Quo venis? Metadata for Common Scientific ASCII Files

Muhammed Bayram, Frank Tristram

Karlsruhe Institute of Technology

Currently most scientific information is stored in files and some in databases or repositories. The diversity of file formats is a challenge for all software systems and automated workflows that try to process the information therein. While the final analysis level of scientific content gets a lot of attention in context of machine learning and data analysis, the other information layers are less in focus, although they are the basis for the other progress. There is no efficient way we can link and compare scientific information over scales, disciplines or sometimes even different labs in the same discipline, if we cannot merge information automatically. Here we show a perspective to structure and enrich unstructured data with metadata as a prerequisite step to combine information from heterogeneous sources for the example of ASCII files. Currently it is trivial to bring thousands of file formats into one storage device, but pretty hard to provide the content of a file on a higher level than a stream of characters. For each specific format a special converter is built and for unspecific formats like .dat or .txt there is even no solution to clarify what converter is needed. The first steps from an unknown file format to "information" need to cut the file into meaningful sub-objects, interpreting a file more as a container. This is very analogue to a "table of content" for a book. We argue that a meaningful file segmentation, that allows to provide object names and object types (like table or key-value pairs), headers and other relations as defined and interlinked data objects and not as plain string is a major step. On such objects, an information crawler can pull information out of dark data archives.

1 Introduction

Clusters of excellence and other large research collaboration must solve the problem of creating an added value from bringing together diverse data, expertise, infrastructures, and motivations. This is mainly relevant for linking cross-disciplinary research to a core topic, which is more difficult, if data needs to be connected across different institutes and reused interoperably. In experimental clusters, hundreds of measurement devices, software solutions and objectives are scientifically connected with each other, which makes an overarching interpretation of all data difficult. Interfaces and standardizations are often

Publiziert in: Vincent Heuveline, Nina Bisheh und Philipp Kling (Hg.): E-Science-Tage 2023. Empower Your Research – Preserve Your Data. Heidelberg: heiBOOKS, 2023. DOI: https://doi.org/10.11588/heibooks.1288.c18076

missing as well as overlapping knowledge and experiences (Stocker et al. 2020). A common starting point is data that is stored on central storage systems but is not FAIR. In order to make every piece of information available to every cluster member, the representation of this data needs to be standardized. The HDF5 format offers a flexible container for structuring all conceivable data in a standardized way and making it available. However, it is not straightforward to convert an arbitrary ASCII file into HDF5, such that there is a major improvement in the amount of structuredness.

2 Reading of ASCII Files

While many specific file formats like .tiff or .pdf at least offer metadata about the creation of the file, plain ASCII files offer no information. Even worse, there is no common interpreter that can, for example, distinguish between different .dat or .txt structures to extract specific information. We are developing a method that one could call file segmentation. As in the image segmentation, this process does not deliver the final file analysis to answer a specific question (e.g. to identify a person or animal), but an intermediate result consisting of distinct, condensed components. Our most important insight was, that an information extraction can never be complete, but needs to be "fit for purpose" like the current image analysis (see Figure 1). It does not deliver all information about an image, but enough to answer questions like "What is in there?" and might add value for one or several specific information dimensions (from which there are practically infinite).

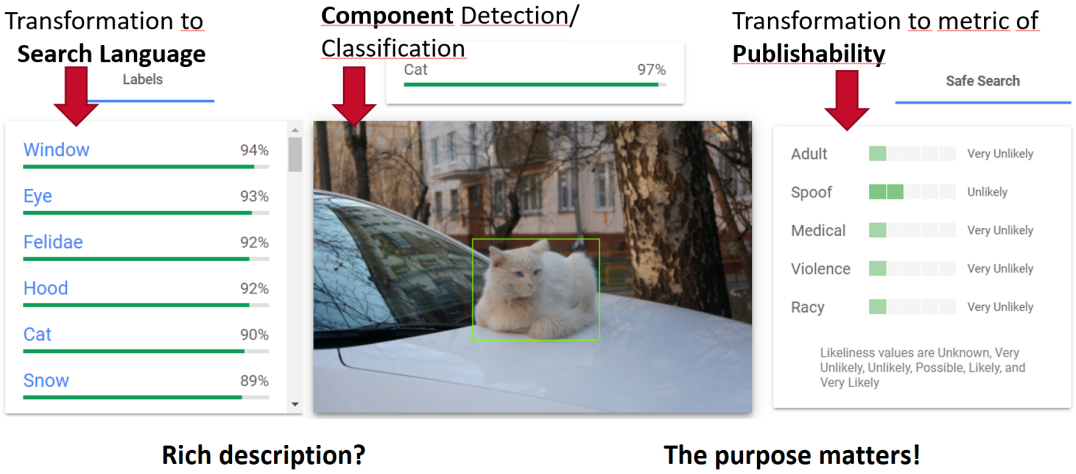

Figure 1: Google Information extraction from images: It still is pretty incomplete, but serves some intended purposes.

Of general importance is a component detection. The components in an ASCII file can be a table, a set of information (like an address), a collection of key-value pairs, some longer text block etc. We identify describing metadata for these components. We then not only know that there is a distinct object, but also object properties, especially focusing on the underlying data structure. Imagine someone would store a 2D-Array of integers into an ASCII file. We try to guess this initial data structure and put this into an HDF5. This intermediate result within the HDF5-file opens up several interesting perspectives:

- For each file format "subclass" (e.g. a special .dat file coming out of a device) a unique fingerprint of the structural subcomponents can be built.
- Values inside very different files can be visualized, compared and analyzed by existing HDF5 tools without the need to write wrappers for each file format.
- It is possible to create a unique interface with persisting functionality, almost independent from the underlying formats (or versions).

We are motivated by these points, but saw these can be of interest for other communities, too. In any use of this, one needs to check the output structure manually to control if it is like one would expect. However, this check is still much less effort, then creating standardized structures "by hand" and can be applied automatically on future use cases. In Figure 2, you can see, that already the most simple example creates an ambigious structural result between a table and key-value pairs.

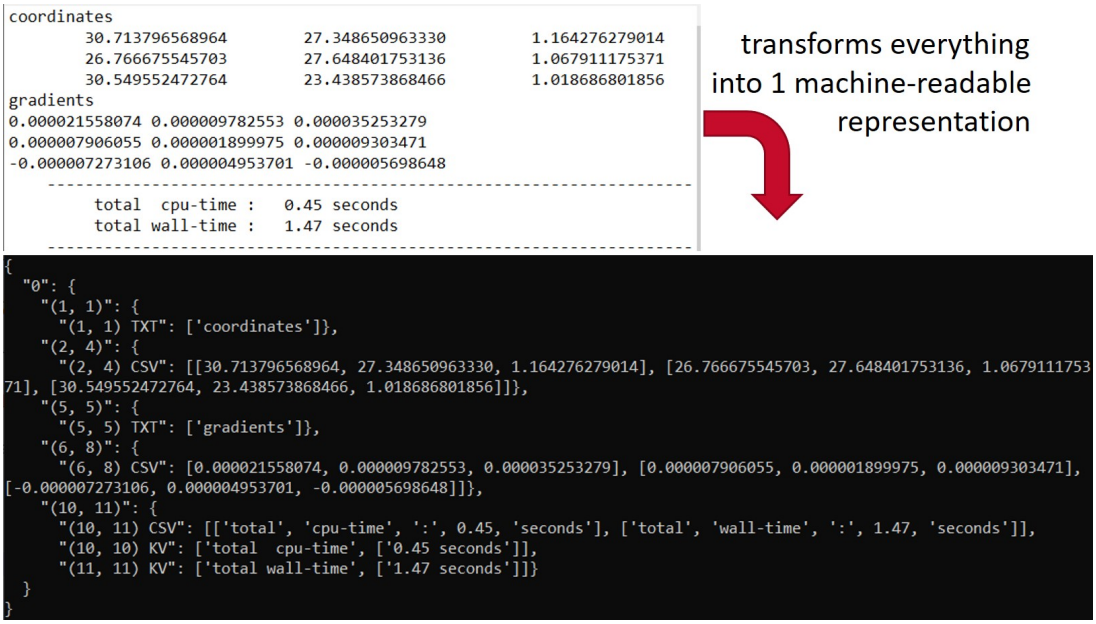

Figure 2: Most basic example what is done by converting a human readable ASCII file into a nested machine-readable form. Here, the machine detects two options to interpret the CPU time content: It could be a table with 5 columns or two key-value pairs. Final decisions must always be done by the user.

3 Harvesting Zenodo

We have downloaded 10.000 scientific data files from material science from Zenodo, to run some real-word analysis with our new information extractor. We only downloaded .json, .txt, .dat and .csv files. We included "prestructured" files: .csv(n=3126), .json(n=259) as well as "unstructured" files: .dat (n=1693) and .txt(n=4935) However, we could only identify 9860 encodings (98.6%) automatically. Manuel inspection revealed, that most remaining files, were likely to be misnamed. So e.g. a readme.txt was in fact a readme.doc

file that could be opened with any common text processing program, but counts as "unreadable encoding" in this statistic. We found that the unstructured formats had very different origins and compositions. For them we identified over 100 structurally different fingerprints. Clustering along these fingerprints created large subclusters of e.g. "readme-similar" files, bullet-lists, data tables and combinations of such elements. We also found a long tail of structures that seem to exist just in one data source.

4 Conclusion

The diversity of subformats raises the question, how to standardize a description of such files. Therefore we attend the E-Science-Tage to discuss about real applications in this area. The library as as well as the computing center community showed interest in what we are doing and more future work, because currently, there is simply nothing that could harvest information from unstructured files and perform some structuring. From these discussions and our own experience we suggest here, to develop a metadata schema that is close to what we know as a "table of content" for books. A table of content is in principle a nested short description of meaningful blocks. Although such blocks are somewhat arbitrary, they are a step forward for structuring and findability. Probably a web portal that provides something like this table of content for any uploaded ASCII (and some more) files, could force a standardization in this untouched area by practical usage. Another idea for a use case that came up, was to apply this on REST-APIs for repositories to identify what you are really looking for. Each API is similar but a bit different and therefore cumbersome to connect to in a way that returns the object of interest automatically - and persistent, even if something within the repository changes. These are just very recent ideas how to move on from where we are and it is not unlikely that a solution in some years will exactly work like this, whether it is developed by us or by others.

Acknowledgements

The work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – 2082/1 – 390761711.

References

Stocker, Markus, Louise Darroch, Rolf Krahl, Ted Habermann, Anusuriya Devaraju, Ulrich Schwardmann, Claudio D'Onofrio, and Ingemar Häggström. 2020. "Persistent Identification of Instruments". *Data Science Journal* 19 (1): 18. DOI: https://doi.org/10.5334/dsj-2020-018.