Robust and Efficient Fuzzy Match for Online Data Cleaning

S. Chaudhuri, K. Ganjan, V. Ganti, R. Motwani

Presented by
Aaditeshwar Seth

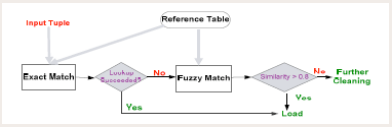
1

Motivation

- Data warehouse: Many input tuples
- Tuples can be erroneous
  - Spelling mistakes
  - Different syntactic representation
- How to clean them automatically?
- Assumptions
  - Tuples are structured: Eg. schema matching has already been done
  - Some tuple fields are supposed to contain the same values

2

Methodology



- Clean tuples are stored in reference table
- Fuzzy matching done to find best matching clean tuples

3

Challenges

- Scalability
  - Reference table can be very large
  - Volume of input tuples can be very large
- Domain specific enhancements should be possible to add
- Should be able to build upon existing relational DBMS
  - No complex data structures for persistence

4

Solution outline

- Transform input tuple into reference tuple
- Similarity metric = 1 – (transformation cost)
- Flat edit distance not good
  - Within a field, cannot distinguish between more and less informative tokens
    - Intuitively know *Boing* is more informative than *Corporation*
    - Hence, match on *Boing* should mean high similarity
    - Implies, should be expensive to transform input token into *Boing* than into *Corporation*
- Attach weights to transformation costs for each token

5

Solution outline: Cont…

- Use inverse token frequency for weight assignment
  - Tokens like *Boing* occur less than tokens like *Corporation*
  - Just a heuristic – pathological cases can exist
- Optimizations
  - Do not compute *exact* transformation cost
    - Match on sets of substrings instead
  - Design efficient index on reference table

6

Fuzzy matching similarity function

- Input tuple u, reference tuple v
- Split into tokens
  - Each token has weight
    $w_{TOKEN} = \log(|Ref\ Set| / freq($
- Find tc = edit distance for each token
- tc(u, v) = Summation of tc for each token
- fms(u, v) = 1 – min(tc(u, v) / Sum(w_{TOKEN}), 1.0)

How is the minimum edit distance calculated? Many transformation paths are possible.

- Not a problem with fms^{apx}

7

Approximate FMS

- Match on q-grams

B o e i n g

H_1 3 2 1 0

H_2 4 5 6 7

Min hash = {ing, boe}

B u e i n g

H_1 8 2 1 0

H_2 9 5 6 7

Min hash = {ing, uei}

$$sim_{mh}(t_1, t_2) = \frac{1}{H}\sum_{i=1}^{H} I[mh_i(QG(t_1)) = mh_i(QG(t_2))]$$

- Hash functions avoid string comparison

8

Approximate FMS: Cont…

$$fms^{apx}(u,v) = \frac{1}{w(u)}\sum_{t}\sum_{r \in tok(q(t))} w(t)\cdot \max_{r \in tok(v(t))}\left(\frac{2}{q} sim_{mh}(QG(t), QG(r)) + d_q\right)$$

1. Test for similarity of token t with all tokens r in same column, and select *Max*

2. Multiply by weight of chosen *Max* tuple r

3. Repeat over all tokens t in same column

4. Repeat over all columns, and divide by total weight

Cannot compute w(t) from input u if spelling mistake, or ordering diff

$d_q = 1 - 1/q$. Factor of 2 implies $sim_{mh}(QG(t), QG(t)) = 0.5$?

9

Error Tolerant Index

Table 3: An Example ETI Relation

Q-gram	Coordinate	Column	Frequency	Tid-list
oei	1	1	1	{R1}
ing	2	1	1	{R1}
com	1	1	2	{R1,R3}
pan	2	1	2	{R1,R3}
bon	1	1	1	{R2}
orp	1	1	1	{R2}
ati	2	1	1	{R2}
sea	1	2	3	{R1,R2,R3}
ttl	2	2	3	{R1,R2,R3}
wa	1	3	3	{R1,R2,R3}
980	1	4	3	{R1,R2,R3}
004	2	4	1	{R1}
014	2	4	1	{R2}
024	2	4	1	{R3}

- Each *q*-gram belongs to a token
- Each to[illegible]en has a weight
- Token b[illegible]ngs to a column
- Coordi[illegible] indicate[illegible]

What if same q-gram belongs to multiple tokens? Overwriting!

- Index reference set on *tid*
- Index ETI on {*q-gram*, *coordinate*, *column*}

10

Query processing

- Find all tokens of input tuple *u*
- Find min-hash signature of all tokens
- For all *q*-grams in min-hash signature
  - Find ETI(*q-gram*, *coordinate*, *column*)
  - Find token *t* to which *q*-gram belongs, and weight of this token
  - Increment similarity metric of matching *tid* by w(*t*)/|mh(*t*)|
- Fetch best *K* matching *tid*'s with similarity > *c.threshold*

11

Query processing: Cont…

- Optimizations
  - When incrementing similarity metric with matching *tid*'s, only need to do it with new *tid*'s if the maximum score possible with all remaining *q*-grams is > *c.threshold*
- Optimistic short circuiting
  - Order *q*-grams according to their weights
  - Process only the first *i* *q*-grams
  - Fetch matching *tid*'s
  - But only fetch new *tid*'s if $\frac{ss(r_k^i)}{wt(Q_i)} wt(Q_{opt}) > ss(r_{K+1}^i) + (wt(Q_o) - wt(Q_i))$
  - Stop when FMS of all *K* tid's > *c.threshold*
  - If don't stop then increment *i* and repeat

12

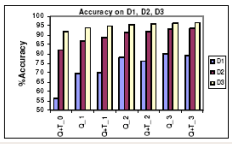
Extensions

- Consider token as another q-gram
  - Split importance equally among itself and its min-hash signature
- Assign weights to columns
  - Domain dependant
- Token transposition

13

Experiments

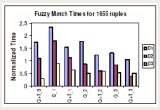
- Clean reference set
- Error injection methods for unclean input tuples
- Accuracy
  - FMS better than edit distance
  - Min-hash signatures are better than token-only
  - Accuracy improves with more hash functions
  - Having tokens does not negatively impact

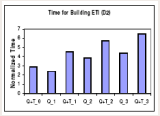


14

Experiments: Cont…

- Efficiency: Processing time
  - Much faster than naïve
  - Query processing time decreases with signature size
  - Use of tokens improves processing time
- Efficiency: ETI construction
  - About 7 times the amount of time taken to process 1 tuple using the naïve algorithm
  - But cost is amortized over repeat queries

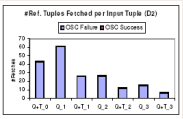




15

Experiments: Cont…

- Average number of *tid*'s fetched per input tuple
  - More *q*-grams decrease set sizes by better distinguishing similarity scores



- Average number of *tid*'s processed per input tuple
  - More *q*-grams increase the number of *tid*'s processed
  - Compensated by decrease in number of *tid*'s fetched

16

Discussion topics

- Relevance: In what scenarios does IDF work and distinguishes between more and less informative tokens
  - Cluster tokens together?
- Does weight calculation become an issue with optimizations and optimal short circuiting?
- How to update ETI with new tuples or outdated tuples?
- What is the role of the factor *2* in fms^{apx}?
- What if same *q*-gram belongs to multiple tokens in the same column?

17