Different sources may have different claims about the same entities. If we send a query to multiple Web sources, we need to:
- Reconcile references to the same entity
- Fuse different representations of the same entity

Use a cache to store record linkage and fusion results performed on previous query results.
Fuse query results with matching records in the cache to improve results quality.
Index the cache to perform fast look-up and updates at query time.

Effective record linkage and fusion cannot be undertaken query by query in isolation.
- Lack of enough evidence to produce a “good” representation of records
- ORLF takes advantage of previously fused query results to improve future query results
- Caching previous record linkage and fusion results

Similarity Search

- Cache is queried with the query records
- Top-K queries are posted to the cache to retrieve similar records
- B*-tree to index and search the cache
  - Fast lookup
  - Fast updates

Motivation

- Datasets
  - Crawled restaurant listings
  - Synthetic personal information
  - Book listings (http://lunadong.com/fusionDataSets.htm)
- We compared ORLF to two systems:
  - Fusion at query time?
  - How to avoid repeating record linkage and fusion operations across queries?
- Complete set of experiments in the paper.

Query Time Record Linkage and Fusion (ORLF) in a Nutshell

- How to efficiently perform Record Linkage and Fusion at query time?
  - Record linkage and fusion are expensive operations
- How to avoid repeating record linkage and fusion operations across queries?
  - Same linkages may occur across queries
- How to benefit from previous query results to improve RL&F of future queries?
  - Can we do RL&F across queries?

Illustrative Example

Index Key Representation Scheme

- Hash the n-grams separately for each attribute
  - Gram vectors components are matched separately

How to determine the optimal number of buckets for each attribute?

- Trade-off between distinguishability and efficiency
- Each attribute may require a different number of buckets
- M attributes in key
- L buckets

Filter matches

- Use a domain-dependent matching function to filter top k records
  - Different domains have different strategies to determine if two records are duplicates

In the ORLF prototype, we have implemented the matching functions for each dataset we used.

Results

- Datasets
  - Crawled restaurant listings
  - Synthetic personal information
  - Book listings (http://lunadong.com/fusionDataSets.htm)
- We compared ORLF to two systems:
  - Fusion at query time?
  - How to avoid repeating record linkage and fusion operations across queries?
- Complete set of experiments in the paper.

Caching System

- Dynamic cache configurations
  - Cache records are allowed to be updated
  - Eviction policies have to be in place
  - Static
  - Cache is read-only
  - Suitable when cache records have been manually cleaned and do not change in time
  - Static with in-place updates
  - Middle ground

Conclusion

- Query-time record linkage and fusion was made possible by taking advantage of smart caching and query locality.
- ORLF can be applied to different domains.