Generating corrupted data sources for the evaluation of matching systems

Fiona McNeill[1][0000−0001−7873−5187], Diana Bental[1][0000−0003−3834−416X], Alasdair J G Gray[1][0000−0002−5711−4872], Sabina Jedrzejczyk[1], and Ahmad Alsadeeqi[1]

Heriot-Watt University, Edinburgh, Scotland
{f.mcneill, d.bental,a.j.g.gray,sj22,aa1262}@hw.ac.uk

Abstract. One of the most difficult aspects of developing matching systems – whether for matching ontologies or for other types of mismatched data – is evaluation. The accuracy of matchers are usually evaluated by measuring the results produced by the systems against reference sets, but gold-standard reference sets are expensive and difficult to create. In this paper we introduce crptr, which generates multiple variations of different sorts of dataset, where the degree of variation is controlled, in order that they can be used to evaluate matchers in different context.

Keywords: Matching · Evaluation · Data Corruption.

1 Introduction

One of the central problems of data matching is the issue of evaluation: when a system returns a set of matches, how are we to determine whether they are correct or not? How exactly do we define what a correct match is, and how do we determine whether the proposed matches fall into that category? If we have a range of different options, how do we determine which is the 'best' match?

In this paper we describe the use of the crptr system to create evaluation datasets for matching. crptr was developed to simulate data quality issues for test datasets used for record linkage evaluation. It can create multiple similar datasets with varying amounts of variation controlled by input settings, and provides a clear mapping back to the original dataset. This creates training and evaluation sets for matchers to run against. We have extended the crptr system to deal with structure in a context where we want to corrupt data sources in order to evaluate the semantic rewriting of queries to unknown data sources.

In Section 2 we describe the crptr system and its original application domain. Section 3 then details how we extended crptr to address corruption of other data sets and of queries. We discuss issues around evaluation in Section 4 and touch on related work in Section 5 before concluding the paper in Section 6.

2 The crptr system

Synthetically generated data is a common approach for evaluating and testing data analysis and mining approaches [9]. However, the use of synthetically generated data fails to capture the messiness of real world data, i.e., they omit data quality issues [5]. To overcome this we developed crptr: a data corruption application that injects data errors and variations based on user requirements. crptr allows the user to control data quality in the generated dataset by simulating and injecting data corruptions into any dataset using known (non-random) methods that mimic real-world data quality issues (errors and variations). Applying these corruptions on a synthetic dataset enables the control of data quality, which makes the synthetic data more realistic and usable for evaluations. crptr contains many corruption methods that mimic commonly found data quality issues, e.g., typing errors, alternative spellings, and missing or swapped attributes, that can be used to simulate different corruption scenarios based on the experiment or project requirements.

crptr works by using a *corruption profile* that controls which methods are used and how much. The idea is that the profile attempts to capture the data quality characteristics of the dataset being modelled. The corruption profile consist of many factors that define the way data need to be corrupted such as the total number of records that need to be corrupted and the corruption methods required to be applied on the data. By controlling the factors of the corruption profile, the user can configure crptr to mimic the data quality characteristics that fit the purpose of the research.

3 Application of crptr to Query Rewriting

The CHAIn system (Combining Heterogenous Agencies' Information) [7] has been designed to support users to successfully query data from a wide range of different data sources, even when these data sources are not known in advance (e.g., data sources of new collaborators). It is primarily aimed at supporting decision makers during crisis response, but is applicable in many domains. Any queries pre-written to extract required information are likely to fail (i.e., not return any data) on these unknown or updated data sources because the queries were not written according to the structure and terminology of the target data source. However, the data sources may well have relevant information that is closely related to the query. CHAIn extracts data from the target source that approximately matches the query (i.e., exceeds a given threshold) and uses this data to rewrite the query so that it succeeds on the datasource. It returns (potentially) multiple rewritten queries, the mappings used to generate them, the data they retrieved and a similarity score $\in [0, 1]$, ranked in order of similarity.

Evaluation in this context therefore means determining whether the scores returned are a reasonable reflection of the distance between the original query and the rewritten query, and hence whether the ranked list is a reasonable ordering of the likely relevance of the responses to what the decision maker actually

wants to know. In this context, the matching is done between the schema of the query and the schema of the target datasource[1]. In order to mimic the process of rewriting a query designed for one data source to succeed on a different data source, we create a query based on the data in a particular data source (i.e., so that it would be able to successfully query that data source) and then introduce corruption reflecting naturally-occurring differences. We can either keep the query fixed and corrupt the data source in multiple ways, or keep the data source fixed and corrupt the query. In practice, we focused on corrupting datasources and then generating corrupted queries from these corrupted datasources - firstly, because it created a more generic process that was able to corrupt both datasources and queries; secondly, because it allows us to more easily focus on the part of the query that is relevant in this context, which is the terminology referring to the target datasource.

We therefore needed to extend the functionality of crptr in two ways. (*i*) We need to consider the domain in which this matching is occurring to determine how terms should be corrupted; (*ii*) Because there is a structural element to schema, we need to consider how this could be corrupted and extend the system to perform this.

In terms of the first requirement, some of the corruptions methods in crptr (e.g., those focusing on spelling errors) are not relevant, whilst others such as abbreviations, need to be adapted, as some kinds of abbreviations (e.g., of first names) are unlikely to occur in our data sources. We need to determine what kinds of mismatches are likely to occur in our domain, and determine what sources we can automatically extract them from. CHAIn is designed to be domain independent, and when addressing the problem of matching different (but similar) data sources in the general case, we need a domain-independent lexical resource to suggest the kinds of synonyms, hyponyms, hypernyms and meronyms that different creators of data sources in a similar domain may naturally use. We therefore turned to WordNet [8], a generic and widely used lexical resource, to allow us to do term corruption. WordNet does provide some information about abbreviations and acronyms which we are able to use in our matching, although additional resources that provide more relevant corruptions in this area would improve performance (but are hard to find).

In terms of the second requirement, we needed to make sure any potential structural change in the schema of a CSV file was considered. This is structurally simple, consisting of columns which are named and ordered, and thus structural changes are restricted to reorganisation (addition, deletion and reordering) of the columns. For SPARQL queries in general there are, of course, many more structural elements (e.g, the potential presence of SPARQL commands such as aggregate functions), and a complete list of potential structural mismatches would be more complicated. As we are only concerned with the terms in the query which correspond with those of the expected data source, we can ignore all of the additional SPARQL structure, stripping out the relevant terms and reinserting the new terms after matching.

[1] Matching at the data level is required when queries are partially instantiated.

4 Evaluation of crptr for different data formats

The quality of the crptr output depends on whether the corrupted data sources it produces are a reasonable facsimile of different but related data sources that would naturally be found. If this is the case then we can infer that the performance of a matching system when matching different data sources created by crptr is a good indication of the matchers performance in real-world settings, and that therefore crptr is a useful matching evaluation tool.

This depends on two things: (*i*) are the terms in the look up table a good approximation of terms that could be used interchangable or in a similar way: is it modelling genuine *semantic* and *syntactic* mismatches?; (*ii*) are the structural mismatches introduced through the corruption process a good approximation of how similar data sources may differ? The first is highly domain dependent. We use WordNet, which is a very widely used lexical resource. It is also likely to be of benefit to also use domain-specific ontologies and lexicographies for each particular domain; however, these are hard to find and often of questionable quality, so this kind of domain-specific corruption may be hard to perform. Matching in such domains is also more efficient for the same reasons. The second aspect is domain independent but format specific. For each format the system is extended to, an analysis of what structural mismatches are possible is necessary in order to demonstrate that the corruptions produced are plausible and thorough.

5 Related work

To the best of our knowledge, a system to generate reference sets (records, queries, RDF data sources, ontologies, etc) in order to evaluate matching in these domains is unique.

Since reference ontologies are expensive to generate and often not available, [6], automatically generated test sets have been used to evaluate ontology matching since the Benchmark Test Set was developed for the Ontology Alignment Evaluation Initiative in 2004 and updated in 2016 [3]. Several other generators were inspired by this, including Swing [4]. These tend to focus on OWL ontologies and are less broadly applicable than crptr. The range of methods they use are in some cases more sophisticated than our techniques, and in domains for which they are relevant, crptr could be improved by incorporated such approaches.

Aside from ontology matching, there is existing work on generating synthetic datasets with structural variations for relational and RDF data for use in benchmarking. The Linked Data Benchmark Council [2] has supported the development of configurable and scalable synthetic RDF datasets with similar irregularities to real data, including structural irregularities, specifically in the domains of social networks and semantic publishing. Existing work on generating structural variatons in RDF data (e.g. [2]) is intended to test the functionality and scalability of searches and the maintenance of RDF datasets. STBenchmark [1] generates test cases for schema mapping systems, taking an original dataset and applying structural and term variations. This is used to create benchmark

data for hand-mapping systems rather than for automated matching or querying. Our work could be extended with similar strategies to these to experiment with greater structural variations.

6 Conclusions

In this paper we have discussed using the crptr system for generating multiple similar datasets for evaluating matchers within different domains. We briefly described how crptr was developed to focus on records and then extended to deal with queries based on CSV files, and could be extended to deal with other kinds of data sources. We discussed what evaluation of these corruption systems means in different contexts.

References

1. Alexe, B., Tan, W.C., Velegrakis, Y.: Stbenchmark: towards a benchmark for mapping systems. Proceedings of the VLDB Endowment **1**(1), 230–244 (2008)
2. Angles, R., Boncz, P., Larriba-Pey, J., Fundulaki, I., Neumann, T., Erling, O., Neubauer, P., Martinez-Bazan, N., Kotsev, V., Toma, I.: The linked data benchmark council: a graph and rdf industry benchmarking effort. ACM SIGMOD Record **43**(1), 27–31 (2014)
3. Euzenat, J., Rooiu, M.E., Trojahn, C.: Ontology matching benchmarks: Generation, stability, and discriminability. Journal of Web Semantics **21**, 30 – 48 (2013). https://doi.org/https://doi.org/10.1016/j.websem.2013.05.002, http://www.sciencedirect.com/science/article/pii/S1570826813000188, special Issue on Evaluation of Semantic Technologies
4. Ferrara, A., Montanelli, S., Noessner, J., Stuckenschmidt, H.: Benchmarking matching applications on the semantic web. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) The Semanic Web: Research and Applications. pp. 108–122. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
5. Hernández, M.A., Stolfo, S.J.: Real-world data is dirty: Data cleansing and the merge/purge problem. Data mining and knowledge discovery **2**(1), 9–37 (1998)
6. Ivanova, V., Bach, B., Pietriga, E., Lambrix, P.: Alignment cubes: Towards interactive visual exploration and evaluation of multiple ontology alignments. In: d'Amato, C., Fernandez, M., Tamma, V., Lecue, F., Cudré-Mauroux, P., Sequeda, J., Lange, C., Heflin, J. (eds.) The Semantic Web – ISWC 2017. pp. 400–417. Springer International Publishing, Cham (2017)
7. McNeill, F., Gkaniatsou, A., Bundy, A.: Dynamic data sharing from large data sources. In: Proceedings of 11th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2014) (2014)
8. Miller, G.A.: Wordnet: A lexical database for English. Commun. ACM **38**(11), 39–41 (Nov 1995). https://doi.org/10.1145/219717.219748, http://doi.acm.org/10.1145/219717.219748
9. Tran, K.N., Vatsalan, D., Christen, P.: Geco: an online personal data generator and corruptor. In: Proceedings of the 22nd ACM international conference on Information & Knowledge Management. pp. 2473–2476. ACM (2013)