D54CDA1A6AAAB2290ECD5986BF2B1719B601463D12F | text/x-po | |
| org/translationproject/C0419E6BC6518647A59573AEB6BE07684BC50C08279E5E1E437844C4A321A9C8 | text/x-po | |
| org/translationproject/BD0CAE3F6DC334998AB3B424380C9756A9957892F36338BA6161E7E900D34751 | text/x-po | |
| org/translationproject/A4138B5F803E0E4B25A70B5FA1D2C73B1E557DC3C6FD547140252FF3E0934AF9 | text/x-po | |
| net/cnki/wuxizazhi/0229541219032066874F666EAE7D16A7DA6F5965EDB9C2F71B80CCE8845E20EC | text/html | 0.69 |
| jp/ac/tsukuba/www/7ECEA6E465699A482F9C50BFFD3C2DAA267C1076572D75BB765E2D6AC63BB2AF | text/x-c | |
| jp/ac/tsukuba/www/43C3D961B7626D9C0930AD61322FFEAD95B7FC65103C9D908288FF1770A1F05B | text/x-c | |
| gov/noaa/arh/aprfc/62464DACBE1BA9486210FA7B09FB868FABA84084C8977826276336051058D9B0 | text/plain | 0.04 |
| gov/nasa/jpl/mls/9B81C7A56AC3C6E6836EF97D9A06463EB277FB37E28227E1966F806BB7F34B25 | text/plain | 0.81 |
| gov/nasa/jpl/mls/80BC2238611041B040B057801EA25E43FC517C9D014B6B61E9EA2D5F7F683914 | text/plain | 0.68 |
| gov/nasa/jpl/mls/392A51636B31FAF5CC1F13D19F66DCE65A5881354BD88DD9872A9150111B4C0C | text/plain | 0.68 |
| gov/nasa/jpl/mls/0D38EC54631D967C0371C54108DFE6860A04AB26DD6CD8350DEF094F4051B40C | text/plain | 0.69 |
| com/accuweather/vwidget/17EB685A4E92D69041CB1BCA6CD77DC14502D7268F26AE2D5553E1A6FD119F7C | application/vnd.ms-cab-compressed | |
| cn/sh/library/www/04B0FA20E767A6ED01E63132395CA26A8BA5D243928124BB0CB4B6100D07890A | text/html | 0.72 |
| cn/sh/library/www/E1BC0070D8B0768AC5FC9E61738EB2441ADBB61C0C10298C49DD35494AC5ED98 | text/html | 0.82 |
| cn/sh/library/www/BE408B3FD9BFD5C5D8F0FB3FCE48CEF7DEB1369D3F05F400309109521ABB3397 | text/html | 0.86 |
| cn/gov/nstl/www/8D8C5E9A1E377E6F3992246F573340D9B424CFB8E9ED38538568EB2A59980ABF | text/html | 0.54 |

Table 1: MIME detection difference b/w Tika and Mac OS X MIME detector

8. We found the correlation coefficient in comparing the test files (25%) with BFA signatures and FHT signatures. The following table shows the best method to identify the files of respective types:

| **MIME type** | **Best method to identify** |
|---|---|
| application/zip | FHT(4 bytes) |
| application/x-executable | FHT(16 bytes) |
| application/pdf | FHT(8 bytes) |
| application/xhtml+xml | BFC |
| application/x-sh | FHT(8 bytes) |
| application/xml | FHT(16 bytes) |

| | |
|---|---|
| image/jpeg | FHT(16 bytes) |
| image/png | FHT(16 bytes) |
| audio/mpeg | FHT(8 bytes) |
| video/quicktime | FHT(4 bytes) |
| text/html | BFC |
| text/plain | BFC |
| text/x-matlab | BFC |
| message/rfc822 | BFC |

Table 2: FHT v/s BFC- Best ways to classify MIME types

## VI. [Extra Credit] Tika Similarity using Cosine Distance and Edit Distance

A workflow of clustering and visualization using Tika Similarity is as follows:

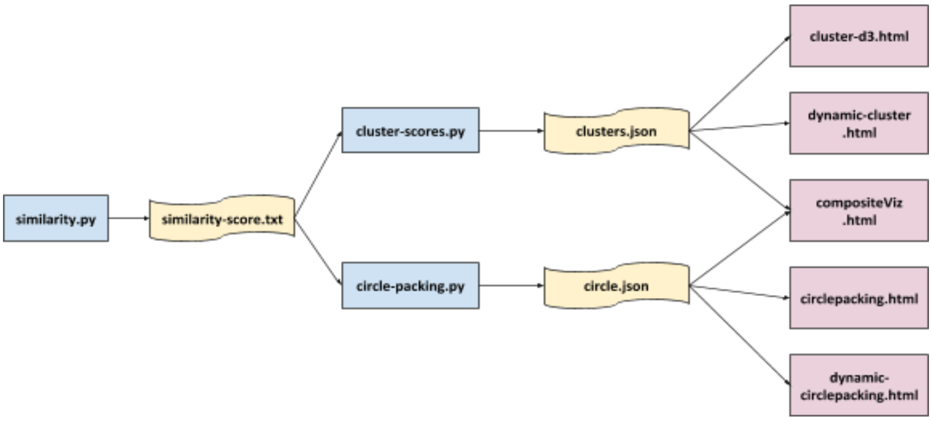


Fig 5: Workflow of clustering and visualization using Tika Similarity

1. By default, Tika Similarity uses Jaccard similarity. However, there is no such concept using Cosine similarity and Edit distance. There is an implementation of K-means clustering already in the project which can be modified to suit our needs.
2. K-means clustering uses Euclidean distance (existing) and outputs the clusters.json file. It transforms each file to a feature vector and calculates distance between each two vectors during the clustering process. We do the following to complete the modification.
   a. Add a function to calculate Edit distance between two vectors. Also modify the feature structure to contain enough data to calculate the distance.
   b. Modify the code that does K-mean clustering so we can specify which distance measure to use. Also modify centroid selection when using distance measure other than Euclidean.
   c. Add a script to create circle.json by taking clusters.json (output from 2.b) as an input.

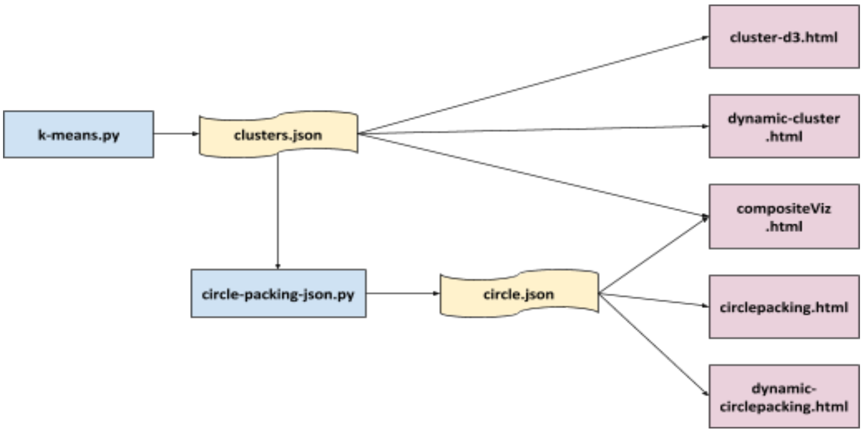


Fig 6: Workflow of clustering and visualization after adding Cosine and Edit Distance

Analysis on clustering was done using 2 datasets, a smaller one containing about 50 files and the larger containing about 500 files. Screenshots of clusters and circle packing of smaller dataset are as follows:

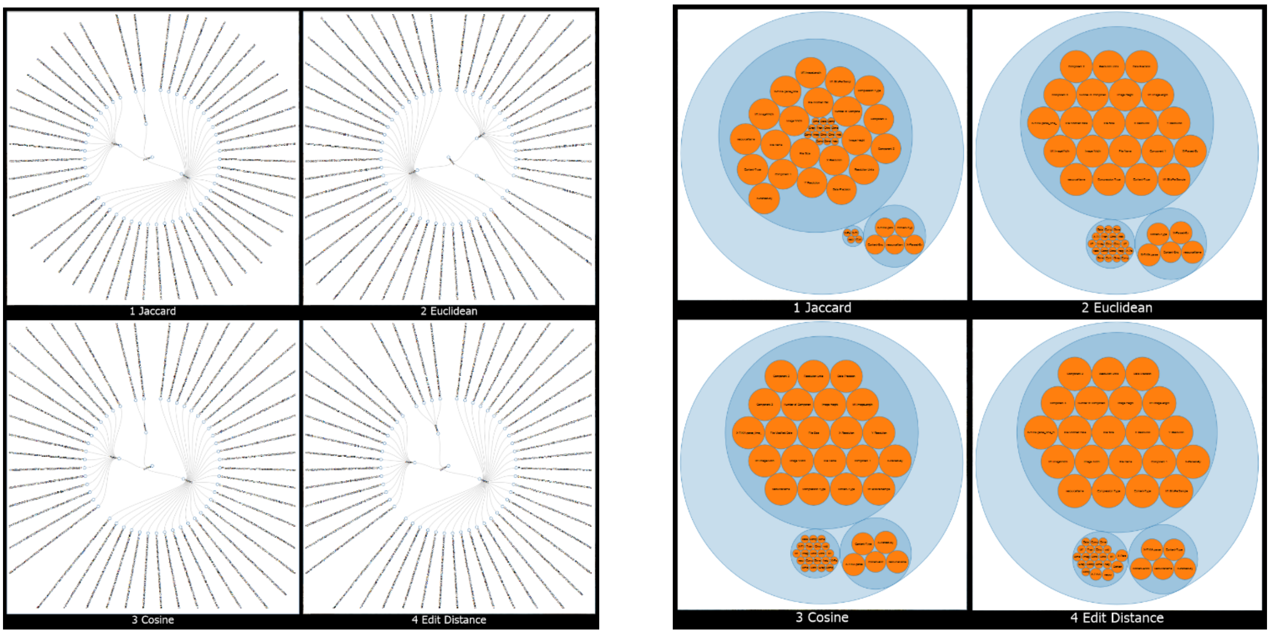


Fig 7: Clustering and Circle Packing of smaller dataset

**<u>Some Observations</u>:**

1. Jaccard similarity resemblance value of each file is calculated from metadata key, so a file type that has the same metadata key should produce similar resemblance, thus will be in the same cluster. Clustering using Euclidean and Cosine distance should be quite the same because both of them use the length of metadata values as features. Edit distance use actual metadata values so the cluster might be different.
2. If we consider the type detected from Tika as each file's actual type and try to classify each file type in each cluster to be the same type as its cluster majority. For smaller dataset we can see that each cluster indicates its type neatly. For larger dataset, although it is not as neat as in smaller dataset but it still resembles the type. The detailed results are as follows.

| **Smaller Dataset ("/com/ytimg")** | | | | |
|---|---|---|---|---|
| **Jaccard** | | | | |
| **Cluster** | **Type** | **Majority** | **Size** | **Accuracy** |
| cluster0 | text/plain | 13 | 15 | 86.66667 |
| cluster1 | image/gif | 2 | 2 | 100 |
| cluster2 | image/jpeg | 31 | 31 | 100 |
| **Overall** | | **46** | **48** | **95.83333** |
| **Euclidean** | | | | |
| **Cluster** | **Type** | **Majority** | **Size** | **Accuracy** |
| cluster0 | text/plain | 13 | 15 | 86.66667 |
| cluster1 | image/gif | 2 | 2 | 100 |
| cluster2 | image/jpeg | 31 | 31 | 100 |
| **Overall** | | **46** | **48** | **95.83333** |
| | | | | |
| **Cosine** | | | | |
| **Cluster** | **Type** | **Majority** | **Size** | **Accuracy** |
| cluster0 | image/jpeg | 31 | 31 | 100 |
| cluster1 | text/plain | 13 | 15 | 86.66667 |

| **Larger Dataset ("/info")** | | | | |
|---|---|---|---|---|
| **Jaccard** | | | | |
| **Cluster** | **Type** | **Majority** | **Size** | **Accuracy** |
| cluster0 | application/xhtml+xml | 1 | 1 | 100 |
| cluster1 | image/gif | 9 | 10 | 90 |
| cluster2 | application/xhtml+xml | 288 | 521 | 55.27831 |
| **Overall** | | **298** | **532** | **56.01504** |
| **Euclidean** | | | | |
| **Cluster** | **Type** | **Majority** | **Size** | **Accuracy** |
| cluster0 | application/xhtml+xml | 7 | 8 | 87.5 |
| cluster1 | application/xhtml+xml | 218 | 253 | 86.16601 |
| cluster2 | text/html | 29 | 44 | 65.90909 |
| cluster3 | text/html | 97 | 227 | 42.73128 |
| **Overall** | | **351** | **532** | **65.97744** |
| **Cosine** | | | | |
| **Cluster** | **Type** | **Majority** | **Size** | **Accuracy** |
| cluster0 | text/html | 45 | 85 | 52.94118 |
| cluster1 | application/xhtml+xml | 226 | 262 | 86.25954 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| cluster2 | image/gif | 2 | 2 | 100 | cluster2 | text/html | 81 | 185 | 43.78378 |
| **Overall** | | **46** | **48** | **95.83333** | **Overall** | | **352** | **532** | **66.16541** |
| **Edit Distance** | | | | | **Edit Distance** | | | | |
| **Cluster** | **Type** | **Majority** | **Size** | **Accuracy** | **Cluster** | **Type** | **Majority** | **Size** | **Accuracy** |
| cluster0 | image/jpeg | 31 | 31 | 100 | cluster0 | text/html | 62 | 135 | 45.92593 |
| cluster1 | text/plain | 13 | 13 | 100 | cluster1 | text/html | 99 | 192 | 51.5625 |
| cluster2 | image/vnd.microsoft.icon | 2 | 4 | 50 | cluster2 | application/xhtml+xml | 199 | 205 | 97.07317 |
| **Overall** | | **46** | **48** | **95.83333** | **Overall** | | **360** | **532** | **67.66917** |

Table 3: Accuracy in Tika Similarity using various distance measures (Edit Distance gives highest accuracy in the larger dataset)

## VII. [Extra Credit] Content Based MIME Detector

1. In “filetypeDetection” project, the author used R scripts to train a neural network model to classify whether a file is from a specific type or not by using its byte frequency distribution. Classes to read this model and do type prediction based on the model are also implemented in Tika.
2. To train a model, first we prepare the training dataset. We built the model to classify "application/xhtml+xml" type. We select 50,000 files of this type to be positive examples. We also select 10,000 files from each other 5 file type ("application/pdf", "image/jpeg", "image/gif", "text/html", "text/plain") to be negative examples. The validation and test dataset are built the same way.
3. Byte frequency distribution of each file is calculated as feature vector and also labelled positive or negative according to its type. We feed this data to our modified R script and a trained neural network model. We then feed the model into Tika (using *org.apache.tika.detect.NNExampleModelDetector*) and compare the result against Tika default MIME detector.
4. The accuracy of training, validation and test dataset are 95.645, 88.033 and 84.027 percent respectively which conform to the result of training script. Detailed result is as follows:

| **Train Dataset** | | **Actual** | | **Accuracy (%)** |
|---|---|---|---|---|
| | | **application/xhtml+xml** | **others** | |
| **Predicted** | **application/xhtml+xml** | 48567 | 2922 | 97.134 |
| | **others** | 1433 | 47078 | 94.156 |
| | | | | 95.645 |
| **Validate Dataset** | | **Actual** | | **Accuracy (%)** |
| | | **application/xhtml+xml** | **others** | |
| **Predicted** | **application/xhtml+xml** | 45610 | 7577 | 91.22 |
| | **others** | 4390 | 42423 | 84.846 |
| | | | | 88.033 |
| **Test Dataset** | | **Actual** | | **Accuracy (%)** |
| | | **application/xhtml+xml** | **others** | |
| **Predicted** | **application/xhtml+xml** | 41176 | 7149 | 82.352 |
| | **others** | 8824 | 42851 | 85.702 |
| | | | | 84.027 |

Table 4: Content based MIME Detector