CASE STUDY: ENTITY RESOLUTION

In this case study, we will do a case study on entity resolution and blocking, especially in the customer/commercial domain.

Opening prompt / problem statement

Entity Resolution or ER is about determining when references to real-world entities are equivalent (refer to the same entity) or not equivalent (refer to different entities). Linking is appending a common identifier to reference instances to denote the decision that they are equivalent. Identity resolution, record linking, record matching, record deduplication, merge-purge, and entity analytics all represent particular forms or aspects of ER. There are many tradeoffs involved when designing a good ER system, which is still not a solved problem despite many decades of research.

Context and opportunity

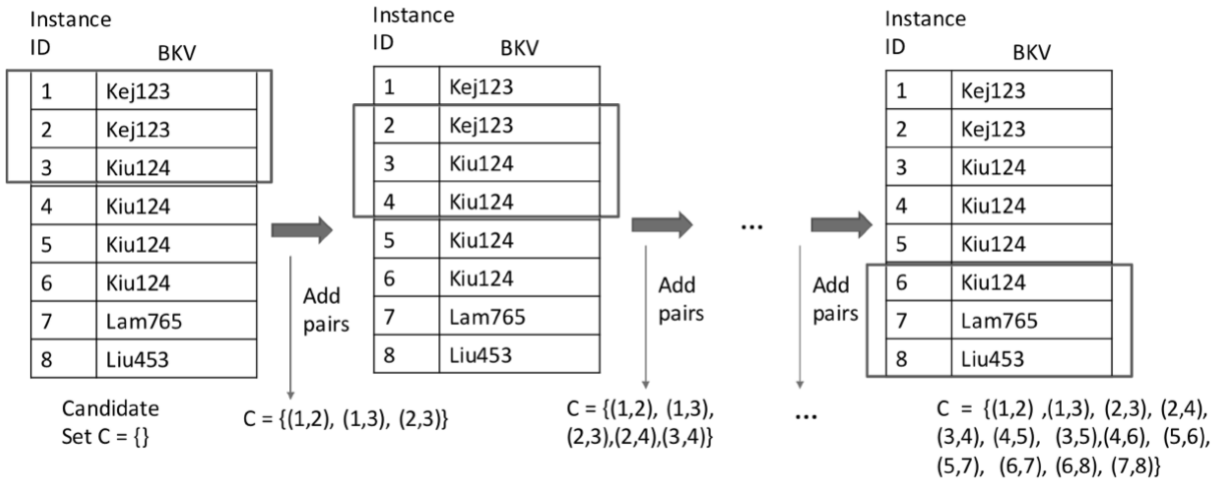
Many businesses need to do ER. Imagine, for example, that you are an e-commerce company like Amazon that has acquired another business. Both the acquired company and Amazon have records of customers who have registered on their website or made purchases in the past. There will likely be many common customers. Some of these customers may have gotten married since, or changed their name etc. In some cases, details may be slightly different, likely because Amazon and the acquired company did not collect data using the same form or mechanism to begin with. To avoid overcounting the customers, and for many other reasons as well, ER will have to be done. In some cases, such as with banks and financial institutions, social security numbers can be used, but such numbers are not always applicable, especially in an international context. With e-commerce, emails could potentially be used, but people have many emails. All of this illustrates the heterogeneity and domain specificity of the problem, especially if near-perfect and scalable solutions are required, as they often are in commercial and medical domains.

Blocking

In class, we discussed several kinds of blocking including sorted neighborhood, traditional blocking and canopies. Make sure to review these. Here are some brief notes. These notes may only make sense in the context of the lecture notes.

8.4.1.1 Traditional Blocking We can generalize the way in which the blocking scheme in example 8.4.2 was generated from a blocking key. Specifically, given a blocking key K, an obvious solution is to generate the candidate set C as the set $\{(m_i, m_j)|m_i, m_j \in M \wedge m_i \neq m_j \wedge K(m_i) \cap K(m_j) \neq \{\}\}$. Note that the definition of C as a set further implies that m_i and m_j may share multiple BKVs. It is only necessary for two mentions to share *at least* one BKV for them to be paired and added to the candidate set. As previously described, C is not guaranteed to be transitive. This makes the method nonrobust, especially to the aforementioned problem of data skew.

Sorted Neighborhood (illustration only):



Canopies:

In the Canopies framework, each canopy represents a block. Concerning the choice of the distance function, the method has been found to work well with (the distance version of) a number of token-based set similarity measures, including Jaccard and cosine similarity (on tf-idf vectors). Such measures are quite robust to a number of issues (e.g., tf-idf based cosine similarity is insensitive to stop words and Jaccard is more sensitive to the number of unique tokens in a text fragment rather than the overall number of tokens, which allows it to discount frequently repeated words). However, token-based measures also have their blind spots, and not every information set or attribute associated with an entity can be decomposed into token sets to begin with. In practice, multiple distance functions and measures may make sense in order to correctly cluster entities with both high precision and recall. It is not completely clear whether one can extend Canopies in a way that seamlessly accommodates multiple functions. A systematic method might be multiview clustering, but at the risk of sacrificing the efficiency and simplicity of the original Canopies algorithm.

Case Questions

i) Which Linked Data principle is most closely related to Entity Resolution and why?

ii) Blocking introduces a tradeoff in an ER pipeline. Discuss the nature of that tradeoff. Under what theoretical situation would we not want to use blocking?

iii) Draw a diagram showing how blocking and similarity inter-relate with one another.

iv) Returning to our customer example in Context/Opportunity, suppose you had to set up a baseline ER system in one day. Describe some string similarity functions you would use for (i) name, (ii) address, (iii) date of birth, (iv) educational status.

(v) What would be an example of a weighted rule you could use to combine outputs from the string similarity function into a probability of a match?

(vi) What would be a good blocking function for this domain?

(vii) How do you evaluate the goodness of blocking? What does goodness even mean in this context? List some metrics, and why they are relevant to blocking.