# Open Data Link

## A dataset search engine for open data

Paul Ouellette and Justin Fargnoli

# Open Data Link

- Dataset search engine for open data.
- Search methods:
  - Semantic keyword search
  - Joinable table search
  - Unionable table search

# Motivation

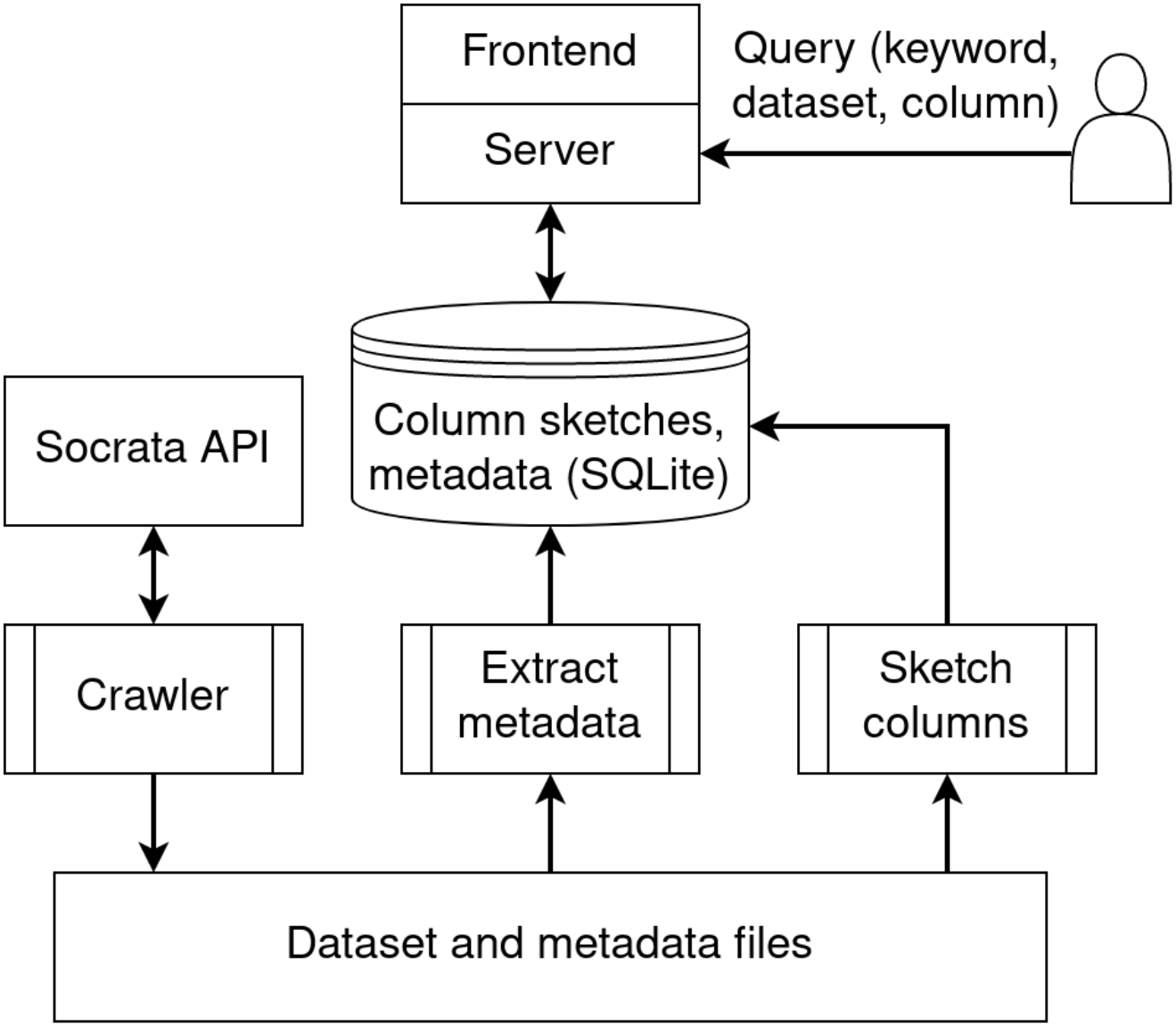
- Governments and other organizations publish a lot of open data, but discovery is still difficult.
- Data scientists can identify ways to integrate datasets.
- Data publishers can see the wider context of their data.

# Demo

# Outline

# Outline

# System overview



# Dataset crawl

- 10k of 42k datasets on Socrata.
- 172k columns.
- Most datasets are small.
- Largest datasets have over 100 million rows.

# Outline

# Minhash[2]

- Data sketch for estimating Jaccard similarity of sets.

$$J(S, T) = \frac{|S \cap T|}{|S \cup T|}$$

- A minhash signature is composed of the results of a number of minhashes.
- The probability that the minhashes for two sets are the same equals the Jaccard similarity of the sets[1].
- Minhash LSH hashes similar signatures to the same bucket.

[1] Mining of Massive Datasets, Chapter 3.

[2] A. Broder, "On the Resemblance and Containment of Documents", Compression and Complexity of Sequences 1997.

# LSH Ensemble[3]

- Set containment is a better measure for computing joinability.

$$C(Q, X) = \frac{|Q \cap X|}{|Q|}$$

- We can convert Jaccard similarity to containment, given the sizes of the domains.
- The size of the indexed domain is not constant, so domains are partitioned by cardinality.
- A minhash LSH index is constructed for each partition.

[3] Erkang Zhu, Fatemeh Nargesian, Ken Q. Pu, Renée J. Miller, "LSH Ensemble: Internet-Scale Domain Search", VLDB 2016.

# Outline

# Unionable table search

- The LSH Ensemble index is queried for each column of the query table.
- Candidate tables are those that appear in $\geq 40\%$ of the joinability queries.
- Candidates are ranked by alignment: the fraction of candidate columns that are unionable with a query column.

# Outline

# Semantic Keyword Search

- Problem: Given a list of keywords, return datasets which are more similar than threshold $t$.
- Motivation: Data scientists want a simple way to find new and insightful datasets

# Our Approach

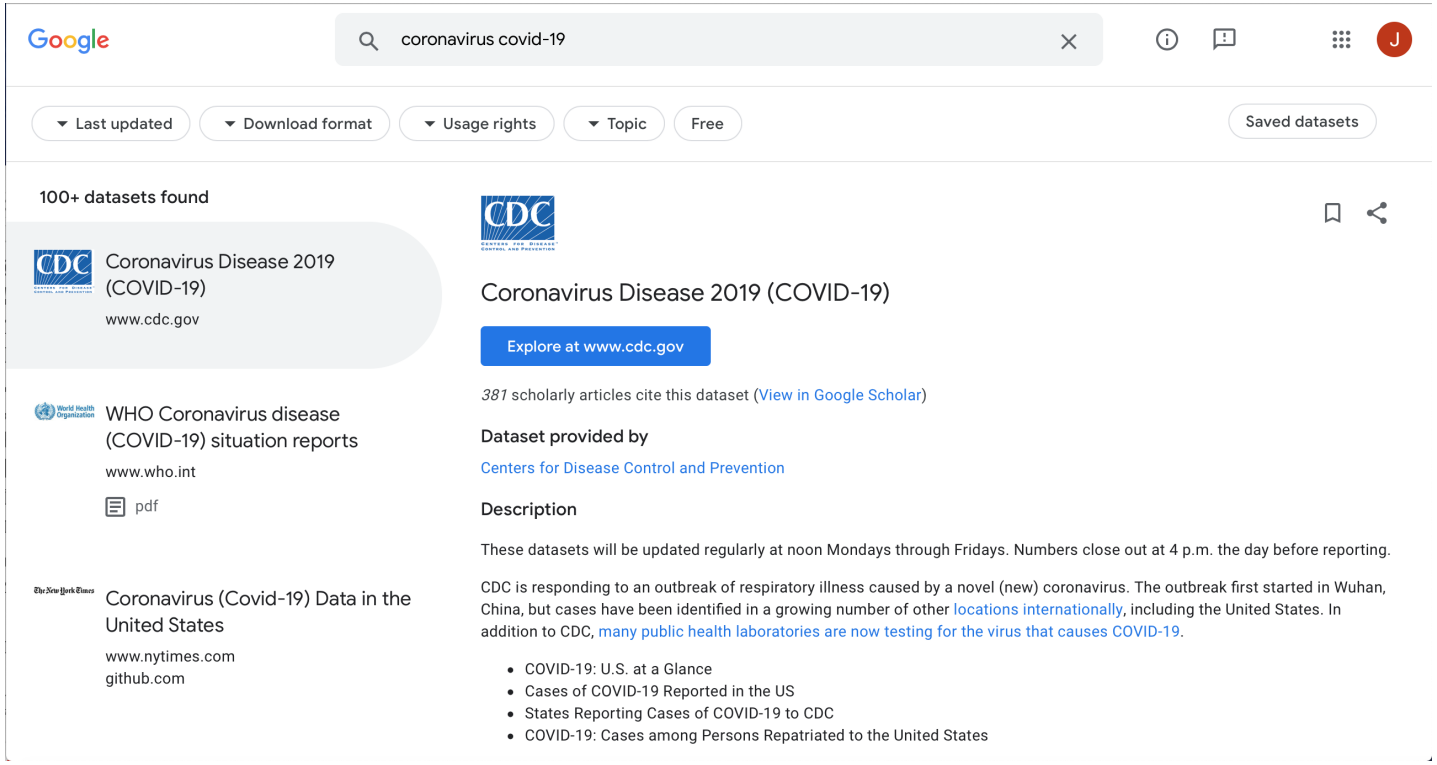
- Search on the metadata, not on the data in the dataset
  - Data in dataset is too noisy
- Metadata that we have:
  - Dataset description
  - Column description
  - Datasets tags

# Our Approach (Cont.)

- Use semantic NOT syntactic similarity
  - Example: Fish & Seafood
  - Example: Coronavirus & Respitory System

# Others Approach

- Google Dataset Search



# System Overview

- FastText: word in dataset's metadata -> embedding vector
- SimHash: embedding vector -> bit vector
- Locality Sensitive Hashing (LSH): build index on the bit vector of each word

# FastText[4]

- Word in dataset's metadata -> embedding vector
- Embedding vector represent the semantics of words
- Embedding vectors are learned from wikipedia articles

[4] A. Joulin, E. Grave, P. Bojanowski, T. Mikolov, *Bag of Tricks for Efficient Text Classification*

# FastText (Cont.)

- closeness or similarity of vectors := Cosine-Similarity
- Closer a pair of vectors, closer the semantics of the two words
  - PICTURE

# Simhash

- Embedding Vector -> Bit Vector
  - PICTURE

# Locality Sensitive Hashing (LSH)

- Underlying data structure: Hash Table
  - Predefined # of buckets
- Insert SimHashed embedding vectors into hash table
- Collitions in hash table buckets are candidate pairs.

# SimHash LSH (Cont.)

- Perdefined # of hash tables
- Query each L hash table for M candidates
  - $M \geq k$
- Order M candidates into a top-k list by the cosine similairty of embedding vectors

# Problem with SimHash LSH

- The # of hash tables and # of buckets in each hash table must be **hand tuned**
- Must be retuned when data size significantly changes
- PICTURE

# LSH Forest

- Underlying data structure: Prefix Tree or Trie
- Similar to LSH
  - Predefined # of prefix trees
  - Query each L hash table for M candidates
    - $M \geq k$

# LSH Forest (Cont.)

- Variable length hashing in prefix tree solves LSH's problems
- PICTURE
- Prefix Tree expands and contracts to account for # of embedding vectors
  - Thus, no hand tuning

# Answering Queries

- Query the index with each keyword in the keyword list
- Add the results to a list
- Rank datasets by how often they appear in the list

# Problems

- No semantic relationships **between** words
  - Example: Keyword List := "traffic violations"
    - Produces good results for "traffic" and "violations", but not "traffic violations"

# Outline

# Future work

- Improve ability to see semantic relationships between words
- Organize datasets into a directory structure
- Use semantic similarity of column names in unionable table search.
- Similar dataset search based on metadata similarity.
- Keyword search over data values.