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

System overview

Joinable table search

Unionable table search

Semantic Keyword Search

Future work

Outline

System overview

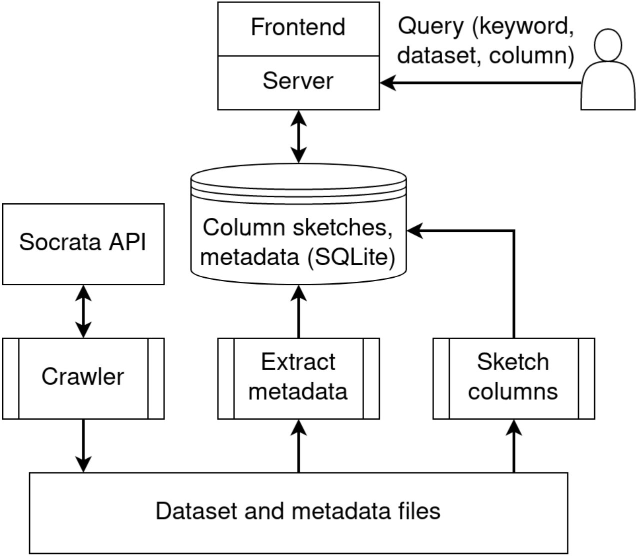
Joinable table search

Unionable table search

Semantic Keyword Search

Future work

System overview



Dataset crawl

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

Outline

Joinable table search

JOIN

ACTION_SOUGHT	DEPARTMENT	CLIENT_ID	CLIENT_NAME	LOBBYIST_ID
AMENDMENT TO PLANNED DEVELOPMENT	TRANSPORTATION	2201517748	FRIEDMAN PROPERTIES	6842
DISCUSSED FY 2020 BUDGET				
CONTRACT APPROVAL				
CONSTRUCTION RELATED				
LANDMARKING, CLASS L APPROVALS AND RIGHT OF WAY ISSUES				
EVENT APPROVA				
ZONING				

YEAR	LOBBYIST_ID	SALUTATION	FIRST_NAME	MIDDLE_INITIAL	LAST_NAME
2011	3958	MR	STEPHEN		FORTINO
2012	3789	MR	NEIL	A	PRITZ
2013	3773	MS.	ERICA	N.	BLAND
2012	3964	MRS	MARY KAY		BONOMA
2011	3764	MS	JENNIFER		CLARK
2013	4332	MR.	MICHAEL	J	PAULOS
2014	3802	MR.	JAMES	L	TERMAN

Joinable table search

- Attributes are treated as sets.
- Sets are encoded with minhash data sketches.
- A table T is joinable with the query U if $Containment(X \in T, Q \in U) \geq t$.
- We use an LSH index for fast querying.

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

Candidate Name	Source Type	Source Name	Date	Amount
Abbett, Richard	Candidate	Abbett, Richard	09/29/2016	20.00
Abercrombie, Neil	Other Entity	Facebook, Inc.	04/01/2014	65.16
Aiona, Sam	Candidate	Aiona, Sam	06/30/2015	6415.49

UNION

Candidate Name	Contributor Type	Contributor Name	Date	Amount
Ige, David	Individual	Ohori, Yoshiko	09/11/2014	99.05
Ige, David	Individual	Perry, Nolan	10/13/2014	50.00
Herkes, Robert	Individual	Nip, Celeste	02/04/2008	200.00
Hannemann, Mufi	Individual	Murakami, Ross R.	04/15/2008	500.00
Hannemann, Mufi	Individual	Dinsmore, Jeffrey C.	07/20/2009	1000.00
Hooser, Gary	Individual	SHERMAN, WENDY L.	06/10/2010	500.00
Hannemann, Mufi	Individual	Miyashiro, Alton K.	10/09/2014	2000.00
Hannemann, Mufi	Individual	Konishi, Glen S.	07/22/2010	150.00
Hong, Ted	Individual	Malasek, Vojtech	10/29/2008	4000.00
Hannemann, Mufi	Individual	Takara, Russell H.	09/08/2008	1000.00
Hokama, Riki	Individual	Matsuda, Eric	06/25/2013	225.00
Hannemann, Mufi	Individual	McIntyre, Gregory T.	06/30/2007	250.00
Ige, David	Individual	Lincoln, Faye	11/10/2014	500.00

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 top-k similar datasets
- Motivation: Data scientists want a simple way to find new and insightful datasets

Our Approach

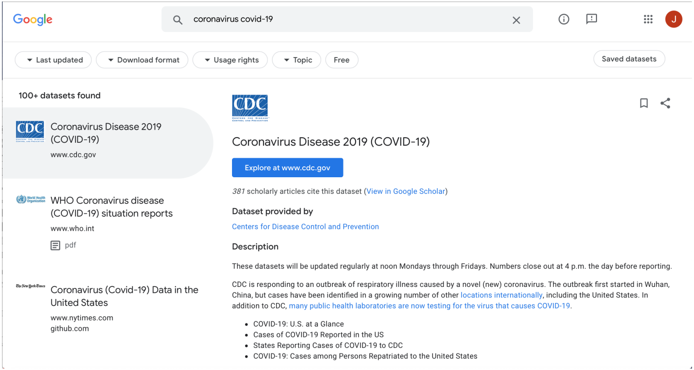
- Search on the metadata, not on the data in the dataset
  - Data in dataset is too noisy
- Metadata that we have:
  - Dataset description
  - Column description
  - Datasets tags

Our Approach (Cont.)

- Use semantic 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

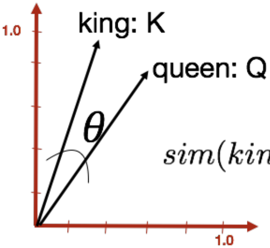
FastText[4]

- 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.)

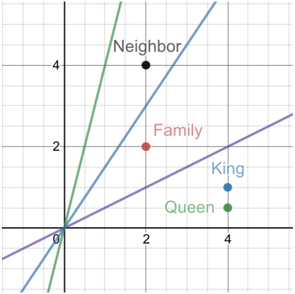
- closeness or similarity of vectors := Cosine-Similarity
- Closer a pair of vectors, closer the semantics of the two words
- Example: King and Queen have high cosine similarity



$$sim(king, queen) = Cosine(\theta) = \frac{\vec{K}.\vec{Q}}{\|\vec{K}\|.\|\vec{Q}\|}$$

Simhash

- Embedding Vector -> Bit Vector
- Word: [puple, blue, green]
  - Neighbor: [1, 1, 0]
  - Family: [1, 0, 0]
  - King: [0, 0, 0]
  - Queen: [0, 0, 0]



Locality Sensitive Hashing (LSH)

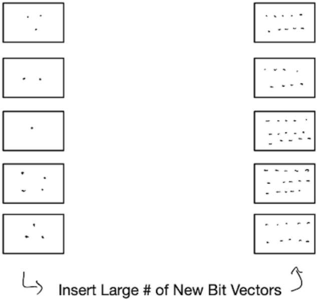
- Underlying data structure: Hash Table
  - Predefined # of buckets
- Insert SimHashed embedding vectors into hash table
- Collitions in hash table buckets are candidate pairs.

SimHash LSH (Cont.)

- Predefined # of hash tables
- Query each 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



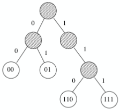
LSH Forest[5]

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

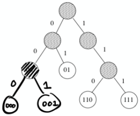
[5] Mayank Bawa, Tyson Condie, and Prasanna Ganesan. 2005. *LSH forest: self-tuning indexes for similarity search*

LSH Forest (Cont.)

- Variable length hashing in prefix tree solves LSH's problems



- Add Bit Vector: [0, 0, 1, . . .]



- Prefix Tree expands and contracts to account for # of embedding vectors
  - Thus, no hand tuning

Review

- FastText: Word in Metadata -> Embedding vector
- SimHash: Embedding Vector -> Bit Vector
- LSH Forest: Index on Bit Vectors

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

System overview

Joinable table search

Unionable table search

Semantic Keyword Search

Future work

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.