PROJECT OVERVIEW
Problem Statement

Build an Information and Retrieval System that will act as a search engine to support ranking, searching, browsing and recommendations for two large collections of data:

- 14M Tobacco Settlement Documents
- 30K ETDs Electronic Theses and Dissertations
Requirements for Elasticsearch

- Ingest data provided by the CME and CMT teams into Elasticsearch in the correct format.
- Decide the relevancy and importance of fields related to the ETD and tobacco dataset and provide feedback on the same.
- Incorporate additional data from TML team related to text summarisation, name entity recognition, sentiment analysis, and clustering information.
- For enhanced search accuracy, perform boosting to assign higher weights to important fields.
- Implement nested queries for in-depth search inside each document.
- Establish connection with Kibana to support searching, browsing and information visualisation.
- Implement automatic ingesting and updating scripts to monitor a designated directory on ceph for new incoming files.
Contribution to Other Teams

- CME
  - Studied the metadata and full text provided
  - Prepared the data schema
  - Ingested into Elasticsearch

- CMT
  - Studied the metadata and full text provided
  - Prepared the data schema
  - Ingested into Elasticsearch

- FEK
  - Provided the data schema, data type and usage recommendation
  - Provided different search query examples
  - Helped establish connection with Kibana

- TML
  - Designed expected format for text summarization, cluster information, sentiment analysis and NER data
  - Tested a subset of text summarization data ingested into Elasticsearch
Achievements

CME
- 99.8%
- 30,925 Electronic Thesis Documents ingested including metadata and full text.
- Fully searchable documents
- Can be filtered and sorted.
- Prepared automated script for addition of new documents

CMT
- 99.9%
- 5,595,936 Tobacco Settlement Documents metadata ingested (81 failed); including 100,000 metadata and full text.
- Fully searchable documents
- Can be filtered and sorted.
- Prepared automated script for addition of new documents.

TML
- Text Summarization, Sentiment Analysis, Named-Entity Recognition, Cluster Data
- Work in progress.
- Tested the text summarisation format.
- Receiving data from TML.
- Work in progress.
DESIGN & IMPLEMENTATION
Concept Map for Elasticsearch

Elasticsearch

- Index name, schema and queries
- User activity log
- Logs, indexed data
- Cluster result, text summarization, NER and sentiment analysis
- Loaded into Elasticsearch database and indexed accordingly
- Metadata and data
- Metadata and data
- Inserted into CEPH

Kibana

- Support every phase
- Pass on the feedback received from the TML team

TML

- Pass on the feedback received from the TML team

ETD (Mongo DB)

Tobacco Settlements (MySQL)
## Tobacco Data Schema

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Elasticsearch Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access</td>
<td>URL/String</td>
<td>Text</td>
</tr>
<tr>
<td>Adverseruling</td>
<td>String-(Alphanumeric)</td>
<td>Text</td>
</tr>
<tr>
<td>Area</td>
<td>String</td>
<td>Text</td>
</tr>
<tr>
<td>attending</td>
<td>List&lt;String&gt;</td>
<td>Text</td>
</tr>
<tr>
<td>Author</td>
<td>List&lt;string&gt;</td>
<td>Text</td>
</tr>
<tr>
<td>availability</td>
<td>String</td>
<td>Text</td>
</tr>
<tr>
<td>Bates</td>
<td>ID-Alphanumeric</td>
<td>Text</td>
</tr>
<tr>
<td>Batesalternate</td>
<td>ID-Alphanumeric</td>
<td>Text</td>
</tr>
<tr>
<td>batesmaster</td>
<td>List&lt;ID&gt;-Alphanumeric</td>
<td>Text</td>
</tr>
<tr>
<td>Box</td>
<td>Number</td>
<td>Numeric</td>
</tr>
<tr>
<td>Brand</td>
<td>String</td>
<td>Text</td>
</tr>
<tr>
<td>Case</td>
<td>ID-Alphanumeric</td>
<td>Text</td>
</tr>
<tr>
<td>Cited</td>
<td>String</td>
<td>Text</td>
</tr>
<tr>
<td>Collection</td>
<td>String</td>
<td>Text</td>
</tr>
<tr>
<td>Copied</td>
<td>List&lt;String&gt;</td>
<td>Text</td>
</tr>
<tr>
<td>Country</td>
<td>String</td>
<td>Text</td>
</tr>
</tbody>
</table>
## ETD Data Schema

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Elasticsearch Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>dc.contributor.author</td>
<td>String</td>
<td>Text</td>
</tr>
<tr>
<td>dc.date.accessioned</td>
<td>Date/Time</td>
<td>Date</td>
</tr>
<tr>
<td>dc.date.available</td>
<td>Date/Time</td>
<td>Date</td>
</tr>
<tr>
<td>dc.date.issued</td>
<td>Date/Time</td>
<td>Date</td>
</tr>
<tr>
<td>dc.identifier.other</td>
<td>String-(Alphanumeric)</td>
<td>Text</td>
</tr>
<tr>
<td>dc.identifier.uri</td>
<td>URL</td>
<td>Text</td>
</tr>
<tr>
<td>dc.description.abstract</td>
<td>String-(Alphanumeric)</td>
<td>Text</td>
</tr>
<tr>
<td>dc.format.medium</td>
<td>String</td>
<td>Text</td>
</tr>
<tr>
<td>dc.publisher</td>
<td>String</td>
<td>Text</td>
</tr>
<tr>
<td>dc.rights</td>
<td>String</td>
<td>Text</td>
</tr>
<tr>
<td>Field Name</td>
<td>Field Type</td>
<td>Field Demo</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Case</td>
<td>text</td>
<td>Minnesota v. Philip Morris Inc.</td>
</tr>
<tr>
<td>Brands</td>
<td>text</td>
<td>Marlboro</td>
</tr>
<tr>
<td>Witness_Name</td>
<td>text</td>
<td>&quot;Wyant, Timothy (affiliation: Decipher; expertise: Statistical analysis; job_title:</td>
</tr>
<tr>
<td>Topic</td>
<td>text</td>
<td>advertising; health effects</td>
</tr>
<tr>
<td>Person_Mentioned</td>
<td>text</td>
<td>Burns, David Michael, M.D</td>
</tr>
<tr>
<td>Organization_Mentioned</td>
<td>text</td>
<td>R.J. Reynolds Tobacco Co.</td>
</tr>
<tr>
<td>Description</td>
<td>text</td>
<td>&quot;The plaintiffs expert witness, a statistician, was deposed&quot;</td>
</tr>
<tr>
<td>Title</td>
<td>text</td>
<td>&quot;Deposition of TIMOTHY S. WYANT, Ph.D., August 19, 1997</td>
</tr>
<tr>
<td>Date_Added_UCSF</td>
<td>text</td>
<td>20 January 2006</td>
</tr>
<tr>
<td>Document_Date</td>
<td>text</td>
<td>19 August 1997</td>
</tr>
<tr>
<td>Cluster</td>
<td>text/keyword</td>
<td>321</td>
</tr>
<tr>
<td>page</td>
<td>text/keyword</td>
<td>5</td>
</tr>
<tr>
<td>content</td>
<td>text/keyword</td>
<td>Paper details</td>
</tr>
</tbody>
</table>
ETD Data Structure

Searching
- Degree level (Apply Filter)
  - Master’s Thesis
    - Department
    - Title
    - Abstract
  - Doctoral Thesis
    - Author
    - Committee Chair
    - Committee Member

Sorting
- Match Query
- Bool Query
  - AND (Must)
  - OR (Should)
  - NOT (Should not)
  - Relevance
    - Based on Score
  - Date issued
    - Ascending
    - Descending
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Field Type</th>
<th>Field Demo</th>
</tr>
</thead>
<tbody>
<tr>
<td>degree-level</td>
<td>text</td>
<td>masters</td>
</tr>
<tr>
<td>contributor-department</td>
<td>text</td>
<td>Computer science</td>
</tr>
<tr>
<td>contributor-author</td>
<td>text</td>
<td>Tony Stark</td>
</tr>
<tr>
<td>Contributor-committee chair</td>
<td>text</td>
<td>John wick</td>
</tr>
<tr>
<td>Contributor-committee co-chair</td>
<td>text</td>
<td>Chris scott</td>
</tr>
<tr>
<td>Contributor-committee member</td>
<td>text</td>
<td>David knight</td>
</tr>
<tr>
<td>date-available</td>
<td>date</td>
<td>2017-01-23</td>
</tr>
<tr>
<td>date-issued</td>
<td>date</td>
<td>2018-02-21</td>
</tr>
<tr>
<td>degree-name</td>
<td>text</td>
<td>MS or P.hD</td>
</tr>
<tr>
<td>description-abstract</td>
<td>text</td>
<td>This field conveys the abstract of the thesis in 10-15 lines</td>
</tr>
<tr>
<td>Author Email</td>
<td>text</td>
<td><a href="mailto:tony_s@stark.com">tony_s@stark.com</a></td>
</tr>
<tr>
<td>subject-none</td>
<td>text</td>
<td>Soils -- Aluminum content Cations</td>
</tr>
<tr>
<td>title-none</td>
<td>text</td>
<td>Hydrolysis of aluminum in synthetic cation exchange</td>
</tr>
<tr>
<td>type-none</td>
<td>text</td>
<td>Dissertation</td>
</tr>
</tbody>
</table>

Fields for Searching and Filtering:

ETDs

For all field types of ‘Text’, use field_name for searching and field_name.keyword for filtering or sorting
Indexing Methods

**Metadata**
Stores the records detail that describes and gives information about the source data.

**Data**
Stores the text content of the ETD and tobacco settlement datasets (page-wise).

**Generated Data**
Data generated by the TML team consists of cluster ID, text summary, sentiment analysis and NER keywords.

---

```
# Ingestion of documents using curl command
curl -H 'Content-Type: application/x-ndjson' -XPOST '10.43.38.7:9200/t_fixed/doc/_bulk?pretty'
--data-binary @tobacco_data.json

# Ingestion for ETD documents
res = es.index(index = '30k', id = doc['identifier-uri'], body = doc)
    OR

# Ingestion for tobacco settlement documents
for lineNum in range(0, len(JSONDocs), numLines):
    res = es.bulk(body = '\n'.join(JSONDocs[lineNum:lineNum + numLines]))
```

Executable python script on ceph in els directory.
Ingesting by Elasticsearch-Python Client

Parsing files into designed format for ingesting

Assign the ID and the name of index

Logging errors (document ID and error messages)
Searching Query

```
GET /tobacco/_search
{
  "query": {
    "term": {"availability": "restricted"}
  }
}
```

```
GET tobacco/_search
{
  "query": {
    "query": {
      "multi_match": {
        "query": "Information Retrieval",
        "fields": ["Title", "summary"]
      }
    },
    "must_not": {
      "range": {
        "Document_Date": {
          "gte": "1910",
          "lte": "1980"
        }
      }
    },
    "should": [
      {"term": {"Title": "Study"}}
    ],
    "minimum_should_match": 1,
    "boost": 1.0
  }
}
```
Full Text Search: Nested Query

Tobacco Doc 1:

Full Text content:

Chapter/Page 1

Chapter/Page 1

Chapter/Page 2

Chapter/Page 2

Chapter/Page 3

Chapter/Page 3
Search Preference: Boosting

Elasticsearch rank searching results based on a designed score.

The scores are calculated by a similarity model based on Term Frequency (TF) and Inverse Document Frequency (IDF) as well as using the Vector Space Model (VSM) for multi-term queries.
Search Preference: Boosting

Field 1, with no boost
Field 1
Field 2, with boost weight = 2
Field 2
Field 3, with boost weight = 0.5
Field 3

Score = field_1 + 2 * field_2 + 0.5 * field_3

{ETD Doc 1: field_1: A,
  field_2: None,
  field_3: None}
{ETD Doc 2: field_1: None,
  field_2: A,
  field_3: None}
{ETD Doc 3: field_1: None,
  field_2: None,
  field_3: A}

Searching for A:
score_2 > score_1 > score_3
Logging

User Logs:

User-oriented information: username, timestamp, query content, IP, cookie, user-agent, etc.

Recommendation, detecting malicious user behaviors, website data analysis.

Index: .logging-yyyy/mm/dd
System Logs:

Event/request recording: timestamp, cluster.name, node.name, cluster.uuid, request/event message.

```
PUT /tobacco/_settings
{
  "index.search.slowlog.threshold.query.warn": "1s",
  "index.search.slowlog.threshold.query.info": "1s",
  "index.search.slowlog.threshold.query.debug": "2s",
  "index.search.slowlog.threshold.query.trace": "500ms",
  "index.search.slowlog.threshold.fetch.warn": "1s",
  "index.search.slowlog.threshold.fetch.info": "800ms",
  "index.search.slowlog.threshold.fetch.debug": "500ms",
  "index.search.slowlog.threshold.fetch.trace": "200ms",
  "index.search.slowlog.level": "info"
}
```
We discussed various ways of implementing recommendation with the TML team. Based on the anticipated cluster information, we implemented a two-step searching process to recommend related documents to the user.

Number of records matching the author name 'Jeong-Ah':
{'value': 1, 'relation': 'eq'}

UserID of the matched record:
1

Number of records matching the clusterID from previous search:
{'value': 2, 'relation': 'eq'}
Incorporating TML Data

We are able to modify, update the desired field in an existing index because we pre-configured the following fields in both datasets as plain text fields.

1. Text Summarization (97,484 for tobacco settlement documents)
2. Sentiment Analysis (765,530*, for tobacco settlement documents)
3. Named-Entity Recognition (213,883 for tobacco settlement documents)
4. Cluster Data (N/A, only for ETDs)

As of 03:14 AM, 12/10/2019
The data files can be processed as:

- Plain text file.
- Named after document ID.

```python
from elasticsearch import Elasticsearch
import sys
import glob

def main():
    updateTobaccoTextSummaries("/mnt/ceph/tml/text_summary/summary/")

def updateTobaccoTextSummaries(path):
    es = Elasticsearch(["10.43.54.87:9200/"])
    print(es.ping())
    files = glob.glob(path + "*.txt")
    for file in files:
        with open(file, 'r') as f:
            textSummary = f.read()
            textid = file.replace('.txt', '').replace('/mnt/ceph/tml/text_summary/summary/', '')
            print(textid + '\n')
            try:
                es.update(index = 'tobacco', id = textid, body = {"doc": {"summary": textSummary}})
            except Exception:
                continue

if __name__ == '__main__':
    main()
```
Index Lifecycle Management

- Indices should be properly managed over time.
- Different indices should be managed differently given their nature
  - Tobacco Settlement Documents: constantly queried, seldom updated
  - ETDs: constantly queried, periodically updated
  - Logs: periodically queried, extensively updated
Index Lifecycle Management - cont.

- Determine appropriate policy for different dataset
  - Tobacco Settlement Documents - Stay in *warm* stage as long as possible and keep in one segment
  - ETDs - Stay in *warm* stage as long as possible and keep in one segment
  - Logs - Stay in *hot* stage, with a limited size of storage and limited life span
Index Lifecycle Management - cont.

```
"tobacco_ETD": {
  "version": 1,
  "modified_date": "2019-12-08T03:25:24.119Z",
  "policy": {
    "phases": {
      "warm": {
        "min_age": "0ms",
        "actions": {
          "forcemerge": {
            "max_num_segments": 1
          }
        }
      },
      "hot": {
        "min_age": "0ms",
        "actions": {
          "forcemerge": {
            "max_num_segments": 1
          },
          "rollover": {
            "max_size": "50gb",
            "max_age": "30d"
          }
        }
      },
      "delete": {
        "min_age": "90d",
        "actions": {
          "delete": {}
        }
      }
    }
  }
}
```

```
"logs": {
  "version": 1,
  "modified_date": "2019-12-08T03:28:23.187Z",
  "policy": {
    "phases": {
      "warm": {
        "min_age": "30d",
        "actions": {
          "forcemerge": {
            "max_num_segments": 1
          }
        }
      },
      "hot": {
        "min_age": "0ms",
        "actions": {
          "forcemerge": {
            "max_num_segments": 1
          },
          "rollover": {
            "max_size": "50gb",
            "max_age": "30d"
          }
        }
      },
      "delete": {
        "min_age": "90d",
        "actions": {
          "delete": {}
        }
      }
    }
  }
}
```
The file 'test.json' appeared in directory '.' via 'MOVED_TO'

Automatic Script
Unit Tests

```
nick@DESKTOP-9MB6T2N:~/Fall2019/els/scripts$ /usr/bin/python3 /home/nick/Fall2019/els/scripts/CMEIndexTest.py
/usr/lib/python3/dist-packages/requests/__init__.py:80: RequestsDependencyWarning: urllib3 (1.25.6) or chardet (3.0.4) doesn't match a supported version!
  RequestsDependencyWarning)
.
Ran 1 test in 0.208s

OK
```

```
nick@DESKTOP-9MB6T2N:~/Fall2019/els/scripts$ /usr/bin/python3 /home/nick/Fall2019/els/scripts/CMTIndexTest.py
/usr/lib/python3/dist-packages/requests/__init__.py:80: RequestsDependencyWarning: urllib3 (1.25.6) or chardet (3.0.4) doesn't match a supported version!
  RequestsDependencyWarning)
.
Ran 1 test in 0.056s

OK
```
CONCLUSIONS AND FUTURE WORK
Deliverables

1. Data schema for ETD and tobacco datasets has been provided to the FEK, TML, CMT and CME teams
2. 30k - index for ETD dataset
3. Tobacco - index for tobacco settlements dataset
4. Facet names, field types, usage recommendation, and field examples provided to the FEK team for filtering, searching and visualization
5. Search query format with example
   a. Ordinary search (FEK)
   b. Nested search with page hit (FEK)
   c. Boosting
   d. Recommendation script (FEK)
6. Automated scripts
   a. Shell script for monitoring new files
   b. Python script for Ingestion and updating
7. Search log (Slow log) on Kibana
8. Unit test scripts
9. Ingesting and indexing data received from the TML team (ClusterID, summary, sentiment, NER)
Future Work

- Continue to ingest the rest of the documents into Elasticsearch
  - Increase space in Elasticsearch

- Improve the recommendations by working with TML team
  - Text Summaries
  - Sentiment Analysis
  - NER
  - Clustering

- Add support for user logs and recommendations
  - User-Specific Logs with FEK team
  - Index Logs / Store in CEPH
ACKNOWLEDGEMENTS
Thank you!