A High Quality Data Pipeline for Reasonable-Scale Machine Learning

David Faragó, Innoopract Informationssysteme GmbH, Karlsruhe

Abstract Data quality (especially correctness) plays a critical role in the success of a machine learning (ML) project. This paper describes a data pipeline for creating high quality data, using as example Key Information Extraction (KIE) from invoices – one of the most popular tasks in Intelligent Document Processing (IDP). The tasks of each data pipeline step are listed, showing the decisions and technology involved.

The focus is on practicality: doing ML at reasonable-scale, i.e. with as little cost (people and hardware) as possible, and a concern for practice more than achieving high scores on a metric that is not grounded in practical use.

Contributions:

1. an extended list of quality dimensions, with simple definitions
2. overview of a data pipeline, examplified on KIE
3. for each pipeline step a list of tasks, showing decisions, pitfalls, and technology involved
4. in particular, how to use the state of the art contrastive model CLIP to solve difficult selection and reduction tasks on images
5. a tool for labeling key information on images
6. a labeling guide for invoices.

Most contributions can easily be transfered to other supervised learning tasks.

Keywords: *data quality, data-centric AI, data pipeline, reasonable-scale ML, IDP, KIE on invoices*

1 Introduction

1.1 Motivation

Data quality has many dimensions – correctness dimensions and class balance are particularly relevant for ML. As data is used for training and validating ML models, its quality strongly influences whether the model is suitable and performant. Thus data quality is a critical part of quality assurance in ML, in spite of being the most under-valued and de-glamorised aspect of AI [16].

1.2 Data Pipeline

But how do you achieve high quality data? There are many decisions, technologies, and pitfalls across the data pipeline, from creating datasets by selecting, reducing, labeling, and preprocessing data, to validating the quality of the datasets by measuring it directly or by splitting and augmenting it to validate and test it via a trained model, see Fig. 1.

The decisions (e.g. size and type of dataset, how to label and how to counter low data quality) strongly depend on whether you are in Big Tech, or in a smaller company, or in academia: Big Tech has sufficient resources to create huge datasets for training huge models on huge hardware. Academia has sufficient time and knowledge to focus on more fundamental challenges than specific industrial problems, and on a concerning [14] publication practice on achieving "ever higher scores on ever higher benchmark tasks", but the scores and benchmarks often do not reflect practice. Smaller companies do not have the resources and thus need to focus on solving practical problems to achieve a high Return On Investment and early break-even.

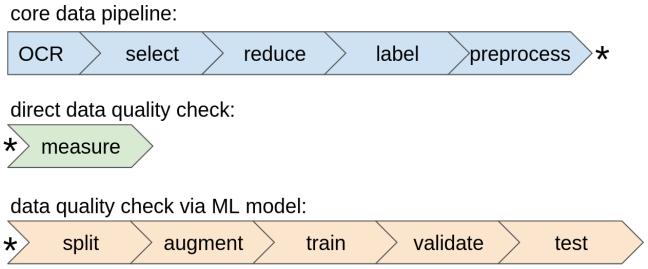

Figure 1: Core data pipeline and two possible extensions for data quality checks: directly or via ML model

This paper focuses on small to reasonable-scale companies and thus reasonable-scale ML, where data is king, open-source solutions should be prefered, and limited time should be invested into DevOps as external services become affordable [20].

Though ethic and legal concerns are also important [14], they are not covered in this paper.

1.3 Case Study: KIE on Invoices

This paper presents a data pipeline for KIE from images of invoices, which is the most prominent use-case in IDP. The biggest challenge in KIE of invoices is the lack of high quality labeled datasets [2].

The invoices (see middle of Fig. 3 for an examplary invoice) are

- template-free: an invoice can have its individual layout, ruling out template-based approaches
- schema-full: the following key information should be extracted: *COMPANY, INVOICE-NO, CUSTOMER-NO, TOTAL, IBAN*.

We use IOB tagging, which marks the first token of a key information as beginning (B), the rest as inside (I), and all tokens not containing key information as OTHER (O). For instance, in Fig. 3, "DE49" is labeled as B-IBAN, the tokens "6129" to "09" as I-IBAN, and "Kreissparkasse" as OTHER.

Our mobile online banking app uses KIE to relieve the user from the manual input for wire transfers: the user can instead take a picture of an invoice, and also focus the camera on certain regions of the invoice if necessary. For data security, inference is performed on the device instead of sending pictures to the cloud.

2 Data quality

In spite of being the most under-valued and deglamorised aspect of AI [16], one should obsess over data quality [20]. Luckily, there is now a paradigm shift taking place, focusing more on data creation [8]: data-centric AI, the discipline of systematically engineering the data used to build an AI system [7].

Since most work in ML is not data-centric, many papers in ML that mention their datasets as having "high quality" mean that the datasets contain values from high quality sensors, e.g. high-resolution images [18]. But this is just one aspect of dataset quality, and selecting only high quality sensor data leads to low-variance and covariate shift if the customer's sensors will not always be of such high quality.

Table 1 lists seven data quality dimensions. They merge and extend [17, 3, 1, 5]. Since simplicity and understandability was a major goal, the quality dimensions are stated negatively (e.g. inconsistency instead of consistency), leading to simpler definitions.

The first two, three or four dimensions are sometimes summarized as data correctness or data accuracy. Class noise is only sensible in supervised machine learning; class imbalance can be generalized to unsupervised learning with infered classes (e.g. clusters). Data redundancy is the same dimension as uniformity and uniqueness [17, 5]. Distribution noise is a generalization of covariate shift or training-serving skew [3] to arbitrary datasets because diverging distributions can also happen within training datasets from different sources, also indicating some deviation from the ground truth. These dimensions are all model-independent, and dimensions that depend on the model, e.g. sufficiency, are excluded.

Dimension	Definition	Quality degrading example
feature noise	percentage of incorrect feature values (wrt ground truth)	ground truth "50€" OCRed as "5i€", or dirty background OCRed as "."
class noise	feature noise on the target feature (i.e. the class)	a value incorrectly labeled as CUSTOMER-NO instead of INVOICE-NO
distribution noise	distribution distance between the dataset and the ground truth	training-serving skew with the training dataset being invoices from B2B, but serving for clients with B2C invoices
incompleteness	percentage of values missing	ground truth "50€" skipped (missing in dataset) due to OCR error
inconsistency	percentage of values with more than one representation	TOTAL "50" and "50€" occurring in the dataset
redundancy	percentage of (non-exact) duplicates	two identical invoices, or with (almost) the same key information
class imbalance	average pairwise size difference between classes	most text of an invoice is no key information, leading to a much larger class OTHER

Table 1: Seven data quality dimensions for ML, each with a definition and an example from KIE of invoices

There are many ways to compute the distance between two dataset distributions d_1, d_2, e.g. Kullback–Leibler divergence, cosine similarity, confidence level of a χ^2 test, or $\max_{v \in V} |P_{d_1}(v) - P_{d_2}(v)|$ over all observed values V [3].

[5] computes quality metrics as macro-scores over features, i.e. firstly computes the metric per feature and then takes the arithmetic average over all features in the dataset. Alternatives are micro-scores (i.e. not distinguishing the features) or per feature metrics.

There are often trade-offs between these dimensions, e.g. oversampling to reduce class imbalance increases redundancy and distribution noise.

3 Data Pipeline

Fig. 1 depicts the core data pipeline (blue, top line), which ingests new invoice images and produces new datasets. As is typical for IDP, both Natural Language Processing (NLP) – especially in the preprocessing pipeline step – and Computer Vision (CV) – especially in all previous steps – are involved. These steps can be adopted in other NLP resp. CV supervised learning tasks.

The two pipeline extensions perform different quality checks for the produced data:

- either the data is validated through direct data measurements (green, middle line)
- or through model training, validation, testing (orange, bottom line).

Each step performs one or multiple tasks and is described in Table 2 and the following subsections.

Pipeline step	Task	Technologies
OCR	native mobile OCR and rotation	ML KIT (Android) & Vision (iPhone)
select	separate German from rest	Python Polyglot
	separate invoice types (giro, QR code)	CLIP, Python, Huggingface
reduce	variance-preserving size reduction	CLIP, Python, Huggingface
	remove invalid invoices	CLIP, Python, Huggingface
label	collaborative annotation guide	Google Docs
	label total/customer&invoice no/IBAN/...	Kotlin multiplatform, Jetpack Compose
preprocess	normalize	Python, Huggingface
	sanitize	Python, Huggingface
	abstract features (numbers, recipient)	Python, Huggingface
	tokenize	Python, Huggingface
	crop (for non-focus mode)	Python, Huggingface
	word embedding	Python, Keras or Fasttext
measure	statistics or review on dataset	e.g. Huggingface DataMeasurementTool, Labeling Tool
	schema inference and validation	e.g. Great Expectations, Tensorflow Data Validation
split	train/valid/test split without data leakage	Python, Huggingface
augment	translate bounding boxes	Python, Huggingface
	shuffle words tagged OTHER	Python, Huggingface
	permute sequence order	Python, Huggingface
	oversample non-OTHER fields	Python, Huggingface
	vary number encoding	Python, Huggingface
train	train BiLSTM or Transformer model	Python, Tensorflow, Keras, Huggingface
validate	F½ model performance (boxes & fields)	Python, sklearn, seqeval, W&B
test	deploy on mobile device	Kotlin multiplatform, TFLite
	deploy in own labeling tool	Kotlin multiplatform, TFLite
	error analysis	Kotlin multiplatform, Jetpack Compose

Table 2: Tasks and applied technologies for each pipeline step

The following two subsections describe the two technology stack items from our core data pipeline with the strongest effect on the data: CLIP and our own labeling tool.

3.1 Data Selection And Reduction Via CLIP

We receive thousands of images to be ingested in our data pipeline, from various sources, with variable style and quality, e.g.

- from various suppliers to various recipients
- some (non-exact) duplicates, while other images are very different and unusual invoices

- some images are not invoices at all (but e.g. maps, pictures of cars, emails, or adds, or empty)
- some are giro transfer forms (e.g. German "SEPA-Überweisung"), others are regular invoices, with or without QR code, single- or multi-page, in German or another language.

We want to reduce and select images to filter and group invoices of specific styles, to create multiple datasets that can later be combined flexibly. This enables suitable data for various use-cases (e.g. training giro transfer forms, using QR code scanning for labeling, testing multi-page KIE).

Furthermore, we want to remove non-exact duplicates: invoices that have very similar style, especially the same layout (usually from the same supplier and the same recipient, with about the same number of items). This quality over quantity approach improves the redundancy quality dimension (useful for training Neural Networks on small datasets [5]), but, more importantly, it reduces the labeling effort.

These semantic reductions and selections on images requires a state of the art computer vision model. We use OpenAI's CLIP [15] (Contrastive Language Image Pre-training), which performs contrastive learning on images and captions, i.e. learns to predict which caption goes with which image, see Fig. 2. It encodes captions (via text tranformer) and images (via vision transformer) into the same latent space, enabling similarity checks between images and captions.

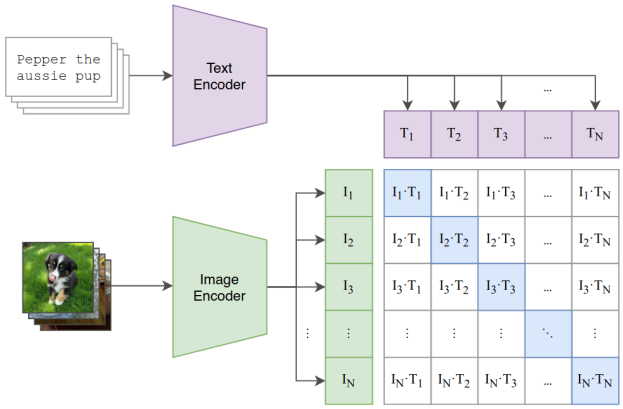

Figure 2: CLIP's contrastive pre-training, see [15]

CLIP has very good zero-shot transfer to downstream tasks, often with competitive model performance (e.g. OCR with zero-shot accuracy of 88%). We employ CLIP for

1. selecting giro transfer forms or invoices with QR code, using another image of a giro transfer form or invoice with QR code as similarity check
2. removing similar images, using pairwise image similarity checks on the original dataset
3. removing images that are not sensible invoices, using as similarity check the caption "Image of an invoice page containing a company name, an invoice number, a customer number, a total amount, and an IBAN.".

Using 2. and 3. together reduces the original datasets to about 10% of their original size. However, the reductions and selections need to be performed semi-automatically: the similarity checks are applied multiple times with manual reviews of their results. This is necessary mainly because the similarity thresholds vary strongly. For instance, two different images of giro transfer forms likely have two different thresholds t_1, t_2 fulfilling the following property: almost all of the images from the original dataset with similarity larger t_i are giro transfer forms, and almost all of the images with similarity smaller t_i are not.

3.2 Labeling Tool for KIE on Images

Though there are many labeling tools available [12], we developed our own tool. It has a very intuitive UI for KIE on images, and we can easily tailor it to our own needs. For instance, it can use a trained model to aid in labeling and for testing. Our tool also has a scan mode that reflects the ability of our online banking app to focus the scan on a single key information of the invoice (e.g. IBAN) in case the key information was not correctly extracted from the full invoice scan (due to errors in OCR or in our AI).

Fig. 3 shows a screenshot of our tool: After choosing a dataset (top left), you can go through the images (left) to label key information (*COMPANY, INVOICE-NO, CUSTOMER-NO, TOTAL, IBAN* for our task) as well as indicator keywords (*TAG-INVOICE-NO, TAG-CUSTOMER-NO, TAG-TOTAL, TAG-IBAN* for our task) on each image. You label by marking OCR bounding boxes with a lasso tool (see blue lasso and 3 boxes marked green) and then clicking on the corresponding label in the top tool bar (or pressing a hotkey). Progress for the dataset and individual images is shown on the left navigation pane, for the current invoice also on the right by listing the OCRed text of each labeled box. Following best practice [21], our labeling guide [11] is directly reachable from the tool ("?" on the top right).

4 Data Quality Measurements

Table 3 lists data quality measurement methods and tools, which either measure on the dataset directly (first 4 lines), or indirectly via ML models trained on the data (last 3 lines). Many tools are just arising out of the data-centric AI movement [7].

Direct measurements can be conducted after any step of the data pipeline: even though the most relevant data quality measurements are at the end of the core data pipeline, shifting left (to measure all but class noise and class imbalance) before labeling can save a lot of cost. Measurements via ML model required the extended pipeline (bottom line of Fig. 1).

Since most of the listed tools are young and cannot

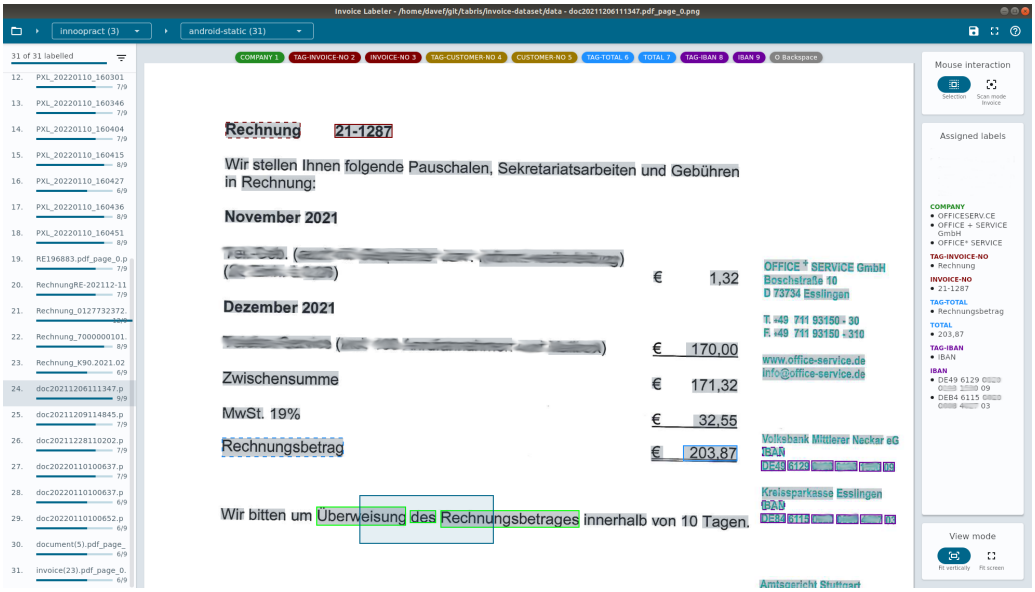

Figure 3: Labeling tool for KIE on images: navigation & progress (left), current invoice, labeled values (right)

Method	Quality Dimensions	Measurements from	Exemplary technology
manually	all but distribution noise	**dataset**	labeling or reviewing tool
comparison to given gold standard	all, mainly class noise	**dataset**	class noise interrater agreement[6,22]
statistics on datasets (distribution distance, outlier detection, ...)	signals for all but inconsistency	**dataset**	Huggingface's Data Measurements Tool[8]
statistics via schemas (inferred or specified manually)	signals for all dimensions	**datasets** & schemas	Great Expectations[10], Tensorflow Data Validation[3]
model performance	signals for all, mainly class noise	the trained **model** the dataset was created for	see previous slides on data quality check via ML model
model confidence	signals for correctness dimensions	the trained **model** the dataset was created for	confidence learning tool Cleanlab[13]
predictions from quality prediction models	signals for correctness dimensions	quality prediction **models**	Consensus Filter[1, 4, 19]

Table 3: Methods and exemplary technology to measure data quality, for the listed dimensions, directly on the dataset or via trained model

handle our nested datastructures, we focus on other data quality measurments in the next two subsections: indirect measurements via model performance and direct measurements via manual quality review.

4.1 Quality Measurement via Model Performance

Though bad data quality leads to bad model performance, the inverse implication does not hold since there are many other reasons for bad model performance, e.g. a badly chosen model architecture.

Model performance should be measured by a metric that is suitable for practice: it should reflect the KPIs for your business case and incorporating risks [9] and what the users expect. Otherwise, validations of the model and of the data quality will be misleading.

Since our app offers a focus mode to correct bad results, accurracy is not so relevant as long as the user easily detects a bad result, which is the case if the field remains empty. Thus precision is more relevant than recall, in line with the assessed risk: an incorrect wire transfer is much worse than an incomplete form that causes user interaction or an aborted wire transfer.

But completely ignoring recall could lead to a cheating model that classifies everyting as OTHER. Thus we incorporate also recall in our metric, but half as strong as precision, leading to the F1/2 score [19].

Another source of misleading measurements is data leakage, i.e. information that should only be available in one dataset (e.g. test set) is leaked into another dataset (e.g. training set), causing too optimistic measurements. For instance, [2] randomly splits one dataset containing many invoices, but all from only eight supplies. Thus the model has likely been trained on all eight invoice layouts, so the validation metric will not measure whether your model is able to generalize to unseen invoice layouts. But this generalization is necessary in our business case (otherwise a template-based approach likely yields better results anyway). Thus we create the split into test, validation, and training set by creating datasets per recipient. This reflects practice, where each recipient (each online banking app user) has some common invoice layouts, which the model has already been trained on, and some uncommon ones, which the model has likely not been trained on and thus requires generalization. Since augmentation can lead to even stronger data leakage, we only use augmentation on the training set.

key information	F½	precision	recall
COMPANY	0.82	0.84	0.76
INVOICE-NO	0.76	0.83	0.61
CUSTOMER-NO	0.66	0.67	0.61
TOTAL	0.78	0.94	0.5
IBAN	0.97	0.98	0.93

Table 4: Average $F\frac{1}{2}$ score for our BiLSTM model

Table 4 shows the model performance for our BiLSTM model, since our transformer models either have lower model performance or are too large for in-

ference on the edge. The table shows performance issues for CUSTOMER-NO, but is this due to data quality issues? Next we measure data quality directly.

4.2 Quality Measurement via Review

Table 5 shows the results of a manual data quality review of 60 labeled invoices by two data scientists. Manual reviews work well on feature noise, class noise, and incompleteness since issues in those dimensions can easily be spotted by inspecting a single invoice. Since redundancy, distribution noise, and class imbalance require the inspection of multiple or all invoices, their manual measurement is too difficult (we did notice one redundancy). Inconsistencies between two invoices are also too difficult, but local inconsistencies, i.e. within a single invoice, have been reviewed, too.

quality dimension	COMPANY	TAG-INVOICE-NO	INVOICE-NO	CUSTOMER-NO	TAG-TOTAL	TOTAL	TAG-IBAN	IBAN
feature noise	10			1			16	11
class noise	12	1	1				2	1
incompleteness	3	1			2	1	2	2
inconsistency	49					1		

Table 5: Issues found by a manual data quality review of 60 invoices (only local inconsistencies reported)

On average, there are 2 quality issues per invoice, but only 6 issues are due to human labeling errors, all others due to OCR. This suggests focusing more on OCR improvements. Furthermore, most issues occur multiple times and are minor, e.g. IBAN containing a trailing ",". Our online banking app fixes some of these minor errors in a post-processing step after KIE, so these minor feature noise issues can alternatively be considered inconsistency issues. Since these minor errors occur so often, the fixes should be moved to the preprocessing pipeline step to remove these quality issues (especially the high amount of feature noise for IBAN) before model training and inference.

To be able to differentiate these minor issues from severe quality issues, the metrics for feature noise, incompleteness, and inconsistency should be weighted, e.g. via the Levenshtein distance, similar to the inconsistency metric in [5].

The low data quality for COMPANY is caused by the complexity and variety of company logos and explains the bad performance of our focus mode on company logos. The low model performance for CUSTOMER-NO (see Subsection 4.1) is not reflected in Table 5, since only one issue was found. The bad performance is likely caused by too little data for CUSTOMER-NO, and class imbalance, suggesting to label more invoices that do contain CUSTOMER-NO.

5 Summary

Seven data quality dimensions were introduced, and methods to measure them, with two depicted in detail: via model performance and via manual review. The dimensions and measurements gave valuable insights and are relevant to all ML projects.

An overview of a data pipeline to achieve high data quality was presented, examplified for KIE from invoices. Two of the most relevant techniques of the data pipeline are depicted in detail: how to use CLIP for semantic reductions and selections of images, and our labeling tool. The tool is relevant for any task performing KIE on images, everything else can easily be transfered to any supervised learning tasks.

A solid data pipeline with data quality measurements is a step towards data-centric AI and worth the investment, even for reasonable-scale ML.

6 Future Work

Our data quality measurements showed that we should improve OCR and label more invoices that contain CUSTOMER-NO, which is both future work. More generally, we plan to improve the metrics for feature noise, incompleteness, and inconsistency to differentiate minor from major quality issues.

Though there are now several data quality measurement tools emerging from the data-centric AI movement, they are still young (e.g. some cannot handle the nested data structures required for KIE and Named Entity Recognition) and it is unclear how well each of them measures the different data quality dimensions and detects root causes for bad quality [3]. We plan to investigate this on our datasets. Hopefully, it will give us useful automations to further improve our data pipeline and data quality.

7 Acknowledgments

Thanks go to Jochen Krause, CEO of Innoopract Informationssysteme GmbH, for offering the opportunity to work on this interesting task, and to my colleagues for the great work environment – to Moritz Post also for creating the labeling tool.

References

[1] Al-Sabbagh, Khaled Walid et al. *Improving test case selection by handling class and attribute noise.* Journal of Systems and Software 183. 2022.

[2] Baviskar, Dipali et al. *Multi-Layout Unstructured Invoice Documents Dataset: A Dataset for Template-Free Invoice Processing and Its Evaluation Using AI Approaches..* IEEE Access, vol. 9. 2021.

[3] Breck, Eric, et al. *Data Validation for Machine Learning.* MLSys. 2019.

[4] Brodley, Carla, and Friedl, Mark. *Identifying and eliminating mislabeled training instances.* Proceedings of the National Conference on Artificial Intelligence. 1996.

[5] Budach, Lukas, et al. *The Effects of Data Quality on Machine Learning Performance.* arXiv preprint. 2022.

[6] Cohen, Jacob. *A coefficient of agreement for nominal scales.* EPM. 1960.

[7] Datacentric AI Resource Hub: https://datacentricai.org.

[8] https://huggingface.co/blog/data-measurements-tool.

[9] Foidl, Harald, and Michael Felderer. *Risk-based data validation in machine learning-based software systems.* Proceedings of the 3rd ACM SIGSOFT. 2019.

[10] Great Expectations Blog Post. *You Are What You Eat: Why Data Quality Matters for Machine Learning.* https://greatexpectations.io/blog/why-data-quality-matters-for-machine-learning.

[11] *Concise Invoice Labeling Guide.* http://tinyurl.com/InvoiceLabelingGuide.

[12] Mariana Neves, Jurica Seva. *An extensive review of tools for manual annotation of documents.* Briefings in Bioinformatics, Volume 22, Issue 1, January 2021, Pages 146–163, https://doi.org/10.1093/bib/bbz130. 2021

[13] Northcutt, Curtis, Lu Jiang, and Isaac Chuang. *Confident learning: Estimating uncertainty in dataset labels.* Journal of Artificial Intelligence Research 70. 2021.

[14] Paullada, Amandalynne, et al. *Data and its (dis) contents: A survey of dataset development and use in machine learning research.* Patterns 2.11. 2021.

[15] Radford, Alec, et al. *Learning transferable visual models from natural language supervision.* International Conference on Machine Learning. PMLR. 2021.

[16] Sambasivan, Nithya, et al. *"Everyone wants to do the model work, not the data work": Data Cascades in High-Stakes AI.* Proceedings of the CHI Conference on Human Factors in Computing Systems. 2021.

[17] Scannapieco, Monica. *Data Quality: Concepts, Methodologies and Techniques. Data-Centric Systems and Applications.* Springer. 2006.

[18] Scheuerman, Morgan Klaus et al. *Do datasets have politics? Disciplinary values in computer vision dataset development.* Proceedings of the ACM on HCI. 2021.

[19] Sluban, Borut, Gamberger, Dragan, and Lavra, Nada. *Advances in class noise detection.* ECAI. 2010.

[20] Tagliabue, Jacopo. *You do not need a bigger boat: recommendations at reasonable scale in a (mostly) serverless and open stack.* Fifteenth ACM Conference on Recommender Systems. 2021.

[21] Tseng, Tina, Amanda Stent, and Domenic Maida. *Best Practices for Managing Data Annotation Projects.* arXiv preprint arXiv:2009.11654. 2020.

[22] Barrett, Leslie, and Michael W. Sherman. *Improving ML Training Data with Gold-Standard Quality Metrics.* 2019.