CheetahER: A Fast Entity Resolution System for Heterogeneous Camera Data

Nan Deng, Wendi Luan, Haotian Liu, Bo Tang (Supervisor) (SIGMOD Programming Contest 2020)
†Department of Computer Science and Engineering, Southern University of Science and Technology
*Contact: {11711004, 11712532, 11613015}@mail.sustech.edu.cn, tangb3@sustech.edu.cn

SUSTech Southern University of Science and Technology

Task Overview

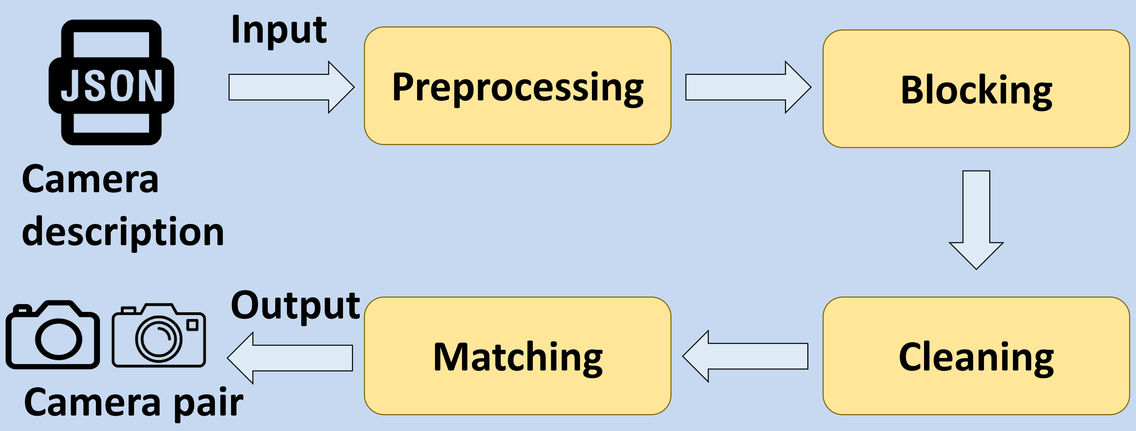
Entity Resolution on a camera dataset:

Input: 30k~ camera description extracted from e-commerce websites in json format

Output: all pairs of camera manifestations that refer to the same real-world camera model, in csv format

Measurement: F1 score (precision, recall), running-time

System Architecture



System Framework

- **Two phases: Blocking** and **matching**
- Complex blocking to do the best-effort entity classification.
- Simple matching only do inner-block combination.
- Two core blocking operations: **merge** and **split**

Preprocessing

Data Refining:

- Filter non-relevant attributes, focused on useful attributes (e.g. <page title>)
- Reduced memory consumption.

In-memory index (hashing table):

- key: identifier of each product
- value: the address of the reserved attributes structure.

Blocking by Brand

Brand List Collection:

- Extracted from distinct <brand> labels
- Refined by regular rules: **e.g. contains only alphabet and white space, the number of white space not more than 1**.

Merge Blocks with different expressions:

- Blocks of same brand in different expressions are merged by **edit distance** (e.g. "Cannon" and "Canon")
- Given two brands A and B, the edit distance between A = {A_1, A_2, ... , A_n } and B = {B_1, B_2, ... , B_m } in recurrence is ($1 \le i \le n$, $1 \le j \le m$):

$$ed_{ij} = \begin{cases} ed_{i-1,j-1} & \text{for } A_i = B_j \\ \min \begin{cases} ed_{i-1,j} + w_{\text{del}}(B_i) \\ ed_{i,j-1} + w_{\text{ins}}(A_j) \\ ed_{i-1,j-1} + w_{\text{sub}}(A_i, B_j) \end{cases} & \text{for } A_i \neq B_j \end{cases}$$

Blocking by Model

Block Collection:

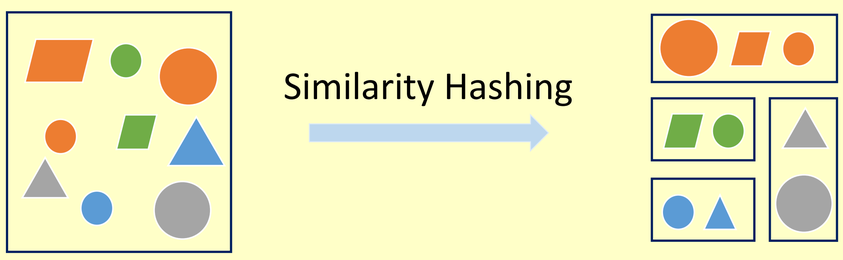
- Regular rules to extract all block names.
  - **e.g. [a-zA-Z][0-9a-zA-Z-]+$**
- Refined by regular rules. **e.g. 'and' cannot be model name.**
- **To minimum False Negatives in Blocking:** A manifestation may be assigned to several blocks.

Splitting:

- Separate blocks with special postfix with regular rules.
  (e.g. 'E0S 5D' -> 'EOS 5D I' - 'EOS 5D II')

Merging:

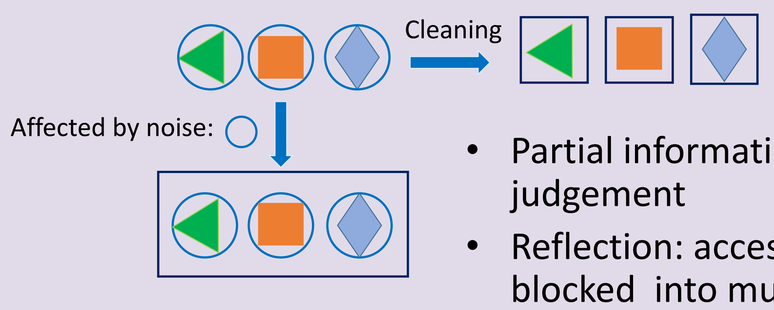
- Co-appearance merging
  → Multiple models appear in the same product description.
  e.g. many products of Canon have both 'ELPH 300' and 'IXUS 220' in their page titles.
  → Merging frequent co-appeared blocks.
- Model Similarity Hashing
  → Regular rules: hash a model to its most **representative** expression, like prefix/suffix elimination.
  e.g. remove prefix 'EX-' from 'EX-FH20'
  → Other auxiliary model-equivalences also participate in function building.



Cleaning

Our Solution: Reverted List

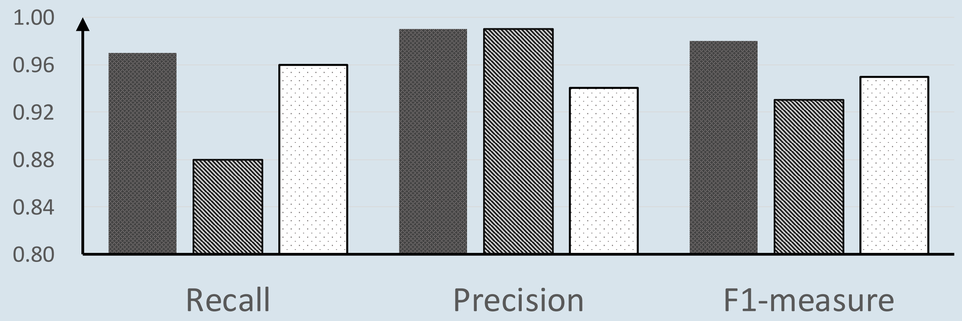
- A product-blocks map recording the assigned blocks of each product
- Noise cleaning: delete a product from all its assigned blocks if the size of list is larger than a threshold.



- Partial information may affect overall judgement
- Reflection: accessories always blocked into multiple models.

Accuracy Benchmark

Main Optimization Strategies: block merging & splitting



Time cost: 17 seconds
Time consuming part: Blocking Procedure (13s)
Hardware: 4x Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz