Introduction

Entity resolution is the task of identifying and matching records that refer to the same entities from different databases. Traditionally, this task is applied in batch-mode and on static databases, for example to find records that relate to the same patient in different health databases, or duplicate records in bibliographic databases. Most research in entity resolution has concentrated on improving the matching quality and scalability to large databases, or reducing the manual efforts required in the entity resolution process.

Many organisations are however increasingly faced with the challenge of having large databases containing entities that need to be matched in real-time with a stream of query records also containing entities, such that the best matching records are retrieved. Example applications include identity verification for online services and benefits, national security databases, and digital libraries.

The aim of real-time entity resolution is to match query records containing entities as fast as possible to one or several databases that contain records about existing entities. The approach must facilitate approximate matching; scale efficiently to large databases that contain many million records; and generate a match score that indicates the likelihood that a matched record in the database refers to the same entity as the one of the query record.

Indexing for Real-time Entity Resolution

The idea of our similarity-aware index is to calculate similarities between unique attribute values once during the build phase, so they don’t need to be calculated for every query record. Different to other approximate query matching techniques, our approach allows any similarity comparison function, and any ‘blocking’ (encoding) function, both possibly domain specific, to be used.

The index consists of three data structures: the block index BI contains encodings and their corresponding attribute values (known as ‘blocks’ in record linkage); the similarity index SI holds the pre-calculated similarities within each block; and the record identifier index RI contains attribute values and their identifiers.

Query records can either refer to an entity stored in the index or to a new, unknown entity. It is assumed that query records can contain variations, errors, out-of-date and missing values. For values not stored in the index, the similarities between attribute values need to be calculated at query time.

For each query record, the matching process returns a ranked list of potential matches and their similarities with the query record. A true match is achieved if one of the top ranked records refers to the same entity as the query record.

Experimental Evaluation

We compared the similarity-aware index approach with standard blocking (basic inverted index) as traditionally used for entity resolution. Both were implemented in Python, and experiments were conducted on an idle Linux server with two quad-core 2.33 GHz 64-bit CPUs and 8 GBytes of memory.

The experiments were conducted using a data set of nearly 7 million records containing surnames, postcodes and suburb (town) names sourced from an Australian telephone directory from 2002. A given name attribute was added to this data set based on a list of around 80,000 unique given names and their frequencies.

To evaluate scalability, we created test data sets of four different sizes containing 10%, 40%, 70% and 100% of the full data set. Both index approaches were queried with five sets of query records containing zero to four manual modifications per record.

Conclusion and Future Work

We presented a novel index approach for real-time entity resolution, and evaluated it experimentally on a real-world data set. The experiments showed that this approach can match query records more than two orders of magnitude faster than a basic index approach traditionally used for entity resolution.

Future work includes improving the accuracy of the proposed approach, a proper analysis of its time and space complexity, improving scalability and query matching time, and conducting experiments on various other large data sets.