KATARA: Reliable Data Cleaning with Knowledge Bases and Crowdsourcing

Xu Chu[1*] John Morcos[1*] Ihab F. Ilyas[1*]
Mourad Ouzzani[2] Paolo Papotti[2] Nan Tang[2] Yin Ye[3*]

[1]University of Waterloo [2]Qatar Computing Research Institute [3]Google

{x4chu, jmorcos, ilyas}@uwaterloo.ca {mouzzani, ppapotti, ntang}@qf.org.qa yye@google.com

ABSTRACT

Data cleaning with guaranteed reliability is hard to achieve without accessing external sources, since the truth is not necessarily discoverable from the data at hand. Furthermore, even in the presence of external sources, mainly knowledge bases and humans, effectively leveraging them still faces many challenges, such as aligning heterogeneous data sources and decomposing a complex task into simpler units that can be consumed by humans. We present KATARA, a novel *end-to-end* data cleaning system powered by knowledge bases and crowdsourcing. Given a table, a KB, and a crowd, KATARA (i) interprets the table semantics *w.r.t.* the given KB; (ii) identifies correct and wrong data; and (iii) generates top-k possible repairs for the wrong data. Users will have the opportunity to experience the following features of KATARA: (1) *Easy specification:* Users can define a KATARA job with a browser-based specification; (2) *Pattern validation:* Users can help the system to resolve the ambiguity of different table patterns (*i.e.,* table semantics) discovered by KATARA; (3) *Data annotation:* Users can play the role of internal crowd workers, helping KATARA annotate data. Moreover, KATARA will visualize the annotated data as correct data validated by the KB, correct data jointly validated by the KB and the crowd, or erroneous tuples along with their possible repairs.

1. INTRODUCTION

Many attempts have been made to improve data quality using, for example, integrity constraints [1, 6], statistics [8], or machine learning [9]. However, it is usually hard, if not impossible, to guarantee the accuracy of the data cleaning process without verifying it via experts or external sources [5].

The advent of knowledge bases (KBs), both general-purpose and within enterprises, and crowdsourcing marketplaces are providing new opportunities to achieve higher accuracy at a larger scale. However, there is no *end-to-end* data cleaning system that would effectively bridge KBs and crowdsourcing, thus enabling reliable data cleaning for a wide range of applications.

We recently proposed KATARA [2], a novel *end-to-end* data cleaning system that effectively leverages prevalent trustworthy KBs and crowdsourcing for data cleaning. Its main functionalities are to interpret table semantics, identify correct and wrong data, and generate top-k possible repairs for wrong data. This demo will demonstrate the following three key features of this system.

(1) *Easy specification.* KATARA provides a succinct GUI that allows users to declare the target table and the reference KB. Moreover, KATARA offers to use the crowd either through a local crowd platform or through Amazon MTurk.

(2) *Pattern validation.* Given the selected table and the KB, KATARA will compute and identify top-k table patterns, which will then be visualized to the users via a simple graphical representation. Since it is hard for a system to automatically pick the best table pattern, the users, by studying the given table, may select the most appropriate table pattern that explains the semantics of the table.

(3) *Data annotation.* Once the table pattern is selected, KATARA scans all the tuples and for each tuple, it marks the tuple as correct if the information in the KB covers all values in the tuple, *i.e.,* the KB is complete vis-à-vis the tuple. Otherwise, since the KB cannot cover all values in the given tuple, an ambiguity is raised about whether this is caused by the incompleteness of the KB or the data is simply wrong. To resolve such ambiguity, KATARA will post questions to the crowd. After getting answers from the crowd, KATARA can more accurately annotate the data.

Related Work. Table understanding, including identifying column types and the relationship between columns, has been addressed by several techniques [7]. KATARA differs from them in two main aspects. First, we focus on finding coherent table patterns for the purpose of data cleaning. In other words, instead of explaining table semantics at the schema level, we need to find table patterns that can align information at the instance level. Second, existing techniques do not explicitly assume the presence of dirty data.

The widely studied integrity constraint (IC) based data cleaning aims to find a consistent database that satisfies the given ICs with a minimum cost. The corresponding heuristic solutions do not usually ensure full accuracy [5]. To address such a shortcoming, several approaches have been proposed such as involving experts as first-class citizen [10]

*Work partially done while at QCRI.

Proceedings of the VLDB Endowment, Vol. 8, No. 12

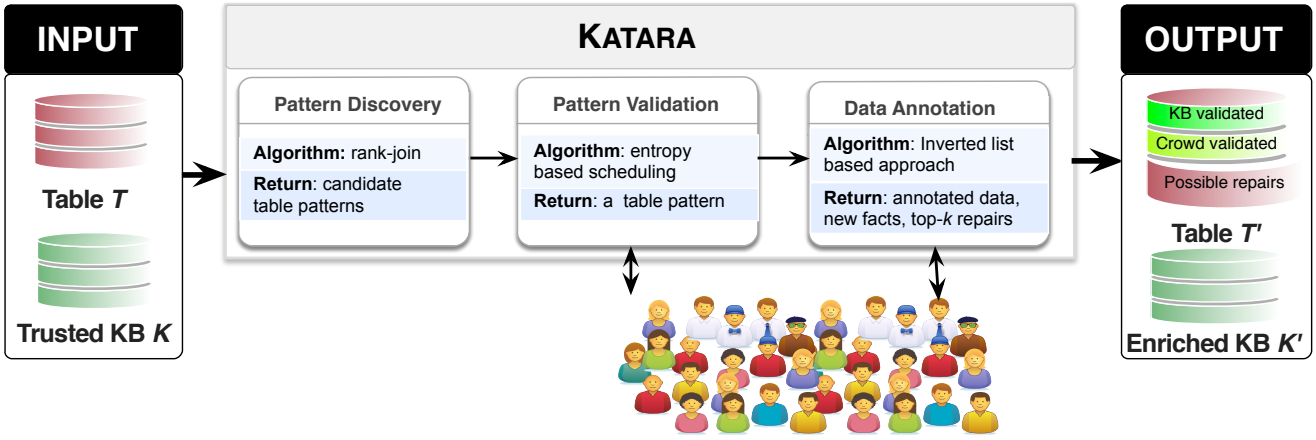

Figure 1: Workflow of KATARA

leveraging high quality reference data [5], and using user-provided confidence values [4]. KATARA stands from these approaches along two aspects: (i) It does not require experts to give high quality data quality rules as input and (ii) it does not require experts to guide the repair process. Instead, KATARA explores a crowd of experts using a general pay-as-you-go approach, leading to lower costs, and leverages KBs, either those readily available in an enterprise setting [3] or general KBs such as Yago and DBpedia.

In this work, we assume that both KBs and expert sourcing are more trusted than the data we have at hand. Improving the accuracy of KBs or the crowd is orthogonal to KATARA, and much effort has been spent on this aspect [3]. Furthermore, some KBs, especially in enterprise settings, are usually carefully curated with considerable manual work. Though not error-free, these are much more reliable than the data we have at hand and thus can be treated as relatively trusted resources. In addition, we do not assume that the KBs to be complete since this is not the case in most practical cases.

2. KATARA ARCHITECTURE

KATARA has three modules (see Fig. 1), namely, pattern discovery, pattern validation, and data annotation. The *pattern discovery* module discovers table patterns between a table and a KB. The *pattern validation* module allows users to select the best table pattern. Using the selected table pattern, the *data annotation* module interacts with the KB and the crowd to annotate the data. It also generates possible repairs for the erroneous tuples. Moreover, new facts verified by the crowd are used to enrich KBs.

(1) Pattern discovery. KATARA first discovers table patterns that contain the types of the columns and the relationships between a table and a KB. A table pattern is represented as a labeled graph where a node represents an attribute and its associated type. A directed edge between two nodes represents the relationship between two attributes.

Note that a relational schema may not be easily aligned with an ontology for reasons such as cryptic naming conventions. We use an *instance based* approach to discover table-KB mappings. Such an approach does not require the availability of meaningful column labels. For each column A_i of table $\mathcal{T}$ and for each value $t[A_i]$ of a tuple t, we map that value to several resources in the KB $\mathcal{K}$ whose type can then be extracted. To this end, we issue SPARQL queries that return the types and supertypes of entities whose label (*i.e.,* value) is $t[A_i]$. The relationships between two values are retrieved in a similar fashion. To rank candidate types of a column A_i, we use a normalized version of tf-idf. Also, to avoid enumerating all candidates, we rely on early termination with a rank-join formulation of the problem.

(2) Pattern validation. Depending on the data at hand and the complexity of the reference KB, the number of candidate table patterns can vary from a handful of patterns to dozens. If the number of the candidate table patterns is small, as will be demonstrated in this demo, we simply visualize them for the users to pick the right table pattern for the table at hand. We assume that the users can easily understand the tuples of the table relative to the reference KB. On the contrary, if the size of the candidate table patterns is large, we have proposed in [2] methods to decompose a table pattern into smaller patterns. We then use such small patterns to formulate simpler (sometimes binary) questions, which crowd workers are known to be good at answering.

(3) Data annotation. Given a table pattern, KATARA annotates each tuple with one of the following three labels:

- *Validated by the* KB $\mathcal{K}$. If we match a tuple to $\mathcal{K}$ over all the attributes in the pattern, the tuple is semantically correct *w.r.t.* the table pattern and the KB.
- *Jointly validated by the* KB *and the crowd.* If there is only a partial match from a tuple to $\mathcal{K}$, either $\mathcal{K}$ is incomplete or the tuple is simply erroneous. To find out we ask the crowd to verify the non-covered data.
- *Erroneous tuple.* For the erroneous tuple confirmed by the crowd, KATARA extracts information from $\mathcal{K}$ and join them to generate a set of possible repairs for this tuple.

In general, the number of possible repairs for an error can be large. Most automatic repair algorithms use minimality as a guiding principle to pick among multiple repairs that make the tuple conforming to the patterns. The intuition is that a repair with a smaller number of changes is preferable to others with more changes, as less changes preserve more values from the original instance. We thus rank possible repairs based on the number of changes in ascending order.

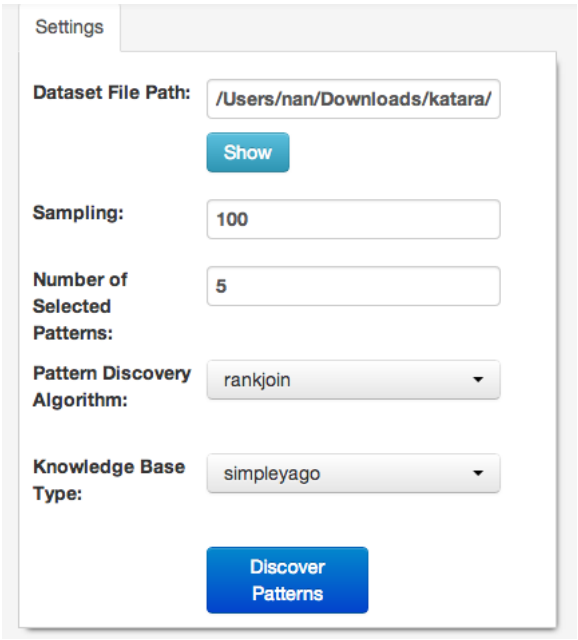

Figure 2: KATARA specification

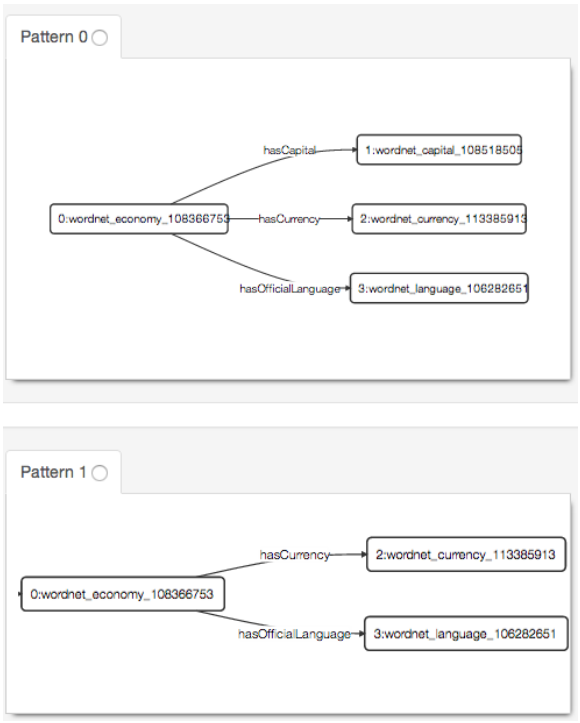

Figure 3: Validating top-k patterns

We expose the top-k possible repairs to the users (or crowd) for selection.

3. DEMONSTRATION OVERVIEW

In this demonstration, we will offer users the opportunity to experience the following features of KATARA: (1) Job specification: Users specify both the input data and reference KB, and the method and parameters supported by KATARA. (2) Pattern validation: KATARA computes and visualizes the top-k table patterns such that the users can pick the one that they consider as the most appropriate. (3) Data annotation: Users will experience how KATARA seamlessly bridges KBs and crowdsourcing to effectively annotate data with high fidelity. We mainly use Web tables, as given in [2]. We will show the following results by running KATARA.

(1) Job specification. Figure 2 displays the KATARA GUI for specifying a job. Users can specify their input tabular table file and see the data, select the reference KB, tune some parameters such as the number of sampling tuples needed and the number k for the top-k table patterns, and select a pattern discovery algorithm.

One of the datasets we shall use in the demonstration is similar to the table below. In this proposal, we illustrate KATARA using this table.

Angola	Luanda	Angolar	Portuguese
Bahrain	Manama	Bahraini dinar	Arabic
Cambodia	Phnom Penh	Riel	Khmer

(2) Pattern validation. After the users provide a job specification as described in (1), KATARA will invoke the selected algorithm to compute the top-k table patterns. The space of all candidate patterns is very large (up to the Cartesian product of all possible types and relationships in the KB), thus making it expensive for human verification. Hence, we rank these candidate patterns and only return the top-k most meaningful ones for final human validation. Since columns are not independent of each other, our scoring function captures the coherence of column types and their relationships. Please see [2] for more details.

Given the above table and the specification shown in Fig. 2, KATARA will find the top-5 patterns. We only show two of them in Fig. 3 due to space constraints. For example, the first pattern (*i.e.,* Pattern 0) states that the four columns are economy, capital, currency and language in SimpleYago, and that the relationships between the columns are hasCapital, hasCurrency and hasOfficalLanguage. The semantics of Pattern 1 can be explained similarly. Let us assume for the rest of the explanation that the users will pick Pattern 0 as the correct semantics of the given table.

(3) Data annotation. After the pattern is selected by the users as discussed in (2) above, KATARA works as follows.

(i) KB *annotation.* For each tuple, if it is fully covered by the given KB such that each value has the given type in the KB and each pair of values has the selected relationship in the KB, the tuple is considered as correct. Otherwise, if the tuple is not fully covered, the cause might be that either the KB is incomplete or the tuple contains some errors.

Consider the pattern selected in (2) above, the annotated tuples are shown in Fig. 4. Here, the tuples with no background color are considered to be fully covered by the KB. For example, the following information exists in SimpleYago: Armenia is an economy, Yerevan is a capital, Dram is a currency, Armenian is a language, and the relationships from Armenia to Yerevan is hasCapital, from Armenia to Dram is hasCurrency and from Armenia to Armenian is hasOfficialLanguage.

The tuples that are not fully covered by the KB are marked

C0Name	C1Capital	C2Currency	language
Angola	Luanda	Angolar	Portuguese
Armenia	Yerevan	Dram	Armenian
Azerbaijan [Europe]	Baku	Manat	Azerbaijani
Bahrain	Manama	Bahraini dinar	Arabic
Bangladesh	Dhaka	Taka	Bengali
Bhutan	Thimphu	Bhutanese ngultrum	Dzongkha
Brunei	Bandar Seri Begawan	Brunei dollar	Bahasa Melayu
Cambodia	Phnom Penh	Riel	Khmer

Figure 4: KB annotation

ID	Question
225	**Is CzechRepublic an instance of economy?** ◯Yes ◯No
230	**Do Chile and Spanish have the relationship of hasOfficialLanguage?** ◯Yes ◯No
235	**Is Turkmen new manat an instance of currency?** ◯Yes ◯No
240	**Do Australia and English have the relationship of hasOfficialLanguage?** ◯Yes ◯No

Figure 5: Crowd questions

C0Name	C1Capital	C2Currency	language
Angola	Luanda	Angolar	Portuguese
Armenia	Yerevan	Dram	Armenian
Azerbaijan [Europe]	Baku	Manat	Azerbaijani
Bahrain	Manama	Bahraini dinar	Arabic
Bangladesh	Dhaka	Taka	Bengali
Bhutan	Thimphu	Bhutanese ngultrum	Dzongkha
Brunei	Bandar Seri Begawan	Brunei dollar	Bahasa Melayu
Cambodia	Phnom Penh	Riel	Khmer

Figure 6: KB & crowd annotation

with a pink background, meaning that either some attribute values do not have the type specified in the table pattern, *e.g.,* Angolar is not a currency in SimpleYago, or two attribute values of the same tuple are not linked by the relationship in the table pattern, *e.g.,* there is no hasCurrency from Angola to Angolar in SimpleYago.

(ii) Crowd data validation. In the case that a tuple is not fully covered by the KB, there are two possible reasons. The KB is incomplete but the tuple is correct, or the tuple itself contains errors. To resolve such ambiguity, we generate binary questions for the crowd workers. There are two classes of questions: "Is A an instance of type" for type validation, and "Do A and B have the relationship of relation" for relationship validation? Please refer to Fig. 5 for sample questions. These questions can be posted to either an internal crowd, or MTurk. In this demo, we will post it to internal crowd so that attendee can easily play with KATARA.

(iii) KB and crowd annotation. After collecting the answers, KATARA will further show the annotated result (see Fig. 6). In this figure, the tuples that are fully covered by the KB will remain with no background color, *e.g.,* Armenia and Bahrain. The tuples that are identified to be correct by the KB and the crowd are marked in green, *e.g.,* the users confirm that Angolar is a currency, which is missing from SimpleYago. The tuples that are considered to contain errors are marked in red, *e.g.,* the currency of Cambodia should be standardized as Cambodian Riel. The tuples that still contain ambiguity remain with the pink background, which means that more questions should be answered by the crowd to further resolve the ambiguity.

Moreover, for each tuple that is identified to contain errors, we will generate a set of possible repairs from the KB (more details can be found in [2]) It is then up to the user to select a repair from the candidate list.

Summary. This demonstration exhibits how KATARA can easily help bridge KBs and crowdsourcing to achieve reliable data cleaning. Our proposal focuses on (a) A GUI that provides a set of user-tunable parameters for specifying a KATARA job; (b) Visualizing the most meaningful patterns such that users can easily pick the most appropriate one; and (c) KATARA's ability to annotate data and visualize them to users. In addition, a by-product of KATARA is that data annotated by the crowd as being valid, and which is not found in the KB, provides new facts to enrich the KB. Last but not least, these well annotated data are an important asset for several data analytic jobs such as data mining, machine learning, and data profiling.

4. REFERENCES

[1] X. Chu, I. F. Ilyas, and P. Papotti. Holistic data cleaning: Putting violations into context. In *ICDE*, 2013.

[2] X. Chu, J. Morcos, I. F. Ilyas, M. Ouzzani, P. Papotti, N. Tang, and Y. Ye. KATARA: a data cleaning system powered by knowledge bases and crowdsourcing. In *SIGMOD*, 2015.

[3] O. Deshpande, D. S. Lamba, M. Tourn, S. Das, S. Subramaniam, A. Rajaraman, V. Harinarayan, and A. Doan. Building, maintaining, and using knowledge bases: a report from the trenches. In *SIGMOD Conference*, 2013.

[4] W. Fan, J. Li, S. Ma, N. Tang, and W. Yu. Interaction between record matching and data repairing. In *SIGMOD*, 2011.

[5] W. Fan, J. Li, S. Ma, N. Tang, and W. Yu. Towards certain fixes with editing rules and master data. *VLDB J.*, 21(2), 2012.

[6] Z. Khayyat, I. F. Ilyas, A. Jindal, S. Madden, M. Ouzzani, P. Papotti, J.-A. Quiané-Ruiz, N. Tang, and S. Yin. Bigdansing: A system for big data cleansing. In *SIGMOD*, 2015.

[7] G. Limaye, S. Sarawagi, and S. Chakrabarti. Annotating and searching web tables using entities, types and relationships. *PVLDB*, 2010.

[8] C. Mayfield, J. Neville, and S. Prabhakar. ERACER: a database approach for statistical inference and data cleaning. In *SIGMOD*, 2010.

[9] M. Yakout, L. Berti-Equille, and A. K. Elmagarmid. Don't be SCAREd: use SCalable Automatic REpairing with maximal likelihood and bounded changes. In *SIGMOD*, 2013.

[10] M. Yakout, A. K. Elmagarmid, J. Neville, M. Ouzzani, and I. F. Ilyas. Guided data repair. *PVLDB*, 2011.