Semi-automatic staging area for high-quality structured data extraction from scientific literature

Luca Foppiano, Mato Tomoya, Terashima Kensei, Suarez Pedro Ortiz, Tou Taku, Sakai Chikako, Wang Wei-Sheng, Amagasa Toshiyuki, Takano Yoshihiko, Ishii Masashi, et al.

RESEARCH PAPER

Semi-automatic staging area for high-quality structured data extraction from scientific literature

Luca Foppiano[a,b], Tomoya Mato[a], Kensei Terashima[c], Pedro Ortiz Suarez[d], Taku Tou[c], Chikako Sakai[c], Wei-Sheng Wang[c], Toshiyuki Amagasa[b], Yoshihiko Takano[c], Masashi Ishii[a]

[a]Materials Modelling Group, Data-driven Materials Research Field, Centre for Basic Research on Materials, NIMS, JP; [b]Knowledge and Data Engineering, Centre for Computational Sciences, University of Tsukuba, JP; [c]Frontier Superconducting Materials Group, MANA, NIMS, Tsukuba, JP; [d]FKI GmbH, DE

ARTICLE HISTORY
Compiled September 7, 2023

ABSTRACT
In this study, we propose a staging-area for ingesting new superconductors experimental data in SuperCon that is machine-collected from scientific articles. Our objective is to enhance efficiency of updating SuperCon while maintaining or enhancing the data quality. We present a semi-automatic staging area driven by a workflow combining automatic and manual processes on the extracted database. An anomaly detection automatic process aims pre-screen the collected data. Users can then manually correct any errors through user interface tailored to simplify the data verification on the original PDF documents. Additionally, when a record is corrected, its raw data is collected and utilised to improve machine learning models as training data. Evaluation experiments demonstrates that our staging area significantly improves data quality with an increase of 40% in F1-score when comparing to the traditional manual approach of reading PDF documents and recording information in an Excel file. This improvement is primarily attributed to a reduction in missing or overlooked information, resulting in a 6% increase in precision and a 50% increase in recall.

KEYWORDS
materials informatics, superconductors, machine learning, database, tdm

1. Introduction

The emergence of new methodologies using machine learning for materials exploration has given rise to a growing research area called materials informatics (MI) [1]. This field leverages the knowledge of the materials data accumulated in the past to efficiently screen candidates of the materials with desired properties. As a matter of course, such an approach requires a larger number of material-related data for training models. Researchers have been developing large aggregated databases of physical properties generated by first-principles calculations based on Density Functional Theory (DFT), such as Materials Project [2], JARVIS [3], NOMAD [4], that played a role of a strong driving force for the development of materials informatics. Using DFT data

Corresponding authors: Luca Foppiano (luca@foppiano.org) and Masashi Ishii (ISHII.Masashi@nims.go.jp)

for machine learning (ML) in materials science has become popular since, in principle, it allows researchers to simulate and obtain various types of physical properties of the target materials only by knowing the crystal structures of the subjects. Those DFT codes are designed so that they reproduce/simulate the physical properties that should be observed by experiment in reality. Nonetheless, caution must be exercised while utilising these computed figures for constructing ML models aimed at steering experiments. This caution arises due to the potential lack of validity in their predictions when dealing with specific simplifications of the interactions between atoms and electrons in solids, such as electron-electron Coulomb correlation, spin-orbit coupling, and similar factors.

Au contraire, accumulated datasets of experimental data from scientific publications are still scarce, despite abundant availability of publications, and exponential growth in materials science [5]. Currently, only a few limited resources exist, such as the Pauling File [6] and SuperCon [7], necessitating reliance on manual extraction methods. This scarcity can be attributed to inadequate infrastructure and a shortage of expertise in the field.

SuperCon [7] was built manually from 1987 [8] by the National Institute for Materials Science (NIMS) in Japan and it is considered the gold standard in superconductors research. Despite being praised for its excellent quality in numerous reports [9–12], the updates of SuperCon have become increasingly challenging due to the high publication rate. However, in response to the need for a more efficient approach to sustain productivity, we embarked on the development of an automated system for extracting material and property information from the text contained in relevant scientific publications [13]. This automated process enabled the rapid creation of SuperCon^2, a comprehensive database of superconductors containing around 40000 entries, within an operational duration of just a few days. Matching the level of quality seen in SuperCon while simultaneously automating the extraction of organised data can be achieved with a properly designed curation process. We define as *curation* the general term indicating the correction and validation of records in a database as a whole, and *correction* as the specific process of modifying the values of one or more properties in a single record. At the moment of writing this article, we are not aware of any other curation tool focusing on structured databases of extracted information. There are several tools for data annotation, such as Inception [14], and Doccano [15] which concentrate on text labelling and classification.

In this work we have designed a workflow along with a user interface crafted to simplify the curation procedure, encompassing the continuous and engaged oversight of data throughout its pertinent life cycle. This framework is custom-tailored to our superconductors database, yet it holds the potential for being adapted to alternative data frameworks. We aim to produce structure data of a similar or superior quality to the one obtained by the classical manual method of reading PDF documents and noting information in an Excel file.

Our contributions to the field can be summarised as follows:

- A user interface and a workflow acting on a machine-collected database. The interface exploits database navigation and an enhanced document viewer, similar to [16]. We demonstrated our solution improves the quality of the curation as compared with the manual method (Section 5.3).
- We propose a mechanism that selects training data based on corrected records, and we demonstrate that such selections are rapidly improving the ML models (Section 5.2).

- We devise an integrated anomaly detection process for the identification of outliers in the materials-properties database which results in a lower rejection rate (false positive rate) from domain-experts (Section 5.1).

The subsequent section (Section 2) presents the data ingestion process. Section 3 describes the curation workflow, and Section 4 the user interface on top of it. Finally, we discuss our evaluation experiments in Section 5.3.

2. Ingestion process

The ingestion process (Figure 1) is designed using an Extract-Aggregate approach we briefly introduced in our previous work [13]. In this section, we summarise the architecture (Section 2.1) and focus on the data transformation from the PDF document to the structured database (Section 2.2).

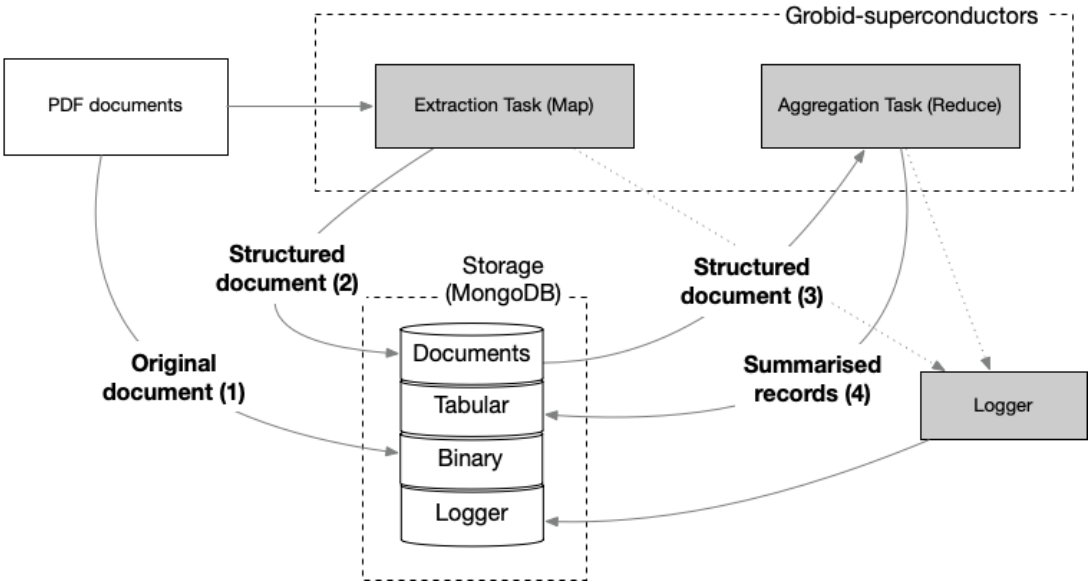

Figure 1. Ingestion process

The "Extraction Task" takes as input PDF documents, stores them, and then processes them with Grobid-superconductors. Grobid-superconductors transforms the PDF documents into a rich representation document containing the original text and the extracted information in JSON format, which we will refer to as *structured documents*. The "Aggregation Task" takes in input the *structured document* and transforms it to a table format where each row contains one material, its T_c and their related properties (we refer to as *summarised record*, or, simply *record*).

2.1. Architecture

The two ingestion tasks are implemented in separate Python scripts that can run asynchronously either with a scheduler or using a publish-subscriber triggering mechanism. The storage is implemented using MongoDB[1], an open-source document database. We

[1] `https://www.mongodb.com`

design the database in five collections:

- **binary**: contains the original PDF documents
- **documents**: contains the *structured documents*
- **tabular**: stores the *summarised records*
- **logger**: contains information on the processing status of each document, including errors and status codes from the Grobid-superconductors API (Section 4.3).
- **training_data** collects the raw information that can then be exported as training data (Section 3.3)

We compute the unique signature for each original document by using the first 10 characters of the MD5 hash function on the binary content. We use this information to link the original document, the *structured document* and the *summarised records*.

2.2. Data formats

The *structured document* contains three main sections: a) bibliographic data (authors, DOI, title, publisher, journal, year of publication), b) runtime execution time from the server side (excluding the network and database delay), and c) a list of text passages, each representing a sentence. Each passage is then composed of the following attributes (identified in orange in Figure 2):

- the text of the passage
- the type of passage: sentence, or paragraph
- the main section: body, header, or annex
- the subsection within the section: title, abstract, paragraph, caption, etc.
- the list of spans where each span represents one entity extracted from the text (in blue in Figure 2)
- a list of layout tokens, which contains, for each token in the passage the layout information such as font size, font name, superscript, subscript, bold, and italic.

Each span aggregates a complex set of information illustrated in an example in Figure 2 and identified in blue: links (in green), attributes (in red), PDF "boxes" coordinates (in yellow), and annotation reference to the sentence (in light blue). The general information is composed of a unique identifier (calculated on certain span attributes), the *text* contains the value of the extracted entity (e.g. *FeSe*), the entity type or class (e.g. `<material>`). Furthermore,*Linkable* indicate whether the span respects the criteria to be linked to other entities and *source* the ML model from which the entity was extracted.

The links (green) indicate linked entities and their type (in our example the material FeSe is linked to 13K of type `<tcValue>`. The type indicates the algorithm used for extracting the relation and the "targetId" indicates the unique id of the relation target span (13 K).

The attributes are stored as a key-value and contain additional information extracted by the subsequent models such as the Material Parser. For example, the chemical formula is stored both as a string and as structured composition (a list of elements and their amount in the formula), the material class, and so on.

The "PDF boxes coordinates" (yellow) are expressed as lists of objects containing page number, top-right and bottom-left coordinates[2]. This information is sufficient to

[2]Details on coordinates is available in the official grobid documentation `https://grobid.readthedocs.io/en/latest/Coordinates-in-PDF/#coordinates-in-teixml-results`

draw a set of rectangles on the PDF document ("the boxes") that encapsulate groups of tokens belonging to each annotation. An example of the final result can be seen in Figure 7.

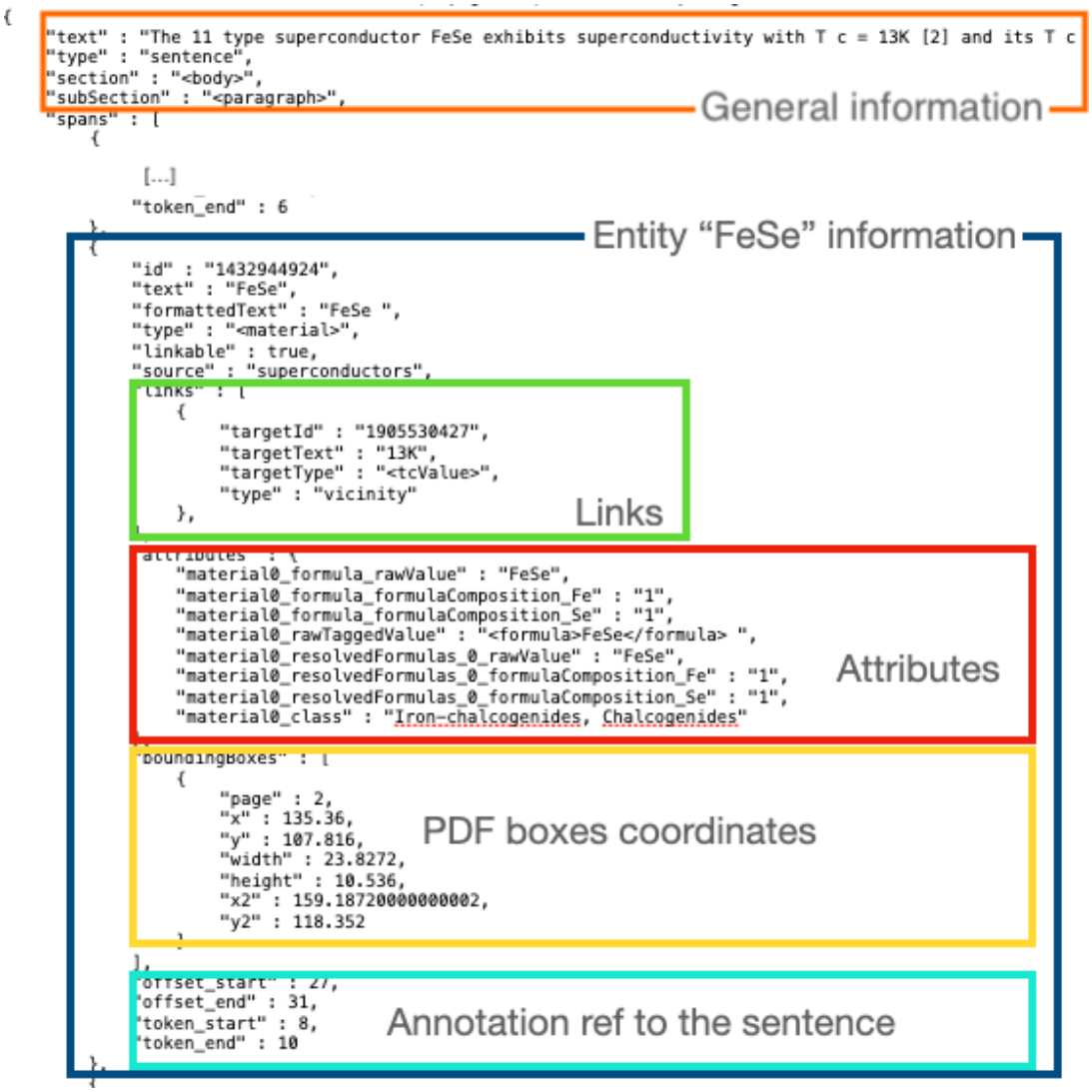

Figure 2. Example of span encoding information of the extracted material *FeSe*

The "Aggregation Task" transforms the "structured document" in input to a tabular format comprising as many rows as the extracted materials records. Each column of the table represents the related information and properties. We format each record in JSON format required by MongoDb as illustrated in Figure 3.

The aggregation pivots around the relation materials-Tc and then attaches additional elements to it. A detailed description of the main fields can be found in our previous work [13]. The span objects corresponding to the entities linked together are repeated in this structure and are used to visualise the passage decorated with the extracted entities in this record.

For complex material names corresponding to multiple entities are split and stored in several records. For example, a material name containing substitution variables like M Fe O (M=La, Cu) will result in two records having materials such as "La Fe O"

```
{
  "_id": ObjectId("63dcae91e4d716dd10dd5a7d"),
  "rawMaterial": "FeSe",
  "materialId": "1432944924",
  "name": null,
  "formula": "FeSe",
  "doping": null,
  "shape": null,
  "materialClass": "Chalcogenides, Iron-chalcogenides",
  "fabrication": null,
  "substrate": null,
  "variables": null,
  "criticalTemperature": "13K",
  "criticalTemperatureId": "1905530427",
  "measurementMethod": "",
  "measurementMethodId": "",
  "appliedPressure": null,
  "appliedPressureId": null,
  "section": "body",
  "subsection": "paragraph",
  "sentence": "The 11 type superconductor FeSe exhibits superconductivity with T c = 13K [2] and its T c reaches
  37K under high pressure (4-6 GPa) [3,4].",
  "spans": [
    {
      "id": "1432944924",
      "text": "FeSe",
      "type": "<material>",
      "linkable": false,
      "offset_start": 27,
      "offset_end": 31,
      "token_start": 0,
      "token_end": 0
    },
    [...]
  ],
  "hash": "f70a71214f",
  "type": "automatic",
  "timestamp": ISODate("2022-11-24T09:02:53.256Z"),
  "status": "new",
  "title": "Evidence of Inhomogeneous Superconductivity in FeTe1-xSexby Scotch-Tape Method",
  "doi": "10.1143/jpsj.81.113707",
  "authors": "Hiroyuki Okazaki, Tohru Watanabe, Takahide Yamaguchi, Yasuna Kawasaki, Keita Deguchi, Satoshi
  Demura, Toshinori Ozaki, Saleem. J. Denholme, Yoshikazu Mizuguchi, Hiroyuki Takeya, Yoshihiko Takano",
  "publisher": "Physical Society of Japan",
  "journal": "Journal of the Physical Society of Japan",
  "year": 2012
}
```


Figure 3. Example of the aggregated record corresponding to the *FeSe* material.

and "Cu Fe O", respectively. In many other cases inferring the correct interpretation is left to curators. For example, the expression "Zn and Cu doping La Fe B" possibly has two meanings. Namely, it means LaFeB doped with both Zn and Cu at the same time, or LaFeB doped with Zn, and LaFeB doped with Cu.

3. Curation workflow

The curation process can be delineated as a structured workflow, wherein each record undergoes a series of transitions between distinct states. These transitions are determined by the actions that are executed on the record, encompassing various stages of refinement and enhancement. Figure 4 illustrate the concept in detail. A record enters the workflow at the ingestion process (Section 2) and its state can change by manual action or automatic process. In this workflow, the automatic process is "anomaly detection" (Section 3.2) which aims to automatically mark outliers. There are four types

of manual actions:

- **mark as valid**: when a record is considered corrected, without the need for any correction,
- **mark as invalid**: the record is considered potentially invalid
- **remove**: the record is considered invalid
- **manual correction**: the record is updated
- **reset**: the statuses previously changed (e.g. marked as valid or invalid) are reset

.

Given an action that updates a field in a record, we define the *original record* as the record **before the modification**, and the *updated record* as the new record **after the modification**. The record data is persisted in every state of the workflow: in case of modifications both the "original" and "updated" records are kept and the "original" record is hidden in the user interface. Similarly, when a record is removed the data is kept but the record is hidden, while records marked as valid or invalid are kept visible. Such an approach is justified by the need to generate the history of each record over time (Section 4.3) and support the implementation of the undo/redo functionality if needed in future.

The workflow establishes also that when a record is manually corrected, the raw data from which the record has been extracted, are collected as training data. This policy is indicated as "(*)" in Figure 4 and is described in detail in Section 3.3.

3.1. Workflow control

The action performed on the record combined with the curation status determines the next stage of the workflow. The curation status characterises each state in the workflow and is encoded by a combination of two fields: *type*, and *status* described in Section 3.1.1. In addition, when a record is corrected we collect details regarding the type of error, and this process is described in Section 3.1.2).

3.1.1. Curation status

The curation status is defined by two internal fields of each record: "status" and "type", and they are illustrated in Figure 4, inside each rounded shape.

The field "status" indicates the record state in the workflow. Some of the "status" values are visible to the users (e.g. validated, curated, invalid) while others (e.g. obsolete, removed) are used internally. Internal statuses are assigned to records that need to be hidden in the user interface (e.g. removed records are no longer visible).

The status definitions can be summarised as follows:

- **new**: default status when a new record is created.
- **curated**: the record has been amended manually.
- **validated**: the record was manually marked as valid.
- **invalid**: the record is wrong or inappropriate for the situation (e.g., T_m or T_{curie} extracted as superconducting critical temperature).
- **obsolete**: the record has been updated and the updated values are stored in a new record (internal status).
- **removed**: the record has been removed by a curator (internal status).

The field "type" indicates whether the record has been modified by a manual or an automatic process. For example, the value "automatic" is provided when the data

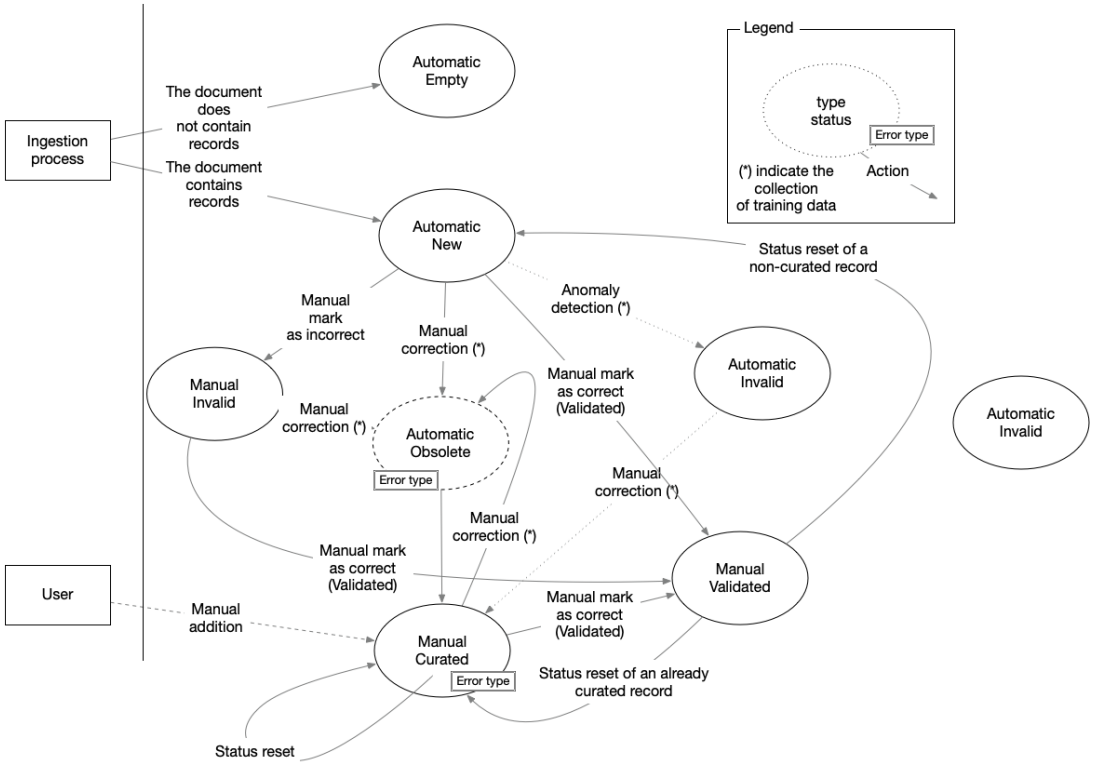

Figure 4. Schema of the curation workflow. Each state is characterised by two properties: type and status, and one action, as indicated in the top right corner. "Error type" indicates the action of storing the error type for that specific action.

is ingested (Section 2) or when the "anomaly detection" detects incorrect values and performs operations such as marking the record "invalid". The "type" can change from "automatic" to "manual" but never in the opposite direction, because automatic operations are never applied on validated or curated records.

3.1.2. Error types

Error types were first introduced in [13] while performing manually the end-to-end evaluation. They were combined with the evaluation to provide a more detailed explanation of the reasons why certain extracted values were not correct. Since such statistics demonstrated to be useful during the development, we have extended their scope and added additional values related to data curation and validation.

Selecting the *Error Type* is mandatory in each of the manual actions defined in Figure 4 and it is stored in the "original" record. In the case of an automatic process, anomaly detection sets the *Error Type* with a special status "anomaly detection" to indicate the origin of the modification.

The error type values can be summarised as follows:

- **From table**: the entities Material → Tc → Pressure are identified in a table. At the moment, table extraction is not performed
- **Extraction**: The material, temperature, and pressure are not extracted (no box) or extracted incorrectly.
- **Linking**: The material is incorrectly linked to the Tc given that the entities are

correctly recognised.

- **Tc classification**: The temperature is not correctly classified as "superconductors critical temperature" (e.g., Curie temperature, Magnetic temperature...).
- **Composition resolution**: The exact composition cannot be resolved (e.g., the stoichiometric values cannot be resolved).
- **Value resolution**: The extracted formula contains variables that cannot be resolved, even after having read the paper. This includes when data is from tables
- **Anomaly detection**: The data is automatically modified by the anomaly detection script.
- **Curation amends**: The curator is updating the data which does not present issues due to the automatic system.

3.2. Anomaly detection

Anomaly detection is the process of identifying unusual events or patterns in data. In our context, this means identifying data that are greatly different from the expected values. This post-process was introduced in a limited scope to draw attention to certain cases during the curation.

The anomaly detection uses a rule-based approach and marks any record that matches the following conditions:

- the extracted T_c is greater than room temperature (273 K), negative, or contains invalid characters and cannot be parsed (e.g. "41]")
- the chemical formula cannot be processed by an ensemble composition parser that combines Pymatgen [17], and text2chem [18]
- the extracted applied pressure cannot be parsed or falls outside the range 0 - 250 GPa.

Records identified as anomalies are then a) marked as "invalid", and b) attached with a special "error type" called "anomaly detection" for easy identification. Since this process may find false positives, its output requires validation from curators. For example, in certain contexts, T_c values above room temperature or applied pressure up to 500 GPa may be valid in researchers' hypotheses, calculations, or simulated predictions.

When we ran the anomaly detection on the full SuperCon[2] database, it identified 1506 records with invalid T_c, 5021 records with a chemical formula that was not parseable, and 304 records with invalid applied pressure. We also identified only 1440 materials that have been linked to multiple T_c values. Further analysis and cross-references with this information may be added in future development.

3.3. Automatic training data generation

The curation process is a valuable endeavour demanding significant knowledge and human effort. It is crucial to maximise the use of this time for collecting as much information as possible. For this reason, we integrated an automatic procedure in the curation process for accumulating examples that can be used for training data in ML models.

Since the training data generated are related to manual corrections, they are targeting real mistakes from the ML model. In Section 5.2 we demonstrate they have

a relevant impact on improving the ML model with a small number of examples as compared with the training dataset.

3.3.1. Training data collection

In the event of a change (update, removal) in a database record, this process retrieves the corresponding raw data: the text passage, the recognised entities (spans), and the layout tokens information. This information is sufficient to be exported as training examples, which can be examined and corrected, and feedback to the ML model.

In detail, the process performs the following actions:

- The updated record is prepared and stored.
- The raw data originating the updated record is identified. First, the corresponding structured document is retrieved from the document collection using the document identifier (the hash). Then, the exact text passage in the structured document is located using a unique id assigned to each material in the database records.
- If the raw data has already been collected, it is skipped. This is the case when multiple records belonging to the same text passage are corrected.
- Otherwise, the raw information comprising the text string, the spans, and the layout tokens are collected and saved in a separate collection.
- The data collected is then sufficient to generate workable instances in different output formats and the related feature files.

3.3.2. Training data management

We designed a specific page of the interface (Section 4) to manage the collected data (Figure 5) in which each row corresponds to a training example, which includes the decorated text showing the identified entities, the document identifier, and the status. The users can examine the data, delete it, or send it to the annotation tool to be corrected. Depending on which state the records are in, the status can be: "new" when the data is added, "in progress" after the data is sent to the annotation tool, and "exported" when the corrected training data is downloaded. We integrated Label-studio [19] for the correction of the collected. Label-studio is an open-source, python-based, and modern interface supporting many different TDM tasks (NER, topic modelling, image recognition, etc.).

4. Curation interface

The workflow is operated through the user interface, which offers several key features to facilitate the data curation process. It provides a comprehensive view of materials and their related properties as a table which includes search, filtering, and sorting functionality (Figure 6). The schema consists of two main classes: material information (material names, formulas, shape, etc.), properties (T_c), and conditions (applied pressure, measurement method, etc.). The complete list including examples is reported in our previous work [13].

During the curation process, it is often necessary to switch back and forth between the database record and the related context in the paper (the related paragraph or sentence). Our interface provides a viewer for individual documents, which visualises in the same window a table with the extracted records and the original PDF docu-

Figure 5. Training data view

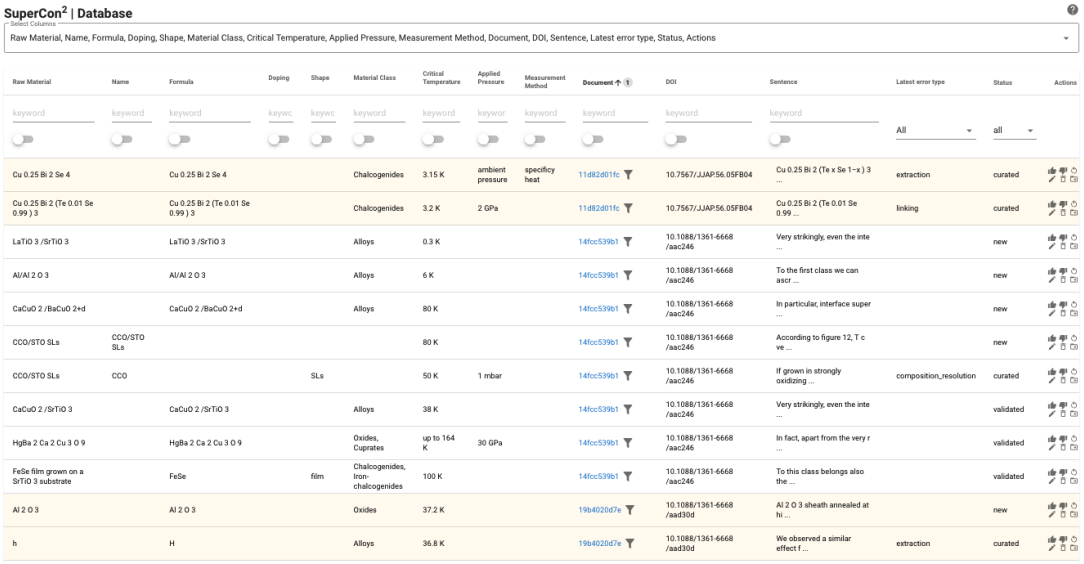

Figure 6. Curation interface showing the database as a table

ment (Figure 7). The PDF document is decorated with annotations that identify the extracted materials and properties, enabling users to easily locate and reference the extracted information within the document.

Through the interface, users can add, amend, remove, or mark each of the records in the database. Marking a record imply declaring explicitly the record as invalid or valid. Adding new records is limited to documents already in the database. When a record is added to a document, the record's bibliographic data are copied from other records in the same documents and the user has to only care filling up the correct experimental information (material, Tc, etc.).

4.1. Manual curation approach

Manual curation is still indispensable for developing high-quality structured data since the data extracted automatically may contain incorrect information. We have set up

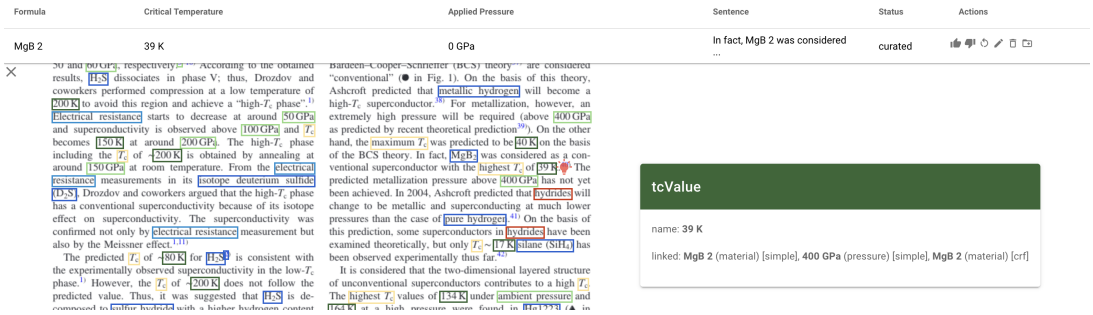

Figure 7. PDF document viewer showing an annotated document. The table on top is linked through the annotated entities. The user can navigate from the record to the exact point in the PDF, with a pointer (the red bulb light) identifying the context of the entities being examined.

an automatic process for anomaly detection (Section 3.2) which can help to speed up the process but it only detects "potential" problems and requires anyway a manual validation.

To certify the gold-standard data quality, we employ only curators from domain experts in the field. Although, even among experts, experience plays an important role (Section 5.3) To avoid worker dependence and ensure robustness in the process, we took two approaches. First, we used a double-round approach where the data is initially corrected by one person, and validated in a second round, by a different person. Second, we have built documentation for the curation as a form of guidelines through an iterative loop of processes, as discussed in our previous work on the construction of the annotated dataset SuperMat [20].

The loop includes four steps:

- collect rules, based on observation and reasoning,
- curation following those rules,
- retrospective including analysis and discussions based on curators' feedback, and
- take decisions and update the guideline

4.2. Curation guidelines

The guidelines consist mainly of two parts: the general principles and the correction rules with examples of solutions. The guidelines are designed to provide general information applied to corrections and very basic explanations containing illustrations for a faster understanding (e.g. the meaning of the colours of the annotations). This would help new curators to catch up with the required level of curation precision quickly. There are two main components described in the correction rules: the record that is being corrected and its context. The context of a record can be obtained by examining the extracted annotated text or the PDF document area.

The correction rules are described based on the error type mentioned in Section 3.1.2, and in the guideline, the description of rules is accompanied by sheets that explain five points to the curators, as illustrated in Figure 8:

- **Sample input data**, a screenshot of SuperCon^2 record in the interface
- **Context**, a screenshot of the related part of the document (either from the PDF or from plain text sentences) that contain the extracted data to be curated,
- **Motivation**, describes the issue with the examined extracted data,
- **Action** to be taken,

- **Expected output**, a screenshot of the expected SuperCon^2 record, after correction

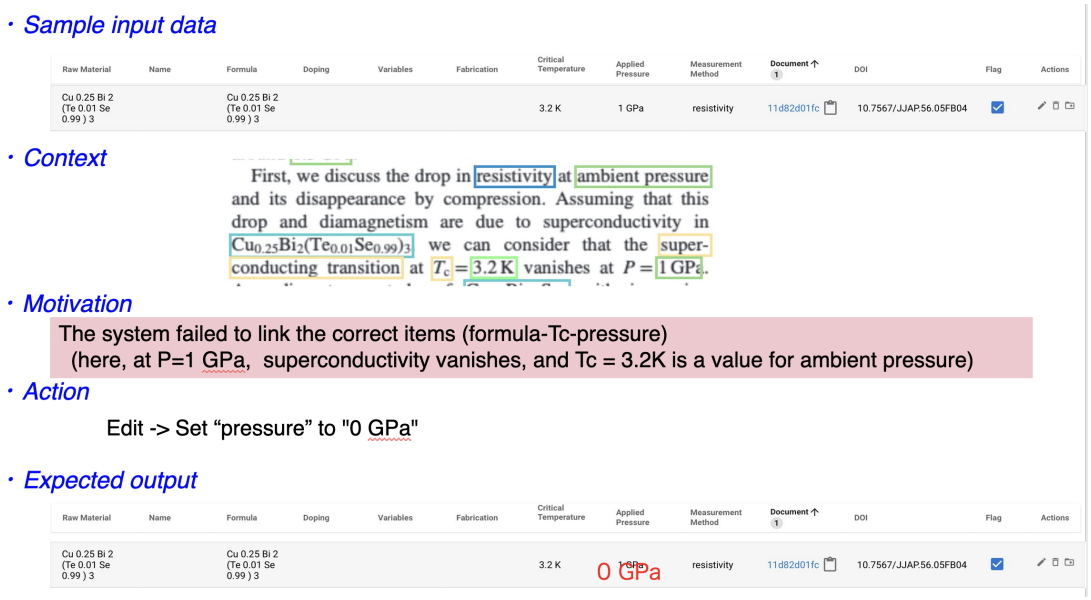

Figure 8. Example of curation sheet. As discussed, it's written with simple language assuming the curator may not be familiar with the task.

4.3. Curation and processing logs

The Supercon^2 interface gives access to information regarding the ingestion (processing log) and the curation process (curation log). The processing log is filled up when the data is ingested, it was built to have minimal functions able to explain why certain documents haven't been processed (Figure 9). For example, sometimes documents are failing because they don't contain any text (image PDF documents) or they are too big (more than 100 pages).

The curation log provides a view of what, when and how a record has been corrected (Figure 9).

5. Results and evaluation

In this section, we illustrate the experiments we have run to evaluate our work. The evaluation is composed of three sets of results. The anomaly detection rejection rate (Section 5.1) indicates how many anomalies were rejected by curators after validation. Then, we demonstrate that the training data automatically selected contributed to improving the ML model with a small set of examples (Section 5.2) Finally, we evaluated the quality of the data extraction using the interface (and the semi-automatic TDM process) against the classical method of reading the PDF articles and noting the experimental information in an Excel file. In Section 5.3 we find that using the interface improves the quality of the curated data and in particular in terms of recall, where the interface helps reduce the missing experimental data.

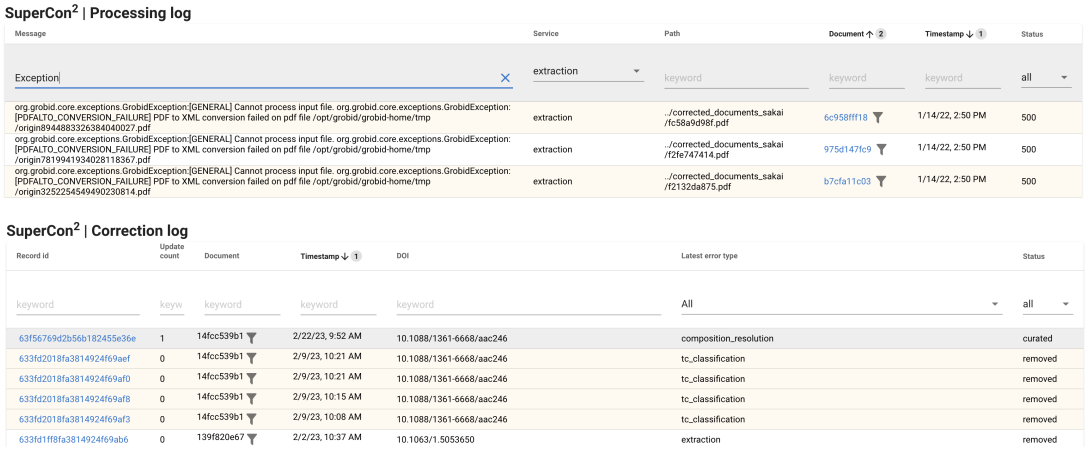

Figure 9. On the top: Processing log, showing the output of each operation (process document, create a record) and the outcome with the exception or error that should have occurred. On the bottom: Curation log, indicating each record, the number of updates, and the date/time of the last updates.

5.1. Anomaly detection rejection rate

We evaluate the anomaly detection and examined the detected anomalies rejected by human validation. We considered a small subset of the database containing 667 records. The detection found 17 anomalies in T_c, 1 anomaly in applied pressure, and 16 anomalies in the chemical formulas. The percentage of anomaly detection results rejected by curators after rechecking was 23% for Tc, 37% in chemical formulas and 0% for applied pressure. This indicates appropriate effectiveness and a relatively low rate of false positives, although a detailed study might be needed due to the small sample size.

5.2. Training data generation

We selected around 400 Supercon^2 extracted records initially marked by the anomaly detection process. Following the guidelines, we corrected these records to exclude false positives wrongly identified by anomaly detection. At the same time the interface collected examples of training data based on our corrections. Then, after we corrected the obtained set of raw data we obtained a small set of 352 training data examples for our superconductors ML models. We call the obtained dataset *curation* to be distinguished from the original SuperMat dataset which is referred to as *base*.

We prepared our experiment using the SciBERT [21] implementation, and we fine-tuned the model for our downstream NER task discussed in detail in [13]. We trained five models, and evaluate them using a fixed holdout dataset from SuperMat, and we averaged the results for smoothing out the fluctuations in the results. We use the DeLFT (Deep Learning For Text) [22] library for training, evaluating, and managing the models for prediction. A model can be trained with two different strategies:

(1) *"from scratch"*: when the model is initialised randomly. We denote this strategy with an *(s)*.
(2) *"incremental"*: when the initial model weights are taken from an already existing model. We denote this strategy with an *(i)*.

The latter can be seen as a way to "continue" the training from a specific checkpoint.

We thus define three different training protocols:

(1) **base(s)**: using the *base* dataset and training from scratch (s).
(2) **(base+curation)(s)**: using both the *base* and *curation* datasets and training from scratch (s).
(3) **base(s)+(base+curation)(i)**: Using the *base* dataset to train from scratch (s), and then continuing the training with the *curation* dataset (i).

We merge "curation" with the base dataset because the curation dataset is very small compared to "base", and we want to avoid catastrophic forgetting [23] or overfitting.

The trained models are then tested using a fixed holdout dataset that we designed in our previous work [13] and the evaluation scores are shown in Table 1.

Table 1. F1-score from the evaluation of the fine-tuning training of SciBERT. The training is performed with three different approaches. The *base* dataset is the original dataset described in [13], and the *curation* dataset is automatically collected based on the database corrections by the interface and manually corrected. *s* indicate "training from scratch", while *i* indicate "incremental training". The evaluation is performed using the same holdout dataset from SuperMat. The results are averaged over 5 runs.

	base(s)	**(base+curation)(s)**	**base(s)+curation(i)**
Nb total examples	16902	17254	16902(s), 17254 (i)
`<class>`	70.41	**73.02**	71.86
`<material>`	79.37	80.09	**80.37**
`<me_method>`	66.72	66.57	**66.95**
`<pressure>`	46.43	**48.42**	47.23
`<tc>`	80.13	**80.92**	80.34
`<tcValue>`	78.29	78.41	**79.73**
All (micro avg.)	76.67	77.44	**77.48**
Δ avg. w/ baseline	-	+0.77	**+0.81**

Table 2. Data support: number of entities of a specific type in the training dataset.

	base	**base+curation**	**Δ**
`<class>`	1646	1732	86
`<material>`	6943	7580	637
`<me_method>`	1883	1934	51
`<pressure>`	274	361	87
`<tc>`	3741	4269	528
`<tcValue>`	1099	1556	457
Total	15586	17432	1846

This experiment demonstrates that with only 352 examples (2% of the SuperMat dataset) comprising 1846 additional entities (11% of the entities from the SuperMat dataset) (Table 2), we obtain an improvement from 76.67%[3] to an F1-score between

[3]In our previous work [13] we reported 77.03% F1-score. There is slight a decrease in absolute scores between DeLFT 0.2.8 and DeLFT 0.3.0. One cause may be the use of different hyperparameters in version 0.3.0 such as

77.44% (+0.77) and 77.48% (+0.81) for (base+curation)(s) and (base(s)+curation(i)), respectively.

This experiment gives interesting insight relative to the positive impact on the way we select the training data. However, there are some limitations: the *curation* dataset is small as compared with the *base* dataset. This issue could be verified by correcting all the available training data and repeating this experiment, and studying the interpolation between the size of the two datasets and the obtained evaluation scores. A second limitation is that the hyperparameters we chose for our model, in particular, the learning rate and batch size could be still better tuned to obtain better results with the second and third training protocols.

5.3. Data quality

We conducted an experiment to evaluate the effectiveness and accuracy of data curation using two techniques: a) the user interface, and b) the traditional manual method involving reading PDF documents and populating an Excel file.

We opted for a sample dataset consisting of 15 papers, which we divided among three curators — a senior researcher, a PhD student, and a master's student. Each curator received 10 papers, with an equal distribution between using the *interface* and working with *pdf* files. There was an overlap of 5 papers between curators, where the opposite method was applied. For instance, if curator A used the *interface* to correct paper 1, curator B, who also had the same paper, corrected it by reading the *pdf* document. After curation, a fourth individual manually reviewed the curated content. The revisions made during this process were then employed to calculate the evaluation scores.

We assessed the two approaches using a dual perspective: efficiency, by contrasting the time needed for curation, and accuracy, quantified through precision, recall, and the F1-score.

5.3.1. Discussion

The comparison of the time taken revealed no significant difference between the interface and the traditional method. Specifically, the total time was only 4 minutes longer with the interface (188 minutes compared to 184 minutes). The time difference did not demonstrate any consistent trend, suggesting the need for a larger dataset in future experiments.

When we examined the accuracy of the extracted data, we observed an improvements of +5.54% in precision and a substantial +50.79% in recall when using the interface (Table 3). The F1 score improves by 40.62%.

Table 3. Evaluation scores (P: precision, R: recall, F1: F1-score) between the curation using the SuperCon 2 interface (Interface) and the traditional method of reading the PDF document (PDF document).

	P (%)	**R (%)**	**F1 (%)**
PDF document	87.83	45.60	52.66
Interface	**93.37**	**96.39**	**93.28**

batch size and learning rate. However, the most probable cause could be the impact of using the Huggingface library which is suffering from quality issues in relation to their tokenizers implementation `https://github.com/kermitt2/delft/issues/150`.

The disparity in experience significantly influenced the accuracy of curation, particularly in terms of high-level skills. Senior researchers consistently achieved an average F1-Score approximately 20% higher than other curators (see Table 4). Furthermore, we observed a modest improvement between master's students and PhD students. These findings indicate also that for large-scale projects, employing master students instead of PhD students may be a more cost-effective choice. Thus, using only a few senior researchers for the second round of validation (Section 4.1).

Table 4. Evaluation scores (P: precision, R: recall, F1: F1-score) aggregated by experience

Experience	**P (%)**	**R (%)**	**F1 (%)**
Master student	90.03	66.10	66.40
PhD student	83.33	65.69	69.45
Senior researcher	**98.45**	**81.22**	**83.08**

Finally, the collected data suggest that all three curators had overall more corrected results by using the interface as illustrated in Table 5.

Table 5. Evaluation scores (P: precision, R: recall, F1: F1-score) listed by experience (Master student, PhD student, and Senior researcher), and method (PDF document, Interface)

Experience	**Method**	**P (%)**	**R (%)**	**F1 (%)**
Master student	PDF Document	94.58	36.55	48.67
	Interface	83.19	95.83	88.25
PhD student	PDF Document	70.00	48.51	50.78
	Interface	96.67	82.86	88.11
Senior researcher	PDF Document	**100.00**	55.56	61.03
	Interface	97.42	**98.33**	**97.78**

The results of this experiment confirmed that utilising the interface in conjunction with an automated system required a comparable amount of time for curating SuperCon data compared to the "traditional method." However, it significantly improved the quality of the extracted data. Additionally, the following observations were made during the curation process:

- The interface requires a finite adaptation time, in particular at the beginning of the work. The curators that were starting the evaluation from the interface tended to ask questions about the usage, primarily due to their lack of familiarity.
- The interface demonstrated a substantial increase in recall. Our intuition suggests the interface overcomes the tendency to overlook information when reading the plain PDF document.

6. Code availability

This application is freely available at `https://github.com/lfoppiano/supercon2`, the repository contains:

- the code of the SuperCon 2 curation interface for visualising and editing material and properties extracted from superconductors-related papers.
- The ingestion workflow to create process PDF documents with Grobid-superconductors and produce a database of materials and properties.
- the guidelines, accessible at `https://supercon2.readthedocs.io`

7. Conclusions

We built a staging area for SuperCon to allow the production of manually-curated high-quality data collected from scientific articles using an automated TDM process. The data is extracted from PDF documents using Grobid-superconductors [13] and is stored in a structured database. We designed a curation workflow that leverages both automatic and manual operations: anomaly detection automatically identifies outliers and individuals can manually correct records through a user interface specifically tailored to optimise quality and mitigate mistakes. The interface combines the best practices in user interaction design and provides, among many other features, an enhanced PDF document visualisation and rapid transitions from the database records the related section in the original document. We reported that our interface achieves higher precision while requiring the same time for curating, as compared with the traditional manual process. The interface also automatically collects data related to the correction that can be used to feed back the ML models as training data. We demonstrated that the feedback loop based on corrected data can substantially improve the machine learning models with training data targeting incorrect recognition by the ML models.

There are several planned features and improvements in the pipeline. Some of these include:

- Undo/redo functionality: The ability to undo and redo changes made to records will be added, to make it easier to correct mistakes.
- Document versioning: A versioning system will be implemented to track changes to documents over time.
- Improved search: The search functionality will be improved to make it easier to find records based on specific criteria.
- Additional record types: SuperCon^2 currently supports records for material and property information, but additional record types will be added.

Acknowledgements

Our warmest thanks to Patrice Lopez, the author of Grobid [24], DeLFT [22], and other open-source projects for his continuous support and inspiration with ideas, suggestions, and fruitful discussions. We thank Pedro Baptista de Castro for his support during this work.

Funding

This work was partly supported by MEXT Program: Data Creation and Utilization-Type Material Research and Development Project (Digital Transformation Initiative Center for Magnetic Materials) Grant Number JPMXP1122715503.

Notes on contributors

LF wrote the manuscript. LF and POS discussed the ML results and experiments. LF implemented the workflow as a standalone service, and TM wrote the front end of the user interface. LF designed the user interface experiment with KT, TT and WS as curators. KT lead the materials-science work on the data with CS, TT and WS. KT, TA, YT and MI revised the paper. YT and MI supervised the work of the respective teams.

References

[1] O. N. Oliveira and M. J. Oliveira. Materials discovery with machine learning and knowledge discovery. *Front. Chem.*, 10, 2022.

[2] Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, and Kristin A. Persson. Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL Materials*, 1(1):011002, 2013.

[3] Stefano Curtarolo, Wahyu Setyawan, Gus L.W. Hart, Michal Jahnatek, Roman V. Chepulskii, Richard H. Taylor, Shidong Wang, Junkai Xue, Kesong Yang, Ohad Levy, Michael J. Mehl, Harold T. Stokes, Denis O. Demchenko, and Dane Morgan. Aflow: An automatic framework for high-throughput materials discovery. *Computational Materials Science*, 58:218–226, 2012.

[4] C. Draxl and M. Scheffler. The nomad laboratory: From data sharing to artificial intelligence. *J. Phys. Mater.*, 2:036001, 2019.

[5] Thuraiyappah Pratheepan. Global publication productivity in materials science research: A scientometric analysis. *Indian Journal of Information Sources and Services*, 9(1):111–116, Feb. 2019.

[6] Evgeny Blokhin and Pierre Villars. *The PAULING FILE Project and Materials Platform for Data Science: From Big Data Toward Materials Genome*, pages 1–26. Springer International Publishing, Cham, 2018.

[7] National Institute for Materials Science. Supercon. `https://doi.org/10.48505/nims.3837`, 2022.

[8] Masashi Ishii and Koichi Sakamoto. Structuring superconductor data with ontology: reproducing historical datasets as knowledge bases. *Science and Technology of Advanced Materials: Methods*, 3(1):2223051, 2023.

[9] B Roter and SV Dordevic. Predicting new superconductors and their critical temperatures using machine learning. *Physica C: Superconductivity and its applications*, 575:1353689, 2020.

[10] Valentin Stanev, Corey Oses, A. Kusne, Efrain Rodriguez, Johnpierre Paglione, Stefano Curtarolo, and I. Takeuchi. Machine learning modeling of superconducting critical temperature. *npj Computational Materials*, 4, 2017.

[11] Huan Tran and Tuoc N Vu. Machine-learning approach for discovery of conventional superconductors. *arXiv preprint arXiv:2211.03265*, 2022.

[12] Tomohiko Konno, Hodaka Kurokawa, Fuyuki Nabeshima, Yuki Sakishita, Ryo Ogawa, Iwao Hosako, and Atsutaka Maeda. Deep learning model for finding new superconductors. *Physical Review B*, 103(1):014509, 2021.

[13] Luca Foppiano, Pedro Baptista Castro, Pedro Ortiz Suarez, Kensei Terashima, Yoshihiko Takano, and Masashi Ishii. Automatic extraction of materials and properties from superconductors scientific literature. *Science and Technology of Advanced Materials: Methods*, 3(1):2153633, 2023.

[14] Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. The INCEpTION platform: Machine-assisted and knowledge-oriented interac-

tive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9, Santa Fe, New Mexico, 2018.

[15] Hiroki Nakayama, Takahiro Kubo, Junya Kamura, Yasufumi Taniguchi, and Xu Liang. doccano: Text annotation tool for human, 2018. Software available from https://github.com/doccano/doccano.

[16] Sheng-Fu Wang, Shu-Hang Liu, Tian-Yi Che, Yi-Fan Lu, Song-Xiao Yang, Heyan Huang, and Xian-Ling Mao. Hammer pdf: An intelligent pdf reader for scientific papers, 2022.

[17] Shyue Ping Ong, William Davidson Richards, Anubhav Jain, Geoffroy Hautier, Michael Kocher, Shreyas Cholia, Dan Gunter, Vincent L. Chevrier, Kristin A. Persson, and Gerbrand Ceder. Python materials genomics pymatgen : A robust open-source python library for materials analysis. *Computational Materials Science*, 68:314–319, 2 2013.

[18] Olga Kononova, Haoyan Huo, Tanjin He, Ziqin Rong, Tiago Botari, Wenhao Sun, Vahe Tshitoyan, and Gerbrand Ceder. Text-mined dataset of inorganic materials synthesis recipes. *Scientific Data*, 6(1):203, October 2019.

[19] Maxim Tkachenko, Mikhail Malyuk, Andrey Holmanyuk, and Nikolai Liubimov. Label Studio: Data labeling software, 2020-2022. Open source software available from https://github.com/heartexlabs/label-studio.

[20] Luca Foppiano, Sae Dieb, Akira Suzuki, Pedro Baptista de Castro, Suguru Iwasaki, Azusa Uzuki, Miren Garbine Esparza Echevarria, Yan Meng, Kensei Terashima, Laurent Romary, Yoshihiko Takano, and Masashi Ishii. Supermat: construction of a linked annotated dataset from superconductors-related publications. *Science and Technology of Advanced Materials: Methods*, 1(1):34–44, 2021.

[21] Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China, November 2019. Association for Computational Linguistics.

[22] DeLFT contributors. Delft. `https://github.com/kermitt2/delft`, 2018–2023.

[23] James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *CoRR*, abs/1612.00796, 2016.

[24] GROBID Contributors. Grobid. `https://github.com/kermitt2/grobid`, 2008 — 2023.