ABSTRACT

Entity resolution aims to identify descriptions of the same entity within or across knowledge bases. In this work, we present the Minoan ER platform for resolving entities described by linked data in the Web (e.g., in RDF). To reduce the required number of comparisons, Minoan ER performs blocking to place similar descriptions into blocks and executes comparisons to identify matches only between descriptions within the same block. Moreover, it explores in a pay-as-you-go fashion any intermediate results of matching to obtain similarity evidence of entity neighbors and discover new candidate description pairs for resolution.

1. DESCRIPTION

Over the past decade, numerous knowledge bases (KBs) have been built to power large-scale knowledge sharing, but also as an entity-centric Web search, mixing both structured data and text querying. These KBs offer comprehensive, machine-readable descriptions of a large variety of real-world entities (e.g., persons, places, products, events) published on the Web as Linked Data (LD). Although KBs (e.g., DBpedia, Freebase) may be derived from the same data source (e.g., a Wikipedia entry), they may provide multiple, non-identical descriptions of the same real-world entities. This is mainly due to the different information extraction tools and curation policies employed by KBs, resulting to complementary and sometimes conflicting entity descriptions. Entity resolution (ER) aims to identify descriptions that refer to the same real-world entity appearing either within or across KBs [2, 3]. Compared to data warehouses, the new ER challenges stem from the openness of the Web of data in terms of number of entity types and properties) and the importance of relationships among entity descriptions (not committing to a particular schema defined in advance). In particular, in addition to highly similar descriptions that feature many common tokens in values of semantically related attributes, typically met in the center of the LOD cloud and heavily interlinked mostly using owl:sameAs predicates, we are encountering somehow similar descriptions with significantly fewer common tokens in attributes not always semantically related, that appear usually in the periphery of the LOD cloud and are sparsely interlinked with various kinds of predicates. Plainly, the coming up of highly and somehow similar semi-structured entity descriptions requires solutions that go beyond those applicable to duplicate detection. A promising area of research in this respect is cross-domain similarity search and mining [8, 7], aiming to exploit similarity of objects described by different modalities (i.e., text,
Blocking and Meta-blocking in Minoan ER: We use blocking as a pre-processing step for ER to reduce the number of required comparisons. Specifically, blocking places similar entity descriptions into blocks, leaving to the entity matching algorithm the comparisons only between descriptions within the same block.

Typically, token-based blocking algorithms place highly similar descriptions (having many common tokens) in many common blocks; intuitively, the more common blocks two descriptions share, the more likely it is that they match. This leads to many repeated comparisons between the same pairs of descriptions. To overcome this problem, we accompany blocking with meta-blocking, which prunes such repeated comparisons. Moreover, meta-blocking aims at discarding comparisons between descriptions that share few common blocks and are thus less likely to match. In Minoan ER, to support a Web-scale resolution of heterogeneous and loosely structured entities across domains, we use algorithms for blocking and meta-blocking that disregard strong assumptions about knowledge of the data schema and rely on a minimal number of assumptions about how entities match (e.g., when they feature a common token in their descriptions or URIs) within or across sources. For doing so, we exploit the parallel processing power of a computer cluster via Hadoop MapReduce, as presented in [5, 4].

Progressive Entity Matching in Minoan ER: Blocking approaches in the Web of data, especially when handling somehow similar descriptions appearing in the periphery of the LOD cloud, may miss highly heterogeneous matching descriptions featuring few common tokens [5]. To overcome that, we focus on exploiting the partial matching results as a similarity evidence for their neighbor (i.e., linked) descriptions. Since this inherently iterative process entails an additional overhead, we are interested in maximizing its benefit, given a computational cost budget. So, we need to estimate which part of the graph is the most promising to explore in the next iteration, in a progressive way.

In this respect, Minoan ER focuses on extending the typical ER workflow with a scheduling phase, which is responsible for selecting which pairs of descriptions, that have resulted from blocking, will be compared in the entity matching phase and in what order. The goal of this new phase is to favor more promising comparisons, i.e., those that are more likely to increase the targeted benefit. This way, those comparisons are executed before less promising ones and thus, higher benefit is provided early on in the process. The update phase propagates the results of matching, such that a new scheduling phase will promote the comparison of pairs that were influenced by the previous matches. This iterative process continues until the cost budget is consumed.

In contrast to existing works in progressive relational ER (e.g., [1]), which consider the quantity of entity pairs resolved, as the benefit of ER, we explore different aspects of data quality, improved through ER. In particular, we are interested in characterizing the quality of the resolved pairs, with respect to the number of descriptions resolved, corresponding to the same real-world entity (targeting attribute completeness), the number of real-world entities resolved (targeting entity coverage), and the number of real-world entity graphs resolved (targeting relationship completeness).

Acknowledgements: This work was partially supported by the EU H2020 PARTHENOS (#654119) and FP7 SemData (#612551) projects.

2. REFERENCES