BEER: Blocking for Effective Entity Resolution – A Literature Review

Ahmad Bacha

Tianyi Hao

Xinyi Yue

Illinois Institute of Technology

Introduction

With the development of technology, data resides in various systems and applications throughout an organization. It can be difficult to find all the records associated with one entity or customer. Each record may not have a unique identifier that identifies which record in one source corresponds to a record in another source. In addition, records that represent the same entity may contain different information.

Entity Resolution (ER) refers to a technology that identifies data records in datasets that describes the same entity. It aids organizations to derive inferences from vast amounts of information in multiple systems and applications by merging sets of data records that correspond to the same entity. The problem of ER is its time complexity. It has quadratic time complexity because all pairs of records need to be compared to identify a match. The increasing volume of datasets and various availability of data sources make this problem even worse. Therefore, blocking is typically added as a pre-processing step before ER is performed. To reduce the number of records pair comparisons, blocking groups similar records into different blocks and only compares records within the same block in subsequent steps.

In this paper, the authors present Blocking for Effective Entity Resolution (BEER). This is the first end-to-end system that uses an intermediate ER output as feedback to refine the blocking results to apply effective entity resolution.

Method Analysis

Before deep diving into the new blocking method, it is necessary to examine the limitations of the traditional blocking method. The author demonstrates these shortcomings using the following example.

r_1^{c6}: 'chevy corvette c6'
r_2^{c6}: 'chevy corvette c6 navigation'
r_3^{c6}: 'chevrolet corvette c6'
r_1^{z6}: 'corvette z6 navigation'
r_1^{ma}: 'chevy malibu navigation'
r_3^{ma}: 'chevrolet malibu'
r_1^{ci}: 'citroen c6 navigation'

Table 1[1]

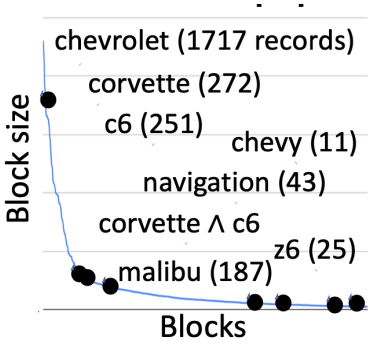

Figure 1[1]

Table 1 contains records where r_i^e refers to the i-th record of entity e. There are four different entities for sample records which are Chevrolet Corvette C6 (c6), Chevrolet Corvette Z6 (z6), Chevrolet Malibu (ma), and Citröen C6 (ci).

Figure 1 is the size of blocks in sample records.

A sample car dataset as is shown in Table 1 corresponding to car models and car descriptions, takes three steps for a common blocking strategy to group the records. The first step is block building. It uses text token t to build different blocks. All descriptions containing t are placed on the same block. The second step is block cleaning, which employs a size threshold to shrink the block. This leads to a trade-off between effectiveness and efficiency. Using a low size threshold could remove useful pairs, causing missing matching pairs whereas using a high size threshold includes too many non-matching record pairs causing inefficiency. The third step is comparison cleaning. Traditional techniques are not capable of creating a small size block that ensures capturing all matching record pairs.

In order to address these limitations, the authors have introduced a new approach, BEER, for performing blocking and ER. The new method begins with blocking aggressively, which is efficient, but not always effective. After computing ER results from the pairs selected by aggressive blocking, it sends these partial results from the ER phase back to the blocking phase, thus producing a loop that improves blocking effectiveness. These feedbacks are used by blocks to assess their qualities and create new blocks intended to capture more matching record pairs. In this way, blocking is gradually self-adjusting and adapting to the characteristics of each record with no manual configuration involved. To obtain a better understanding of how BEER provides solutions, it is necessary to examine this approach in different stages for detail.

Block Building

Block Building returns a number of blocks from input records by assigning each record multiple blocks. As mentioned previously, traditional blocking techniques such as standard blocking generates a separate block for every token t so that all records containing t are part of the block, which produces many large blocks containing plenty of non-matched record pairs. BEER builds a hierarchy of intersection blocks with multiple levels to ensure its efficiency. The first level is the same as the output generated by other technologies, but it has multiple levels whereas the ith level contains the intersection of i different blocks in the first level. The authors called these blocks refined blocks representing the intersection of parent blocks.

Block Cleaning

Block Cleaning is a process that sets a score for each block and cuts back the blocks with the lower scores. There are various criteria for assigning scores. Common technologies such as TF-IDF determine scores regarding the block size but do not consider the block quality. BEER develops a block scoring method that forms informative blocks according to their ability to obtain records from a cluster. In contrast to traditional cleaning algorithms that value block size over other factors, BEER's cleaning algorithm favors blocks having a high fraction of matching pairs and fewer clusters.

Comparison Cleaning

BEERS creates a graph for the input records, ensuring each pair of records in the same block is connected by an edge. Also, every edge must be assigned a weight representing the possibility of matching measured by the weighted Jaccard similarity of the record pair. All lightweight edges are removed from the block. Besides the graph and edge weight, it also calculates the similarity of each pair of records to conduct pairwise comparisons in subsequent steps.

Pair Matching and Clustering

Taking the resemblance of data pairs as input, BEER creates a matching algorithm by prioritizing the pairs. In BEER, one of the current using progressive ER algorithms is available that prioritizes data pairs enhancing the progressiveness of the ER process. BEER uses an active learning-trained random forest classifier for pre-loaded datasets to classify pairs of data as matching or non-matching. As an alternative to training data, BEER offers the capability to use predefined similarity-based rules or manually input new matching rules. To ensure the effectiveness of the pipeline, BEER employs the random graph toolkit. It produces a set of clusters based on the comparison results.

Feedback

BEER employs the union-find data structure to sustain the records of different entities and produces records from each block to estimate the block scores. When the matching result of the pipeline achieves one percent of the new blocked pairs, the feedback of the matching result is sent back to blocking so that blocking can use the feedback to update its blocking scores and revise the structure of blocked pairs.

Performance Comparison

The author tested the performance of BEER on the dataset by presenting a comparative test and setting different parameters as shown in Figure 2. Technique 1 uses feedback whereas Technique 2 does not.

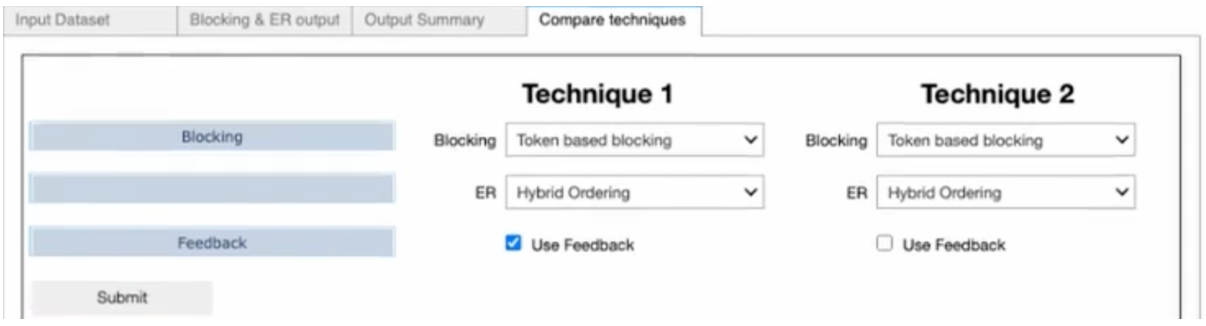

Figure 2[1]

The result (Figure 3) shows that Technique 1 has a higher recall and F-score than Technique 2 does. The F-score is a measure of a model's accuracy on a dataset in statistical analysis. It is

calculated by combining the precision and recall of the test. The precision is calculated by the number of true positive results divided by the number of all positive results. The recall is the number of true positive results divided by the number of all records that should have been recognized as positive [2]. BEER using feedback has better quality.

Quality	Quality	Quality
Precision	0.98	0.98
Recall	0.95	0.65
F-score	0.96	0.78

Figure 3[1]

Figure 4 demonstrates the number of pairs compared during the blocking process. It is obvious that Technique 1 with feedback triggers a smaller number of record comparisons than Technique 2 without feedback.

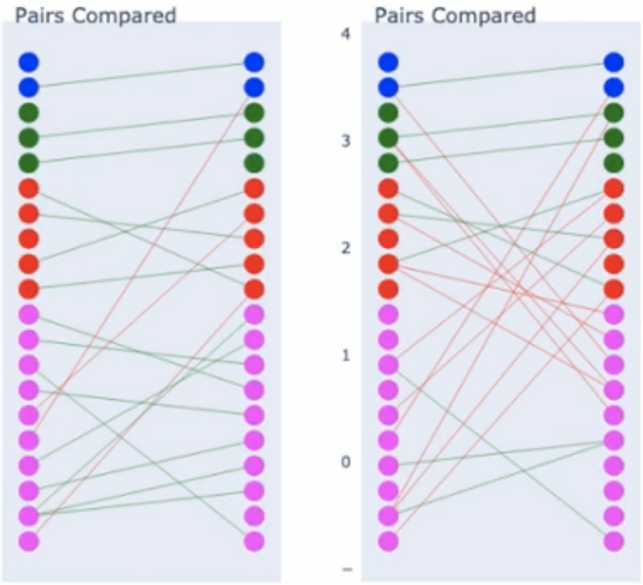

Figure 4 [1]

References

[1] Galhotra, Sainyam, Firmani, Donatella, Saha, Barna, & Srivastava, Divesh.. *BEER: Blocking for Effective Entity Resolution*. *SIGMOD*. Retrieved from https://par.nsf.gov/biblio/10286530.

[2] Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27, 861-874. doi: https://doi.org/10.1016/j.patrec.2005.10.010