Modeling and querying possible repairs in duplicate detection

Robert Surówka

1. Overview

The main article describes a method of dealing with duplicate records in an unclean database. The approach presented is to find some subset of possible deterministic repairs (where repair is produced by determining which records are duplicates and merging them in some arbitrary way (not discussed by authors)), store them in a convenient format, and then answer queries using them. The subset is found by employment of slightly modified versions of hierarchical clustering algorithms, of which two were used by the authors: Hierarchical Linkage-based Clustering and Hierarchical Nearest Neighbor Clustering. Those found repairs, which are deemed to be "reasonable", are saved in a lossless way as a U-Clean Relation (together with computed probabilities of them being the correct ones and history from merging of what records each of them was obtained), which also allows efficient answering of some queries, like e.g. select-type ones. Relational queries are defined over U-Clean Relation using the concept of possible worlds semantics. Also new types of queries were introduced, which allow user to take into account meta-information acquired during cleaning process which is mainly probability of the repairs. Also some algorithms were proposed to answer queries over multiple possible repairs in shorter time, main concern were aggregation-type queries and queries over views produced by other queries in this setting, yet some of those algorithms were only mentioned. In overall this paper presents new method of performing de-duplication of a datasets which retains far more information about the input unclean instance than method used for comparison by authors, allows easy back-tracking of from what merges answer to any query was produced and gives to the user ability to formulate queries that are able to take into consideration the uncertainty of the set of found repairs. Basing on tests performed by prototypical implementation of the method it's usage proves to be practically feasible.

Another issue presented stems from the second article, and concerned work that is in progress. The problem is: given an XML Document and a DTD that this document is not conformant to find a repair of it that would be in that DTD and also won't "corrupt" the information in the given document. To define what "corruption" means, user would be given a language through which imposing constraints (like e.g. don't delete a node if it has a child of some type) on a repair process would be done. Then "corruption" of data would be equal to breaking any of those constraints, therefore a repairing algorithm would have to find a repair following all user's restriction or answer that it's impossible. Due to at least exponential time complexity of even determining if the repair in given setting is possible to obtain, an approximate or probabilistic algorithm is looked for.

2. Detailed comments

Comparing the method developed by the authors to the one presented as contrast in their paper, the former in my opinion is significantly superior. Yet there are some drawbacks that I believe should be stressed more in the article. Firstly, the human input, through

parameterization, to the repairing process is huge. Values and methods that must be chosen arbitrary (basing on input unclean database) include: metric for computing distances between records, minimum distance between any pair of records guaranteeing that they represent different entities and probability function (defined in passage 3.1). Mistake in fixing the first two of them can result in wrong repairs (that is less probable repairs would be found by the algorithm, while the really probable ones would be ignored) while mistake in any of the three of those will result in wrongly computed probability rates for found repairs. Furthermore because the clustering algorithms used are heuristic, even with a best practically possible parameterization it is possible to not to find the best possible repairs. Those concerns were the main relevant issues raised during discussion of the paper. Others that were brought up concerned clarification of the presented method or issue why to use any method like this at all while maybe one could just foresee all possible scenarios needing repair and write an algorithm that would be specifically prepared for all of them – which at this point of technological advancement would practically mean performing more work than just manually repairing all the datasets that need cleaning.

The core ideas of the paper are easy to follow and make perfect sense. The flow of the article is intuitive, and the issues discussed neatly stem from each other eventually creating a well formed body. Some parts of implementation section seemed to be not transparent enough and hard to understand to me but I think I should blame myself for it, not consider it a drawback of the paper. What is a little troublesome yet, is the fact that on many occasions authors refer reader to different articles even when a subject referred to seems to be quite an internal part of the method presented. But taking into consideration that enclosing all those issues would greatly lengthen the article and they seem to be complicated and to high degree self-enclosed is in my opinion a sufficient argument for such a choice. The depth of the paper is fully satisfying if a minor flaw mentioned above and not stressing enough drawbacks of proposed method are excused. But I believe that any reader with sufficient background would easily identify the latter, what also happened during discussion of the article. In paper authors compare their work with a deterministic approach for finding and merging duplicates. It is clearly shown that their method supplies user with far vaster information. It not only makes the process of dealing with uncertainties significantly less likely to conceal the right data but also lets user employ the metadata accumulated by the algorithm to further enhance odds of choosing the right repair. On the other hand the proposed method consumes more space and time than the original one (especially when it comes to producing queries' results), but taking into account the advantages it has over its predecessor in my opinion it is not a too high price to pay.

Due to a brief presentation of the second set of slides the discussion was slight, which mostly came down to the introduction of XML and DTD concepts and some clarification of the arguments presented – particularly the need behind idea of introducing constraints to the problem.

Because of being one of the authors of the XML-repairing work presented I don't feel it would be appropriate for me to grade how deep or sound it is.