Incremental Schema Recovery

Poonam Kumari
University at Buffalo, SUNY
poonamku@buffalo.edu

Gourab Mitra
University at Buffalo, SUNY
gourabmi@buffalo.edu

Oliver Kennedy
University at Buffalo, SUNY
okennedy@buffalo.edu

ABSTRACT

Abstract goes here

KEYWORDS

Stuff

ACM Reference Format:
Poonam Kumari, Gourab Mitra, and Oliver Kennedy. 1997. Incremental Schema Recovery. In *Proceedings of ACM Woodstock conference (WOODSTOCK'97).* ACM, New York, NY, USA, Article 4, 2 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

It's a story as old as time: A student gathers data, makes a graph with the data, writes a paper about the data. Then the student graduates and the data languishes, without so much as a wiki page or README file documenting it. The next student to use the data needs to spend hours, days, or even weeks reverse-engineering it. Then they also graduate and the whole process can start over again.

As a way to break this tragic cycle of data abandonment, we propose *Label Once, and Keep It* (**LOKI**), a data-ingest middleware for incremental, re-usable schema recovery. **LOKI** allows users to assemble schemas on-demand, both (re-)discovering and incrementally refining schema definitions in response to changing data needs. To accomplish this, **LOKI** is built around a knowledge-base of both approximate, as well as exact schema labelings. First, approximate labelings derived from existing open-data sets, user-feedback, and expert-provided heuristics, jump-start the labeling process. When a user first points **LOKI** at a new tabular data set, **LOKI** provides users with a preliminary, default schema. As users confirm and/or override parts of the proposed schema, **LOKI** preserves the labels for the dataset's next user.

In this paper, we detail on our initial efforts to prime the **LOKI** knowledge-base with existing governmental open-data. Specifically...

2 SYSTEM DESIGN

The overall goal of **LOKI** is to streamline the process of developing schemas for existing unlabeled or poorly labeled data sets. As illustrated in Figure 1, **LOKI** lives alongside an existing RDBMS or Spark deployment, and takes as input tabular data in the form of a URL or HDFS file path. **LOKI** provides users with two modes of interaction: (1) A *labeling* interface that assists users in assigning names to existing columns of data, and (2) A *discovery* interface that helps users to search for columns representing particular concepts of interest. Both interfaces are supported by a knowledge-base that combines expert-provided heuristics, learned characteristics, as well as historical feedback gathered from users about already-loaded datasets. Once the user has labeled or discovered a sufficient set of columns, **LOKI** generates appropriate data loading/initialization code (e.g., a `CREATE TABLE` or Spark DataFrame initializer).

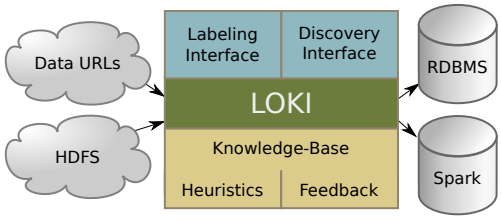

Figure 1: System Overview

2.1 Labeling and Discovery

- Declare notation: Set of columns, Rule for column labeling, etc...
- Express the labeling problem in terms of this
- Express the discovery problem in terms of this

2.2 Approximate Matching Rules

Outline approximate matching rules: Expert heuristics, rules, etc...

- How are approximate KB entries encoded?
- How do we unify different types of heuristics?

2.3 Exact Matching Rules

Outline exact matching rules: Data identity, recording/querying feedback. Merging conflicts.

- How is feedback saved in the KB.
- What happens in response to conflicting feedback.

3 KNOWLEDGE-BASE

Overview sketching data for similarity.
Data type taxonomy

- Numeric Data (Sketch = Distribution)
- Textual Data (N-Gram distribution?,)
- Enum Types (Overlap, Concept-Similarity)
- ...?

3.1 Numeric Data

Focus on the challenges of sketching numeric types

4 EXPERIMENTS

Experiments

WOODSTOCK'97, July 1997, El Paso, Texas USA
2016. ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4

5 RELATED WORK

[4] uses data similarity between two attributes for a join operation by counting the number of times each value from one attribute appears in the other and a histogram is constructed from the counts for all of the values.

Wrangler [3] infers the data type of a column and highlights errors based on inconsistent data types. Although wrangler creates a new column for extracted data as part of unfold operation, but does not infer the column name, analyst has to name the new column manually.

Potter's wheel [6] let's user define custom domains and uses inclusion function match to identify values in the domain.

Unfold operation âĂİunflattensâĂİ tables; it takes two columns, collects rows that have the same values for all the other columns, and unfolds the two chosen columns. Values in one column are used as column names to align the values in the other column.

Yago [1] provides a model which helps express entities, facts, relations between facts and properties of relations. This property could help list ontologies and find relation between columns.

A data summary called the data describer is used in [5]. The data types, correlations and distributions of the attributes in a private dataset are listed. Each attributei s categorized into either numerical or non-numerical. If non-numerical attribute cannot be parsed as datetime then it is considered to be a string.

In order to understand the layout and meaning of data, PADS [2] explores data and accumulators track the number of good values, the number of bad values, and the distribution of legal values. Typical questions answered through this approach include: how complete is the description of the syntax of the data source, how many different representations for âĂIJdata not availableâĂİ are there, what is the distribution of values for particular fields, etc.

There is an increasing number of datasets in which well-structured attributes (with or without a name) can be identified, each containing a set of values called a domain. There is lack of schema description in most of the datasets. LSH Ensemble is used in [7] to find domains that maximally contain a query dataset, which can help to find datasets that best augment a given set of data.

6 FUTURE WORK

Future work...

- Meta-queries for columns that *could* be in a query.
- Discovery of meta-data (e.g., units)
- Automatic translation/transformation (units, structure – GPS vs Textual)

REFERENCES

[1] MS Fabian, K Gjergji, WEIKUM Gerhard, et al. 2007. Yago: A core of semantic knowledge unifying wordnet and wikipedia. In *16th International World Wide Web Conference, WWW*. 697–706.

[2] Kathleen Fisher and Robert Gruber. 2005. PADS: a domain-specific language for processing ad hoc data. In *ACM Sigplan Notices*, Vol. 40. ACM, 295–304.

[3] Sean Kandel, Andreas Paepcke, Joseph Hellerstein, and Jeffrey Heer. 2011. Wrangler: Interactive visual specification of data transformation scripts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 3363–3372.

[4] Arnab Nandi, Lilong Jiang, and Michael Mandel. 2013. Gestural query specification. *Proceedings of the VLDB Endowment* 7, 4 (2013), 289–300.

[5] Haoyue Ping, Julia Stoyanovich, and Bill Howe. 2017. DataSynthesizer: Privacy-preserving synthetic datasets. In *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*. ACM, 42.

[6] Vijayshankar Raman and Joseph M Hellerstein. 2001. Potter's wheel: An interactive data cleaning system. In *VLDB*, Vol. 1. 381–390.

[7] Erkang Zhu, Fatemeh Nargesian, Ken Q Pu, and Renée J Miller. 2016. LSH ensemble: Internet-scale domain search. *Proceedings of the VLDB Endowment* 9, 12 (2016), 1185–1196.