On Linking Heterogeneous Dataset Collections

Mayank Kejriwal and Daniel P. Miranker

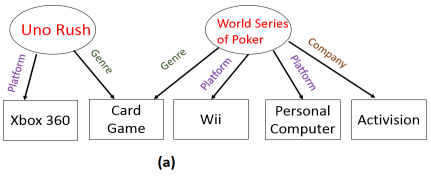
University of Texas at Austin
{kejriwal,miranker}@cs.utexas.edu

Abstract. Link discovery is the problem of linking entities between two or more datasets, based on some (possibly unknown) specification. A blocking scheme is a one-to-many mapping from entities to blocks. Blocking methods avoid $O(n^2)$ comparisons by clustering entities into blocks, and limiting the evaluation of link specifications to entity pairs within blocks. Current link-discovery blocking methods explicitly assume that two RDF datasets are provided as input, and need to be linked. In this paper, we assume instead that two heterogeneous dataset collections, comprising arbitrary numbers of RDF and tabular datasets, are provided as input. We show that data model heterogeneity can be addressed by representing RDF datasets as *property tables*. We also propose an unsupervised technique called *dataset mapping* that maps datasets from one collection to the other, and is shown to be compatible with existing clustering methods. Dataset mapping is empirically evaluated on three real-world test collections ranging over government and constitutional domains, and shown to improve two established baselines.

Keywords: Heterogeneous Blocking, Instance Matching, Link Discovery

With the advent of Linked Data, discovering links between entities emerged as an active research area [2]. Given a link specification, a naive approach would discover links by conducting $O(n^2)$ comparisons on the set of n entities. In the *Entity Resolution* (ER) community, a preprocessing technique called *blocking* mitigates full pairwise comparisons by clustering entities into blocks. Only entities within blocks are paired and compared. ER is critical in data integration systems [1]. In the Semantic Web, the problem has received attention as scalably discovering *owl:sameAs* links between RDF datasets [5].

In the Big Data era, *scalability* and *heterogeneity* are essential components of systems and hence, practical requirements for real-world link discovery. Scalability is addressed by blocking, but current work assumes that the *dataset pairs* between which entities are to be linked are provided. In other words, datasets A and B are input to the pipeline, and entities in A need to be linked to entities in B. Investigations in some important real-world domains show that pairs of dataset *collections* also need to undergo linking. Each collection is a *set* of datasets. An example is government data. Recent government efforts have led to release of public data as batches of files, both across related domains and time, as



Subject	Genre	Platform	Company
World Series of Poker	Card game	Wii; Personal Computer	Activision
Uno Rush	Card game	Xbox 360	*null*

(b)

Fig. 1. The property table representation. For subjects that don't have a property, the reserved keyword *null* is entered. *;* is a reserved delimiter that allows each field value to have set semantics.

one of our real-world test sets demonstrates. Thus, there are (at least) two scalability issues: at the collection level, and at the dataset level. That is, datasets in one collection first need to be mapped to datasets in the second collection, after which a blocking scheme is learned and applied on each mapped pair. The problem of blocking two collections is exacerbated by *data model heterogeneity*, where some datasets are RDF and the others are tabular.

We note that data model heterogeneity has larger implications, since it also applies in the standard case where two datasets are provided, but one is RDF and the other, tabular. In recent years, the growth of both Linked Open Data and the Deep Web have been extensively documented. Datasets in the former are in RDF, while datasets in the latter are typically relational. Because of data model heterogeneity, both communities have adopted different techniques for performing link discovery (typically called *record linkage* in the relational community). There is a clear motivation, therefore, in addressing this particular type of heterogeneity, since it would enable significant cross-fertilization between both communities. We will show an example of this empirically.

The intuition behind our proposed solution to data model heterogeneity is to represent the RDF dataset as an *information-preserving table*, not as a set of triples or a directed graph. The literature shows that such a table has previously been proposed as a *physical* data structure, for efficient implementation of triple stores [6]. An example of this table, called a *property table*, is shown in Figure 1. We note that this is the first application of property tables as logical data structures in the link-discovery context. The table is information-preserving because the original set of triples can be reconstructed from the table.

Note that the property table builds a *schema* (in the form of a set of properties) for the RDF file, regardless of whether it has accompanying RDFS or OWL metadata. Thus, it applies to arbitrary files on Linked Open Data. Secondly, numerous techniques in relational data integration can handle datasets with different schemas (called *structural heterogeneity*). By representing RDF datasets in the input collections as property tables, data model heterogeneity is reduced to structural heterogeneity in the tabular domain.

Figure 2 shows the overall framework of link-discovery. The first step, proposed in this paper for collections, is called the *dataset mapping* step. It takes two collections A and B of heterogeneous datasets as input and produces a set

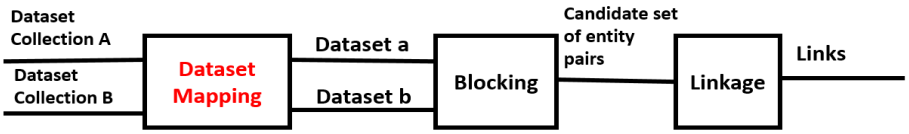


Fig. 2. The overall link-discovery framework. Dataset mapping is our contribution.

of mappings between datasets. Let such a mapping be (a, b) where $a \in A, b \in B$. For *each* such mapping, the subsequent blocking process is invoked. Blocking has been extensively researched, with even the least expensive blocking methods having complexity $O(n)$, where n is the total number of entities in the input datasets. Blocking generates a *candidate* set of entity pairs, Γ, with $|\Gamma| << O(n^2)$. Thus, blocking provides complexity improvements over brute-force linkage. To understand the savings of dataset mapping, assume that each collection contains q datasets, and each dataset contains n entities. Without dataset mapping, any blocking method would be at least $O(qn)$. With mapping, there would be q instances of complexity $O(n)$ each. Since Γ depends heavily on n, the savings carry over to the final quadratic process (but which cannot be quantified without assumptions about the blocking process). We empirically demonstrate these gains. An added benefit is that there is now scope for parallelization.

The mapping process itself relies on *document similarity* measures developed in the information retrieval community, by representing each dataset as bag of tokens. Intuitively, mapped datasets should have relatively high document similarity to each other. Empirically, we found a tailored version of cosine similarity to work best. Many packages exist for efficiently computing it. Computing similarities between all pairs of datasets, we get a $|A| \times |B|$ matrix. A straightforward approach would use a threshold to output many-many mappings, or a graph bipartite matcher to output one-one mappings. The former requires a parameter specification, while the latter is cubic ($O(q^3)$). Therefore, we opted for a *dominating strategy*, which can be computed in the same time it takes to build the matrix. Namely, a mapping (a, b) is chosen if the score in the cell of (a, b) *dominates*, that is, it is the highest in its constitutent row and column. This has the advantage of being conservative against false positives. The method applies even when $|A| \neq |B|$. In our experiments, we used cosine document similarity combined with the dominating strategy.

Experiments: Some results are demonstrated in Figure 3. We use three real-world test cases. The first two test cases (a and b in the figure) comprise RDF dataset collections describing court cases decided in Colombia and Venezuela respectively, along with Constitution articles. The third test set consists of ten US government budget dataset collections from 2009 to 2013 [1]. Other such collections can also be observed on the same website, providing motivation for dataset mapping. We have released publicly available datasets on a single page[2] with

[1] `http://www.pewstates.org/research/reports/`
[2] `https://sites.google.com/a/utexas.edu/mayank-kejriwal/datasets`

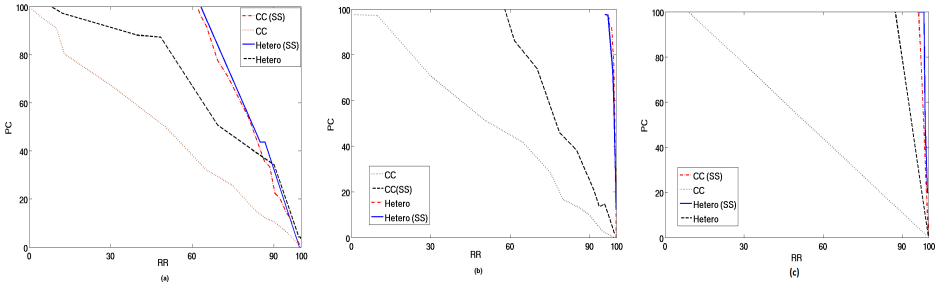


Fig. 3. *RR* (Reduction Ratio) quantifies *efficiency* and is given by $1 - |\Gamma|/|\Omega|$, with Ω the set of all $O(n^2)$ entity pairs, while *PC* (Pairs Completeness) measures *recall* of Γ with respect to the ground-truth set, Ω_m ($\subseteq \Omega$). *SS* indicates if dataset mapping (equivalently denoted *Source Selection*) was used.

ground-truths. We used two popular methods as baselines, a state-of-the-art unsupervised clustering method called *Canopy Clustering* (*CC* in figure) [4] as well as an extended feature-selection based blocking method (*Hetero* in figure) [3]. The gains produced by dataset mapping are particularly large on *CC*. More importantly, we found that the dataset mapping algorithm was able to deduce the correct mappings without introducing false positives or negatives, and with run-time negligible compared to the subsequent blocking procedures.

Future Work: We continue to investigate dataset mapping, including other document similarity measures, task domains and mapping strategies. We are also investigating supervised versions of the problem, particularly in cases where token overlap is low. Finally, we are investigating the property table further.

References

1. P. Christen. *Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection*. Springer, 2012.
2. R. Isele and C. Bizer. Learning expressive linkage rules using genetic programming. *Proceedings of the VLDB Endowment*, 5(11):1638–1649, 2012.
3. M. Kejriwal and D. P. Miranker. An unsupervised algorithm for learning blocking schemes. In *Data Mining, 2013. ICDM'13. Thirteenth International Conference on*. IEEE, 2013.
4. A. McCallum, K. Nigam, and L. H. Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 169–178. ACM, 2000.
5. F. Scharffe, Y. Liu, and C. Zhou. Rdf-ai: an architecture for rdf datasets matching, fusion and interlink. In *Proc. IJCAI 2009 workshop on Identity, reference, and knowledge representation (IR-KR), Pasadena (CA US)*, 2009.
6. K. Wilkinson, C. Sayers, H. A. Kuno, D. Reynolds, et al. Efficient rdf storage and retrieval in jena2. In *SWDB*, volume 3, pages 131–150, 2003.