Technical Perspective: TASHEEH: Repairing Row-Structure in Raw CSV Files

Matthias Boehm
Technische Universität Berlin
matthias.boehm@tu-berlin.de

Open science and data exchange in general rely on standardized and interoperable file formats. Comma-separated value (CSV) files are probably the most versatile, simplest, and widely-used file format for tabular data. For example, the FAIR data principles of research data management promote findable, accessible, interoperable, and reusable data and metadata. In this context, CSV files ensure accessibility and interoperability because of its simple structure and text-based format, making them amenable for long-term storage. An analysis by the Google Dataset Search team found that `https://schema.org/` contained almost 30M datasets of which 37% are tables in CSV or XLS format [1].

CSV in Practice: The RFC 4180 document [6] formally defines the "text/csv" MIME type. A CSV file is a list of records, separated by line breaks. Records and the optional header in turn contain fields, separated by commas. Fields containing line breaks, double quotes, or commas should be enclosed in double quotes, and double quotes can be used for escaping (e.g., double quotes). This definition is simple and standardized. In practice, however, there exist a number of variants as well as data corruptions. First, tab-separated files are also very common, and many data systems nowadays support arbitrary single- or multi-character field delimiters, custom quotes, and multi-line headers. Second, common corruptions include incorrect quoting, additional meta data, and inconsistent number of fields. Most data systems and data frame libraries raise errors on clearly identifiable corruptions but leave it up to the user to correct these issues with custom transformation programs.

Existing Work: In contrast to basic data type detection (e.g., via sampling and regular expressions) as well as semantic type detection (e.g., via machine learning) [4], the literature on reliably detecting (and correcting) corrupted CSV files is relatively sparse. First, there was early work on gracefully completing large-scale Jaql jobs despite errors through declarative compensation plans for erroneous records per operator [5]. Silently corrupted records remained undetected though. Second, later work also focused on detecting the CSV dialect and formatting through a consistency measure of pattern (number of fields per record) and type (data type per column) scores [7]. Third, there is a large body of methods on data cleaning (e.g., outliers, attribute swaps, missing values, duplicates), many of which require already loading the data first. Identifying and correcting fine-grained corruptions was largely an unaddressed problem.

Paper Context (SURAGH): In order to address this open problem of identifying ill-formed CSV records, a team from HPI proposed SURAGH [2] in an earlier EDBT'22 paper. This work formulated the problem of pattern schemas and classifying ill- or well-formed records, and introduced an algorithm for identifying ill-formed records through value mapping to these patterns. Moreover, the authors introduced and shared an annotated benchmark dataset with 131 files (210,550 rows and various errors) from open data repositories for evaluating both efficiency and effectiveness.

Paper Contributions (TASHEEH): Building on top of SURAGH, the authors then introduced TASHEEH [3]—for standardizing ill-formed into well-formed records—which also received the EDBT'24 best paper award. Records are iteratively classified as ill- or well-formed, and patterns are derived for groups of ill-formed records. The ill-formed records are further classified as wanted or unwanted through an elegant sequence alignment technique according to dominant patterns found in the set of well-formed records. Finally, the wanted, ill-formed records are repaired with a set of pattern transformation operators. Overall, TASHEEH shows—on the extended benchmark dataset—very compelling accuracy in terms of F1 measure at moderate runtime requirements.

Final Remark: Already at an "HPI and friends" dinner during VLDB'21 in Copenhagen, Mazhar, Gerardo, and I were talking about ideas behind these works. It is great to see that our data management community appreciates such data engineering methods with high practical relevance.

1. REFERENCES

[1] O. Benjelloun, S. Chen, and N. F. Noy. Google dataset search by the numbers. In *ISWC*, 2020.
[2] M. Hameed, G. Vitagliano, L. Jiang, and F. Naumann. SURAGH: syntactic pattern matching to identify ill-formed records. In *EDBT*, 2022.
[3] M. Hameed, G. Vitagliano, F. Panse, and F. Naumann. TASHEEH: repairing row-structure in raw CSV files. In *EDBT*, 2024.
[4] M. Hulsebos et al. Sherlock: A deep learning approach to semantic data type detection. In *SIGKDD*, 2019.
[5] C. Kanne and V. Ercegovac. Declarative error management for robust data-intensive applications. In *SIGMOD*, 2012.
[6] Y. Shafranovich. Common format and mime type for comma-separated values (csv) files. RFC 4180, 2005.
[7] G. J. J. van den Burg, A. Nazábal, and C. Sutton. Wrangling messy CSV files by detecting row and type patterns. *Data Min. Knowl. Discov.*, 33(6), 2019.