**University of Southern California**

Viterbi School of Engineering

CSCI 599: Content Detection and Analysis for Big Data

**Instructor**:Dr. Chris Mattmann

Assignment 2: Evaluating the Content Analysis Text Retrieval Conference (TREC)

Polar Dynamic Domain Dataset

**TEAM 22**

**GitHub repository**: *https://github.com/harshfatepuria/Evaluation-of-Content-Analysis-on-TREC-Polat-DD-Dataset*

**Github.io website**: [*http://harshfatepuria.github.io*](http://harshfatepuria.github.io)

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

Date: 05/02/2016

Report submitted by:

Harsh Fatepuria, *fatepuri@usc.edu*

Warut Roadrungwasinkul, *roadrung@usc.edu*

Rahul Agrawal, *rahulagr@usc.edu*

(Graduate Students, Department of Computer Science)

1. **Common Codes for all the following Parsers**
2. **TikaExtractedTextBasedParser**: An abstract class extended from Tika AbstractParser. Parsers that perform on text extracted from documents using Tika should extend this class. This class also provide a method to get extracted text from a document. Most of the parsers created in this assignment extend this class.
3. **AbstractParserRunner**: A utility class to run a specific parser using all the documents in a specified folder. It can be configured through different parameters. Start parsing by invoking runParser() method.

|  |  |
| --- | --- |
| baseFolder | base folder of documents to be parsed |
| resultFolder | base folder to keep result files |
| markerFile | marker, for skipping the files that are already been parsed |
| overwriteResult | set to overwrite existing results (default : false) |
| documentsInCborFormat | setting that tell whether documents to be parsed are in CBOR Format, for parsing data in common-crawl (default : false). CBOR support can be used separately using cbor.CborReader |

1. **Classification path from Request to Content**
2. Categories of pages part of the request
3. What named entities were present on the arrived at page
4. Add D3
5. Did the crawler find the most relevant webpages? Why why not?
6. Sample Code snippet/ JSON ?? can be removed.

{

}

Snippet 1: Sample code snippet

1. **File size diversity of Common Crawl (CCA) dataset by MIME type**
2. The file size ratio of the Solr Index and the original file sizes were computed and averaged over each MIME type.
3. The D3 visualization depicting the same is shown below: Add D3
4. A sample JSON using which the above D3 was built is shown below: // or code snippet or the formula// or high level algorithm can be added here.

{

}

Snippet 2: Example JSON showing

1. **Parser Call Chain**
2. Apache Tika was run over all the files of the CCA dataset. During the run, various parsers are called as depicted in the visualizations below. Minor changes were made in the Tika code to store the parser information at each level, and their counts were used to draw conclusions.
3. The amount of Text retrieved per MIME type : Add D3
4. Amount of Metadata retrieved per MIME type: Add D3
5. **Language Identification and diversity**
6. Apache Tika was run over all the files of the CCA dataset. NLTK, the Natural Language Tool-Kit was used to obtain the languages of the content in each file. The language diversity of the dataset is shown below: //Add D3
7. Sample code snippet showing how NLTK was used: //maybe??

**java** …

1. **Maximal occurring topics in the dataset**
2. After running various parsers built during the course of the project, including measurement parser, SWEET ontology parser, and other metadata extractors, a set of maximally occurring words were obtained for each of the parsers. The results are combined and presented in the form of one Word Cloud below:
3. //Add D3 word cloud
4. **Named Entity Recognition toolkits:**
5. NLTK, CoreNLP, openNLP and Grobid Quantities are toolkits used to extract named entities for the data. While NLTK, CoreNLP and openNLP are more generic extractors, Grobid Quantities is a more specific measurement quantities recognizer which can extract measurement related data like ’25 centimetre’.
6. Each of the above parsers were used to parse the Polar Dynamic Domain full-dump dataset, and the counts fo the various extracted entities were stored separately.
7. [Extra Credit] Grobid Quantities:
8. A Tika NER implementation that invokes Grobid Quantities via its REST service was developed. A pull request to merge the implementation is submitted.
9. The NER wiki for Tika was updated with description of how to use Grobid Quantities.
10. An algorithm was developed to compute the maximal join agreement between the four algorithms. The algorithm is as follows:

**Algorithm for Maximal Join agreement..**

**…**

1. The above algorithm was encoded as a CompositeNERAgreementParser in Tika. A pull request for adding the above parser in Tika has been submitted.
2. A D3 visualization for evaluating the maximal joint agreement NER between the most frequently occurring entities is shown below:
3. Part g // discuss: Analyze whether your new joint agreement produces any update NER for your metadata records, and if so, add new maximal joint agreement NER to the metadata records in Solr
4. **Spectrum of Measurements**
5. A program was developed which takes the various measurement extractor entities- developed as a part of Assignment 2, and the measurements extracted using Grobid Quantities and generates JSON in three formats:
6. For Max, Min and Average for all the measurements. Add D3..
7. For all measurements clustered based on domain: A dictionary of high level domain and related measurement units was manually and meticulously curated.

A sample entry in dictionary looks like:

{

"TEMPERATURE":["kelvin", "celsius", "celcius", "fahrenheit", "degreeC", “degreeF” … ]

}

Using this mapping, the extracted measurements and their counts were sorted into bins as shown in the visualization below:

//Add D3

1. For all measurements clustered based on the MIME types: A script was developed to map each file in the dataset to its MIME type. This was generated using a list generated as a part of Assignment 1 which maps each MIME type to the list of file-paths it refers to. Then the measurements extracted for each files were clustered based on the MIME type of that file. A Sample JSON for the D3 visualization is shown below:

{

“name”: Measurement Range Clustered on MIME Type”,

"children": [{

“name”: “application/x-sh”,

"children": [

{"name": "hour", "size": 3},

{"name": "kelvin", "size": 2},

{"name": "month", "size": 1}

…

} … ]

…

}

Snippet 5: Solr Index. MEMEX GeoParser used to generate the world map shown above

//Add D3 clustered on MIME Type

1. **Contributions to** [**http://www.polar.usc.edu**](http://www.polar.usc.edu) **and Video Demonstration:**
2. We have previously contributed our visualizations to polar.usc.edu in both, Assignment 1 and Assignment 2. We have also added a pull request for the visualizations developed as a part of this assignment.
3. Video demonstrating a screencast walk-through to the tasks is available at: ……..
4. **Important Observations**:

Write a short 4 page report describing your observations, and answer the following questions, using your visualizations, and knowledge gained through the semester:

1. Is your MIME detection good? Define “good”.

2. Are your parsers extracting the right text? Define “right”.

3. Are we selecting the right parser? Define “right”.

4. Is your Metadata appropriate? What’s missing? You can use your Metadata score generated from assignment #2 here, and also your results from this assignment.

5. How well is my language detection performing? Comment based on the diversity of the languages derived in this assignment. Are there mixed languages? Did it affect your accuracy?

6. Do your Named Entities make sense?