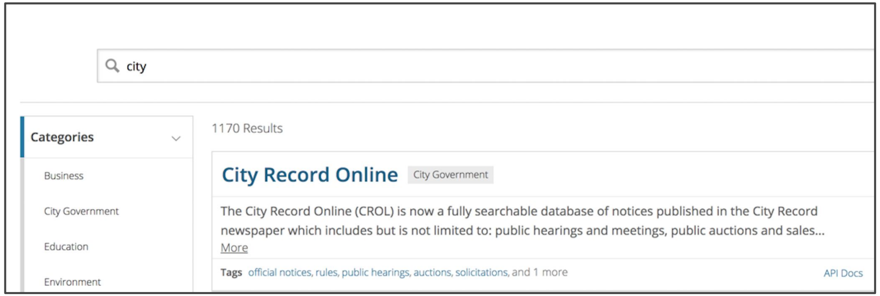
Search Engine in PySpark

Big Data Term Project
Spring 2018

Project Goal: search engine for structured data

NYC OpenData



1170 Results

City Record Online City Government

The City Record Online (CROL) is now a fully searchable database of notices published in the City Record newspaper which includes but is not limited to: public hearings and meetings, public auctions and sales...
More

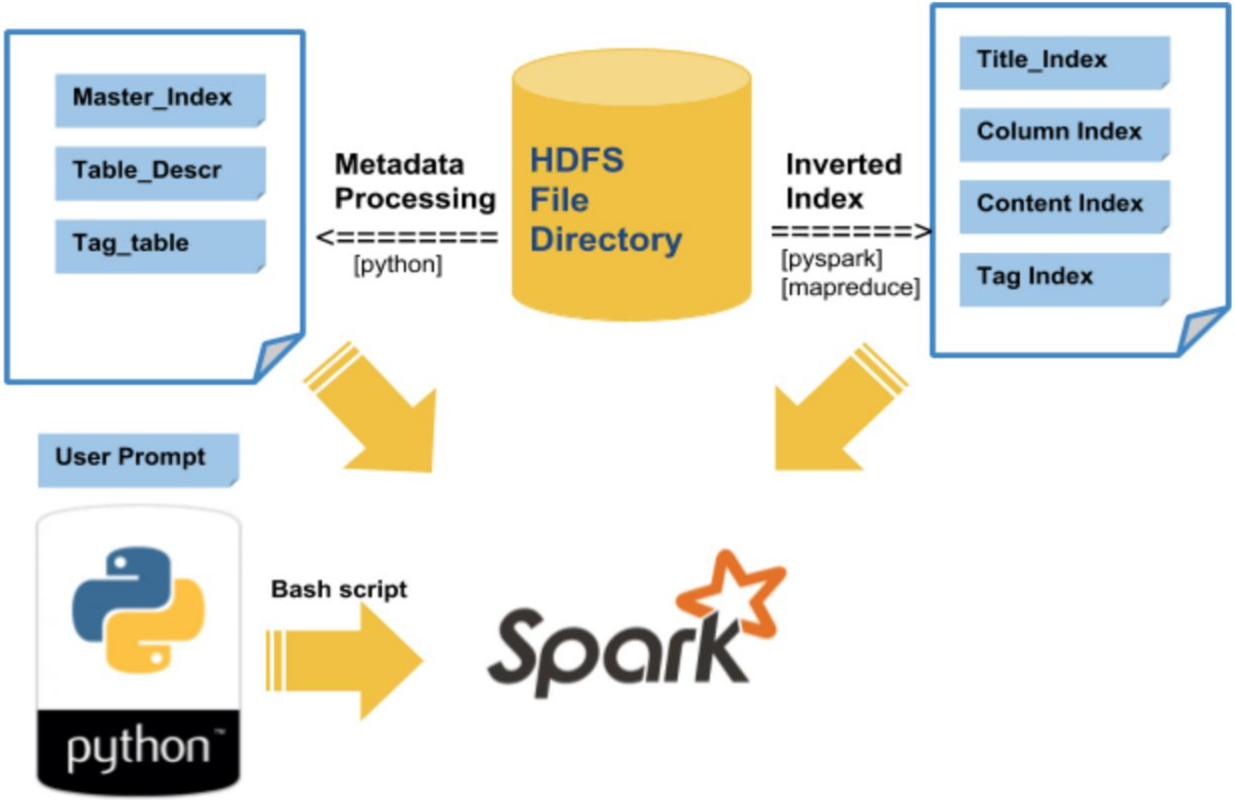
Tags official notices, rules, public hearings, auctions, solicitations, and 1 more API Docs

Only Title Search Available!

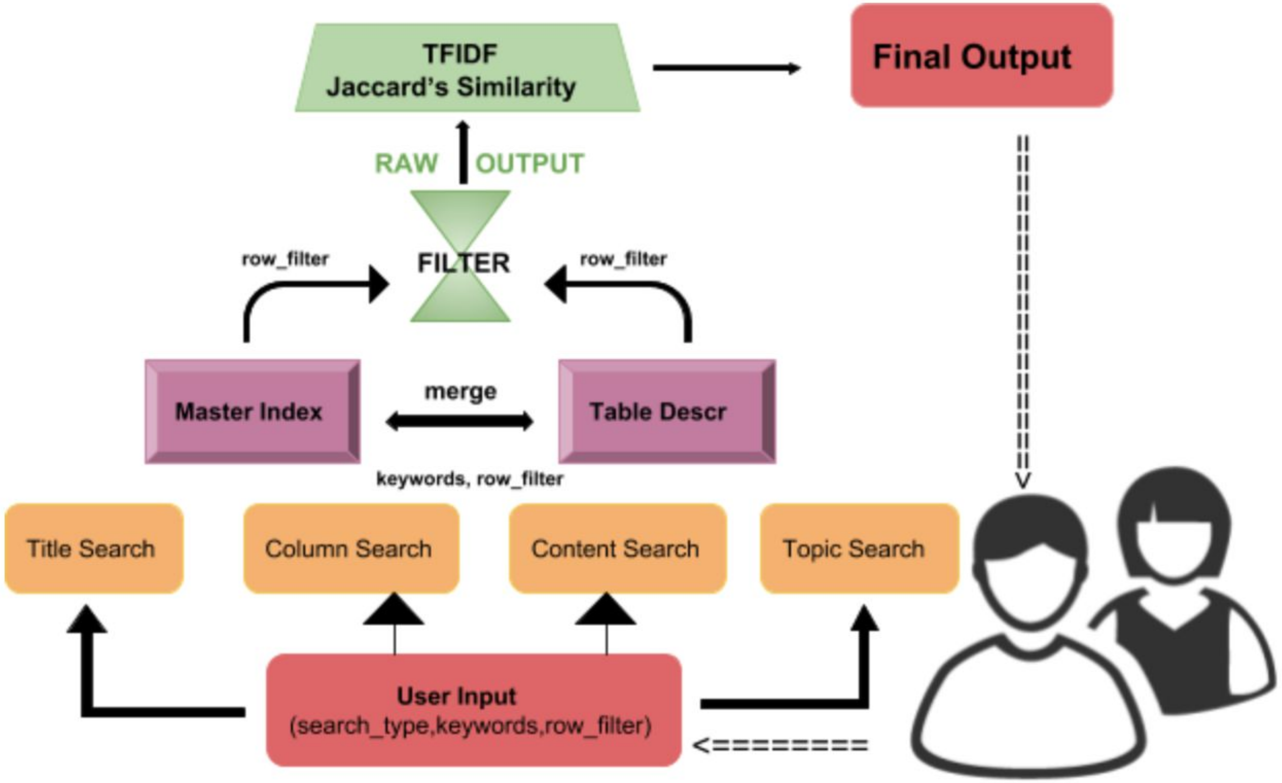
We will offer search by:

- Title
- Column
- Content
- Topic
- Filter by table length

Architecture & Design: Method Design



Architecture & Design: Application Design



Json Processing and Other Tables

HDFS Directory

metadata

Tag_table (Doc_ID, tags) (generate tag_index)

Table_Desc (Doc_ID,Category, Description)

Table_Schema(used to generate column_index)
(Doc_ID,Schema)

data

Master_Index(used to generate title_index)
(Doc_ID,Table_Name,Table_Length)

All tables in .csv format for content index

Inverted Indices: title, column, context, tags

1. **Collect** the data
2. Parse collection of tables
    a. **Lowercase** all words: “SPARK” -> “spark”
    b. **Tokenize**: “search+engine” -> “search”, “engine”
    c. Filter out **stop words**: “the”, “a”, “an”, etc.
    d. Stem each token using **Porter Stemmer Algorithm**: “fisher”, “fishing”, “fishes”, “fished” -> “fish”

Inverted Index

- Building the index using **MapReduce**
  - **Key**: vocabulary term
  - **Value**: postings list (tables where the term appears & in-table term frequency)

We compared two methods for constructing indices

- Built **two versions** of inverted indices
  - PySpark
  - Hadoop
- Decided to go with **PySpark**
  - Built-in functionality for querying

Querying

User Queries → Spark.SQL Queries

Python

- Search_Type:

  ---1,2
- Keywords:

  ---cab,taxi,city
- Row_Filter:

  ---100000

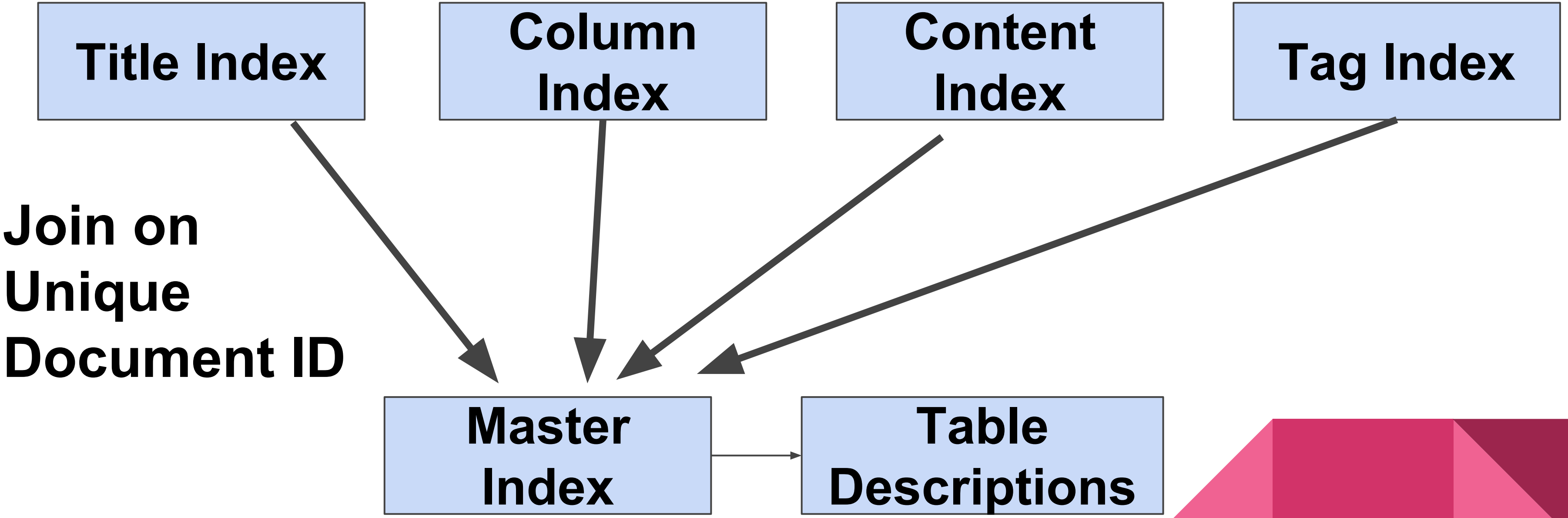
Spark

title_search([cab,taxi,city],100000)

column_search([cab,taxi,city],100000)

Processing Queries

Spark SQL



Join on Unique Document ID

Search Ranking: TF-IDF

- Term Frequency & Document Frequency

$$TF(t, d) \qquad\qquad DF(t, D)$$

- TF.IDF Scoring

$$IDF(t, D) = \log \frac{|D| + 1}{DF(t, D) + 1}$$

$$TFIDF(t, d, D) = TF(t, d) \cdot IDF(t, D).$$

Search Ranking: Vector Space Model

- Represent each document as a vector, with each entry being the TF.IDF weight of the corresponding term *t* in document *d*
- Represent each query as a vector in the same vector space as the documents
- Cosine similarity: the relevance between each document and the query

$$sim(q,d) \quad = \quad \frac{\vec{V}(q) \cdot \vec{V}(d)}{|\vec{V}(q)||\vec{V}(d)|}$$

Search Ranking

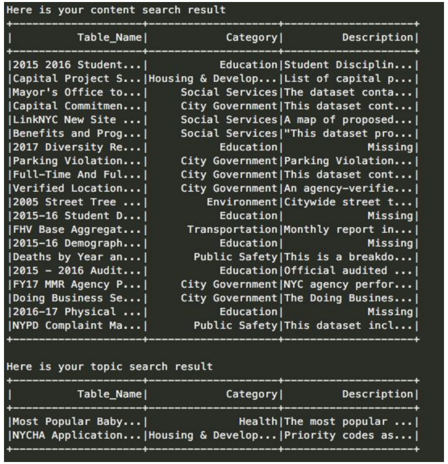
Preprocessing the tables

- Calculate TF.IDF scores from the inverted index files
- 4 inverted index files -> 4 TF.IDF models
- Save the models into files

Searching in run-time

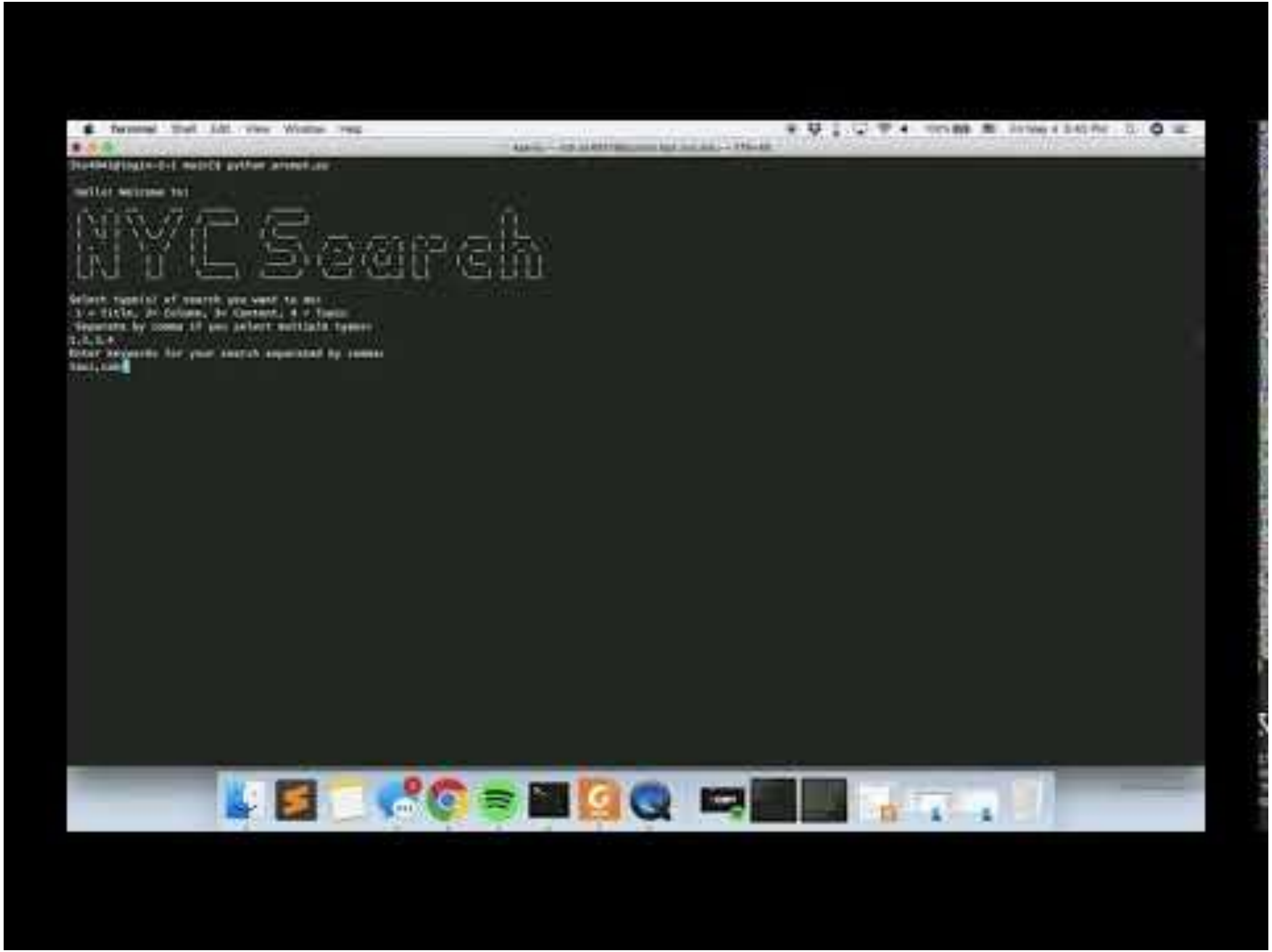
- Get the query & Parse
- Convert into vector
- Read the TF.IDF models
- Calculate the cosine similarities
- Rank & Output

Demo



```
Here is your content search result
+--------------------+--------------------+--------------------+
|          Table_Name|            Category|         Description|
+--------------------+--------------------+--------------------+
|2015 2016 Student...|           Education|Student Disciplin...|
|Capital Project S...|Housing & Develop...|List of capital p...|
|Mayor's Office to...|     Social Services|The dataset conta...|
|Capital Commitmen...|     City Government|This dataset cont...|
|LinkNYC New Site ...|     Social Services|A map of proposed...|
|Benefits and Prog...|     Social Services|"This dataset pro...|
|2017 Diversity Re...|           Education|             Missing|
|Parking Violation...|     City Government|Parking Violation...|
|Full-Time And Ful...|     City Government|This dataset cont...|
|Verified Location...|     City Government|An agency-verifie...|
|2005 Street Tree ...|         Environment|Citywide street t...|
|2015-16 Student D...|           Education|             Missing|
|FHV Base Aggregat...|      Transportation|Monthly report in...|
|2015-16 Demograph...|           Education|             Missing|
|Deaths by Year an...|       Public Safety|This is a breakdo...|
|2015 - 2016 Audit...|           Education|Official audited ...|
|FY17 MMR Agency P...|     City Government|NYC agency perfor...|
|Doing Business Se...|     City Government|The Doing Busines...|
|2016-17 Physical ...|           Education|             Missing|
|NYPD Complaint Ma...|       Public Safety|This dataset incl...|
+--------------------+--------------------+--------------------+

Here is your topic search result
+--------------------+--------------------+--------------------+
|          Table_Name|            Category|         Description|
+--------------------+--------------------+--------------------+
|Most Popular Baby...|              Health|The most popular ...|
|NYCHA Application...|Housing & Develop...|Priority codes as...|
+--------------------+--------------------+--------------------+
```



```
[illegible] python prompt.py

 Hello! Welcome to!
NYC Search
Select type(s) of search you want to do:
 1 = Title, 2= Column, 3= Content, 4 = Topic
 Separate by comma if you select multiple types:
1,2,3,4
Enter keywords for your search separated by comma:
taxi,cab
```

Results

- Improved **Search Capabilities**
  - Title, Column, Content, Tags, Row Filter
- **Efficient Ranking**
  - TF-IDF using MLlib
- **Scalable** Implementation
  - Can built additional applications on top