By the end of this activity, you will be able to:

1. Import and query text documents with Lucene
2. Perform weighted queries to see how rankings change
3. View the Term Frequency-Inverse Document Frequency (TF-IDF)

**NOTE: if you get the error *Exception in thread "main" java.lang.NoClassDefFoundError*when running the commands in this activity, you will need to download Lucene by running these commands:**

1

2

3

cd $HOME/Downloads

wget http://archive.apache.org/dist/lucene/java/5.5.0/lucene-5.5.0.tgz

tar -xvzf lucene-5.5.0.tgz

**Step 1.** **Open a terminal shell.** Open a terminal shell by clicking on the square black box on the top left of the screen.



Change into the vector directory:

1

cd Downloads/big-data-2/vector

Run *ls* to see the scripts and data directory:

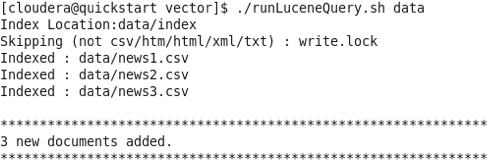


The data directory contains three CSV files, which contain textual data from the news.

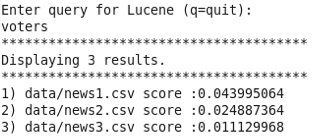
**Step 2. Import and query text documents.**Run *runLuceneQuery.sh data* to import the documents in the data directory:

1

./runLuceneQuery.sh data

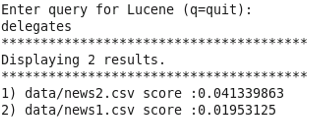


Enter *voters* to query for that term:



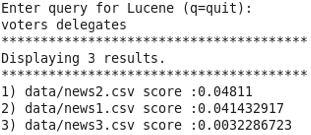
The output shows the rankings and score for each of the three CSV files for the term *voters*. This shows that *news1.csv* is ranked first, *news2.csv* is second, and *news3.csv* is third.

Next, enter *delegates* to query for that term:



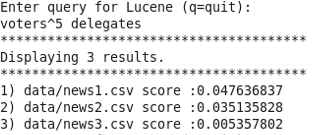
The output shows that *news2.csv* is ranked first, *news1.csv* is ranked second, and *news3.csv* is not shown since the term *delegates* does not appear in this document.

We can query for multiple terms by entering them together; enter *voters delegates* to query for both terms:



The output shows that *news2.csv* is ranked first, *news1.csv* ranked second, and *news3.csv* ranked third.

**Step 3. Perform weighted queries.** We can perform a weighted query (or "boosting") to give one term more importance than the others. Enter *voters^5 delegates* to give the term *voters* a boost factor of 5:



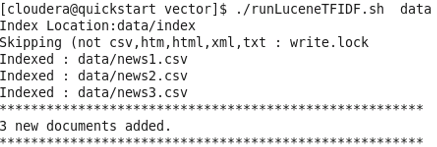
The output shows that *news1.csv* is ranked first and *news2.csv* is ranked second. Note that these two rankings are reversed from when we performed the same query without boosting.

Enter *q* to quit this script.

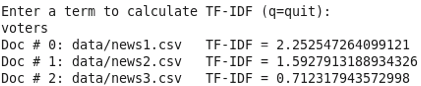
**Step 4. View the TF-IDF.** Run *runLuceneTF-IDF.sh data* to see the TF-IDF for terms in the documents:

1

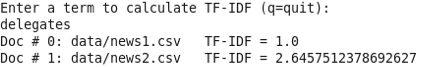
./runLuceneTFIDF.sh data



Enter *voters* to see the TF-IDF for that term:



Enter *delegates* to see the TF-IDF for that term:



Enter *q* to quit this script.