The authors describe how a crowdsourcing service such as Amazon’s Mechanical Turk might be better integrated into the workflows of database management. Toward this end they create a query processing system called Qurk that provides a flexible interface and optimization capabilities for crowd-based filtering, join, and sort operators. An experiment is conducted demonstrating that crowdsourcing utilizing this reconfigured workflow reduces costs. The authors describe their approach as the first systematic study of the use of these operators in a crowd-based system.

The join operator asks human users to compare between elements forming joined relations. In the experiment this features as a comparison between two images. The results of these comparisons are used to evaluate the satisfaction of the HIT joined condition. The interface will present the user with the two elements to be joined and the condition predicate for their comparison. These join functions take the form of simple join, naive batching, and smart batching. Optimization of these join operators is pursued through a feature-based (attribute) method by which a user can set conditions that must be matched in order for a join to be established. The performed experiment found all three to work well, with smart batching the most efficient. Additionally, feature-based filtering allowed a cost reduction from $67.50 to $27. The sort operator utilizes crowdsourcing to provide an ordered arrangement of elements. Humans provide pairwise comparative or rating information to the Qurk algorithm so that it might then sort the elements accordingly. The experiment found that ratings provide a cheaper albeit slightly more inaccurate means of crowd-based sorting.

Pros

1. Crowdsourcing via Qurk allows for cost savings through optimization features
2. As the smart batch was found efficient, it reveals that more complex operations are easily performed by crowd-based methods and thus suggests the potential for more powerful approaches using human-computation.
3. Accomplishes HITs that would be much more complex and costly for computational methods alone

Cons

1. Although recognized by the authors, scaling the datasets might offset the efficiencies found using Qurk. As a consequence, this paper provides a conceptual rather than a practical method for crowd-based join/sort.
2. In the join operation, cost reduction was partially achieved by feature filtering. While the images they provided (celebrity images) lend themselves to recognizable features, other datasets may not provide equally usable features and thus crowdsourcing might become less efficient than purely algorithmic operations.
3. The use of crowd-based methods bring in questions of population diversity. For example, celebrity images shown here might be familiar to western workers but not other demographics. While filtering mitigates this problem, the content of different datasets will have a potentially negative effect on efficiency if they present workers with socio-culturally esoteric HITs.

Possible Improvements: The experiments might be strengthened in their external validity if an additional dataset were used (e.g. something other than celebrity pictures).