

Project Statement for Milestone 2

B2J

Jacob Lin, James Ellis, Yiheng Bai

Overview:

At this stage of the project, teams should have prepared the data, including cleansing, transforming and reducing the data, for further steps. Teams should also have decided on a non-relational data model that represents the data well. Teams may have looked into NoSQL databases that may be compatible with the data model.

In this report, team should focus on the dataset preparation and data model description. They should also provide appropriate statistics on the reduced and transformed data.







Report Topics:

The report should cover the following subtopics and answer the questions listed:







1. Data Preparation and Data Reduction:
 - a. Describe data cleansing and data transformation steps you have performed so far. Include pseudo-code you have implemented for these steps.
 - A. The data in all but the relations, entities, and others had any data with missing info removed after the column reduction.
 - b. It is recommended that you reduce the data to a reasonable size for faster development and testing. Described the reduced data set using basic statistics - number of files, storage size (KB/MB/GB), number of records (rows), number of attributes (columns), etc.
 - A. Reduction in the columns of information in every csv, which alone reduced the data to around a third of the original size in KB. This was implemented using the .drop() function from the pandas library in python.
 - B. Entities and others cannot be reduced by this yet so it will be postponed until later.
 - A. reasoning: entities row: 814345 -> 0; others row: 2989 -> 0
 - C. current reduction stats are as follows
 - A. addresses row: 402246 -> 99390
 - B. addresses column: 8 -> 4
 - C. entities row: 814345 -> 814345
 - D. entities column: 21 -> 18

- E. intermediaries row: 26768 -> 12598
- F. intermediaries column: 10 -> 4
- G. officers row: 771315 -> 470263
- H. officers column: 7 -> 3
- I. others row: 2989 -> 2989
- J. others column: 13 -> 9
- K. relationships row: 3336917 -> 3336917
- L. relationships column: 8 -> 7

D. Memory of the original is in the images below

| | | | | |
|----|--|------------------|----------------------|------------|
| A. |  nodes-addresses.csv | 3/9/2023 9:26 AM | Microsoft Excel C... | 70,735 KB |
| |  nodes-entities.csv | 3/9/2023 9:26 AM | Microsoft Excel C... | 194,260 KB |
| |  nodes-intermediaries.csv | 3/9/2023 9:26 AM | Microsoft Excel C... | 3,969 KB |
| |  nodes-officers.csv | 3/9/2023 9:26 AM | Microsoft Excel C... | 88,887 KB |
| |  nodes-others.csv | 3/9/2023 9:26 AM | Microsoft Excel C... | 390 KB |
| B. |  relationships.csv | 3/9/2023 9:26 AM | Microsoft Excel C... | 274,468 KB |

E. Memory of the reduced is in the image below

| | | | | | |
|----|--|----------------------------|-----------|----|------------|
| A. |  reduced-addresses.csv | Microsoft Excel Comma S... | 2,255 KB | No | 8,368 KB |
| |  reduced-entities.csv | Microsoft Excel Comma S... | 27,486 KB | No | 137,011 KB |
| |  reduced-intermediaries.csv | Microsoft Excel Comma S... | 200 KB | No | 671 KB |
| |  reduced-officers.csv | Microsoft Excel Comma S... | 5,834 KB | No | 15,847 KB |
| |  reduced-others.csv | Microsoft Excel Comma S... | 44 KB | No | 162 KB |
| |  reduced-relationships.csv | Microsoft Excel Comma S... | 20,726 KB | No | 185,261 KB |

- F. Overall the data hasn't been condensed into one or two csv files yet due to their immense size.
- c. You may have developed a parser to transform the raw data into the format/tools you are using. Briefly describe the functions of the parser you have implemented so far.
 - A. No real functions for the parser have been made yet, however one pass was to have all the data with missing data from the addresses, intermediaries, and officers be removed to significantly reduce the size of the files.
 - B. The transformation of the raw data has only seen a change to the column count so far. There may be a file filled with problematic data points later that is determined from relations no longer pointing to existing nodes.

2. Data Model:

- a. Describe the data model you are using to represent the dataset? Justify why the data model is an appropriate one for the dataset. Note: You should be using a non-relational data model for this project.

We are using a graph data model to represent the dataset. The other options such as relational data model does not work well with our data set as the data set has really complex relationships, and it tends to have less efficiency when the complexity grows. There is also a document data model that we can use JSON or XML (semi-structured) and Key-Value data model. And again both of the models have the same issue of bad handling of complex relationships. Among all the data model options graph data model will be the best choice.

- b. Report the following statistics for your (reduced) dataset:

- A. addresses row: 9390
- B. addresses column: 4
- C. entities row: 814345
- D. entities column: 18
- E. intermediaries row: 12598
- F. intermediaries column: 4
- G. officers row: 470263
- H. officers column: 3
- I. others row: 2989
- J. others column: 9
- K. relationships row: 3336917
- L. relationships column: 7

If you are using a graph data set: how many nodes and edges? How many attributes are there for the nodes/edges? Is it labelled? Directed?

We currently have 1300195 of nodes. It is label With 4 labels, Entity, Intermediary, Officers, Other. 20 different records. It is directed relationships since the relationships.csv provide start_node_id and end_node_id for me to establish the relationships. There is other edges we consider to add. Such as Intermediary -> Entity, Officer -> Entity with edge officer_of. And also address. But in the current project we encounter the address.csv dirty bit that hasn't been cleansed yet which means it will not be able to establish all the relationships. And creating the relationships between nodes based on the relationships.csv takes hours to load and still in progress. I am not sure if my code has a problem or maybe it is just because there are more than 1 million entries (I will provide cypher code in the zip file).

1. Database:

- a. What database are you considering for storing the reduced data? Would it scale for storing and processing the original dataset?

We are using Neo4j as the primary database option. Other databases we will be looking into are Titan and Cassandra. Both are open-source where Titan offers graph database systems and Cassandra offers NoSQL database management systems. Which means by using Titan as a graph model and backend with Cassandra to query.

Scaling:

It is possible to scale up the Neo4j database as we already encountered during the importation of the dataset there was the problem of bottleneck of transaction size and heap size. We find the solution by changing the configure file of the database to bump the size up will resolve the error. So as long as the hardware can handle the scale it is possible to have larger and larger sets of data. With processing the original data will still be possible. One of the method can be using the “LOAD CSV” command to create Node. And with “MATCH” and “DELETE” we can delete the duplicate to perform a data set update.

1. Source Code:

- a. Provide source code of your data preparation, data reduction and data transformation steps in a Zip file.

Peer Evaluation:

All team members should complete the CATME Peer Evaluation survey.

Grading:

- 15 pts: Team has successfully performed data preparation, data reduction and data transformation steps. Team has provided accurate description and statistics of the reduced dataset.
- 5 pts: Project milestone document provides all information with relevant diagrams, pseudo-code and sample dataset.