Towards Efficient Annotation Databases

René Heinzl[1], Markus Nissl[2] and Emanuel Sallinger[2,3]

[1]Building Digital Solutions 421 GmbH, Vienna, Austria
[2]TU Wien, Vienna, Austria
[3]University of Oxford, Oxford, United Kingdom

Abstract
Recent advances in machine learning have increased the demand for efficient annotation data management for machine learning applications by organizations. In this paper, we address this challenge through an industrial collaboration centered around the unification of data for training and prediction workflows by enabling fast analytical processing through summarization. Beyond this specific solution, we provide a very concrete real-world scenario and solution to the data management community as inspiration for further theoretical and practical research. Finally, we report on the open scientific challenges that remain in this field.

1. Introduction

Answering the call specifically pushing for *"papers in real-world contexts"* we represent a paper on a real-world application in the area of waste separation, that is, in the context of the pressing societal issues of circular economy and meeting the UN sustainable development goals (SDGs). This presents ongoing research based on an award-winning in-production large-scale deployment in multiple countries.

Context. The core of this paper is focused on annotation data management for machine learning, a critical part of data management for machine learning [1]. Industrial implementations, such as Amazon SageMaker [2], VGG Image Annotator [3] or Anafora [4] exist, but stop at the level of annotating data for training purposes or at the management of the training process itself. They have limited support for metadata management, lacking support for real-time data management and analytical querying. Yet, we know that in the data management community, we have ample studies on *metadata management* [5, 6, 7, 8, 9] and *annotation databases* [10, 11, 12] – though in quite different contexts than what is required for annotation data management in machine learning.

In this paper, we describe the concrete solution to this problem which we developed for this widely deployed real-world application. Our solution is centered around the unification of data for training and prediction workflows by enabling fast analytical processing through summarization. This is especially important when real-time data is used in reporting systems and automated machine learning processes. Beyond this specific solution, the most important aspect of this paper is giving a very concrete real-world scenario and solution to the data management community as inspiration for further theoretical and practical research.

15th Alberto Mendelzon International Workshop on Foundations of Data Management, May 22–26, 2023, Santiago, Chile

Application. In the following we provide the core use cases of our business partner for the domain of interest, demonstrating the need for an advanced annotation and metadata storage for machine learning processes.

Use Case (Object storage). The waste separation business is interesting in detecting impurities in plastic waste such as batteries, metals or cardboard. The company has the requirement that (i) each image should be stored as a possible candidate for training for at least one year for subsequent analysis requests, (ii) each detected label for each version of a machine learning model applied on an image should be stored for (real-time) analytical purposes and (iii) for statistical evidence of correct labeling, the created labels are stored per user.

The storage of the image data alone for this use case with an average of 100GB (or 20.000 Full-HD images) per device generates a large amount of data. While typically the image data is stored in an object storage, still the metadata for each device exceed 7 million entries per year, without considering the details such as the number of labels per model or user. Moreover, usually additional metadata is stored as demonstrated by the following use case:

Use Case (Metadata). The company is interested in storing next to annotation data for training and analytical purposes information regarding the device, such as the model number, camera metadata or location data. This allows the company among others to correlate specific waste information with trucks and household areas for optimization purposes.

There exist different approaches to store such metadata in database systems. Typically annotation databases are built on top of relational databases or NoSQL stores using either separate annotation tables, additional fields in the document table, or as binary data such as serialized JSON or XML data. In some cases, annotations are stored in object stores with a reference to the location in the database. While the first two methods allow *more efficient analytical queries*, the latter two methods allow to deal with *more complex annotation scenarios*, such as frame series annotations, where several thousand records ranging between several MB to several hundred MB are required at once [13]. Systems and theory that support both scenarios do not exist to the best of our knowledge.

Contribution. In this paper, we address this challenge by reporting on

- a *real-world contemporary use case* in the context of the pressing societal issues of circular economy;
- ongoing work for an *efficient annotation data storage* solution that leverages both database systems and object storage;
- *key requirements* that an annotation database for machine learning purposes has to fulfil.

Outline. In the remainder of this paper, we discuss first the requirements, then present the solution and finally conclude by discussing open challenges.

2. Requirements

In this section, we establish several key requirements that an annotation database[1] has to fulfill in order to manage annotation data effectively. Our requirements are based on our use cases from the waste separation company, extended with knowledge from different scenarios established over several years on hands-on experience in the field of machine learning. The requirements are:

- *Integration with machine learning workflows.* An annotation database should integrate seamlessly with machine learning workflows, allowing the use of annotation data in the training and evaluation of machine learning models.
- *Support for search and analysis.* An annotation database should store data in an optimized format that can be efficiently queried and analyzed in real-time. One should be able to navigate through the data, extract insights and trends as well as find specific annotations.
- *Performance and scalability.* An annotation database should be able to handle large volumes of data and support high levels of concurrent access.
- *Flexibility and extensibility.* An annotation database should support a wide range of annotation types and workflows as well as custom annotation types to cover highly specialized annotation tasks.
- *Support of annotation metadata.* An annotation database should allow the storage and management of annotation metadata, such as annotator details, timestamps, model information, location data and additional information related to the annotation.

3. Solution

In this section, we present our solution for the use case to address the established requirements from the previous section. Our approach is structured into three different components: (i) base data ingestion, (ii) machine learning data ingestion, and (iii) real-time analytical component. We provide an overview of each of the components in the following by referring to Figure 1.

Base data ingestion. In our use case, multiple end user devices (in the figure referenced as “Data Collector”) are capturing new (image) data at real-time (each device every few seconds) and inserting them into our annotation database. Thereby we distinguish between raw data (e.g., the image) which is stored in an object storage, and meta data (e.g., timestamps, locations, the path at the object storage of the raw data) which is stored in our meta storage. Already in this step it is crucial to utilise an efficient bucketing schema for the metadata to optimise towards the real-time analytical component – a key shortcoming of some approaches discussed in the introduction.

[1] Note that we concentrate here on the annotation database, not on the annotation management system which includes also additional functionality such as user management, visualization tools and an advanced user interface.

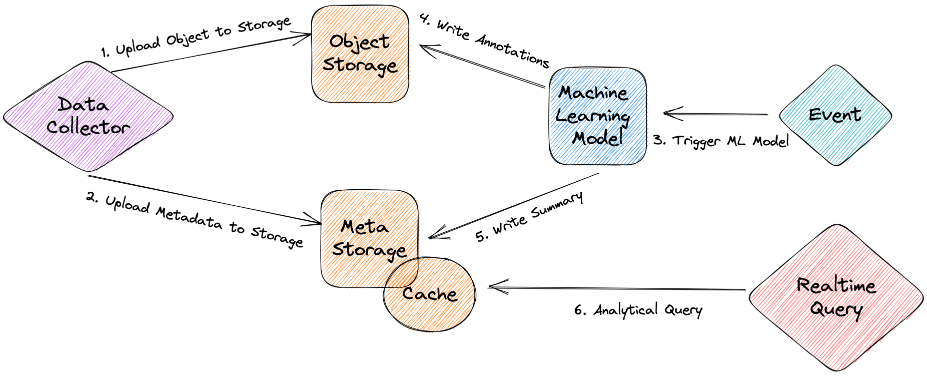

Figure 1: Overview of our Annotation Database Setup

Machine Learning Data Ingestion. Here our main goal is to overcome the – inefficient and costly – separation between training data storage and operational data storage. Operationally, the machine learning process is initiated when different triggers fire. These are, for already deployed models, insertion triggers for computing new annotations (labels) in real-time and, for newly trained models, an on-demand execution over existing raw data in the object storage after deployment of the model. The resulting annotations are written to the object storage, a summary of those annotations are provided to the meta storage. With this, i.e., the storage of the annotation data in the object storage on the one hand, we allow for handling complex annotation scenarios – a key shortcoming of the other approaches discussed in the introduction, and with the summarization on the other hand, we provide the foundation of efficient real-time querying of the meta storage, the second key point raised in the introduction.

Real-time Analytical Component. The last part of the system is the efficient possibility to subscribe to a query of annotation results from the meta storage. For this, we encountered different queries from the business domain, such as how many annotations of one specific label or a combination of labels have been found per day for specific metadata criteria (device, location, machine learning model, and so on). This provides a high number of query combinations, but with only a limited amount of queries being currently actively requested. By combining an efficient analytical real-time database with bucketing (we use buckets based on the timestamp), we are able to cache "old" results and only have to (re)compute the changes in the newest bucket. With a subscription to database changes, the solution is even able to clear and recompute the cache for changes for currently subscribed queries as well as notify in real-time the current subscribed queries with the newest updates. This ensures that the solution meets the second key point raised in the introduction, more efficient analytical queries, and one of the key requirements.

Evaluation. This approach has been evaluated by the stakeholders of the company in real-world production in multiple countries and satisfies all requirements.

4. Conclusion.

We conclude by raising open challenges for our community:

Open Challenges (theory). While the presented solution provides an effective solution for the use, in the data management community we lack (1) a systematic study of this combination of annotation storages and summarization, and (2) theoretical results on the limits of such techniques.

Open Challenges (practice). Here we lack (1) a systematic analysis of different database technologies for the meta storage, and (2) the development of optimized data management systems in the context of resource-limited environments.

In addition, we are particularly interesting in exploring this topic in more detail in the setting of Knowledge Graphs [14, 15, 16, 17] and our Vadalog system [18, 19, 20].

Acknowledgments

This work has been funded by the Vienna Science and Technology Fund (WWTF) [10.47379/VRG18013, 10.47379/NXT22018, 10.47379/ICT2201]; and the Christian Doppler Research Association (CDG) JRC LIVE.

References

[1] M. Schlegel, K. Sattler, Management of machine learning lifecycle artifacts: A survey, SIGMOD Rec. 51 (2022) 18–35.

[2] D. Nigenda, Z. Karnin, M. B. Zafar, R. Ramesha, A. Tan, M. Donini, K. Kenthapadi, Amazon sagemaker model monitor: A system for real-time insights into deployed machine learning models, in: KDD, ACM, 2022, pp. 3671–3681.

[3] A. Dutta, A. Zisserman, The VIA annotation software for images, audio and video, in: ACM Multimedia, ACM, 2019, pp. 2276–2279.

[4] W. Chen, W. Styler, Anafora: A web-based general purpose annotation tool, in: HLT-NAACL, The Association for Computational Linguistics, 2013, pp. 14–19.

[5] P. G. Kolaitis, Schema mappings, data exchange, and metadata management, in: PODS, ACM, 2005, pp. 61–75.

[6] P. A. Bernstein, S. Melnik, Model management 2.0: manipulating richer mappings, in: SIGMOD Conference, ACM, 2007, pp. 1–12.

[7] M. Arenas, J. Pérez, J. L. Reutter, C. Riveros, Foundations of schema mapping management, in: PODS, ACM, 2010, pp. 227–238.

[8] P. G. Kolaitis, Reflections on schema mappings, data exchange, and metadata management, in: PODS, ACM, 2018, pp. 107–109.

[9] P. Edara, M. Pasumansky, Big metadata : When metadata is big data, Proc. VLDB Endow. 14 (2021) 3083–3095.

[10] D. Bhagwat, L. Chiticariu, W. C. Tan, G. Vijayvargiya, An annotation management system for relational databases, VLDB J. 14 (2005) 373–396.

[11] P. Senellart, Provenance and probabilities in relational databases, SIGMOD Rec. 46 (2017) 5–15.

[12] P. Buneman, W. Tan, Data provenance: What next?, SIGMOD Rec. 47 (2018) 5–16.

[13] How to efficiently manage storage for high-volume data annotation projects, https://medium.com/multisensory-data-training/storage-e7f37afba24c, 2023. Accessed: 2023-03-08.

[14] L. Bellomarini, M. Benedetti, S. Ceri, A. Gentili, R. Laurendi, D. Magnanimi, M. Nissl, E. Sallinger, Reasoning on company takeovers during the COVID-19 crisis with knowledge graphs, in: RuleML+RR (Supplement), volume 2644 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 145–156.

[15] L. Bellomarini, L. Bencivelli, C. Biancotti, L. Blasi, F. P. Conteduca, A. Gentili, R. Laurendi, D. Magnanimi, M. S. Zangrandi, F. Tonelli, S. Ceri, D. Benedetto, M. Nissl, E. Sallinger, Reasoning on company takeovers: From tactic to strategy, Data Knowl. Eng. 141 (2022) 102073.

[16] L. Bellomarini, E. Sallinger, S. Vahdati, Knowledge graphs: The layered perspective, in: Knowledge Graphs and Big Data Processing, volume 12072 of *Lecture Notes in Computer Science*, Springer, 2020, pp. 20–34.

[17] L. Bellomarini, E. Sallinger, S. Vahdati, Reasoning in knowledge graphs: An embeddings spotlight, in: Knowledge Graphs and Big Data Processing, volume 12072 of *Lecture Notes in Computer Science*, Springer, 2020, pp. 87–101.

[18] L. Bellomarini, L. Blasi, M. Nissl, E. Sallinger, The temporal vadalog system, in: RuleML+RR, volume 13752 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 130–145.

[19] L. Bellomarini, R. R. Fayzrakhmanov, G. Gottlob, A. Kravchenko, E. Laurenza, Y. Nenov, S. Reissfelder, E. Sallinger, E. Sherkhonov, S. Vahdati, L. Wu, Data science with vadalog: Knowledge graphs with machine learning and reasoning in practice, Future Gener. Comput. Syst. 129 (2022) 407–422.

[20] L. Bellomarini, D. Benedetto, G. Gottlob, E. Sallinger, Vadalog: A modern architecture for automated reasoning with large knowledge graphs, Inf. Syst. 105 (2022) 101528.