Automated Extraction and Curation of Materials Information from Scientific Literature

Luca Foppiano

Automated Extraction and Curation of Materials Information from Scientific Literature

March 2024

Luca Foppiano

Automated Extraction and Curation of Materials Information from Scientific Literature

Graduate School of Science and Technology
Degree Programs in Systems and Information Engineering
University of Tsukuba

March 2024

Luca Foppiano

Abstract

The scientific literature, which contains vast human knowledge, is rapidly expanding, posing challenges in organising and retrieving information. The use of big data techniques aids in uncovering patterns and making predictions and has long been applied in chemistry and biology. While materials science has fallen behind, a new discipline called Materials Informatics (MI) has emerged in recent years, thanks to large collaborative projects such as the Material Project. MI is a discipline that leverages computational power to accelerate research in Materials Science by employing techniques like Density Functional Theory (DFT) computations and Machine Learning (ML). Despite advances, the limited availability of experimental datasets, such as SuperCon or the Pauling File, hinders progress. SuperCon is a database for superconductor materials constructed manually by Japan's National Institute for Materials Science (NIMS). NIMS faces challenges updating SuperCon manually due to high publication rates and scarcity of skilled labour force. Automated processes are needed to extract data from new publications promptly, and manual curation, which is prone to errors, requires careful tools and feature selection.

In this dissertation, we propose an end-to-end pipeline for extracting material information from the scientific literature to improve the efficiency and quality of material databases. Our work aims to combine automated tasks and efficient tooling to reduce the dependency on human intelligence to the minimum. The automatic extraction pipeline processes scientific documents in PDF format and combines ML-based models for recognising complex material-related expressions. Material expressions are further processed by a specialised "Material parser" that decomposes granular information such as name, formulas, doping, shape, etc. The extraction of properties and conditions (e.g., 3K, 24 GPa, 12 atm) is performed by exploiting a general parser for extracting measurements of physical quantities: Grobid-quantities. Grobid-quantities support the identification and normalisation of measurements to the International System (SI) base units and allow the pipeline to support properties and conditions from various domains flexibly. Furthermore, to obtain the necessary high-quality training data for our ML-based processes, we developed SuperMat, a dataset of 164 superconductor articles for evaluation and training. Superconductor researchers constructed and validated the dataset and provided a structure containing annotated entities and relations. We demonstrate the efficacy of our pipeline by processing a large set of scientific articles from the Arixv repository and collecting a database with over 40000 extracted material-properties records. Finally, we proposed a curation workflow to validate the extracted data by combining an enhanced PDF viewer and a comprehensive interface. In general, using such a tool improved the quality of the extracted data with an increased precision by 6% and recall by 47%

compared to the traditional manual approach of reading the plain PDF document and writing the data in an Excel file. The models, dataset, and interface developed in this work will help increase the automation and accuracy of the processing of materials databases such as SuperCon from the scientific literature.

Acknowledgements

I express my deepest gratitude to all those who have contributed directly or indirectly to the completion of this dissertation, as their support and encouragement have been invaluable throughout this academic journey. First and foremost, I extend my sincere appreciation to my advisor, Professor Toshiyuki Amagasa, for his unwavering guidance, patience, and expertise. Their insightful feedback and constructive criticism have played a pivotal role in shaping the content and direction of this research. I am grateful to the members of my dissertation committee, Professor Hiroyuki Kitagawa, Professor Takehito Utsuro, Associate Professor Takashi Inui, and Dr. Masashi Ishii, for their time, expertise and valuable insights. Their diverse perspectives enriched the quality of my work and provided me with a broader understanding of the subject matter.

I want to express my heartfelt gratitude to Dr. Masashi Ishii for his invaluable support and guidance during my time at NIMS (National Institute for Materials Science). This research has greatly benefited from fruitful discussions and collaboration with Sae Dieb and Guillaume Lambard. I owe my ability to steer my career towards academic research in computer science and machine learning to the invaluable guidance, assistance, and support of Dr. Patrice Lopez. For over a decade, Dr. Lopez has been a constant source of inspiration, providing me with the necessary tools and motivation to pursue my academic endeavours. I express my gratitude to Dr. Laurent Romary for his assistance during my time at the National Institute for Research in Digital Science and Technology (INRIA), in Paris. Additionally, I thank Dr. Mikiko Tanifuji for her support during my remarkable journey to Japan.

I have been fortunate to have many individuals who have positively impacted my career, life, and accomplishments. Unfortunately, I am unable to mention each of them individually due to space constraints.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Materials science is a multidisciplinary field at the intersection of physics, chemistry, and engineering, dedicated to understanding, designing, and manipulating materials for many real applications. Central to this discipline is the exploration of the structure, properties, and performance of materials, ranging from metals and ceramics to polymers and composites. By delving into the microscopic and atomic scales, materials scientists seek to uncover the fundamental principles that govern a material's behaviour and tailor its properties to meet specific technological needs. This field plays a critical role in advancing technology and innovation, as discoveries in materials science often lead to the development of new materials with enhanced functionalities, improved performance, and novel applications across industries.

Historically, breakthroughs in materials science often resulted from chance discoveries and unexpected observations, a process driven by serendipity and relying on trial and error. However, with the advent of advanced computational tools, machine learning, and big data analytics, materials informatics (MI) has emerged as a transformative methodology. MI is a multidisciplinary field that leverages computational methods, data science, and informatics techniques to accelerate the discovery and design of new materials, identify patterns, and predict material properties with unprecedented accuracy. MI expedites the discovery of novel materials and enhances our understanding of complex relationships between structure, composition, and performance. This approach allows for more efficient and systematic exploration of the vast material space, potentially leading to breakthroughs in electronics, energy, healthcare, and beyond.

In comparison of MI with well-established fields like computational chemistry and biology, we can discern the unique contributions of each discipline and the

collective progression towards harnessing computational power for transformative advancements in science and technology. Computational chemistry is the earliest among the three, with its roots tracing back to the mid-20th century. The development of computational chemistry gained momentum with the advent of digital computers, and it has evolved significantly over the decades. Computational chemistry involves the application of theoretical methods and simulations to study the structure, properties, and behaviour of molecules, making it a well-established and mature field. Computational biology, on the other hand, gained prominence later, particularly with the explosion of biological data in the post-genomic era. The field began to flourish in the late twentieth century and has since grown rapidly. Computational biology encompasses a broad range of techniques, including bioinformatics, molecular dynamics simulations, and systems biology, to model and analyse biological processes at various levels of complexity. Materials informatics is a more recent entrant compared to computational chemistry and biology. Although the idea of using informatics approaches for material discovery has been around for some time, the field has gained significant traction over the past few decades, especially with the rise of advanced computing capabilities and the availability of large material databases.

Materials Informatics (MI) is founded on two key techniques: Density Functional Theory (DFT) computations and a data-driven approach using Machine Learning (ML). DFT computations, a core aspect of quantum mechanics simulations, delve into a material's electronic structure and properties at the atomic and molecular levels, providing a precise understanding that complements experimental data. Conversely, the data-driven approach utilises ML algorithms, from regression models to neural networks, to analyse extensive material datasets. These algorithms decode patterns and correlations, enabling researchers to predict the properties of new materials or optimise existing ones. This combined approach, which integrates quantum-level insights with data-driven predictive modelling, accelerates the materials discovery process, facilitating the efficient exploration of diverse materials spaces.

Data-driven methods have significantly contributed to the design of new materials across various sub-domains, such as magneto-caloric, thermoelectrics, and superconductor materials. Despite the impactful role of these methods, a notable challenge lies in the limited availability of experimental datasets. Currently, the primary sources of such datasets in inorganic materials are Pauling File [5], Starrydata [6], and SuperCon [7]. The use of these databases highlights the need to broaden and diversify experimental data, improving the effectiveness and applicability of data-driven approaches in materials design. The augmentation of these datasets is crucial to advancing the field, promoting innovation, and expanding the scope of materials discovery through data-driven methodologies. SuperCon is the

standard superconductor materials database developed and manually maintained by Japan's National Institute for Materials Science (NIMS) since 1987. It consists of records of materials reported in scientific experiments and a large set of measured properties.

A superconductor is a material that exhibits zero electrical resistance and expulses magnetic fields from its interior (Meissner effect) when cooled below a critical temperature. This phenomenon, known as superconductivity, occurs in various materials, with many requiring extremely low temperatures, often close to absolute zero, to manifest this unique behaviour. The critical temperature at which superconductivity emerges varies depending on the material. Superconductors have numerous applications, notably in the field of electrical engineering. They are used to create powerful electromagnets for applications such as magnetic resonance imaging (MRI) machines, particle accelerators, and magnetic levitation systems. Superconductors also play a crucial role in developing high-speed, energy-efficient power transmission lines since they can carry large currents without any loss. The potential for quantum computing and more efficient electronic devices is another area where superconductors are actively being explored, showcasing the interdisciplinary nature of their applications in physics, materials science, and technology.

SuperCon hosts about 32,000 superconductor material records and a complex schema with about 200 properties. The data available in SuperCon is both sparse and heterogeneous. Some records include properties that are not directly relevant to superconductivity, while others lack important properties that have not been consistently gathered [8]. For example, the pressure applied to obtain superconductivity was reported in only 16 records; however, its studies and experiments date back to the 1970s, but it became more popular only at the beginning of the 21st century. Nevertheless, SuperCon has been widely recognised for its quality and has been the data source for many attempts to design models that can predict T_c [9–11].

Using ML to predict T_c has some criticality that needs to be considered. The models have an intrinsic applicability domain, meaning predictions are limited to the patterns and trends encountered in the training set. This can lead to significant selection bias, making the models ineffective when applied to materials constituted by different compounds (or belonging to different classes). The variety of training data to be used should contain a relevant amount of non-superconductors materials to avoid training a model biased toward the assumption that a T_c always exists and renders it ineffective when applied to new materials. The definition of non-superconductivity is tricky: if no superconducting critical temperature has been found, it does not mean one does not exist. It might be outside the range of current technological instruments. This leads to a conundrum when delving into the data: ignoring compounds with no reported T_c and maybe disregarding a potentially im-

portant part of the dataset for the sake of simplicity. Simplifying too much could lead to an incomplete understanding of the factors that determine superconductivity and limit the ability to predict T_c accurately. Generally, in addition to these considerations, several SuperCon-related works attempted to integrate complementary data from other datasets. However, at the time of writing, only [8] has aimed to enhance SuperCon. They successfully enhanced 9150 records with crystal structures from experimental results stored in the Inorganic Crystal Structure Database (ICSD). This recent work indicates that the SuperCon dataset continues to play a pivotal role in the landscape of superconductor research.

1.1 Motivation

The number of publications in superconductor research remained stable in the last 10 years[1]. It is reasonable to expect that future breakthroughs in superconductor research will be achieved through materials informatics, given its growing importance, and for that to happen, a necessary condition will be to have an updated SuperCon.

Two main challenges that SuperCon is facing motivate our work. The manual approach employed for compiling SuperCon is proving to be increasingly demanding in keeping up with the influx of new publications on time. The reliance on manual labour for dataset curation becomes a growing obstacle, as it demands specialised skills that are not readily available in the market. As the field of superconductor research evolves, the need for more efficient and scalable methods becomes apparent. An automatic process is necessary to streamline and improve experimental data collection in SuperCon. The outline of the process comprises the ingestion of complete documents or text and the identification of materials and relative properties. In a small-scale preliminary assessment, we analysed the problem, identified several challenges to overcome, and proposed a data extraction framework [12].

The expressions that identify materials are complex and lengthy, and their identification in the scientific literature poses several challenges. Furthermore, the strict definition of "material" cannot be expressed in a synthetic sentence valid for the whole discipline. There are differences between subdomains in materials science that make this process highly dependent on domain experts. A material may be expressed in a multitude of partially overlapping definitions: as a single material, a single sample, a family, or a class. Alternative approaches to naming exist, including the use of commercial names or sample designations, often arbitrarily chosen

[1] analysed on arXiv statistics on cond-mat category https://info.arxiv.org/about/reports/submission_category_by_year.html

by researchers (Sample 1, Paris-sample, etc.). Chemical formulas can be presented as stoichiometric expressions within these three categories, with possibilities such as “Y-111” or “Ba(Fe1-xCox)2As2”. Examples of standard names include Oxygen or Magnesium Diboride, while some adopt abbreviated forms like Yttrium Barium Copper Oxide (YBCO). These varied strategies provide a range of options for nomenclature within the domains of chemistry and materials science, while the conventional method primarily involves the use of chemical formulas. A class of materials refers to a broader category that shares certain fundamental characteristics or properties. Materials in the same class might be characterised by common structural features, electronic configurations, or other common characteristics that make them suitable for investigation. A family usually implies a more specific grouping that shares a closer genetic or compositional relationship. A set of materials with similar chemical compositions, crystal structures, or electronic configurations. A sample is just an arbitrary name chosen by the researchers that can overlap any of the previous definitions.

To make matters more complicated, all these definitions are not strictly enforced and may fluctuate from one laboratory to another. Authors may talk about ”high-T_c cuprates”, referring to materials in the class of cuprates that also have high-T_c. This fallacious definition does not clearly describe which materials are included: *”How high should T_c be to be considered 'high'?”*. The various interpretations that words, terms, and symbols can have in different sub-domains can cause significant confusion. For instance, the abbreviation ”TC” or ”T_c” can refer to either ”Temperature Curie” or ”superconducting critical temperature”, depending on the specific context. However, authors may utilise ”T_C” (C uppercase) to indicate the critical temperature of superconductors. When extracting text, this convention is essentially the same as referring to ”Temperature Curie.” Resolving this ambiguity is only possible through careful examination of the context.

Manual curation remains a necessary component, but it does not guarantee the complete elimination of errors. [8] identified more than 10000 duplicate records within the SuperCon dataset, revealing instances where certain properties were inadequately populated. This underscores the inherent challenges and limitations associated with relying solely on manual efforts for data curation, emphasising the imperative to adopt more robust and automated strategies to enhance the accuracy and comprehensiveness of the SuperCon dataset. There is a need for a curation-centric user interface and tools that allow the reduction of the manual bottleneck to a few minor actions, leaving the tedious tasks of identifying main data in scientific publications to automatic processes.

1.2 Problem definition

NIMS encounters various difficulties updating SuperCon to incorporate the latest research findings due to the growing number of publications every year and the scarcity of a highly qualified workforce. Employing high-skilled and educated individuals who have to work on a task that offers minimal rewards requires a large monetary commitment. To expedite the extraction of information from scientific papers, it is necessary to develop an automated process. Collaborating closely with domain experts in the field is essential to address and overcome the potential pitfalls and challenges that may arise, as they can offer validation and valuable guidance. Despite the potential challenges arising from different work methodologies, this collaboration allows sharing of best practices between people with different backgrounds, leading to a comprehensive and unified solution.

1.3 Contributions

In the following section, we discuss the various contributions of this dissertation, highlighting their main relevancy. We created a novel flow compared to other works that support PDF documents as input data sources. This method enables us to establish a unified interface for all domains and publishers and the capability to retrieve the full-text body (Section 1.3.1). Moreover, the scientific community widely adopts the PDF format as the "de facto" standard for publishing research papers [13]. We rely on ML for its resistance to noise and ability to generalise well on unseen examples. Nevertheless, due to the scarcity of high-quality training data (Section 1.3.2), we constructed SuperMat (Section 1.3.2). SuperMat is a novel dataset of 164 articles containing annotations and relations between entities. SuperMat is a valuable addition to the field of MI as it helps address the limited availability of structured data sources. In the realm of materials science, the information extracted is modelled as a triplet that contains material, properties, and conditions. Both properties and conditions can be expressed as measurements of physical quantities that are relatively uniform in most disciplines. However, while the conditions are constant between domains, the properties are highly domain-dependent. As part of our contributions, we co-developed with Patrice Lopez a separate module related to Grobid based on ML to extract measurements of physical quantities (Section 1.3.3). We then combined all of these techniques to perform a large-scale extraction from documents obtained by the ArXiv repository. The automatically extracted database, SuperCon^2 (Section 1.3.4), is the main contribution of our work, presenting a novel set of properties: "applied pressure" and "measurement methods". Domain experts consider these properties promising complementary information to breakthroughs

in discovering new superconductor materials. The database obtained addresses a critical requirement of NIMS to streamline the extraction of experimental data from scientific documents. Finally, to improve the quality and reduce the impact of manual curation, we developed a user interface that supports domain experts in data curation with a set of advanced functionalities, which we demonstrate substantially improve the quality of the curation output.

1.3.1 Extraction from the full-text body of PDF documents

Unlike other works [14–16] that were predominantly based on web scraping or through the publisher API, we designed a novel data flow for the extraction of material-related information from PDF documents using an existing open-source library, Grobid (Generation of bibliographic data) [17]. While web scraping is rarely allowed (it breaks the usage agreement) or is limited to public metadata information (e.g., abstract, authors, title), API access is often hindered by the necessity of agreements with publishers, making the gathered content difficult to share and challenging to replicate. In both cases, each publisher carries its own limitations and data formats and must be treated separately: mining data directly from PDF documents enables us to emancipate ourselves from the data source type. The increasing abundance of open access literature [18] provides means to access large-scale scientific literature (e.g., ArXiv (www.arxiv.org), ChemXiv (https://chemrxiv.org/), NIMS Material Data Repository (MDR) [19] (https://mdr.nims.go.jp)). We can then access all information from the scientific documents, including full-text body, tables, and figures. Our work focuses only on text, leaving figures and tables to separate projects. Accessing the body differs us from other approaches [20], which are limited to short and synthetic abstracts. The body contains more information that includes experimental results in related works. However, it is acknowledged that abstracts, being synthetic and possessing a simpler structure, also play a role in data extraction. This contribution is discussed in detail in Chapter 3.

1.3.2 ML-based extraction of materials-related expressions from text

The scope of this work is to create databases of experimental data that are reported in the scientific literature. The structure representing this information can be complex and strongly depends on each subdomain's characteristics and conventions. However, we can define the basic data model for experimental data as consisting of three main constructs: material expressions, properties, and conditions. Material expressions are a relatively loose concept that strongly depends on the domain.

On the other hand, properties and conditions can be decomposed into expressions of measurements of physical units. This contribution comprises three main components that are discussed in Sections 1.3.2, 1.3.3, and 1.3.2.

Identification of complex material sequences

The field of materials science encompasses various disciplines, including chemistry, physics, and engineering. Material expressions within this field are typically lengthy and intricate sequences of characters. Materials can be represented by chemical formulas, which may include additional information such as the material's shape (e.g., crystal, single crystal, wire) and doping details (e.g., 2% Zn-doped). The identification and extraction of these entities is extremely challenging. We use machine learning (ML) techniques to implement the named entity recognition (NER) system. This approach offers several advantages, such as context awareness, robustness to noise, and better generalisation to unseen examples compared to rule-based methods. Additionally, we discuss the evaluation of three different architectures: Conditional Random Fields (CRF) [21], Recurrent Neural Networks (RNN) based on Bidirectional chains of LSTM units [22], and Bidirectional Encoder Representation for Transformers (BERT) [23] models.

The novelty of our approach can be summarised in four main components. First, the need for high-quality training data and the lack of resources in material informatics was raised to construct a new dataset called SuperMat (Section 1.3.2) that contains annotated documents from superconductor research. This contribution is discussed in detail in Chapter 5. Second, we wanted to reduce the risk of entities with low probability being overlooked. Our ML training strategy was designed to build models that are geared toward recall rather than precision (Section 4.2). Third, we train and evaluate three different architectures and select the best-performing combination for each model. This flexibility is convenient since each architecture has its strengths and weaknesses. Finally, we needed to extract a large set of entity types; therefore, we devised a two-layer approach where the second layer was specialised to segment only the material entities (Section 1.3.2). The original text is first processed, and the main entities are extracted: materials and classes, properties (T_c), and parametric conditions (applied pressure, measurement methods). Then, the material entities are further processed by a specialised material parser, where different modifiers such as doping, shape, substrate, chemical formula, and name are identified. Additionally, we incorporate the use of characteristics consisting of text patterns (such as uppercase and lowercase) and layout details (such as superscript, subscript, bold, and italic) that are exclusively extracted from the PDF structure.

Construction of SuperMat: an annotated and linked dataset of superconductors research papers

One major limitation of the available MI resources is the scarcity of datasets that combine materials, conditions, properties, and their relations. At present, there are only a limited number of datasets that provide a high-quality unified approach for machine learning models, taking into account the complex and inconsistent terminology common in superconductor research. To address this limitation, we constructed SuperMat, which also serves as the basis for defining a methodology which involves iterative collection and correction of data as a collaborative effort with domain experts. This work contributes to this effort by expanding the dataset for MI. SuperMat comprises, at the time of writing, 164 articles from superconductor research annotated with 2-layers annotations: six types of top-level entities related by three types of relations between entities, and a secondary layer where identified materials are further decomposed in granular element (name, formula, doping, etc.). This dataset has already been used for practical applications such as ML training [24], Large Language Models (LLM) evaluation [25], and weighted clustering [26]. Chapter 5 is centred on the methodology for creating the dataset due to its extensive and intricate nature, which necessitated a detailed and comprehensive explanation.

Parsing and normalising materials sequences

The need to standardise material expressions arises because materials can contain a combination of different types of information, such as chemical formulas, doping, substrate, etc. For example, the expression "La x Fe 1-x O 7 (x=0, 0.1)" represents a formula with substitutions, where x can take values of 0 or 0.1. Other examples are the expressions "La-doped Fe O7" and "2% La-doped Fe O7", which combine the material with the presence or the amount of doping. It should be noted that sometimes only the doping ratio is given, as in the case of "x = 0.1". Related tools such as text2chem [16] and PyMatGen [27] exist; however, they have a low tolerance to noise that limits them to clean formulas. For example, these tools would not correctly parse a non-stoichiometric formula such as "La x Fe 1-x O 7 (x=0, 0.1)". We built a comprehensive "Material parser" that uses an ML model to identify specific information such as formula, name, and doping. We combine the sequence with text2chem and PyMatGen, as mentioned above. As a result, our parser is tolerant of noisy input and can parse and convert intricate and long material expressions to a clean, structured form. This contribution is discussed in Chapter 4.

1.3.3 Extraction of properties and conditions as measurements of physical quantities

Properties and conditions within the realm of science and engineering are often expressed through measurements of physical quantities. These measurements serve as a universal language, providing a standardised way of conveying information about various aspects of the physical world. Whether it is the length of an object, the temperature of a substance, or the intensity of a force, these measurements encapsulate essential characteristics that form the basis for scientific understanding and technological advancements. Measurement of physical quantities forms the backbone of scientific inquiry, fostering a collective language that transcends boundaries and enabling collaborative efforts in the pursuit of knowledge and innovation. Employing a different array of tools for material-related expression identification and property/conditions analysis enhances adaptability towards new requirements or different sub-domains. Although measurements may seem like consolidated information, their handling can be difficult. This is because measurement units are often used in different ways across various disciplines. Additionally, different systems of measurement units, such as using miles instead of meters or ATM instead of Pascal, may be employed for human comprehension but can cause confusion for machines. A well-known example highlighting this issue is the explosion of the Mars Climate Orbiter, which occurred due to unit conversion errors. In this contribution, we present Grobid-quantities, a Grobid module specialised in the identification, extraction, and standardisation of physical quantities and measurement. At the time of development, Grobid-quantities was the first ML-based open-source project aiming to cover multiple disciplines (from biology to materials science) and systems of units (SI base, SI extended, imperial, etc.). This contribution is discussed in detail in Chapter 6.

1.3.4 Large scale collection of experimental data from scientific literature: SuperCon^2 Database

The contributions discussed above are combined in a comprehensive extraction pipeline called Grobid-superconductors. We demonstrated a significant advance in data extraction efficiency and scalability by creating the SuperCon^2 Database, collecting 40324 records in a few days, indicating a substantial increase in the amount of data processed. The original SuperCon database, cultivated for two decades, contained approximately 33000 elements. SuperCon^2 Database serves as an important intermediate step in collecting experimental data related to superconductors, bridging the gap between the original SuperCon database and the new data. The tool's effectiveness is evident not only in its ability to match the scope of the orig-

inal database but also in its ability to surpass it by capturing detailed information such as applied pressure and measurement methods. Once curated and validated, the data within SuperCon^2 can seamlessly enhance the SuperCon database, leading to continuous improvements in the accessibility and depth of the stored scientific information. This contribution is detailed in Chapter 7.

1.3.5 Reducing the impact of manual curation

When using automatically collected data, training machine learning models or other data-driven processes must be done carefully. Automated processes pose significant risks due to the inherent potential for inaccuracies and errors. We have designed a robust solution to address this need that features a staging area ("SuperCon^2 Database") that is accessed through a curation workflow with a user-friendly interface. The interface offers functions to control the content of the database records, triggering the state transition to the underlying workflow. Domain experts required that the underlying data be organised so that any update is incrementally stored and the history of changes in each record is accessible. Our system provides an enhanced visualisation of the original PDF document, allowing users to inspect and cross-reference the data efficiently. Furthermore, our solution incorporates ML capabilities to improve the automated data collection process continuously. The system accumulates training data based on corrections made during the validation process, assuring relevant refinement of the underlying ML models. In this work, we experimented and demonstrated that our workflow and interface can increase the recall of curated information from 45% to 92% and the F1 score from 52% to 92%. The experiments and the description of the curation interface are listed in Chapter 8.

Chapter 2

Related work

This chapter is divided into different contributions and aspects of this research. The chapter is organised from specific (text mining in materials science) to generic (NER, machine learning) works.

2.1 Text and data mining in materials science

[14] propose a method to extract Curie and Néel temperatures from scientific papers. The authors initially considered using the original ChemDataExtractor for this task. Nevertheless, they observed that the rule-based approach reduced precision and recall due to the complexity of the scientific text. As a solution, they enhanced ChemDataExtractor by incorporating a semi-supervised relationship extraction algorithm. This algorithm learns a typical pattern using a modified version of the snowball library. Typical patterns are stored and used to cluster new extracted patterns by distance similarity. The authors claim they extracted 39822 records, consisting of 11340 Néel and 28482 Curie temperature records.

[15] propose another work where they process 74000 web-scraped scientific journal articles and build a database of 20,400 magnetic and superconducting phase transition temperature records and their associated chemical compound names. The NLP pipeline operates as follows: text and tables from target journals were analysed to identify material formulas and temperatures. Subsequently, the identified entities were standardised, temperatures converted to Kelvin, and specific rules resolved doping issues.

SC-CoMIcs [20] describes a new corpus called SC-CoMIcs (SuperConductivity

Corpus for Materials Informatics) tailored for the text mining of superconducting materials. The corpus consists of 1000 manually annotated abstracts related to superconductivity with seven named entity categories—characterisation, process, property, material, element, doping, and value. The IAA (Inter Annotator Agreement) was around 75-85%, similar to other materials science corpora. Experiments using SciBERT for recognising named entities achieved a 77% F1 score comparable to human agreement. The learning curves indicate that the corpus size is sufficient, although some categories could benefit from more data. The corpus was used to build a word search tool based on word vectors, demonstrating its utility for retrieving relevant terms by category. This paper shows the potential of text mining to extract critical information from the superconductivity literature. The corpus provides a valuable resource for developing more capable natural language processing systems for material informatics.

After our work, [28] described a similar pipeline to extract superconductivity information from abstracts from the scientific literature. It uses the SC-CoMIcs [20] corpus of 1000 annotated abstracts on superconductivity to train NER and RE models. The author reports that they extracted material compositions, transition temperatures, doping information, and process details from 48,565 abstracts obtained by querying the Elsevier Scopus API (Application Programming Interface) for a total of more than 43,000 superconducting materials and 24,000 transition temperatures. They follow a similar approach to our work; first, they extract entities and find the relations. However, the abstract limits their extraction density, which has only synthetic information. Considering their search query *"supercond * AND tc OR transition temp"*, they obtain around one superconductor material per abstract.

There are several other works on TDM on experimental information from materials science; the most relevant are discussed in the following paragraphs.

[16] describes a text mining system to extract and segment synthesis recipes from scientific articles. The articles are obtained through web scraping, and the synthesis paragraphs are identified through classification. Subsequently, NER is implemented using RNN (specifically, BidLSTM + CRF) in the identified paragraph. This helps to identify the various components of the recipe, and the precursor, operation, and synthesis conditions are extracted for a total of 19,488 solid-state synthesis reactions. This work was limited to articles after 2000 due to the lack of available HTML/HTM versions for older articles.

[29] proposes a text mining pipeline to extract metal-organic frameworks from scientific papers. In this paper, the authors focus on two main properties, surface area (SA) and pore volume (PV), mainly because they are the main properties related to the absorption properties of MOPs, are very commonly described in papers,

and have a very distinctive signature such as the units used to measure the quantity. The source data was obtained in HTML format, and after parsing, cleaning, and tokenising the text, the tokens were classified into five types using lexicons from the Cambridge structural databases. Moreover, they employ specific terms to recognise appropriate units used unconventionally. Using a rule-based algorithm, the resulting classified tokens were combined (the material was connected to the corresponding attributes). On a sample dataset, the algorithm achieves 90% and 88.8% precision in extracting surface area and pore volume data, respectively. When tested on a larger dataset of 2315 MOF papers, the precision drops to 73.2% for surface area and 85.1% for pore volume. Errors occur due to complex sentence structures and incorrect identification of MOF names.

[30] introduces a nanoparticle-related dataset of 5,154 records extracted from 4.9 million materials science papers containing synthesis recipes and morphological information. The authors successfully extracted 7,608 experimental and 12,519 characterisation paragraphs with compounds, amounts, synthesis actions, sizes, shapes, etc. Limitations include the inability to distinguish targets from other morphologies and the lack of order information for seed-mediated syntheses.

[31] develop an automated approach to extract band-gap values and relate them to chemical compounds in the titles and abstracts of articles. The authors extended ChemDataExtractor with a new BandGapParser to identify band-gap information. On a sample of 415 papers, the system achieved 51.32% correct, 36.62% partially, and 12.04% incorrect extractions. Errors are due to incorrect extraction of chemical entities, band-gap values, and referring entities to values. The approach was applied to 11,939 articles, extracting 10,608 band gap values for 10,292 compounds. The limitation of this study is the evaluation that was performed by reviewing the extracted information rather than comparing it with the ground truth.

[32] presents ChemNLP, a natural language processing (NLP) library and web application for analysing material chemistry text data. It uses the publicly available arXiv dataset of 1.8 million scholarly articles collected over 34 years. The authors analyse publication trends, author names, taxonomy categories, and word frequencies in titles and abstracts. An interactive web application was built to search for articles containing specific chemical elements and compounds. As a demonstration, ChemNLP was applied to identify new superconducting materials by comparing the arXiv data set with a DFT-based superconductor database. Machine learning techniques such as TF-IDF vectors, t-SNE clustering, and classification algorithms were used to categorise "cond-mat" arXiv articles with 80% accuracy. ChemNLP provides an open dataset and tools to apply NLP techniques to the materials science literature for knowledge discovery.

2.2 Machine Learning

Machine learning (ML) can extract valuable implicit knowledge from data [33], providing fresh insights into existing problems. The application of machine learning (ML) in intricate fields like chemistry requires the use of reliable and accurate data. However, it is essential to note that the results obtained from ML models may not always be attributed solely to the domain knowledge incorporated, as there could be other concealed factors within the data itself [33].

Conditional Random Fields (CRF) [21] have been widely used in various ML tasks such as sequence labelling and NER [34]. CRFs have been particularly effective in situations where multiple annotator-embedded label sequences are available, but there is no actual ground truth, as they provide a probabilistic approach to sequence labelling [34]. Furthermore, CRFs have been compared with other machine learning models [35] such as Support Vector Machines (SVMs) [36] and Structured Support Vector Machines (SSVMs) in tasks such as chemical entity recognition, demonstrating their versatility and effectiveness [35, 37, 38].

The transition from CRFs to RNNs represents a shift from probabilistic graphical models to deep learning architectures. While CRFs effectively capture dependencies between input features and make predictions based on them, RNNs excel in capturing temporal dependencies in sequential data due to their recurrent nature. This transition is particularly relevant in tasks where the sequential nature of the data is crucial, such as speech recognition and acoustic event detection [39, 40]. The use of recurrent neural networks (RNNs) has gained importance thanks to Long Short-Term Memory (LSTM) [41] and Gated Recurrent Units (GRU) [42] for text processing tasks: classification and recognition. [43–46] highlight the effectiveness of RNNs in processing sequential data, showcasing their superiority over traditional methods in natural language processing tasks. Furthermore, the combination of RNNs, particularly bidirectional LSTM recurrent neural networks (BidLSTM), with CRFs has been explored for tasks such as word segmentation and morpheme segmentation, highlighting the adaptability of RNNs in sequence labelling [47].

Particularly relevant is the work [22], which proposes a new neural architecture for NER with an architecture based on a bidirectional set of LSTM chains. Bidirectional indicates that the text is computed from left to right (forward LSTM) and from right to left (backward LSTM). Such an architecture can incorporate information usually captured by hand-crafted features or gazetteers. Such architecture takes in input two types of word representation: character-based word representation, calculated from the supervised corpus, and unsupervised word representation, learnt from unsupervised corpora. The unsupervised word representations discussed were word2vec [48], but they can also be used for more recent ones: fastText [49],

GloVe [50].

[51] proposes a novel deep contextualised word representation that addresses two key aspects: a) capturing complex characteristics of word usage and b) incorporating the polysemy of words within linguistic contexts. The proposed approach calculates word representations based on the entire sequence. The network architecture consists of two bidirectional LSTM layers with dimensions 4096 x 512. These layers work in a complementary manner, with one predicting the next token from left to right based on the preceding tokens and the other predicting from right to left. The study also highlights several significant findings: a) instead of solely outputting the last layer, averaging the weights from all layers with a coefficient improves performance in downstream tasks, b) lower layers capture syntactic information while higher layers capture context-dependent aspects (semantic), and c) incorporating ELMo embeddings in both the input and output enhance tasks that utilise the attention layer after the RNN, but does not provide benefits for other task types.

The transition from LSTM-based RNN to transformers marks a significant change in natural language processing. The introduction of Bidirectional Encoder Representations from Transformers (BERT) [23] revolutionised language representation models by leveraging the power of transformers. Transformers, as proposed by [23], replaced RNNs with self-attention, demonstrating the potential of this approach in language processing tasks [52]. BERT, based on bidirectional transformers, offers a simple yet robust architecture that allows efficient pre-training and fine-tuning with minimal task-specific modifications, leading to state-of-the-art performance across a wide range of language processing tasks such as question answering, language inference, and named entity recognition [23, 53]. The bidirectional nature of BERT enables it to capture contextual information from both preceding and subsequent words, which is crucial to understanding the meaning of a word in a given sentence, thus outperforming traditional LSTM-based models [54, 55]. Furthermore, BERT's ability to handle long-range dependencies and its superior performance in sequence labelling tasks, such as NER and RE, has been demonstrated in various studies [56–58]. Furthermore, the effectiveness of BERT in sentiment analysis, misinformation check and text classification tasks has been highlighted, showcasing its versatility and robustness in different NLP applications [59–61]. However, it is essential to note that while BERT has shown superiority in various language processing tasks, there are instances where CNN and LSTM models also perform competitively with BERT-based models [62, 63]. In general, the advantages of BERT-based architectures lie in their conceptual simplicity, empirical power, contextual understanding, and superior performance in diverse NLP tasks, making them a crucial advance in NLP.

In the following sections, we first discuss all essential BERT-related work and

then the BERT-flavoured transformers related to materials science.

2.3 Transformers

[23] introduce BERT (Bidirectional Encoder Representation Transformer). While pre-trained language models have been discussed in previous work [64] and can be exploited in two main ways: feature-based or fine-tuning. BERT aims to improve the fine-tuning approach and to solve the limitation of existing left-to-right language models, where each token can attend only to previous tokens in the self-attention layers [52] of the Transformers architecture. BERT introduces an MLM (Masked Language Model) pre-training objective, where it randomly masks tokens in training examples and tries to predict them using the full context. This enables using both left and right contexts and pre-trains a fully Bidirectional Transformer. Using the original transformer architecture [52], they define $\text{BERT}_{\text{BASE}}$ with L=12, H=768 and A=12 (110M parameters) and $\text{BERT}_{\text{LARGE}}$ as L=24, H=1024 and A=16 (340M parameters) with (L=layers, H=hidden layers, and A=attention heads). The authors define the input as a sequence that can be a natural sentence, a paragraph, or a pair of sentences. They used a special separator [CLS] to start the sequence and convey the result of the final hidden state. For separating two sentences from a pair, they encode the separator with [SEP] and provide sentence embedding, which states whether a token belongs to sentence A or B. Each token is encoded using position embeddings calculated using WordPiece using a dictionary of 30000 tokens vocabulary. The sum of the vocabulary token, the sentence token, and the position embeddings represent the input representation for each token. The authors defined two training objectives: Masked LM (MLM) and Next-Sentence Prediction (NSP). The MLM is prepared by substituting 15% of the token's positions at random: 80% of the time with [MASK], 10% of the time with a random token, and 10% of the time do not substitute. The reason is to provide the model with enough randomness so that the model does not always learn to replace the tokens with [MASK]. The NSP aims to identify whether a sentence follows another sentence. This is justified by the need to perform tasks such as answering questions in which relations between sentences are determined. BERT outperforms the present architectures on most NLP tasks and benchmarks such as GLUE, MNLI and SQUAD.

After BERT was released, a multitude of flavoured models were released. We analyse the most important ones in the following paragraphs.

[65] reproduces the results obtained in BERT, revises the original architecture and proposes RoBERTa (Robust Optimised BERT Architecture). This revised BERT implementation outperforms BERT on downstream tasks such as GLUE,

RACE, and SQUAD benchmarks. The main modifications are summarised as follows. Instead of masking tokens statically only once during preprocessing when the data are duplicated ten times, they apply different masking at each instance of the same example. Dynamic masking helps for less than 1 point percentage. They suggest that changing how the input data is prepared and removing the NSP loss objective improves the performance. In particular, providing complete sentences from the same or different documents without NSP matches the original implementation of providing a pair of segments with NSP. Furthermore, results are even better when only sentences from the same documents are provided, and the NSP is removed. NSP removal was discussed in recent work [66]. Lastly, they propose training for a longer time, and supplying more data can improve the results: the original BERT was trained on 1M steps with a batch size of 250, while the RoBERTa was trained for fewer steps, 31000 using a much larger batch size of 8000 examples. This setting requires a more scalable approach because the required GPU memory is more significant. With these changes, RoBERTa can outperform BERT on most language understanding benchmarks such as SQUAD, GLUE, and RACE.

[67] propose a scientific version of BERT called SciBERT. SciBERT was trained on a random sample of 1.4M articles from Semantic Scholar (data set was not published): 18% from computer science and 82% from the biomedical domain. SciBERT was evaluated against BERT and outperformed BERT. Unlike BioBERT [54], SciBERT was trained using a different tokeniser (SentencePiece instead of WordPiece). Most importantly, the tokeniser was re-trained from scratch using scientific text. Another exciting aspect is that BioBERT was trained from a model initialised with the weight from BERT, trained for a longer time, and on more data overall. However, the limited coverage of the BERT tokeniser on scientific data penalised the performance on various downstream tasks.

[68] introduces a new approach in pre-training a BERT model in two flavours: LinkERT and BioLinkBERT on the general text and biomedical text, respectively. The authors introduce a novel approach which includes text from linked documents, and an additional objective function is to classify the type of link. The pre-training data is aggregated as follows: a) Each example is structured as a pair of text sequences as in the original BERT. b) The pairs are aggregated by randomly selecting the second segment (when available) from the same or a related document. For the LinkBERT, this only works for the Wikipedia corpus, exploiting the links between Wikipedia pages. For BioLinkBERT, the authors select sequences from related documents in the citation graph. The authors replace the NSP objective with the DRP (Document Relation Prediction), which aims to classify the relation type between the two segments, giving three possible classes: contiguous, random, and linked. Evaluation metrics calculated on QA (extractive question answering), GLUE and SQUAD outperform BERT. BioLinkBERT outperforms PubMedBERT [69] on most

NLP tasks.

[69] pre-trained a biomedical-based model called PubMedBERT and evaluated it using a newly assembled dataset called BLURB (Biomedical Language Understanding & Reasoning Benchmark). The vocabulary was trained using a BPE (Byte-Pair Encoding) approach with a length of 30522 and trained on text from PubMed abstracts: 14 million abstracts, 3.2b words. The fact that they only used abstracts could be a significant limitation on the type of information used in the pre-training. They are the first to consider the concepts of "in-domain" (in their case, biomedical text) and "out-of-domain" (everything else). Their strong claim is that domain-specific pre-training from scratch can be superior to mixed-domain pre-training, which contrasts with [70]. From the perspective of vocabulary, it is demonstrated by comparing scores from BioBERT [54] and SciBERT [67] that vocabulary specificity improves performance in tasks applied to scientific text.

[71] tested biomedical-trained BERT models on relation extraction in biomedical data. They demonstrate that downstream tasks on overlapping data used in pre-training bias the results for RE. In classification, they indicate that using only the information in the "CLS" token can be improved by adding complementary information from other layers.

[70] demonstrate that training with a multidisciplinary corpus gives better results than a transformer based on a domain-specific dataset. In addition, they also found that larger models do not always perform better. However, these results might be due to other issues they have introduced with their pre-training process. Previous work by [66] has suggested a similar concept applied to translation tasks, where including cross-training information and, in particular, the addition of data in different languages not only boosts the performance in translation but also helps the performance in non-translation tasks.

2.4 Materials science BERT-based pre-trained transformers

This section analyses the pre-trained BERT-based transformers using materials science text. We provide a comparison evaluation of all of these transformers as a comparative measurement.

[72] propose a material-property extraction for polymers based on a fine-tuned BERT model on 2.4 million abstracts. Their NER system achieves higher results than other material-based pre-trained BERT models. Properties are connected to

materials using a heuristic approach to find the closer entities. With their system, they extracted more than 300000 records of polymers and properties from the 2.4 million abstracts. However, the article does not provide information on data contamination between pre-training and evaluation: we do not know whether there is any overlap between the evaluation datasets, PolymerAbstract [73] and the training set of 768 abstracts. Moreover, whether their model can generalise properly is not indicated, providing out-of-domain information of their evaluation dataset compared with the training set.

[74] is another specialised set of specialised BERT models that focus on scientific articles related to batteries. In this work, the authors experiment with different approaches for fine-tuning, continuing a generic BERT training (batteryBERT), a SciBERT training (batterySciBERT), or training from scratch a new model based only on text from battery research (batteryOnlyBERT). The different vocabularies provide the main differences, where batteryBERT and batterySciBERT are constrained by the original vocabulary. The batteryOnlyBERT vocabulary was trained from scratch and could model the terminology used in the battery research articles more accurately. In document classification, batterySciBERT obtained the highest score. SQuad evaluation (extractive Q&A dataset) surprisingly showed better scores with batteryBERT. This work demonstrated the importance of having a solid base with general text and scientific information to provide a model that can generalise and adapt to multiple situations where it could be used.

[75] proposes a two-stage deep learning framework for extracting chemical formulas and applying a "Role labelling" in the chemical reaction description. Following a standard schema Extract-Link, they first extract all references to chemical compounds and, subsequently, label each compound with its role in the reaction description. They introduce a chemical-focussing pre-trained BERT flavour followed by a task-adaptive encoder (ChemRxnBERT) that provides a role labelling method. The authors created a small dataset for fine-tuning chemical reaction annotations composed of relevant paragraphs containing chemical reactions extracted from articles in various chemistry journals. Role labelling was implemented in cascade by providing the primary material between special tokens ([P], [
P]), which are then linked to the other extracted entities. Since the sequence is shallow, it can link only one material with the rest of the reaction. Therefore, running the process numerous times is needed if multiple products are present in the reaction paragraph.

2.5 Extraction of quantified properties and measurements

NER of physical measurement has a fundamental application in scientific texts and is relevant in other disciplines, including humanistic or social science.

Attempts have been made to extract measurements from text using many approaches. At the time of this contribution, we identified the following related work.

[76] built a tool for the extraction of quantities based on Apache UIMA (Unstructured Information Management Architecture) in combination with pattern matching that circles a Finite-State Automata (FSA). The authors claim that UIMA and FSA are faster and better suited to process a large quantity of text. Their work supports multiple constructs: values, intervals, and enumerations. Units are normalised to the International System of Standard (SI). Units are loaded from a configuration file that includes the normalised format. In the same work, they also combine keyword extraction and text segmentation, which are exciting challenges in the analysis of patent text. The authors used Quantalyze[1], a commercial tool designed to process patents, for comparison. They reported that Quantlyze had limited unit support.

[77] propose a search engine called MQSearch. Their work proposes a rule-based extractor for quantities and units that include the measured object. The extractor models each measurement using the five-tuple (sign, number, error, scientific notation, units), where only numbers and units are mandatory. The data flow comprises four phases: preprocessing to mitigate noisy and incorrect character extraction. Then, to recognise units, they expanded an ontology of units from the OBO Foundry with additional information from external sources and defined one associated rule for each unit. The ontology is exploited to recognise the measured object. Quantities are extracted similarly with a set of regular expressions. The last phase is post-processing, which is used to discard possible wrong or invalid values. They also present a SoLR-based search engine that allows for a search using the extracted units, values, and measured objects.

[78] built an extractor for patents using GATE (General Architecture for Text Engineering). Although GATE provides plugins for machine learning, they implemented their extraction using a lookup to a gazetteer and a database containing the transformation rules between one unit and another. The gazetteer was built from a set of units published by the GNU (Gnu's not Unix) Foundation and comprises 30000 units. In addition, they also recognise references within the text and patent information (e.g. patent number, country code). They built the annotation rules

[1] https://www.quantalyze.com/

using the JAPE language, which is structured as a matching rule and action to perform.

Although most works are based on English, some contributions focus on specific languages: [79] investigated issues applied to Russian-derived languages (Russian and Belarusian). The authors make a consistent analysis of the challenges of Russian-derived languages. They proposed a solution based on finite-state automation with 350 graphs running on top of the NooJ[2] linguistic processor. Grammar covers three of the six grammar constructs (genitive, accusative, and nominative). The evaluation was carried out on a mixed corpus of 100,000 words and resulted in an F1 score of 82%.

[80] combine the use of an Ontological and Terminological Resource (OTR) with the use of ML. The OTR is used as a reference and can be updated with new units that are not extracted correctly or entirely. They aim to use a modified string-matching function to extract measurements and units. Thus, the main problem they try to solve is to reduce the search space using string-matching functions. They proposed a two-step process. First, they use a classifier to determine whether a sentence contains a unit. The classifier is built on a bag of words where the words are counted using three methods. TF, TF/IDF and BM25. The model is then built with three different algorithms, which are then compared: Naive Bayes [81], decision three (J48), support vector machines (SVM) and Discriminative Multinominal Naive Bayes (DMNB). In the second step, they match potential candidates in the OTR and use the string function to select the item that is more likely to represent the extracted one. Based on the similarity, they defined two thresholds that they can consider as variants of an existing or new unit to enrich the OTR and improve future recognition.

[82] describe another rule-based solution in which they outperform the respective ML-based system. They aim to extract targeted information from laboratory test results from diagnostic devices examined by the US Food and Drug Administration (FDA). The authors developed a symbolic information extraction (SIE) system for extracting four types of entities: analytes (substance considered), specimens (where the analyte is measured), units of measure of the analyte, and detection limits of the diagnostic device in the exam. The SIE is based on a combination of rules and dictionaries. First, the candidates are extracted and then ranked to select the most plausible ones from the set. The units of measures targeted in this work are only a subset of all units and are specific to this particular subdomain in biology. They evaluated their SIE against three probabilistic learning approaches: CRF, SVM, and HMM. SIE outperforms ML-based models, except for the unit of measures where the CRF obtained the best scores.

[2] https://nooj.univ-fcomte.fr/

More related to materials science [83] proposes integrating relevant units of measurement when composing a small data set (392 sentences) to develop nanocrystal devices. The data set focuses on identifying several properties, including the property value and common units in the domain. x This work also leverages ML using a set of CRF engines, focussing only on a few entity types applied in a cascade.

[84] describes an integrated measurement extraction that focuses on the scientific literature in earth science. They developed a novel method that extracts context-related information from measurements. Instead of creating their measurement extraction tool, they evaluated three reuse options. Quantalyze showed low recall in both the extracted measurement and the measured object and had technical limitations in integration, such as a lack of REST API. [78], discussed previously, was discarded due to the need to maintain the rules. Therefore, they chose our tool, Grobid-quantities, which had the advantage of requiring only labelled data and providing acceptable precision, recall, and f-score results.

[85] extracts numeric laboratory test expressions from clinical trial eligibility criteria texts. They encode information using the TimeML specification language TimeML. The system presented Valx extracts: a) numeric values and units using regular expressions, b) using a hybrid approach with a knowledge base, they extract variables referred to as quantities. They cover different text representations (e.g., BMI, body mass index, etc.). Valx supports the association of multiple values to the same variable. For example, ”40 and 60 years” associate both values with ”years”. Their tool supports normalisation to convert conventional units to international units.

[86] describe an approach to formalisation of quantities and measurements that aims at extracting information from free text and inferring complex reasoning that includes the measured entities and their quantity. The authors introduce QVR (Quantity Value Representation) with three constituents: Value (which includes values and ranges), units (which characterise the value), and change (which indicates whether there is a modification in the value, e.g. increases). The quantities are extracted using a Semi-CRF and a bank of classifiers. [86] uses four types of features that are calculated for each token (and a window including the three previous and following tokens): a) classification using a lexicon to identify whether it appears as a number, unit, etc., b) determining whether it contains a digit, all digits, etc., c) POS (part of Speech). Units are extracted by assuming they are adjoined to the numeric values. This work reserves a detailed discussion on the entailment of quantities. Given a quantity value and a text, it is a 3-way decision problem: a) ”entail” when the quantity value is supported in the context, b) ”contradicts” when the context, instead, does not support the quantity-value, the Q-V is different, c) no relation. The Quantity entailment aims to clarify whether the context confirms units and if the way

they are described can be equivalent (e.g. two couples, four people). The quantity entailment is relevant for scientific text comprehension, especially for explaining experimental results concisely and containing multiple relative comparisons within the same paragraph, making it hard to understand.

The works of [87] and [88] focused on tables.

[88] describes a comprehensive query system that uses previously extracted quantities, units, and context to answer questions containing measurement conditions. Their data model is described as triples of the form (entity, quantity, context), where the context gives more information and proof to quantity and entity. The triplet is computed offline in several steps: first, the quantity is extracted based on [86], which we have described previously. Entities and quantities are referenced using the Yago knowledge graph. Different columns are related to their previous work [89], and the extracted data is used to create an index with conceptualised quantities. This idea is particularly relevant to our Grobid-quantity work because very few works currently use normalised quantities to search by values and intervals.

[87] assumes that the table columns are already identified and focus on covering unconventional unit names (e.g. LTS as Litres), considering that the columns should have the same normalised unit. The authors evaluate and compare their system (PUC) regarding currencies, data storage, length, mass, and volumes. For example, Grobid-quantities do not support currencies well because they hardly appear in scientific papers. At the moment of writing, Grobid-quantities does not support extraction from tables; however, since there is a plan to improve the table recognition in the Grobid library, this future work could be implemented within the same framework.

Chapter 3

Extraction from the full-text body of PDF documents

3.1 Introduction

PDF is a widely accepted format for sharing and presenting documents, as it ensures that the layout and formatting of the document remain consistent across different devices and operating systems. This is particularly important in scientific publications, where complex figures, tables, and equations must be accurately represented. The PDF format is designed for documents intended for printing, as they can embed fonts and high-quality images, and scientific journals traditionally required high-resolution print-ready files for publication. PDF files can be signed digitally, providing security and authenticity for scientific publications. This is important to ensure the integrity and trustworthiness of research findings, particularly in fields where reproducibility and transparency are paramount. Although isolating trends in the Internet jungle is challenging, PDF documents account for about 85% of the Common Crawl dataset[1] and interest has been growing[2]. However, it is essential to note that while PDF has been widely used in scientific publications, emerging trends and technologies challenge its status as a ”de facto” format. For example, the rise of open-access publishing has led to adopting HTML and XML formats for online articles, allowing greater interactivity, accessibility, and searchability of scientific content. However, those formats are penalised by the lack of standardised approaches, leading to a jungle of flavours of the same original format (e.g., XML-JATS format is one of the most widely used).

[1] https://commoncrawl.github.io/cc-crawl-statistics/plots/mimetypes
[2] https://trends.google.com/trends/explore?date=2000-01-01%202024-01-19&q=PDF

3.2 Proposed approach

Given the critical role of PDF documents in disseminating scientific knowledge, supporting them natively is a necessary step for building effective TDM processes. Our approach uses the Grobid library (Generation of Bibliographic Data) [17], which can parse and structure text from PDF documents, such as scientific publications or patents. Grobid implements a set of pre-trained models supporting different architectures, including Conditional Random Field (CRF) [21], Recurrent Neural Networks [22], and transformer-based [23]. Other tools for document structuring, such as Cermine [90] and ParsCit [91] have been outperformed by the Grobid performances of its out-of-the-box models [92]. Grobid has been successfully extended to support several domain-specific problems, for example, recognition of astronomical entities [93], segmentation of dictionaries [94], software mention [95]. Among the various open-source tools available, at the time of writing, it is still actively developed and used in several large-scale research repositories, such as Mendeley [96], ResearchGate (https://researchgate.com), Scite.ai (https://www.scite.ai) [97].

Grobid offers several advantages. a) It does not require one to make multiple agreements with different publishers to obtain source documents. A growing number of open-access pre-print repositories offer a viable alternative for gathering input data. b) PDF support in Grobid allows one to focus on processing a single format instead of dealing with several XML flavours. c) It is integrated with PDFAlto (https://github.com/kermitt2/pdfalto), a specialised tool for converting PDF to XML, which mitigates extraction issues such as invalid character encoding and incorrectly ordered text flow. PDFAlto supports the resolution of embedded fonts and layout variants such as multi-column. d) The Grobid internal data model *LayoutTokens* access information in PDF documents at a low level for each token: style (italic, bold, superscript, and subscript), font (font type, font size), and coordinates within the visual document. e) By parsing PDF format natively, Grobid gives access to full text, figures, and tables and thus does not limit our work to public information such as abstracts. Finally, f) Grobid includes high-quality, out-of-the-box pre-trained machine learning models.

This and the following contributions are combined in Grobid-superconductors, a novel application of the Grobid library as a text mining pipeline for scientific publications in materials science. Our approach differs from other related works in several aspects. 1) Instead of abstracts [28], we support extraction from full-text body. 2) We are not required to reach an agreement with publishers [16]. 3) We focus on processing a single format instead of dealing with several XML flavours [16] 4) We avoid web scraping [15] and other processes that require addressing each website separately and may require regularly updating the harvester to follow potential website

changes. 5) We use low-level layout details as attributes in our ML models that handle intricate representations of materials. For instance, including superscripts or subscripts (which may not always be present in XML-formatted documents provided by publishers) can be intuitively advantageous in identifying chemical formulas.

3.2.1 Publisher's agreement for source data

Obtaining scientific articles through publisher APIs (Application Programming Interface) or web scraping can be inconvenient for several reasons. Many publishers control access to their API, demanding subscriptions or pay-per-view models, limiting accessibility for those without institutional access or financial means. API limitations, such as the number of requests allowed within a specific time frame, can slow down data retrieval, especially with many articles. The lack of standardisation in metadata and content formatting further complicates the extraction process. As mentioned, each publisher provides its own "JATS-flavoured standard", making it challenging to create a standardised data set from multiple sources. However, web scraping does not guarantee data format stability and may breach the publisher's terms of service, leading to legal and ethical complications. Open access resources, including institutional repositories and pre-print servers, offer a wealth of scientific literature at a reduced cost and in a consistent PDF format that represents a convenient means to access and process articles from these sources due to their accessibility and lack of legal constraints.

3.2.2 Abstract versus full-text

The scientific paper's abstract, a concise summary located at the beginning of the document, serves as a quick overview that summarises the research's objectives, methods, results, and conclusions in 150 to 250 words. Researchers frequently employ abstracts to assess the relevance of a paper to their work and for streamlined database searches. In contrast, the full text of a scientific article delves into comprehensive details, spanning sections such as introduction, methodology, results, discussion, and conclusions. This exhaustive account is crucial for in-depth literature reviews, detailed analysis of methods, data review, and evaluation of the validity of the findings. Taking into account the perspective of TDM, both abstracts and full-texts play distinct roles. Abstracts, due to their succinct nature, are suitable for large-scale automated analyses, offering rapid insights into trends, topics, or keywords across many papers. However, full texts are amenable to more in-depth TDM, allowing for the detailed extraction of information and revealing nuanced relationships between concepts, methodologies, and findings. Although previous studies [20]

have used abstracts, our research focuses on extracting complex information from scientific documents using the entire full-text.

3.3 TDM pipeline

Our TDM pipeline is built as a Grobid module called Grobid-superconductors. Grobid-superconductors is structured as a three-step process illustrated in Figure 3.1. The input is a PDF document converted to an internal model composed of text passages, tokens, and features. Those three elements are then used to perform Named Entities Recognition (NER), which are then linked by a Relation Extraction (RE) process. The output is a structured object serialised in JSON format.

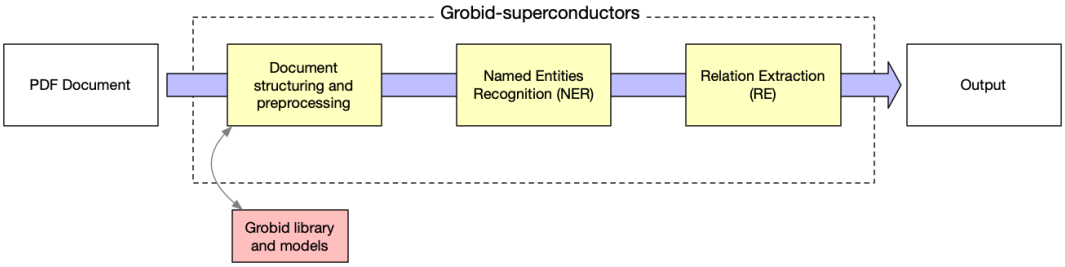


Figure 3.1: Processing pipeline for extracting superconductors materials and properties.

3.3.1 Document structuring and segmentation

The initial stage of our procedure involves using Grobid and PDFAlto to convert the PDF document. This results in segmenting raw data into a basic structure consisting of header, body, and annexes using the *Segmentation ML model*. These three structures are then parsed by a second Grobid model *Fulltext ML model* that identifies paragraphs, section heads, and references for body and annexes. We have created a custom process that analyses the structure and generates an internal model based on a list of text statements, tokens, and features. For example, the content of the tables is excluded, but the table captions are retained. In the process, we can also assign *section* and *subsection* information as the results of the Grobid first level: header, body, and annexe, and the second level (paragraph, table or figure caption, abstract, title) structures.

Header structures are parsed with the *Header ML model* to identify bibliographic data. This aggregated version only includes information that is relevant to our TDM

process. Our process selects a subset of bibliographic information from the header: title, authors, DOI, publisher, journal, and year of publication, and we consolidate them via Grobid to match the publisher's quality (even by processing the "pre-print version" of the publication).

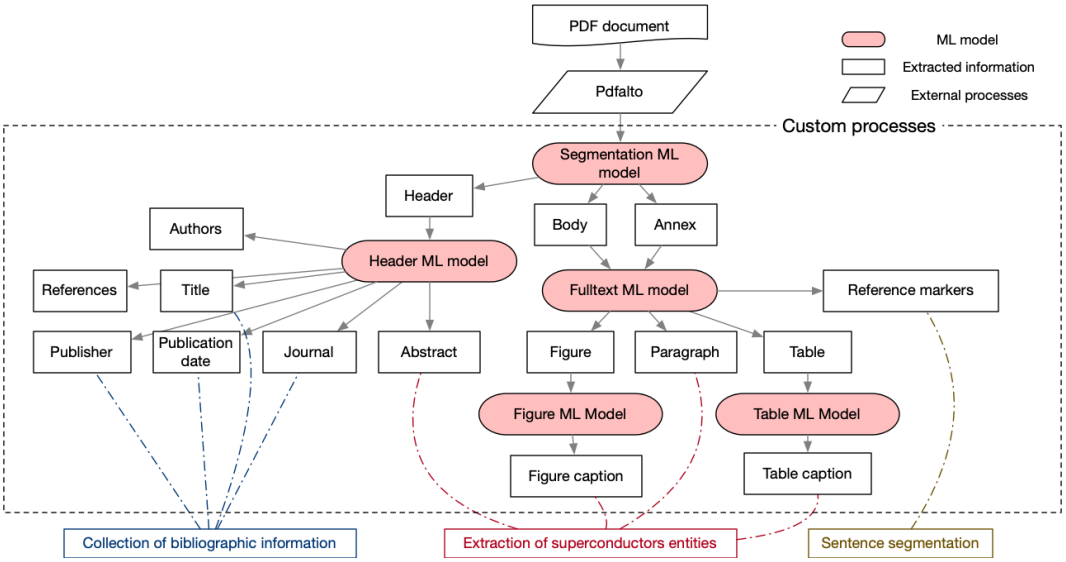


Figure 3.2: Grobid-superconductors extraction processes (bibliographic information, superconductor entity extraction and sentence segmentation) within the Grobid cascade data flow.

3.3.2 Sentence segmentation

An additional query concerning natural language processing (NLP) pertains to the choice between employing sentence-based or paragraph-based text. While paragraphs can be extracted as part of the PDF document's layout, obtaining sentences requires an additional step through a sentence segmenter. Sentences are typically shorter, which offers advantages in processing with large DL models. During training, sentences use less memory and allow us to train models with a larger "batch size", which has been shown to improve efficiency and obtain better results [65].

We used sentence-based segmentation in Grobid-superconductors after performing small-scale preliminary experiments on our task. For the Named Entities Recognition (NER) task, we trained and evaluated a sequence labelling model for each approach (paragraph and sentence based) in four annotated documents (3/1 document partition for training/evaluation) from SuperMat, a dataset that we built [4] and is described in Chapter 5. As indicated in Table 3.1, the F1 score increased by

Table 3.1: Results from cross-validation for sentence-based and paragraphs-based text. Measurements are micro average. P: Precision, R: Recall, and F1: F1-score.

Label	**P**	**R**	**F1**
Paragraph-based	44.44	27.21	33.76
Sentence-based	48.41	50.00	51.70

17.94% points when using sentence-based text.

For the RE task, we found in a previous study [12] that the sentence-based approach would favour higher precision, while the paragraph-based approach would result in greater recall.

Since Grobid supports partitioning documents into paragraphs, a sentence segmenter is necessary. The problem has been long investigated [98–104]; nonetheless, there are still challenges in segmenting paragraphs from scientific text: reference markers (also called *reference callouts*), formulas, and other constructs may contain periods and mislead algorithms. Thanks to Grobid and the ability to recognise reference markers and formulas, we use the collected ones from the text as features to improve paragraph segmentation in sentences: segmentation is aborted if the end of a sentence falls within the boundaries of one of these markers. For example, a sentence such as "*[...]The evaluation from (2021, Foppiano et al.) offers a solid validation for our method*" containing a reference in the form "(2021, Foppiano et al.)". It may be mistakenly segmented in the middle token "[...] et al.", and the new sentence starts with ") offers [...]".

3.4 Conclusions

In conclusion, the application of innovative techniques, exemplified by Grobid, marks a significant milestone in materials science. This effort represents a novelty for integrating such methods within the domain, showcasing the potential for transformative advancements. By sidestepping the complexities and expenses associated with signing agreements with scientific publishers and the intricate landscape of XML formats, this approach not only streamlines the dissemination process but also liberates researchers from the often daunting challenges posed by conventional publishing models.

In the evolving landscape of scientific dissemination, where alternative means gain traction, it is noteworthy that the PDF format remains resilient as the "de

facto" standard. The decision to publish through open access aligns with the trend toward greater accessibility and openness in scholarly communication. The increasing availability of open-access articles further reinforces the utility and relevance of Grobid, positioning it favourably as a valuable tool in the arsenal of researchers seeking efficient and cost-effective methods for sharing scientific knowledge in the materials science domain.

In this contribution, we introduced Grobid-superconductors, and we described the component responsible for reading and structuring PDF documents, preparing the data to be processed by our novel specialised models, which are discussed in Chapters 4 and 6.

This contribution was published in the paper "Automatic extraction of materials and properties from scientific literature" [24].

Chapter 4

ML-based extraction of materials-related expressions from text

4.1 Introduction

This chapter discusses the Grobid-superconductors component responsible for data extraction through an intricate set of ML models encapsulated in data parsers for extracting material-related entities from scientific text. Identifying named entities in the field of materials science poses a formidable challenge, particularly due to the intricate nature of material expressions that are composed of complex and extensive sequences requiring meticulous attention. The definition of materials remains elusive in this domain, lacking a clear-cut delineation (refer to Chapter 1). Moreover, the boundaries within materials science are loose, subject to variations based on specific subdomains, and researchers focus on distinct aspects or phenomena associated with the materials. Material expressions exhibit significant variability within superconductor research, incorporating various types of information within a single sequence. This can range from the chemical formula and doping ratio (”Zn-doped”, ”2% Cu-doped”, for instance) to the shape of the material (e.g., single crystal, wire, polycrystalline), as exemplified by expressions such as `single crystal La x Fe 1 x O 7 (with x = 0.1, and 0.2)`. Analysing non-stoichiometric formulas, in particular, presents a complex task, as these necessitate resolution with substitution values scattered throughout the paper.

The complexity inherent in deciphering such material expressions has propelled

our exploration into the realm of Machine Learning (ML) as a means to model these intricate boundaries. ML has demonstrated its efficacy in various domains, including bioinformatics and chemistry [105–107], often surpassing rule-based or lexicon-based approaches by achieving a delicate balance between flexibility and simplicity. However, it is essential to acknowledge that the success of ML is contingent upon the availability of data. This facet presents a notable challenge in material informatics compared to other domains. This challenge, in turn, has led us to create a novel dataset of annotated and linked text of scientific articles from superconductor research. In Chapter 5, we list the characteristics at the time of writing (number of articles, number of entities, etc.) and the methodology we used to construct it. Moreover, the notation incorporates supplementary text formatting choices, such as subscript and superscript, italic and bold. When these formatting options are converted to text, they are usually standardised to a regular font size, resulting in the omission of information. To tackle this problem, Grobid has been crucial in preserving these data by utilising PDF layout information, as discussed in Chapter 3. This approach improves the accuracy and comprehensiveness of our identification of named entities in materials science.

We propose a novel solution that is organised as a two-stage process: initially, the text that is selected from various structures extracted by Grobid (Chapter 3) we pass them to a set of parsers that apply Named Entity Recognition (NER) and extract significant entities. Next, the extracted entities are processed in pairs by a relation extraction (RE) process, and results are collected.

4.2 Identification of complex materials sequences

The process of identifying complex material sequences is a recently developed method that comprises four novel aspects.

4.2.1 High-quality training data

We have chosen to use machine learning for NER tasks due to the various advantages we have previously outlined. Machine learning techniques are particularly well-suited for extracting complex structures, although they require high-quality training data. We encountered a scarcity of such data, prompting us to create a new dataset. This project was an interdisciplinary collaboration between computer scientists and materials scientists. Domain experts specialising in superconductor materials provided guidance and technically validated the dataset. In Chapter 5 of this manuscript, we provide a comprehensive description of the data model and a

detailed account of the dataset construction process, which we refer to as SuperMat. The primary objective of Chapter 5 is to shed light on the specific characteristics of the domain, including the types of data, providing examples and discussing their relationships.

4.2.2 Positive sampling

The SuperMat training data is prepared using a new strategy to improve the model's ability to identify entities with low probabilities, also known as "positive sampling". This approach prioritises recall over precision, aiming to capture as many entities as possible, even if it means sacrificing precision scores. Although some unrelated entities may be extracted, they are likely to be filtered out in the subsequent step of RE, which was developed as rule-based and thus more conservative. This decision is based on expecting the final result to undergo human validation. RE with a high recall may negatively impact efficiency, as the humans validating the data would need to spend significant time removing numerous incorrect records. On the contrary, extracting a few high-precision records will require less time to be ready for use. From a holistic perspective, the recall-precision bias will be balanced between the two processes, as demonstrated in the end-to-end evaluation results, which will be further discussed in Section 4.3.4. Positive sampling is applied by removing examples that do not contain any entities (negative examples, Figure 4.2a). This approach provided an improvement of 2% in both precision and recall compared to the result without sampling. We tested additional sampling approaches, called active and random sampling, that target highly imbalanced datasets [108]. We tested them using ratios of negative examples of 0.1, 0.25, 0.5 and 1.0. However, they did not provide stable evidence suggesting any improvement, considering that our dataset was balanced.

4.2.3 ML architectures

To determine the most effective architecture for our task, we assess several traditional machine learning (ML) methods supported by Grobid through Wapiti [109] and the DL library DeLFT [110]. The architectures we consider include the linear CRF (CRF) [21], Bidirectional LSTM with CRF (BidLSTM_CRF) [22], Features Bidirectional LSTM with CRF (BidLSTM_CRF_FEATURES) [22], and a fine-tuned BERT-based transformer. For our specific task, we prefer to use the SciBERT [67] encoder with a CRF as the activation layer (SciBERT). Unlike BERT, SciBERT has been trained on the scientific text and has demonstrated superior performance in various NLP tasks, including sequence labelling, classification, and question an-

swering [67]. It is worth noting that the CRF and BidLSTM_CRF_FEATURES architectures utilise features summarised in Table 4.1.

4.2.4 Two-levels approach

To handle a diverse range of entity types, we must address the complexity of the information structures. Since most entities are structured around material names, we have implemented a two-level strategy to manage them effectively. In this approach, after processing the text with the first level ML models in the "Superconductor parser" (Section 4.2.4), a dedicated ML-based "Material parser" for process the material entities (Section 4.2.4). While previous studies such as [15, 16, 83] have mainly focused on extracting formulas, we aim to extract a larger sequence and then deal with material segmentation separately. By adopting this approach, we can address the problems in separate instances, thus reducing their complexity. This parser can handle noisy data and be applied to various fields (Section 4.2.4).

The schema illustrated in Figure 4.1 comprises three main components: the "Superconductor parser", the "Material parser" and the "Post-processing".

Superconductor parser

The "superconductor parser" extracts first-level entities (Table 4.2) by aggregating the resulting entities from two ML models: "Superconductors ML model" and "Quantities ML model". The "Quantities ML model" is reused from a separate Grobid module to extract measurements and physical quantities (Chapter 6) and is limited in scope to target only entities of temperatures and pressures. Overlapping entities of temperatures and pressures are merged, exact duplicates are removed, and the largest entities (in terms of string length) are preserved. The "Superconductors ML model" was trained with SuperMat (Chapter 5), whose schema is summarised in Table 4.2.

Entities of type `<material>`, which may contain heterogeneous mixed information, are passed to the "Material parser", which aggregates ML and rule-based methods.

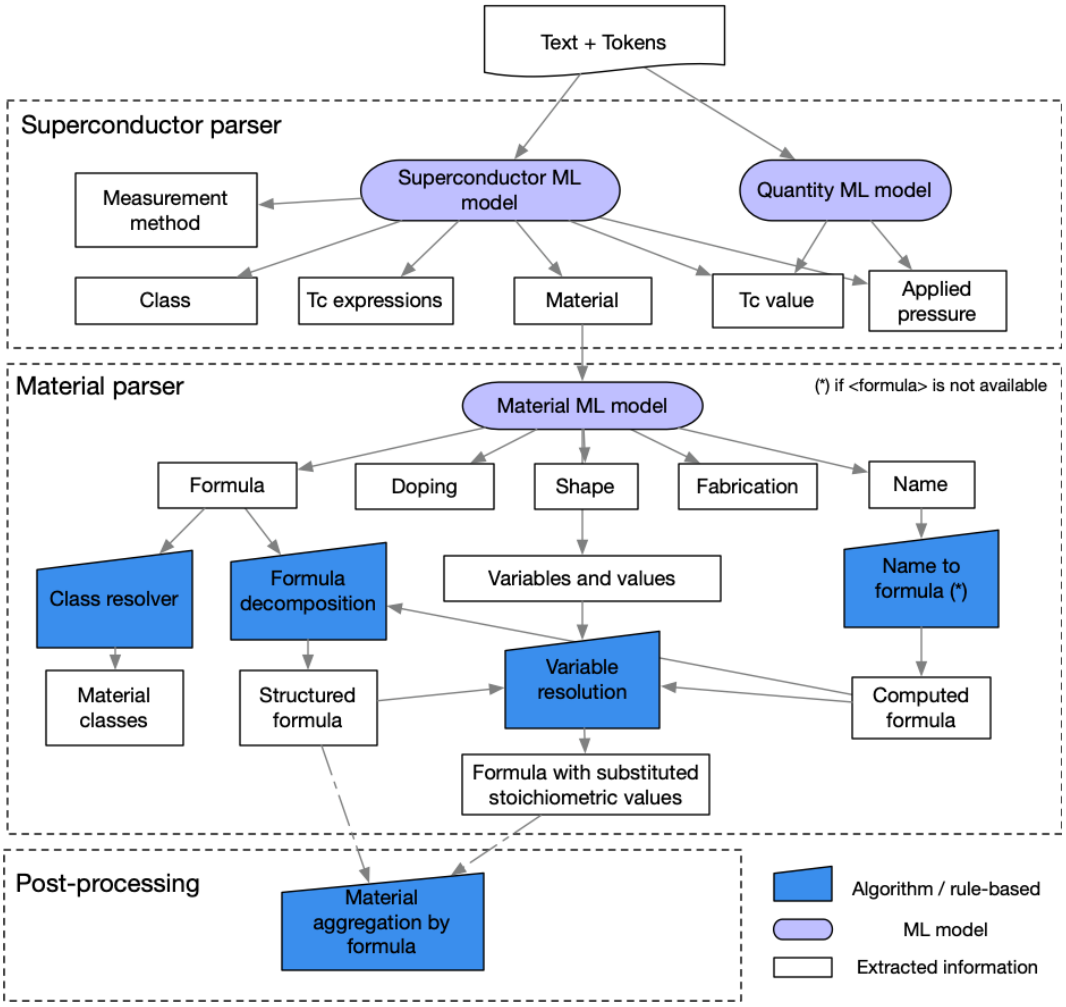


Figure 4.1: 2-level architecture for solving the NER task. The white rectangles indicate the extracted information (described in Tables 4.2 and 4.3).

Material parser

The Material parser is an expression of different approaches. First, the entity is passed through a novel Material ML model to segment and identify its content (Table 4.3). Then, different processes are applied depending on which information is available:

- Formulas are decomposed into a structured composition. We identify each element-stoichiometry pair (e.g., "O": 7.0) using text2chem [16] and PyMatgen [27]; if only the material name is available, we lookup its formula (e.g., hydrogen to H),
- Using heuristics, we classify the formula by assigning multiple classes as they are understood by superconductor researchers, for example, cuprate, oxides, alloys, etc.
- Using the variables and values extracted, we substitute them into partial formulas. For example, in `La 4 Fe 2 A 1-x O 7 (A=Mg,Co; x=0.1,0.2)`, we substitute A and x using their parsed values, and applying permutations, we obtain four *resolved formulas*: `La 4 Fe 2 Mg 0.9 O 7`, `La 4 Fe 2 Mg 0.8 O 7`, `La 4 Fe 2 Co 0.9 O 7` and `La 4 Fe 2 Co 0.8 O 7`.

4.2.5 Post-processing

Finally, after all entities are extracted, the post-processing aggregates different mentions of the same materials using the parsed formulas at the document-level. For example, formula with partial substitutions such as `La 2 Fe 1-x O 7 (x = 0.1, 0.2)` will be aggregated with materials like `La 2 Fe 0.9 O 7` appearing in other sections of the same document.

4.2.6 Extraction of relation from materials-related entities

Relation extraction (RE) links materials and their corresponding properties. We explored different approaches, such as the use of dependency parsing [112–115] or machine learning [116,117]. Unfortunately, these methods were developed or trained using text from news articles, and their performances on scientific text were unsatisfactory. Another aspect is that our dataset was too small, in terms of the number of relations, to train any of these ML methods successfully. In particular, working with dependency parsers was extremely difficult due to their inability to districate in

such complex writing. Compensating such poor performances resulted in the need to write over-complicated rules.

We developed a rule-based algorithm that links together pairs of entities:

- **material-tcValue**: The link between a material and its corresponding T_c.
- **tcValue-pressure**: The link between T_c and its related critical pressure.
- **me_method-tcValue**: The link between T_c and its corresponding measurement method.

Entities of type `<tcValue>` are pre-processed through a classifier that establishes whether or not the temperatures refer to the superconductivity. This excludes temperatures referring to irrelevant properties (e.g., annealing, transition, Curie) that might be incorrectly extracted upstream. This rule-based classifier combines the extracted entities of T_c expressions (label `<tc>`) with a set of predefined standard terms. If a temperature is not considered a T_c, it is excluded from the list of possible linking candidates.

Two scenarios are considered. First, if entities to be linked in the sentence are only two, they are linked automatically, else further rules are applied. If the word "respectively" appears in the sentence, we apply "order-linking". For example, consider the following sentence:

> P-or Ba-122 and Co-doped Ba-122 have lower T_c's of about 30 K and 24 K, respectively, which makes helium-free operation questionable.

It contains the word "respectively", and by applying "order-linking", *P-or Ba122* is assigned to *30 K* and *Co-doped Ba-122* to *24 K*.

Suppose the word "respectively" does not appear in the sentence. In that case, we apply "distance linking" by defining the distance measurement d as a value calculated as the number of characters between the centroid of each entity, measured in the number of characters. Entities surrounded by parentheses are expanded to cover the "whole parentheses", and their centroid is updated. As an example, in the sentence

> We tested two materials, MgB2 (Tc = 39 K) and FeSe (Tc = 16 K).

`39 K` is closer to `FeSe` (d=10) than to `MgB2` (d=11). In this example, however, both temperature entities would be expanded to their containing parenthesis (e.g.

"39 K" to "(Tc = 39 K)". In this case, the centre of the entity "39 K" is shifted to the left, from the initial value of 38 to 35, and the distance from MgB2 is reduced from d=11 to d=8. As a result, the MgB2 entity is correctly linked to "39 K".

Distance calculation is also adjusted with the addition of "penalties" by doubling the calculated distance when specific keywords or punctuations (",", ".", ";", "and", "but', "while', "whereas', "which", "although") appear between two entities because they represent a logical separation of predicates [118]. In the above example, the distance between 39 K and FeSe would be doubled (d=20), and the link would not be made.

4.3 Results

4.3.1 Identification of materials-related entities from text

Experimental settings

We used SuperMat (Chapter 5) for training and evaluation. The holdout set evaluation uses a fixed part of a dataset for validation. The dataset was divided into a training set corresponding to 132 documents (76%) and a holdout set comprising 32 documents (24%).

Selection must be performed to reproduce the same distribution of entities from the original dataset. We assembled the holdout set using stratified sampling and adjusting manually, ensuring it had a similar ratio of examples, entities and unique entities. The remaining was used as the training set (Figure 4.2a). Maintaining the same rate for the distribution of entity types between the two sets was more challenging. On average, we obtained approximately 15-18% labels of each type in the holdout set (Figure 4.2b), except for the <material> label (23%).

We defined the "out-of-domain" ratio as the number of unique entities from the holdout set that were not in the training set. The holdout set "out-of-domain" ratio was, on average, around 72%, which challenges the model generalisation (every 100 entities in the holdout set, 72 were never seen before during training). Most labels had an "out-of-domain" ratio above 50% (Figure 4.3); <material>, the most important label, had the highest ratio (82%) while <me_method> and <pressure> have the lowest (25% and 33%). The low ratio of <me_method> can be explained by their low entity variability (11.44%).

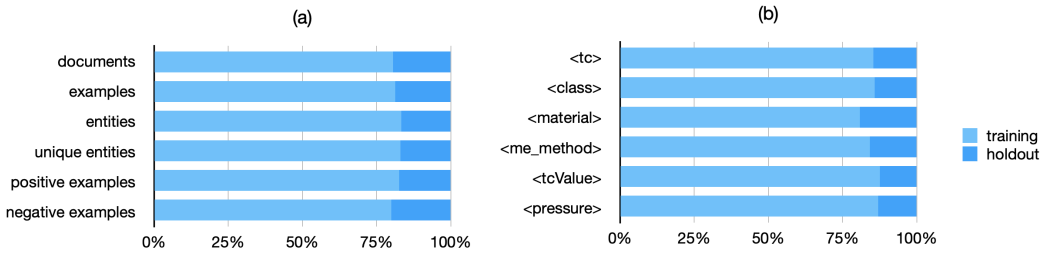


Figure 4.2: Holdout/training set distribution for (a) general metrics and (b) entity labels; entities and unique entities indicate the number of labelled entities with and without value duplicates, respectively, and positive examples (+) and negative examples (-) indicate the number of sentences with at least one entity and with no entities, respectively.

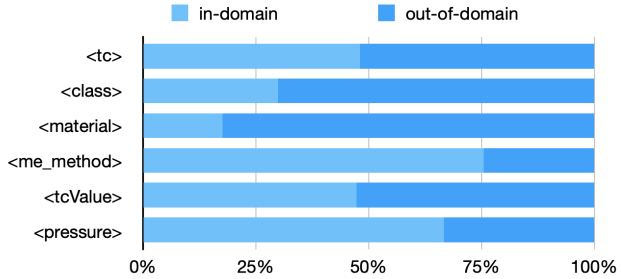


Figure 4.3: Holdout "out-of-domain" rates. The entities from the holdout set that are also in the training set are "in-domain", and the entities that are not in the training set are "out-of-domain".

Results

We evaluated each ML model by repeating the training and evaluation process five times (except for the CRF architecture, which is entirely predictable). Although the training and evaluation datasets are fixed, the way the training examples are provided to the model is randomly shuffled at each step, adding some entropy to the training, which is visible in the evaluation scores. The results are then averaged as reported in Table 4.4.

SciBERT obtained the best results with an F1 of 77.03% and a recall of around 80.69% (Table 4.4). The features did not improve RNN models: BidLSTM_CRF and BidLSTM_CRF_FEATURES resulted in the same F1 score. This result is a surprise because features such as superscript/subscript were expected to be determinants for recognising material sequences.

The <pressure> label had the lowest performance scores in all architectures. We believe that 274 training examples are not sufficiently large. Pressure expressions depend on the context and the property they refer to, so in many occurrences, they may refer to different types of pressure (e.g., annealing pressure). The label with the highest score was <material>, with F1 values of 80.77% and 78.06% for SciBERT and BidLSTM_CRF, respectively. In addition, <material> had the highest "out-of-domain" ratio in the holdout set (greater than 75%, Figure 4.3) and the highest "label variability" (the ratio between unique entities and total entities, about 42%), which suggests that the model recognises correctly materials that have not been "seen" during the training. On the other hand, the <me_method> label, which has lower "label variability" (around 11%) and a low "out-of-domain" ratio, had an F1 score of 66.56% with SciBERT and 65.92% with BidLSTM_CRF. For <tc>, the CRF outperformed the other architectures (F1 score of 83.96%), especially SciBERT (78.35%). This outcome can be explained by the extremely low variability (12.69%) of entities labelled as <tc>.

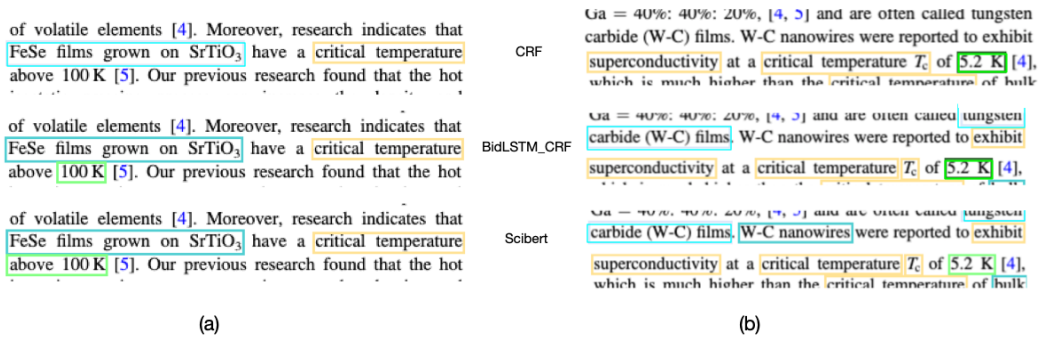


Figure 4.4: Examples taken from two sources [1, 2] of results from three different architectures: CRF, BidLSTM_CRF and, SciBERT. The boxes annotating the text represent the extracted entities (material are indicated in light blue, T_c in green, and T_c expressions in yellow).

SciBERT shows good generalisation capacity for unseen examples or examples appearing in different contexts. For example, in Figure 4.4a, only SciBERT correctly extracts "above 100K", while CRF misses it entirely and BidLSTM_CRF misses "above". In the training data, "above 100K" is not present, but "below 100K" and "$\tilde{1}$00K" are present, and several other entities contain the token "above", and SciBERT can understand that the token "above" is relevant to the temperature. In a second example (Figure 4.4b), only SciBERT can correctly extract "W-C nanowire", which is not contained in the SuperMat training data. Unfortunately, we cannot check whether "above 100K" or "W-C nanowire" are also present in the dataset used in the pre-train of SciBERT by their authors [67] because the data are not available.

4.3.2 Material parser segmentation model

We trained the Material ML model using the second layer annotation set from SuperMat comprising entities of type `<material>`. The resulting entities are segmented into eight pieces of information illustrated in Table 4.3. We trained the same architecture in the same way as the "Superconductors ML model".

Experimental settings

In this model, we created an independent holdout set because manual annotation is performed on smaller chunks of text and requires less effort than annotating entire sentences. We used material data extracted from a dataset of 500 documents ("500-papers") from three publishers: *American Institute of Physics* (AIP), *American Physical Society* (APS) and *Institute of Physics* (IOP) [12].

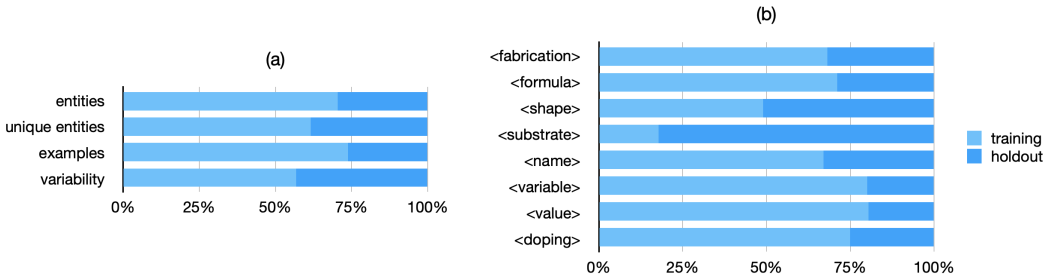


Figure 4.5: Holdout/training set for the Material ML model: (a) general metrics and (b) entity labels.

The resulting holdout set has an average coverage greater than 25% (Figure 4.5) and an average "outside the domain" ratio of 83.93% (Figure 4.6).

Results

SciBERT obtained the best results, with F1 at 84.15% (Table 4.9). The inclusion of features in the BidLSTM_CRF architecture only improved the results by less than 1% (from 83.13 to 83.76%). The label `<fabrication>` did not perform well in any architecture, most likely because it is too generic (Table 4.3), and the content is too heterogeneous. Another label, `<substrate>` has only one-third of the training examples of `<fabrication>` but obtained results that were three times higher with

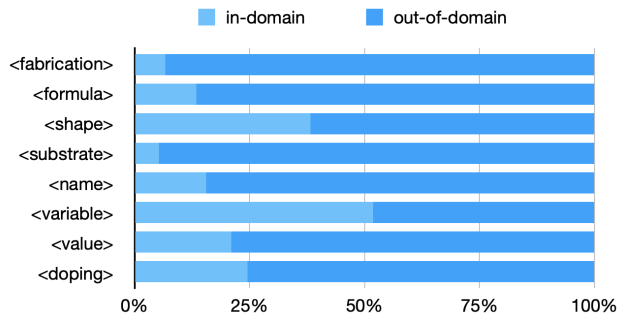


Figure 4.6: Holdout "out-of-domain" rates for the Material ML model. The entities from the holdout set that are also in the training set are the in-domain, and the entities that are not in the training set are the out-of-domain.

SciBERT, suggesting that `<fabrication>` should be split into separate and more homogeneous labels.

4.3.3 RE evaluation

This rule-based linking was evaluated using the linked entities from SuperMat [4] (Table 4.10) and is divided considering each link type. The F1 score for the `material-tcValue` relation was about 80% with a precision of 88.40%. `tcValue-pressure` F1 score was 3% lower than `material-tcValue` considering much less data available (support was 118 compared with 726).

4.3.4 End to end evaluation

End-to-end evaluation (E2EE) measures the system's capacity from the PDF documents until the final linked results. We limited the scope of the E2EE to the triplet 'material-T_c-pressure', which, at the moment, is the backbone upon which the database is built. We performed the E2EE on the "500-papers" dataset where we manually examined the resulting database as follows: 1) we marked invalid records and 2) we identified the cause of failure from a predefined set of five *error types* (Figure 4.7):

- **From table**: the extracted text is wrongly extracted from a table. Although table content is ignored, the error rate from the Grobid library is still relevant due to the lack of training data.

- **Material-related NER**: entities are either not, wrongly, or partially recognised.
- **Properties NER**: quantity entities (pressure, temperature) are not correctly extracted. We measured this error separately to identify the failure that could be shared with the Quantity ML model.
- **T_c classification**: the temperature is wrongly classified as superconducting T_c.
- **RE**: given the initial steps were performed correctly, the resulting entities are not linked correctly.

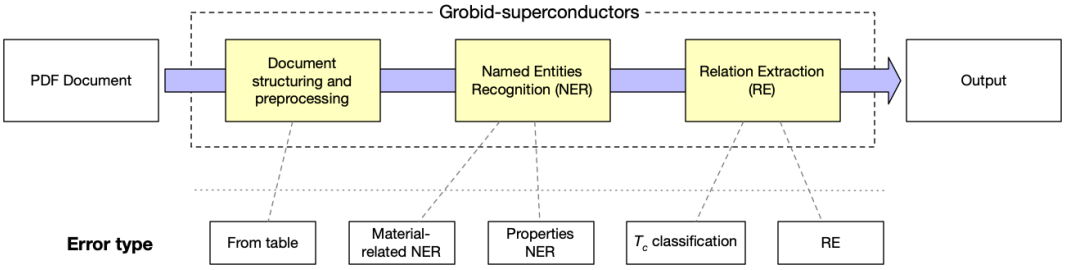


Figure 4.7: *Error types* in the context of the data flow.

The E2EE scores are summarised in Table 4.11. The recall is omitted from the table because it is less relevant. However, we estimated that the recall is between 55% and 60%, considering a sample of articles. The precision score (micro average) was 72.60% for all the subsections, although the precision of figure captions (59.28%) and unknown subsections (57.14%) were lower than those of the other subsections ($>$ 70%). The 'unknown' subsections indicate that the extracted text's structure was not well identified by Grobid, but it was nonetheless aggregated. The overall score increases to 73% when excluding unknown subsections, 75.24% when excluding figure captions, and 79.14% when excluding both. Excluding these two subsections will not affect the amount of text because they account for less than 20% of the total number of subsections.

The error types are summarised in Figure 4.8. The most common failures originate from T_c classification (40%), RE (32%), and Material-related NER (20%). The most common T_c classification failures are incorrect recognition of 1) relative values of T_c (e.g., 1 K higher than material X); 2) values indicating the transition temperature width (ΔT_c); 3) temperature values that are not T_c, for example, material synthesis temperatures (T), other critical transition temperatures that are not superconducting (e.g., T_{Curie}); and 4) values of temperature at which there is no superconductivity (e.g., "at 70 K there is no superconductivity"). "RE errors"

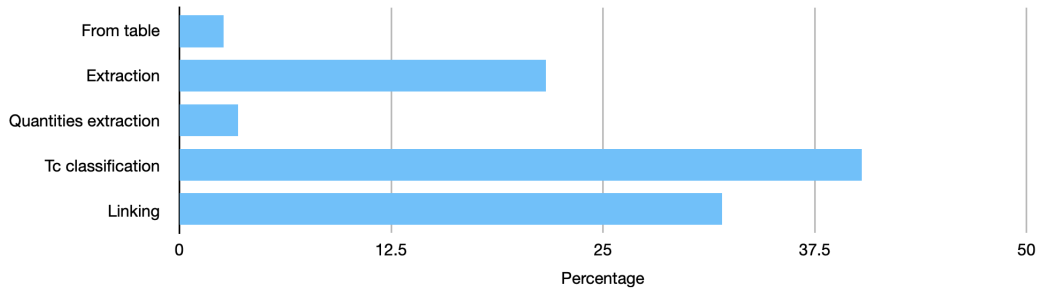


Figure 4.8: Error type distribution in the E2EE of the *500-papers* dataset.

occur mainly when the text compares relative values of T_c using materials as the basis for comparison (e.g., "The Tc = 38 K is similar to the one of MgB_2"). Finally, "Material-related NER" issues mainly originate from: 1) implicit mention of the base material when experimented using different "substrates" combination, and 2) mismatches between `<material>` and `<class>` which, by definition, overlap (see the SuperMat entities analysis in Section 5.5.3).

4.4 Conclusions

This work introduces our proposed method for automatically extracting material-related expressions. This method is an essential component of our project, which aims to create a database of experimental data sourced from scientific literature. We introduced four novelties: a high-quality dataset, SuperMat. The data was prepared using positive sampling, which increases recall in the output models. Each model was trained and evaluated among several architectures, of which the best performing were selected. Finally, we designed a 2-level process where a dedicated model parsed the material entities. We applied this specialised process in an unexplored area, where extracting information is challenging due to the intricate nature of the field and the specific attributes of the entities involved.

The contribution described in this chapter has been published in the paper "Automatic extraction of materials and properties from scientific literature" [24] and partially (more details in Chapter 6) "Automatic identification and normalisation of physical quantities from scientific literature" [119].

Table 4.1: Summary of the features used in the *superconductors* and *material* ML models. *All* under Architecture indicate only BidLSTM_CRF_FEATURES and CRF.

#	Feature	Model	Architecture
1	current token	all	all
2	current token lower cased	all	all
3-6	(four features) current token, prefix characters 1 to 4	all	CRF
7-10	(four features) current token, suffix characters 1 to 4	all	CRF
11	information about capitalisation: first character (INITCAP), all characters (ALLCAPS), none (NOCAPS)	all	all
12	digits content: all (ALLDIGIT), some digits (CONTAINDIGIT), no digits (NODIGIT)	all	all
13	(boolean) the token is composed of a single character	all	all
14	punctuation information and normalisation to placeholders: no punctuation (NOPUNCT), open or end brackets (OPENBRACKET, ENDBRACKET), various punctuation (DOT, COMMA, HYPHEN, QUOTE), open or close quotes (OPENQUOTE, ENDQUOTE), anything else (PUNCT)	all	all
15	Shadow the numbers	all	CRF
16	Shadow any characters: “x” for lowercase, “X” for uppercase, “d” for digits	all	CRF
17	As the previous but compressed	all	CRF
18	Font name	superconductors	all
19	Font size	superconductors	all
20	Font style: standard (BASELINE), superscript (SUPERSCRIPT) or subscript (SUBSCRIPT)	superconductors	all
21	(boolean) if the token style is bold	superconductors	all
22	(boolean) if the token style is italic	superconductors	all
23	(boolean) the token is identified as a chemical compound by ChemDataExtractor [111]	superconductors	all

Table 4.2: Entities extracted by the superconductors parser.

Entity (**tag**)	**Description**
Material (`<material>`)	Materials and samples names, formulas (including non-stochiometric formulas), substitution variables of values and elements, shape, doping, and substrate
Class (`<class>`)	Groups of materials having similar characteristics or common strategic compounds that define their nature
T_c value (`<tcValue>`)	The value of the superconductor critical temperature
T_c expressions (`<tc>`)	Expressions in the text that provides information about the phenomenon of superconductivity related to a value, interval or variation of T_c
Measurement method (`<me_method>`)	Technique used to measure or calculate the presence of superconductivity
Applied pressure (`<pressure>`)	Applied pressure when superconductivity is recorded

Table 4.3: Entities extracted by the material parser.

Entity (**tag**)	**Description**
Name (`<name>`)	The canonical name of a material (e.g., hydrogen, PCCO, carbon)
Formula (`<formula>`)	Chemical formula of the material (e.g., `Pr1.869Ce0.131CuO 4-`, `MgB2`, `La 2-x Sr x CuO 4`)
Doping (`<doping>`)	Doping ratio and doping materials that are adjoined to the material name (e.g., `Zn-doped`, `2% Zn-doped`)
Shape (`<shape>`)	shape of the material (e.g. single crystal, polycrystalline, thin film, powder, film)
Substitution variables (`<variable>`)	Variables that can be substituted in the formula.
Substitution values (`<value>`)	Values expressed in the doping.
Substrate (`<substrate>`)	Substrates as defined in the material name
Fabrication (`<fabrication>`)	Additional information that does not belong to any of the previous tags (e.g., intercalated, electron-doped)

Table 4.4: Evaluation scores (%) for the Superconductor ML model in the four architectures. For the DL architecture, the results are averaged over five runs. Support (Supp) indicates the number of labels in the training data. Values in bold indicate the highest score. P: precision, R: recall.

Label	**CRF**			**BidLSTM_CRF**			**BidLSTM_CRF _FEATURES**			**SciBERT**			**Supp**
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	
<class>	79.74	66.79	72.69	79.01	72.62	**75.66**	77.84	72.40	74.97	72.95	75.28	74.09	1646
<material>	79	72.15	75.42	79.25	76.94	78.06	81.07	75.10	77.94	80.15	81.42	**80.77**	6943
<me_method>	60.25	68.73	64.21	56.41	79.49	65.92	55.86	80.45	65.90	56.26	81.52	**66.56**	1883
<pressure>	46.15	29.27	35.82	49.45	58.05	52.53	50.25	60.49	**54.36**	41.72	52.68	46.51	274
<tc>	84.36	83.57	**83.96**	78.61	82.54	80.48	79.19	82.07	80.60	74.46	82.66	78.35	3741
<tcValue>	69.8	66.24	67.97	70.36	75.16	72.67	68.95	76.56	72.52	70.90	79.74	**75.06**	1099
All (micro avg)	76.88	72.77	74.77	74.59	77.67	76.09	**75.17**	76.79	75.96	73.69	**80.69**	**77.03**	

Table 4.5: Holdout/Training set distribution (%) between training and holdout sets for the Superconductor ML model. Positive examples indicate the number of sentences with at least one entity, and negative examples the number of sentences with no entities.

	training	**holdout**	**% holdout/training**
documents	132	32	24.24%
examples	16902	3905	23.10%
entities	15586	3112	19.97%
unique entities	6699	1372	20.48%
positive examples	8380	1776	21.19%
negative examples	8522	2129	24.98%

Table 4.6: Holdout/Training set distribution (%) between training and holdout sets on different labels for the Superconductors ML model.

label	**training**	**holdout**	**% holdout/training**
`<tc>`	3741	639	17.08%
`<class>`	1646	271	16.46%
`<material>`	6943	1649	23.75%
`<me_method>`	1883	355	18.85%
`<tcValue>`	1099	157	14.29%
`<pressure>`	274	41	14.96%

Table 4.7: Holdout/Training set distribution (%) training and holdout sets for the Material ML model.

	training	**holdout**	**% holdout/training**
examples	13648	5728	41.97%
entities	4512	2817	62.43%
unique entities	9268	3292	35.52%

Table 4.8: Holdout/Training set distribution (%) training and holdout sets on different labels for the Material ML model.

label	**training**	**holdout**	**% holdout/training**
`<fabrication>`	94	44	46.81%
`<formula>`	6301	2569	40.77%
`<shape>`	809	841	103.96%
`<substrate>`	32	148	462.50%
`<name>`	1930	949	49.17%
`<variable>`	1795	449	25.01%
`<value>`	1895	463	24.43%
`<doping>`	792	265	33.46%

Table 4.9: Evaluation scores (%) of the Material ML model with holdout set. Support (Supp) indicates the number of labels in the training data. Values in bold indicate the highest score. P: precision, R: recall.

Label	**CRF**			**BidLSTM_CRF**			**BidLSTM_CRF _FEATURES**			**SciBERT**			**Supp**
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	
<doping>	60.41	55.85	58.04	67.98	62.42	64.95	69.00	62.34	**65.43**	63.58	62.79	63.16	792
<fabrication>	40.00	4.55	8.16	23.61	5.91	9.24	37.33	9.09	14.48	22.51	13.18	**16.52**	94
<formula>	80.81	82.29	81.54	82.59	84.14	83.35	83.83	85.14	84.47	84.53	86.56	**85.53**	6301
<name>	72.2	63.75	67.71	76.29	78.76	77.43	74.51	80.38	77.33	77.18	81.86	**79.44**	1930
<shape>	90.89	92.51	91.69	90.93	95.79	**93.29**	90.33	95.74	92.96	89.67	97.20	93.28	809
<substrate>	37.04	6.76	11.43	54.31	32.43	40.44	60.08	33.38	42.82	56.32	41.22	**47.59**	32
<value>	80.21	83.15	81.65	84.81	89.33	86.99	85.16	90.15	**87.58**	83.14	85.92	84.50	1895
<variable>	96.85	95.98	96.41	95.19	97.77	96.46	96.32	97.90	**97.10**	96.22	96.52	96.37	1795
All (micro avg)	81.15	78.09	79.59	82.76	83.50	83.13	**83.20**	84.33	83.76	83.11	**85.23**	**84.15**	

Table 4.10: Evaluation scores for the Linking. Support (Supp) indicates the number of labels in the training data. Values in bold indicate the highest score. P: precision, R: recall.

Relationship type	**P**	**R**	**F1-**	Supp
material-tcValue	88.40	74.52	80.87	726
tcValue-pressure	85.71	71.52	77.98	118
me_method-tcValue	62.28	65.74	63.96	151

Table 4.11: Summary of the E2EE evaluation scores. Support indicates the number of labels in the training data.

Subsection	**Precision**	**Support**
Title	100	2
Abstract	80.32	61
Paragraph	75.2	623
Figure captions	59.28	140
Unknown	57.14	21
Micro avg.	72.60	847
Micro avg. (excl. figures)	75.24	707
Micro avg. (excl. unknown sections)	73.00	603
Micro avg. (excl. figures and unknown sections)	79.14	657

Chapter 5

SuperMat: Construction of a linked annotated dataset from superconductors-related publications

5.1 Introduction

In this contribution, we introduce SuperMat (Superconductor Materials), an annotated corpus of linked data derived from scientific publications on superconductors, which comprises 164 articles, 18698 entities, and 1524 links that are characterised into two layers. The first layer contains six categories: the names, classes, and properties of materials; links to their respective superconducting critical temperature (T_c); and parametric conditions such as applied pressure or measurement methods. The second layer is dedicated to material entities and contains eight classes: formula, name, doping ratio, etc. The construction of SuperMat resulted from a fruitful collaboration between computer scientists and material scientists, and its high quality is ensured through validation by domain experts. The quality of the annotation guidelines was ensured by a satisfactory Inter Annotator Agreement (IAA) between the different annotators. SuperMat includes the dataset, annotation guidelines, and annotation support tools that use automatic suggestions to help minimise human errors.

5.2 Content acquisition

SuperMat originates from PDF documents of scientific articles related to superconductor research. The original documents were collected from the following sources: (a) the Open Access (OA) version of peer-reviewed articles referenced in the SuperCon database records; (b) articles provided by domain experts containing suitable items and potential links of material names, T_c values, measurement methods, and pressures; (c) articles from "condensed matter" category of arXiv (https://arxiv.org/archive/cond-mat) selected using the search terms of "superconductor", "critical temperature", and "superconductivity".

Pre-print versions of peer-reviewed articles were obtained using a bibliographic data lookup service called biblio-glutton (https://github.com/kermitt2/biblio-glutton) that aggregates data from various sources: the Crossref (https://www.crossref.org/) bibliographic database, the unPaywall (http://unpaywall.org) service, the PubMed Central repository (https://pubmed.ncbi.nlm.nih.gov/), and mappings to other databases. We queried *biblio-glutton* using the bibliographic data of each article referenced in Supercon and subsequently downloaded the pre-print article associated with the retrieved record, if available. Although the published version may differ from the pre-print version of a document, the differences measured by comparing pre-print and peer-reviewed articles in biology [120] measured objective differences to be around 5%

5.3 Preliminary annotation study

A preliminary annotation study was carried out to assess the effort required from the annotators to reach an acceptable Inter Annotation Agreement (IAA >0.7) . We annotated two randomly selected OA papers by using a preliminary version of the guidelines with a limited tag-set of four labels: `<material>`, `<tc>` (expression describing the presence or absence of superconductivity), `<tcValue>` (value of T_c), and `<doping>` (amounts of substitution, such as stochiometric values, usually expressed as functions of x or y). The process was iterated multiple times. Each iteration ended with computing the IAA using Krippendorff's alpha coefficient [121, 122], while annotators discussed the disagreements and updated the guidelines.

Based on the results in Table 5.1, IAA reached a satisfactory level of around 0.9 after the third iteration. Although the average IAA reached 0.7 on three of the four labels in the second iteration, the average agreement was unsatisfactory. When analysing the disagreement, we noticed that the low score in the `<doping>` label was

Table 5.1: Summary of the IAA for each annotation iteration.

Iteration #	IAA	IAA by label	
1	0.45	`<material>`	0.45
		`<tc>`	0.56
		`<tcValue>`	0.50
		`<doping>`	0.21
2	0.65	`<material>`	0.75
		`<tc>`	0.85
		`<tcValue>`	0.85
		`<doping>`	0.39
3	0.89	`<material>`	0.89
		`<tc>`	0.91
		`<tcValue>`	0.88
		`<doping>`	0.94

caused by a heavy overlap with the `<material>` label, which required a more precise definition in the guidelines.

Based on this preliminary study, the following changes were implemented. (a) The label `<doping>` was merged under the `<material>` because, even with detailed documentation, it was too difficult for humans to annotate them consistently. (b) Three more labels were added: measurement methods and pressure (described as parametric conditions in relation to T_c) and class of materials.

5.4 Tag-set design

The tag set (also referred to as *labels*) represents the classes of entities and the type of relations between them, which were designed to be extracted from the text. There are two layers of tag sets: the first layer identifies entities in the text (Figure 5.1), and the second layer defines the parts within material expressions that characterise the different elements.

5.4.1 Top-level Entities

Entities (also referred to as Named Entities, mentions, or surface forms) are chunks of texts that represent information of interest, as follows:

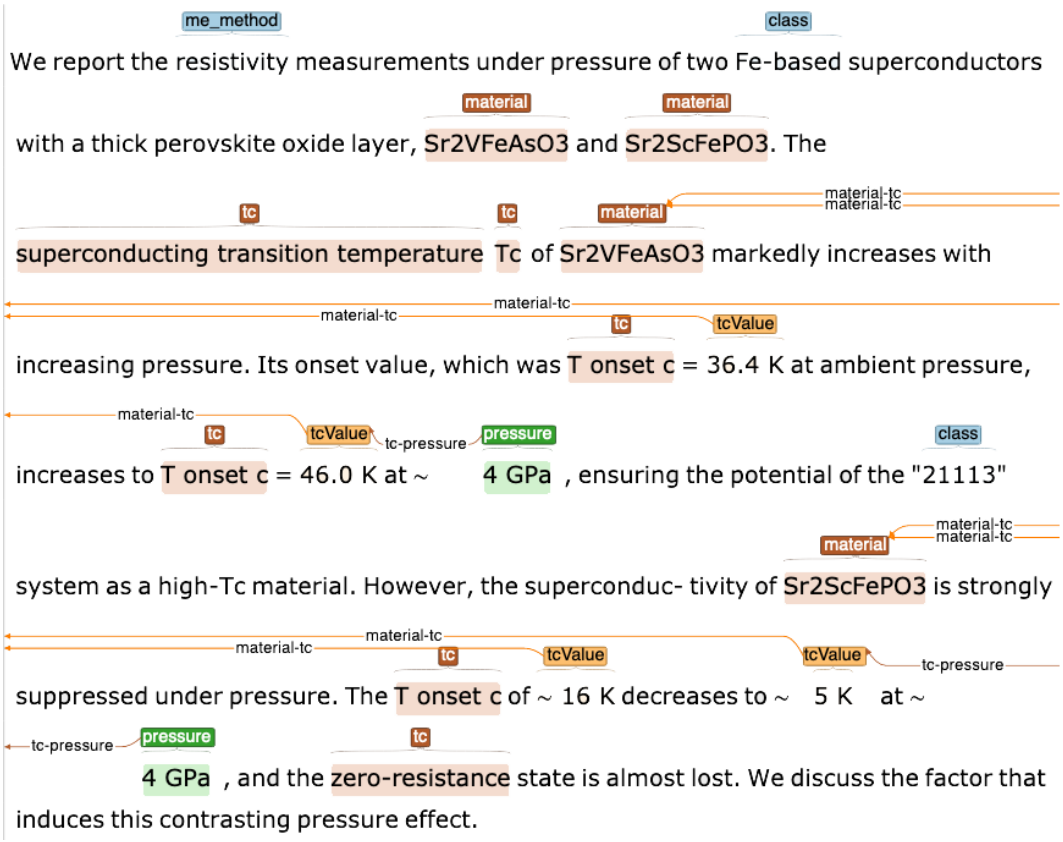


Figure 5.1: Example in the annotated corpus. Excerpt from © 2009 The Physical Society of Japan (J. Phys. Soc. Jpn. 78, 123707)

Class (tag: `<class>`) represents a group of materials defined by certain characteristics. Superconducting materials can be classified according to different criteria, such as composition and magnetic properties. Among publications collected for this study, the domain experts identified three types of classes based on: (a) the composition and crystal structure, (b) material phenomena (e.g. "I-type" and "II-type superconductivity", "BCS superconductors", "nematic", and "conventional/unconventional superconductivity"), and (c) high/low T_c value (e.g. "high-tc" superconductors).

In this work, we only considered the (a) classes, mainly because the material composition and crystal structure do not change with time. For example, a cuprate from 1998 is still called a cuprate today. In comparison, many material phenomena used for (b) are not robust enough and can be biased by the viewpoint of the author(s) or research group, or the measurement methods. Finally, the definition of "high-tc" superconductors (c) is completely relative; i.e., with the progress of research, materials once considered "high-tc" might not be so anymore.

Material (tag: `<material>`) identifies the name of one or more materials. This label is used to collect the following types of information:

- Chemical formula indicating the material by its general or non-stochiometric formula (e.g. $LaFe_{1-x}O_7$, WB_2),
- Compositional name (e.g. `magnesium diboride`) or abbreviations (e.g. `YBCO`),
- The material's shape (e.g. wire, powder, thin film) or form of material (e.g. single/polycrystal),
- Modification by a dopant (`Zn-doped`, `Si-doped`) or by percentage of doping (`2%-doped`). We also considered qualitative expressions such as *overdoped*, *lightly doped*, and *pure* as valid information,
- Substrate information (e.g. `grown on MgO(100) film`) when it was adjacent to the material name or formula in the text,
- Additional information about the sample when it was adjacent to the material name or formula in the text (e.g. `as-grown`, `untwinned`, `single-layer`).

Superconducting critical temperature (tag: `<tc>`) identifies expressions related to the phenomenon of superconductivity. Any temperature mentioned in the text is not necessarily the T_c. Rather, it could refer to the temperature for other processes/events, such as annealing/sintering temperature, specific measurements, and

structural changes. This label identifies the presence or absence of superconductivity at a given temperature. In addition, modifiers of this information (increasing/decreasing T_c) are also retained.

Superconducting critical temperature value (tag: `<tcValue>`) represents the temperature at which the superconducting phenomenon occurs. It can be defined by different experimental criteria, such as the onset, mid-point of resistivity drop, or zero resistivity. This value also considers boundary conditions, such as the *onset of superconductivity* and *zero resistance*.

Applied pressure (tag: `<pressure>`) indicates the applied pressure corresponding to a measured T_c.

Measurement method (tag: `<me_method>`) indicates the method used to measure or calculate the presence of superconductivity. Here, we considered the following categories: resistivity, magnetic susceptibility, specific heat, and theoretical calculations.

5.4.2 Level-2 entities

Extracted materials entities are then segmented according to the Level-2 tagset, illustrated below. The examples are shown as text entities, without font variations such as superscript or subscript.

Material name (tag: `<name>`) represent the canonical name of a material. Examples: `PCCO`, `PCO`, `metal diboride`, `oxygen`, `carbon`

Formula (tag: `<formula>`) identify the material as expressed from the chemical formula. Examples: `Pr1.869Ce0.131CuO 4-`δ, `MgB2`, `La 2-x Sr x CuO 4`

Doping (tag: `<doping>`) identify the doping ratio and doping materials that are adjoined to the material name. Examples: `overdoped`, `underdoped`, `optimally doped`, `bulk`, `pure`, `Zn-doped`, `Zn concentration`, `1% Zn`

Shape (<shape>) Identify the shape of the material. Examples: single crystal, polycrystalline, thin film, powder, films

Variables (<variable>) identify the variables that can be substituted in the formula. Examples:

- La x Fe 1-x O 7 with x < 3, variable: x
- RE x Fe 1-x O 7 with RE=La,Fe, and 1 < x < 3, variable: x, RE
- La x Fe y O 7 with x = 1,2,3, and 4 and y = 2,3,4, variable: x, y

Values (tag: <value>) identify the values expressed in the stoichiometric doping. Examples:

- La x Fe 1-x O 7 with x < 3, value: <3
- La x Fe 1-x O 7 with 1 < x < 3, value: 1<, <3
- La x Fe 1-x O 7 with x = 1,2,3, and 4, value: 1, 2, 3, and 4

Substrate : (tag: <substrate>) identify the substrates as defined in the material name. Examples: PCCO films onto Pr 2 CuO 4 (PCO)/SrTiO 3, substrate: Pr 2 CuO 4 (PCO)/SrTiO 3

Fabrication (tag: <fabrication>) represents all the various information that does not belong to any of the previous tags. Examples: cointercalated, intercalated, synthesized by MBE method, electron-doped, hole-doped

5.4.3 Relations

The relations connect entities of materials or samples to their corresponding properties, conditions, and results. The connecting links are non-directional, and each entity has no restrictions on the number of links. The relation types have been introduced in Section 4.2.6.

5.4.4 Annotation guidelines

Annotation guidelines include the principles and the rules that describe what constitutes all desired information for the SuperMat dataset and how to annotate it. They include detailed descriptions of the specific rules that have been defined for each type of information to be annotated, with one or more definitions and examples illustrating what to annotate in different cases, exceptions, and references. We used an online system to track the discussions and decisions when a question or a comment was raised and provided a link to such issues in the respective description or example. In addition, the guidelines include *linking rules* that provide information on how to connect the entities in a relationship correctly. The guidelines were built using a dynamic markup language (called RestructuredText) and stored in a git (https://git-scm.com/) version control system repository. We deployed them as HTML files via the web, which were updated automatically after each modification. They can be accessed at https://supermat.readthedocs.io.

5.5 Annotation support tools

The task of annotating documents is tedious and requires both attention and subject knowledge from the annotators. Annotation support tools aim to maximise the efficiency of annotators and minimise human mistakes. They comprise a web-based collaborative annotation tool, automatic annotation suggestions, and automatic corpus analysis.

5.5.1 Web-based collaborative annotation tool: INCEpTION

The annotation tool is the platform for creating, correcting and linking annotations. After evaluating several tools, we selected INCEpTION [123,124], a web-based multi-user platform for machine-assisted rapid dataset annotation construction. INCEpTION provides supportive functionalities that include:

- Multi-layer annotation sheets allow different annotation schemas over the same documents,
- Two annotation steps: annotation consists of manually correcting pre-imported documents, while curation allows another user to validate the annotations (Figure 5.5).

- On-the-fly automatic suggestions based on active learning and string matching (Figure 5.5),
- Bulk annotation corrections, and
- Being open-source (Apache 2.0 license), and under active development at the time of this paper (https://inception-project.github.io/).

5.5.2 Annotation suggestions

Previous works have demonstrated that annotation suggestions improve the quality of the output [125–127]. We provide two types of annotation suggestions. (i) *Machine-based annotated data* that were assigned to the documents before loading into the annotation tool. Here, we use a machine learning (ML)-based system from a previously implemented prototype [12] to support our tag-set. (ii). *Active learning recommendations* provided by INCEpTION are assigned on the fly based on previous annotations. The active-learning recommendations are less precise since they aim to increase the recall, and therefore they need to be explicitly accepted by the annotator.

5.5.3 Automatic corpus analysis

Automatic corpus analysis is a set of scripts designed to run after the validation step. These scripts automatically find inconsistencies in the links and entities while extracting the statistics of the corpus. We calculated the inconsistencies by examining every annotated entity and computing the frequency of the same text being annotated with different labels. The script outputs a summary table by visualising each annotation value and its labels and frequencies. We visually inspected this table because the reported inconsistencies can be either obvious mistakes (Table 5.4) or arise from ambiguities (Table 5.3); therefore, their context should be verified.

Although the links are conceptually non-directed, we have defined a practical convention to maintain consistency. For example, *material-tcValue* is always represented as a link between `<tcValue>` and `<material>` entities. The script also computes the statistics (Table 5.2) for the number of entities (total, unique, by class), the number of links (total, intra- and inter-paragraph, between paragraphs), and other statistical information.

Table 5.2: Statistical overview of the dataset. Level-2 entities are considered only for materials $\text{Relations}_{\text{is}}$ indicates the number of relations within the same sentence (intra-sentence). $\text{Relations}_{\text{es}}$ indicate the number of relations from different paragraphs (extra-sentence).

Documents	**Files**	**Paragraphs**	**Sentences**	**Tokens**
	164	2800	20807	1284569
Level-1 Entities	**Entities**	**Unique entities**		**Labels**
	18698	8071		6
Relations	**Relations**	**$\text{Relations}_{\text{is}}$**		**$\text{Relations}_{\text{es}}$**
	1524	985		539
Level-2 Entities	**Examples**	**Entities**	**Unique entities**	**Labels**
	9268	13648	4512	8

Table 5.3: Inconsistencies resulting from human mistakes.

Text	Label 1	#	Label 2	#
superconducting transition	<material>	1	<tc>	61
NCCO	<material>	14	<tc>	1
superconducting transition temperatures	<material>	1	<tc>	11
occurrence of superconductivity	<material>	1	<tc>	1

Table 5.4: Inconsistencies resulting from the overlapping of <material> and <class> labels.

Text	Label 1	#	Label 2	#
LiFeAs	<material>	89	<class>	1
Bi-2212	<material>	34	<class>	1
cobalt oxide	<material>	89	<class>	1
RE-123	<material>	34	<class>	1

5.6 Annotation process

The annotation workflow (Figure 5.2) was designed following the *MATTER* (Model, Annotate, Train, Test, Evaluate, and Revise) schema [128] and other related work [129, 130]. The workflow is composed of five steps (Figure 5.2): *data-preparation*, *correction*, *validation*, *testing and evaluation*, *revision*. This workflow involves three main actors: the automatic process, computer scientists, and the domain experts.

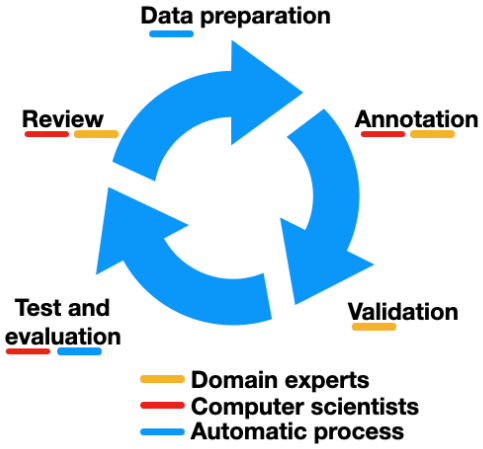


Figure 5.2: Annotation workflow. Different colours illustrate the involvement of each group at each step of the workflow.

The first step of the annotation process involves preparing the machine-based annotated data from the source PDF documents. The PDF files are converted to an XML-based format, and annotation is automatically applied, followed by four more steps:

- Annotation: The human annotator can select a document and manually add, remove, or modify each entity based on the rules defined in the guidelines. Once the annotation is complete, the document is marked ”ready” for validation.
- Validation by domain experts: Annotations from different users are validated and merged into a final document (Figure 5.5). The domain expert (”curator”) can compare the different annotated versions, select the best annotation combinations, or add new ones. This step ensures that the annotations are cross-checked and domain experts validate the document.

- Automatic consistency checks and statistical analysis: This step aims to discover obvious mistakes such as mislabelling or incorrect linking. A sequence labelling model is trained and evaluated using 10-fold cross-validation. The evaluation provides precision, recall, and f-score metrics for all the labels. The resulting model produces machine-based annotated data in the following iteration.
- Review: Retrospective analysis of the past iteration, where unclear cases are discussed and documented in the annotation guidelines.

5.7 Data transformation

There are two processes of data transformation (Figure 5.3): (a) from the source document (PDF) to the dataset format representation (XML-based), and (b) from the dataset format representation to the annotation tool exchange formats (https://inception-project.github.io/releases/0.16.1/docs/user-guide.html#sect_formats) and vice-versa.

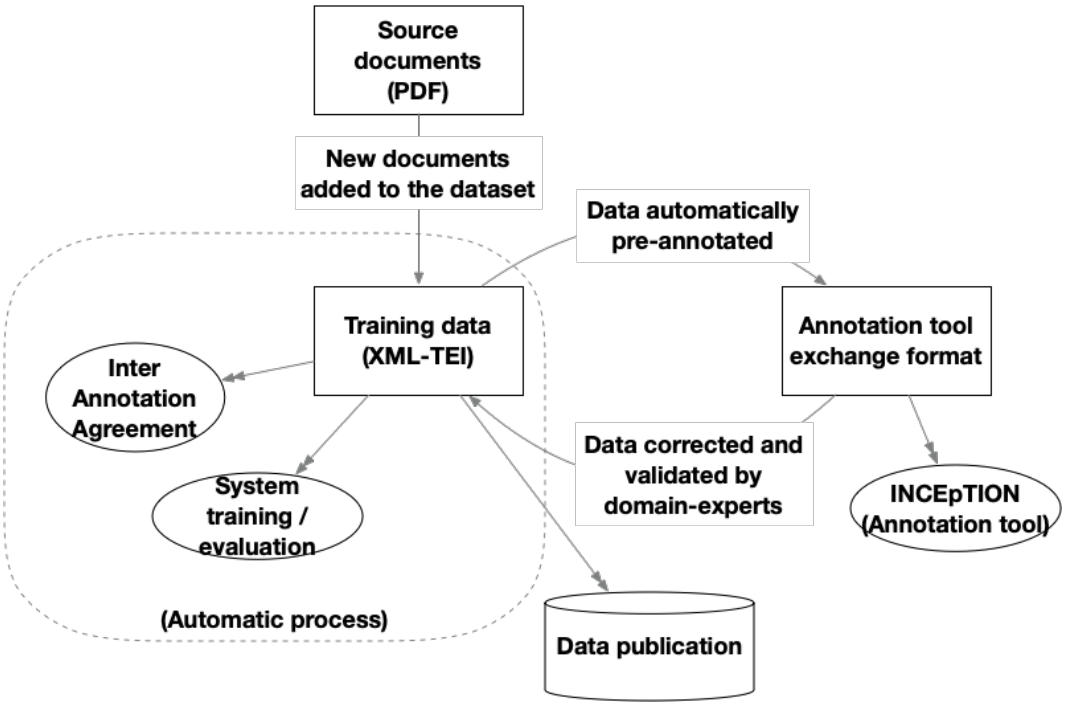


Figure 5.3: Summary of the data transformation flows.

- **PDF to XML-based**: This step converts the PDF source document to the dataset format representation in XML following the Text Encoding Initiative (TEI, https://tei-c.org/) format guidelines. Such transformation is performed by Grobid as described in Chapter 3. The resulting structured document is then encoded in XML as described below.

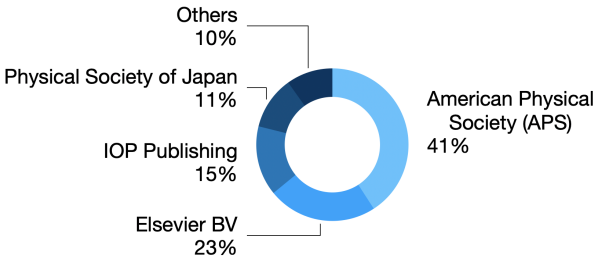
- **XML to the annotation tool exchange formats**: We transform our XML-formatted data into an INCEpTIONS compatible import format, such as the Webanno TSV 3.2 (https://inception-project.github.io/releases/0.17.0/docs/user-guide.html#sect_formats_webannotsv3), and vice-versa using a set of Python scripts. The Webanno TSV 3.2 format is an extension of the CONLL (https://www.signll.org/conll/) format, with additions to the header and column representation.

5.8 Data Record

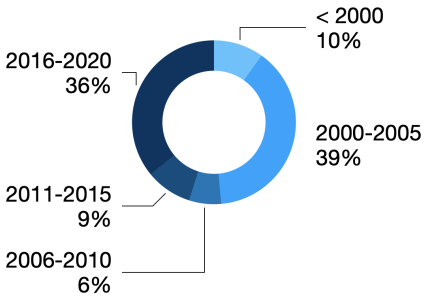
The dataset is composed of 164 PDF documents. Although we used the pre-print version of most of the articles to comply with copyright restrictions, the dataset cannot be distributed and is not publicly available in our repository. The three leading publishers represented in the corpus are the American Physical Society (APS), Elsevier, and IOP Publishing (Figure 5.4a). Figure 5.4b illustrates the distribution by publication date. We summarise SuperMat's content in Table 5.2, with the statistics of documents, entities, and links given separately. In particular, this dataset contains 16052 (7166 unique) entities spread over six labels and 1398 links.

Each document is encoded according to the XML TEI guidelines, a rich format for document representation. We have carried out no specific customisation to remain fully compliant with the general TEI schema. A TEI document has two main parts: the header (within the `<teiHeader>` tags) containing all the document metadata and the body (within the section delimited by the `<text>` tag). The transformed data has the following structure:

```
<TEI xml:lang="en" xmlns="http://www.tei-c.org/ns/1.0">
    <teiHeader>
        <fileDesc>
            <titleStmt>
                <title>[...]</title>
            </titleStmt>
            <publicationStmt>
                <publisher>[...]</publisher>
```



(a) Distribution by publisher.



(b) Distribution by year of publication.

Figure 5.4: Distribution of paper in the dataset by (a) publisher, and (b) year of publication.

```
            </publicationStmt>
        </fileDesc>
        <encodingDesc/>
            <abstract>
                <p>[...]</p>
                <ab type="keywords">[...]</ab>
            </abstract>
        <profileDesc>
        </profileDesc>
    </teiHeader>
    <text>
        <body>
            <p>[...]</p>
            <ab type="tableCaption"> [...] </ab>
            <p> [...] </p>
            <ab type="figureCaption"> [...] </ab>
        </body>
    </text>
</TEI>
```

We transformed the source documents into these TEI-compliant structures using a simplified representation for specific content types. The general objective is to flatten the content into a generic structure where priority is given to the annotations. For instance, the keywords section, which groups together the key terms defined by the author(s) of the paper, is encoded using the generic tag `<ab type="keywords">` as free text instead of the dedicated `<keywords>` element that would typically be part of the header. For both the abstract and the article body, the text is segmented in paragraphs (by means of the `<p>` element). The text is annotated with the generic `<rs>` (referencing string) element adorned with three attributes: `@type` (the entity type), `@corresp` (to provide a link to another annotation such as from *material* to T_c), and `@xml:id` (to uniquely identify the annotation for referencing or RE purposes).

Because only the captions of tables and figures are retained from the original source, a simplified encoding was defined by means of the `<ab>` element characterised by a `@type` attribute; that is, `<ab type="figureCaption">` for figure captions and `<ab type="tableCaption">` for table captions. Here is an example:

```
<p>
    The electron-doped high-<rs type="tc">transition-
    temperature</rs> (<rs type="tc">Tc</rs>) <rs
    type="class">iron-based pnictide</rs>
```

```
    superconductor <rs type="material"
    xml:id="m6">LaFeAsO1-xHx</rs> has a unique
    phase diagram: Superconducting (SC) double domes are
    sandwiched by antiferromagnetic phases at ambient
    pressure and they turn into a single dome with
    a maximum <rs type="tc">Tc</rs> that
    <rs type="tcValue" xml:id="m7"
    corresp="#m6,#9">exceeds 45K</rs>
    at a pressure of <rs type="pressure"
    corresp="#m7">3.0 GPa</rs>.
    [...]
</p>
```

In the above snippet, the entities *"3.0 GPa"*, *"exceed 45K"* and *"LaFeAsO1-xHx"* are linked together via the pairs `@corresp, @xml:id`. This schema supports multiple annotations to any part of the document. For example, the entity *exceed 45K* has a second link with the corresponding identifier (*"#9"*) to an annotation outside this paragraph.

5.9 Practical applications

The dataset was used in the following practical applications:

- Evaluation tasks: SuperMat has been used for evaluation of Large Language Models (LLM) in mining experimental data. In particular, NER and RE were tested and applied to the materials science literature [25]. The dataset not being publicly available counted in a lower chance of being used for pre-training the LLM.
- Automatic extraction: As discussed in Chapter 4, we used this dataset to develop a system for information extraction for superconducting materials.
- Weighted clustering was implemented using SuperMat annotations to identify research papers that discuss specific information of interest in different categories of superconducting materials. The annotations were used to bias the clustering algorithm towards entity similarity to achieve a more targeted clustering toward a specific type of information [26].

5.10 Technical Validation

The following measures were employed to ensure the creation of a high-quality dataset:

- Each document was revised and validated by domain experts,
- The workflow begins by assigning machine-based annotated data. This has improved the annotation task over several aspects: time consumption, error rate, and annotation agreement [125–127].
- On-the-fly automatic annotation recommendations, which provide fresh suggestions based on online decisions made by the annotators.
- The annotators have rapid access to changes in the annotation guidelines.
- The discussions were documented and linked in the guidelines.
- Reviews are discussed and approved collaboratively between domain experts and other annotators.

Table 5.5: Average IAA between the annotated and validated documents

Label	Average
<material>	0.956
<me_method>	0.887
<pressure>	0.723
<class>	0.925
<tcValue>	0.863
<tc>	0.831
Micro avg.	0.911

These guidelines are a vital piece of this work since they contain knowledge accumulated from these activities. However, measuring the completeness of the guidelines is challenging. Assuming that the documents validated by domain experts represent the ground truth, we conducted IAA analysis between different annotators against the ground truth, using the Krippendorf's Alpha metric [121]. Table 5.5 shows the average IAA which is satisfying with a value of approximately 0.9. The highest score is obtained in the <material> entities, while the lowest one is obtained in <pressure>, which appears less frequently in the papers. The disagreement in <tcValue> can appear to be too low as compared with other labels such as

<class>, which is, at first look, more ambiguous. We analysed the different cases and identified three reasons why this happens. First, <tcValue> may depend heavily on the context that requires more human attention, and it is therefore more prone to errors. Second, our suggestions system is challenged in its ability to disambiguate critical temperatures from other temperature data, leading to incorrect or invalid suggestions. Finally, the presence of mathematical symbols (e.g. "~", "<", and ">") or other modifiers ("up to", "exceeds", etc.) before the <tcValue> could generate small disagreements that accumulate in the average score.

Table 5.6: Calculated IAA for annotations produced by domain experts, non-domain experts, and novices compared to the validated version. Annotations from domain experts are cross validated.

Label	Domain experts	Non-domain experts	Novices
<material>	0.969	0.950	0.924
<me_method>	0.890	0.862	0.901
<pressure>	0.836	0.741	0.746
<class>	0.990	0.836	0.899
<tcValue>	0.895	0.734	0.841
<tc>	0.874	0.776	0.830
All labels	0.940	0.882	0.896
# paragraphs	1066	1648	325

To more precisely isolate the impact of the guidelines, we grouped the IAA results by level of domain experience. Table 5.6 displays the IAA between the validated data and the data corrected by (a) domain experts (researchers who conduct superconducting development experiments), (b) non-domain-experts (researchers with no experience with superconducting materials), and (c) novices (students in materials science with limited domain experience). Obviously, the domain experts have the highest agreement and the IAA value (around 0.95) is 0.06 higher on average than that of non-domain experts. Thus, superconducting materials is a complex domain that requires knowledge in materials science to produce high-quality data, while crowdsourcing initiatives such as the Amazon Mechanical Turk might not work well.

Furthermore, we measured the reliability of the guidelines by observing how quickly novices could reach a satisfying agreement with the validation of the domain experts, without any previous training on the guidelines. From Table 5.6, the novices can attain high IAA results by only using the guidelines and our annotation support tools. The average difference in agreement with domain experts (around 0.05) indicates that the guidelines are precise and complete, and that the annotations tools

offer sufficient support.

5.11 Data Availability

The references of the original papers and their bibliographical information, the annotation guidelines sources, and the developed code are available at the GitHub repository https://github.com/lfoppiano/SuperMat. The guidelines are freely accessible at https://supermat.readthedocs.io. The data transformation scripts were written in Python and can be run from the command line. They require BeautifulSoup (https://www.crummy.com/software/BeautifulSoup/), an open-source library for parsing XML and HTML formats. The data analysis scripts were developed as Jupyter notebooks (https://jupyter.org/), which can easily output results and graphs in the browser. The open-source annotation tool is INCEpTION (https://inception-project.github.io/). The OA content was acquired using biblio-glutton (https://www.github.com/kermitt2/biblio-glutton). We computed the IAA using the Java library DkPro statistics (https://dkpro.github.io/dkpro-statistics/) [131].

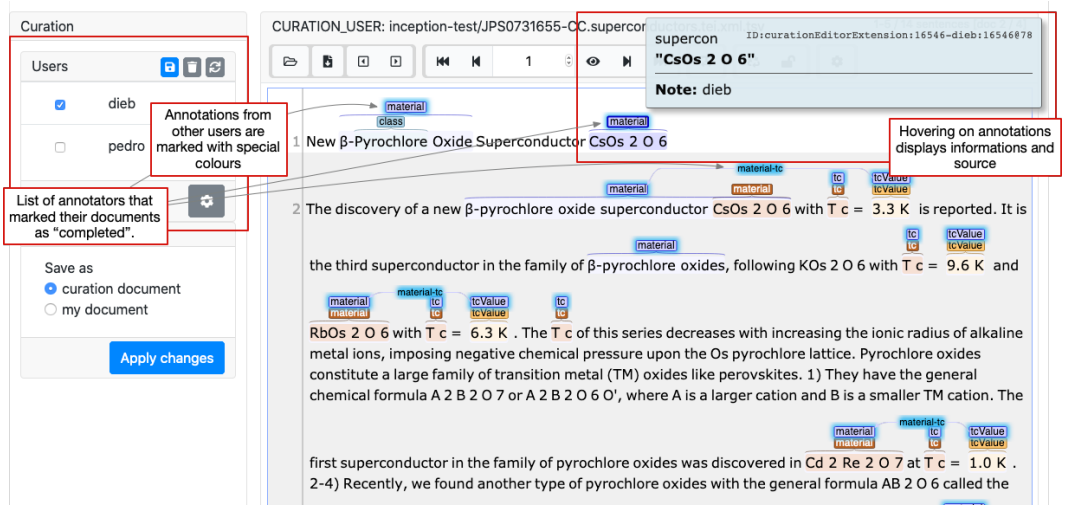


Figure 5.5: INCEpTION curation interface. Excerpt from © 2004 The Physical Society of Japan (J. Phys. Soc. Jpn. 73, 1655-1656)

5.12 Conclusions

This chapter presents the process of creating an annotated linked dataset for scientific publications on the development of superconductors. SuperMat aims to create a reliable infrastructure for improving text and data mining processes in the field of superconductor materials. We annotated 164 full-text articles using a two-layer set of annotations. The first layer contains six categories: the names, classes, and properties of materials; links to their respective superconducting critical temperature (T_c); and parametric conditions such as applied pressure or measurement methods. The second layer is dedicated to material entities and contains eight classes: formula, name, doping ratio, etc. Experts in the field have validated the dataset, which consists of 18698 top-layer entities, with 1524 links between materials, properties, and conditions. This dataset is the building block for the contribution to material extraction (Chapter 4) presented in this dissertation.

This contribution was published in the article "SuperMat: Construction of a linked annotated dataset from superconductors-related publications" [4].

Chapter 6

Extraction of properties and conditions as measurements of physical quantities

6.1 Introduction

In our approach to extracting material properties and conditions, we have opted for diverse components to accommodate the intricacies of different domains. For instance, we delve into critical temperature (Tc), applied pressure, and other pertinent factors in handling superconductor articles. Similarly, we focus on properties like coercivity and remanence when dealing with magnetic materials. This tailored approach is essential to capture domain-specific nuances effectively. However, we also recognise the necessity for a more generalised method alongside this specialised handling. Here, we prioritise a generic approach centred on extracting properties expressed through measurable physical quantities. This dual strategy ensures versatility in our material analysis framework while maintaining precision in capturing specific material behaviours across various domains.

For this task, we developed a Grobid module called Grobid-quantities that aims to provide a general solution to extracting quantities and physical measurements from the scientific literature. Grobid-quantities has been used in practical applications in other domains such as Earth observation data analysis [84] It combines ML-based techniques and lexicon-based memory that aims to cover most of the existing units, unlike other work [76]. Most other approaches lack the generalisation to an extensive corpus or deal mainly with specific languages. [78] addressed identi-

fying the numeric properties of patents using GATE (General Architecture for Text Engineering). [132] investigated issues applied to Russian-derived languages. [80] described an attempt to recognise units by looking up terms from an ontology, using ML combined with pattern matching and string metrics. At the time of development, other ML-based approaches were limited to certain fields [83] or domain [82]. We incorporate a normalisation process to convert measurements to the base units of the international system (SI), a crucial step in harmonising data from various documents, disciplines and domains.

6.2 Proposed solution

The system accepts input in text or PDF files similar to the approach described in Chapter 3. The extracted content is processed in three steps: (a) tokenisation, (b) extraction and parsing of measurements, and (c) normalisation of quantities.

6.2.1 Data model

The data model (Figure 6.1) was based on the concept of *Measurement*, which links an object or a substance with one or more *quantities*. We defined four *Measurements* types: (a) atomic, in case of a single measurement (e.g., 10 grams). (b) interval (*from 3 to 5 km*) and (c) range (100 ± 4 mm) for continuous values, and (d) a list of discrete values. A *Quantity* links the quantitative value and the unit.

Since data extracted from PDF documents unavoidably present irregular tokens from incorrect UTF-8 encoding or missing fonts, we designed this model to allow partial results. The *Value* and *Unit* entities allow three different representations (Figure 6.1): *Raw* as appear in the input, *Parsed* unifies the value into the numerical expression, and the unit with its properties (system, type). Finally, *Normalised* contains the transformed unit and values to the SI system. *Value* object supports four types of representations: numeric (2, 1000), alphabetic (two, thousand), scientific notation ($3 \cdot 10^5$), and time, which is also an expression of measurement. Units objects are organised following the SI, which allows representing units as products of simpler compounds (e.g. m/s to $m \cdot s^{-1}$) further decomposed as triples (prefix, base and power).

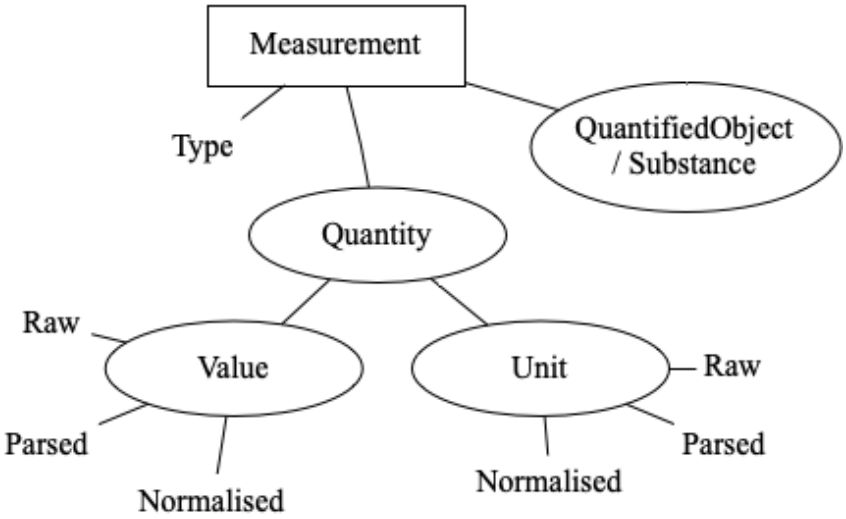


Figure 6.1: Schema of the data model, from the data parsing and normalisation point of view.

6.2.2 Tokenisation

This process splits the input data into tokens. Grobid-quantities uses two-phase tokenisation: (1) first, it splits by punctuation marks, then (2) each resulting token is re-tokenised to separate adjacent digits and alphanumeric characters. Given the example `25m^2`, first returns a list `[25m, ^, 2]` and then recursively divides `25m` as `[25, m]` resulting in `[25, m, ^, 2]`.

6.2.3 Extraction

The tokens are passed through three ML models following the 2-level approach (Section 4.2.4): first, the *Quantities* parser determines the appropriate unit and value tags. The results are further processed by the respective *Units* and *Values* parsers as illustrated in Figure 6.2.

Table 6.1 describes the labels predicted by *Quantities* parser. Note that to reconstruct complex structured objects from the flat sequence generated by the engine, additional labels are necessary (such as `<unitLeft>`, `<unitRight>`, for units).

Previous work presented extensive use of databases or ontologies. In our solution, we used a similar approach. We created a list of units (in English, French, and German) with their characteristics: system (SI base, SI derived, imperial, etc.) and

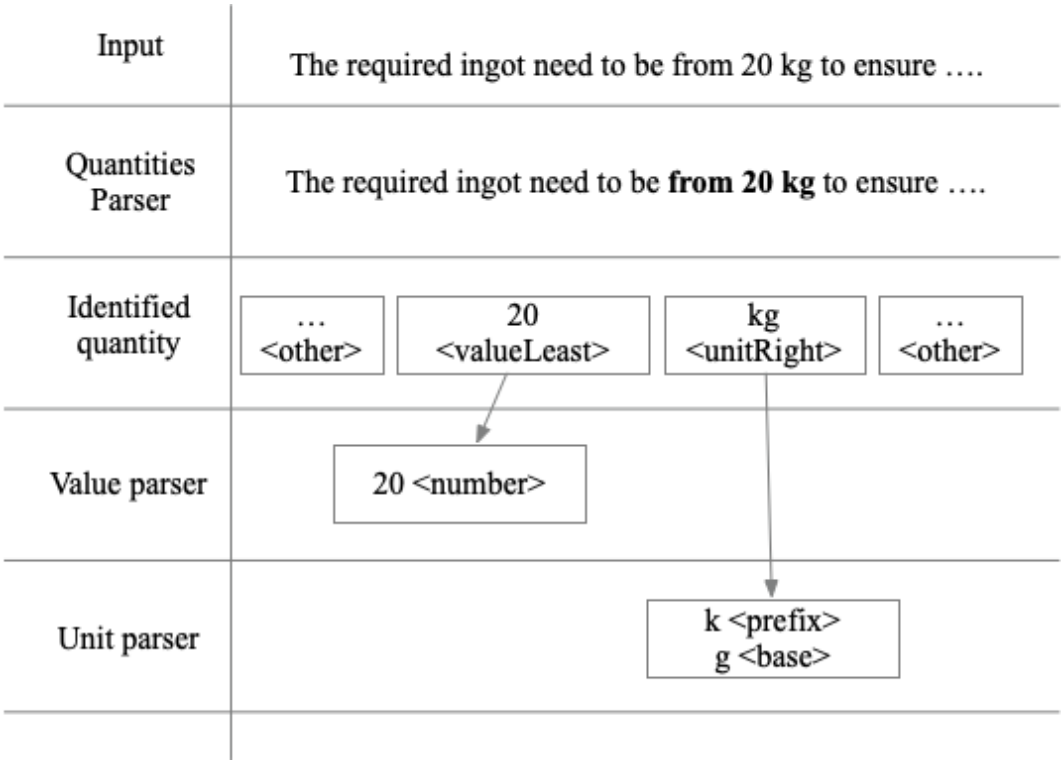


Figure 6.2: The cascade approach of the applied parsers. The Quantities parser recognises values and units passed to Values and Units parsers for further extraction.

Table 6.1: Labels description for the Quantities parser. The values referred to by the label are highlighted in bold.

Label	Description	Example
`<valueAtomic>`	value of an atomic quantity	**2** m
`<valueLeast>`	least value in an interval	from **2** m
`<valueMost>`	max value in an interval	up to **7** m
`<valueBase>`	base value in a range	**20** $\pm$ 7 m
`<valueRange>`	range value in a range	20 $\pm$ **7** m
`<valueList>`	list of quantities	**2**, **3** and **10** m
`<unitLeft>`	left-attached unit	**pH** 2
`<unitRight>`	right-attached unit	2 **m**
`<other>`	everything else	-

type (volume, length, etc.) and their representations: notations (m^3, `m^3`), lemmas (cubic meter, cubic metre) and inflexions (cubic meters, cubic metres). We made this list available through the *Unit Lexicon*, which offers unit lookups by properties (such as notation, lemma, and inflexion). A second gazetteer was created to allow the transformation of alphabetic values into numeric ones (for example, 21 to 21).

Features in the *Quantities* model are generated from preceding and following tokens, presence of capital, and digits. Orthogonal features are obtained through the *Unit Lexicon*, like a *Boolean* indicating whether a token is a known unit. Typographical information (format, fonts, subscript and superscript) are ignored.

The *Units* parser works at the character level and uses the *Unit Lexicon* to highlight known units or prefixes. The input tokens are parsed and transformed into a product of triples (prefix, base, power) as shown in Table 6.2. For example, `Kg / mm`^{2}, corresponds to $Kg \cdot mm^{-2}$ and becomes `[(K, g, 1), (m, m, -2)]` as a product of triples. We define "simple units" as all the units that can be decomposed in a single triplet and complex units when they are decomposed in multiple triplets.

Table 6.2: Labels description for the Units parser. In bold are highlighted specific examples.

Label	Description	Example
`<prefix>`	prefix of the unit	**k**m^2
`<base>`	unit base	k**m**^{2}
`<pow>`	unit power	km$^{\mathbf{2}}$
`<other>`	everything else	-

We then use the structured triples to fetch the corresponding information (system, type) from the *Unit Lexicon* and attach them to the resulting object. This implementation processes the unit characters in right-to-left order. Priority modifiers, such as parentheses, are ignored. They are generally not frequent in unit expressions and require a more complex logic.

In parallel, the *Values* parser unifies the format of the identified values into numerical formats. It supports four types: numeric, alphabetic, scientific notation, and time expression (see Table 6.3). Different techniques are applied for each type: alphabetic expressions are looked up in the word-to-number gazetteer, and scientific notation is parsed and calculated mathematically. Time expressions are further segmented using the Grobid built-in "Date" model.

Table 6.3: Labels description for the Values parser. In bold are highlighted specific examples.

Label	Description	Example
`<number>`	numeric value / coefficient	$\mathbf{2.5} \cdot 10^{\mathbf{5}}$
`<alpha>`	alphabetic value	**twenty**
`<time>`	time expression	in **1970-01-02**
`<base>`	base in scientific notation	$2.5 \cdot \mathbf{10}^{5}$
`<pow>`	exponent in scientific notation	$2.5 \cdot 10^{\mathbf{5}}$
`<other>`	everything else	-

6.2.4 Normalisation

The measurements extracted are transformed to the base SI unit (grams to kg, Celsius to Kelvin, etc.). We used an external Java library called Units of Measurement [133], which provides a set of standard interfaces and implementations to handle units and quantities safely. Manipulating measurements with transformations often leads to common errors due to wrong rounding and approximations.

6.3 Evaluation and results

We trained and evaluated our system's models using a dataset based on 32 scientific publications (English, Open Access (OA)) and three patents (with translation in English, French, and German) randomly selected from different domains such as medicine, robotics, astronomy, and physiology. Training data was generated automatically and then corrected and cross-checked by three annotators. The dataset was then divided into a training set (26 documents) and an evaluation set (9 documents). We used the holdout dataset partition to evaluate each model and produce precision, recall, and f1 scores, as summarised in Tables 6.4, 6.5, and 6.6. As presented in Chapter 4, we report the results of the three architectures introduced in Section 4.2.3: CRF, RNN (with and without layout features), and transformers, fine-tuning a pre-trained SciBERT [67].

The best results of the Quantities ML models were achieved by the SciBERT model, for which we recorded an F1 micro-average of 85.14% with precision and recall of 86.24% and 83.96%, respectively. SciBERT outperforms the CRF model by 8% F1-score. We confirmed the same observation reported in Chapter 4, that layout orthogonal features do not contribute to helping the model to generalise better. In

Table 6.4: Evaluation scores (%) of the Quantities ML model with holdout set. Support (Supp) indicates the number of labels in the training data. Values in bold indicate the highest score. P: precision, R: recall. Deep learning results are averaged over five independent runs of training and evaluation.

Label	CRF			BidLSTM_CRF			BidLSTM_CRF _FEATURES			SciBERT			Supp
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	
<unitLeft>	88.74	83.19	85.87	88.56	92.07	90.28	88.91	**92.20**	90.53	**93.99**	90.30	**92.11**	464
<unitRight>	**30.77**	30.77	30.77	24.75	**30.77**	27.42	21.73	30.77	25.41	21.84	**36.92**	**27.44**	13
<valueAtomic>	76.29	78.66	77.46	78.14	86.06	81.90	78.21	86.20	82.01	**84.50**	**88.19**	**86.31**	581
<valueBase>	84.62	62.86	72.13	83.51	94.86	88.61	83.36	**97.14**	89.72	**100.00**	90.86	**95.20**	35
<valueLeast>	77.68	69.05	73.11	**82.14**	60.63	69.67	80.73	60.63	69.12	81.09	**71.59**	**76.04**	126
<valueList>	45.45	18.87	26.67	62.15	10.19	17.34	**73.33**	8.68	15.33	64.12	**43.78**	**51.64**	53
<valueMost>	71.62	54.64	61.99	77.64	68.25	72.61	77.25	**70.31**	73.58	**81.52**	67.42	**73.71**	97
<valueRange>	**100.00**	97.14	98.55	96.72	**100.00**	**98.32**	94.05	98.86	96.38	99.39	91.43	95.24	35
All (micro avg.)	80.08	75	**77.45**	81.81	81.73	81.76	81.76	81.94	81.85	**86.24**	**83.96**	**85.08**	

general, SciBERT performs better than any other model in most labels. The CRF model better recognises only the label `<valueRange>`. The support information also suggests that `<list>` and `<unitRight>` require more training examples.

Unit expressions are generally short (1-3 characters) and have lower variability, meaning each label tends to have more duplicates than in the training datasets of other models. For example, the expressions 1% and 2% have two different values (1, 2) and the same unit (%), which would appear twice. For this reason, we evaluated the unit ML model with a dataset of 2000 annotated units called UniSCor (Units Segmentation Corpus) [134]. After annotating each unit following the schema discussed in Table 6.2, the resulting corpus contains approximately 700 simple and 1300 complex units. As illustrated in Table 6.5) the overall scores for the Units ML model are lower because the evaluation corpus is larger than the training corpus and contains a substantial percentage of complex units, while in the training dataset, the complex units are less frequent. Table 6.5 shows that features impact positively, and both architectures using them (BidLSTM_CRF_FEATURES and the CRF) obtain the best scores.

Table 6.5: Evaluation scores (%) of the Units ML model with holdout set. Support (Supp) indicates the number of labels in the training data. Values in bold indicate the highest score. P: precision, R: recall.

Label	CRF			BidLSTM_CRF			BidLSTM_CRF _FEATURES			SciBERT			Supp
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	
<base>	**80.57**	**82.34**	**81.45**	56.01	50.34	53.02	59.98	56.33	58.09	61.41	57.08	59.16	3228
<pow>	72.65	**74.45**	73.54	93.70	62.38	74.88	**93.71**	68.40	**78.94**	91.24	64.60	75.60	1773
<prefix>	93.8	84.69	**89.02**	80.31	85.25	82.54	83.21	83.58	83.35	82.10	85.30	83.62	1287
All (micro avg)	**80.73**	80.6	80.66	70.19	60.88	65.20	73.03	**65.31**	**68.94**	73.02	64.97	68.76	

Table 6.6 shows the Value ML model scores. SciBERT outperforms the other

architectures in almost all labels with an average F1 score of 99.23%. Average precision and recall are 99.13% and 99.33%, respectively. We noticed that both `<base>`, `<pow>` and `<time>` have lower f1-score. While `<base>` and `<pow>` require more training data, `<time>` expressions may overlap with `<number>` suggesting more contextual information should be introduced.

Table 6.6: Evaluation scores (%) of the Values ML model with holdout set. Support (Supp) indicates the number of labels in the training data. Values in bold indicate the highest score. P: precision, R: recall.

Label	CRF			BidLSTM_CRF			BidLSTM_CRF _FEATURES			SciBERT			Supp
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	
<alpha>	98.06	96.03	92.02	97.67	**99.53**	98.58	97.82	**99.53**	98.66	**98.59**	**99.53**	**99.05**	126
<base>	**99.91**	92.31	**96.00**	96.92	92.31	94.52	96.92	93.85	95.32	90.40	**98.46**	92.88	13
<number>	97.50	**99.88**	98.36	99.24	99.34	99.29	99.21	99.38	99.30	99.48	99.31	**99.40**	811
<pow>	**100.00**	**100.00**	**100.00**	92.92	92.31	92.47	90.28	93.85	91.90	**100.00**	**100.00**	**100.00**	13
All (micro avg)	95.79	99.27	97.50	98.90	99.17	99.03	98.86	99.25	99.05	**99.13**	**99.33**	**99.23**	

6.4 Conclusions

Our approach to property and conditions extraction emerges as a comprehensive strategy to address the diverse intricacies across different domains. The development of Grobid-quantities underscores our commitment to providing a versatile solution for extracting properties in a flexible way across diverse materials science domains. Our combined approach balances specificity and adaptability, enabling us to navigate the complexities of diverse domains while maintaining a robust foundation. As we continue to refine and enhance our methodologies, we anticipate further advancements in our ability to extract, analyse, and interpret material properties, thus contributing to the broader scientific understanding and application of materials science principles.

This contribution was published in the paper "Automatic identification and normalisation of physical quantities from scientific literature" [119].

Chapter 7

Large scale collection of experimental data from scientific literature: SuperCon^2 Database

7.1 Introduction

This Chapter presents the database we constructed through an automatic process that leverages the work presented in this dissertation. This is the main contribution of our work: SuperCon^2 Database. We processed 37770 research articles in PDF format belonging to the category *cond-mat.supr-cond* in ArXiv[1]. We obtained 40324 records of superconductor materials with their respective properties and conditions, including T_c, applied pressure measurement methods. We presented a novel data flow that ingests PDF documents (Chatpter 3), combining traditional ML architectures with a novel process that extracts materials and properties (Chapters 4 and 6) and construct a database automatically. This work represents a novel approach to TDM for material informatics. Compared with other methods, our work does not need to be adapted to the data source, whether ArXiv, BioRXiv, ChemRXiv, or any proprietary publisher article.

SuperCon^2 Database is an efficient, parallel flow of data ingestion and not a replacement for SuperCon. Considering the strict quality requirements that SuperCon

[1] https://arxiv.org/archive/cond-mat

must meet and the fact that automatic processes are imperfect, the data must be validated before being accepted in SuperCon. Compared to the original SuperCon construction, SuperCon^2 Database was collected in a few days. The updated data model counted new properties: 2052 triplets with applied pressure (*material-T_c-pressure*) and 3602 records with an explicit measurement method (*material-T_c-measurement method*).

In this chapter, we discuss how SuperCon^2 is built, including the technical information about the data format and the processing.

7.2 Database construction

The ingestion process (Figure 7.1) is designed using an Extract-Aggregate approach, implemented through Grobid-superconductors (see chapters 3 and 4).

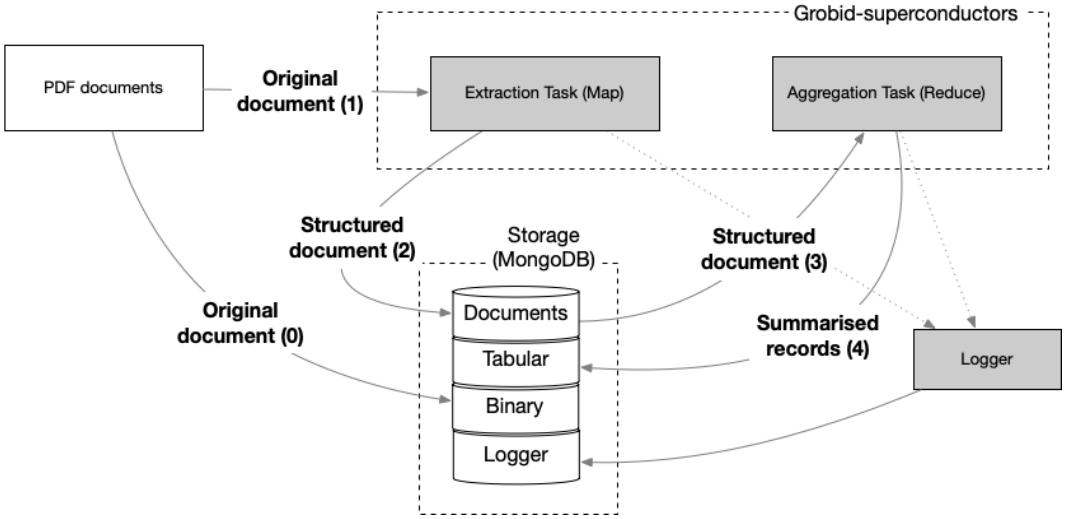


Figure 7.1: Ingestion process. The numbers between parentheses represent the order in which each operation is performed.

Storage is implemented using MongoDB [2], an open-source document database, and the collections are designed as different stages in the processing:

- the **binary** collection contains the original PDF documents
- the **documents** collection contains the structured document

[2] https://www.mongodb.com

- **tabular** collection stores the summarised records, and finally
- **logger** contains detailed information on the status of the process of each document, including possible errors. More details are given in Section 8.3.3.
- **training_data** Used to collect training examples in and discussed in Section 8.2.3.

The "Extraction Task" takes the PDF documents in input and transforms them into a rich representation document that includes text passages (sentences, paragraphs), annotations, and tokens as JSON files, which we will refer to as *structured documents* (Figure 7.2).

Each passage comprises the following attributes: the text, the type (whether a sentence or paragraph), the main section: header, body, and annexe, and the subsections: title, abstract, paragraph, and caption. Furthermore, a passage contains the list of spans representing the extracted entities and a list of layout tokens. The spans are characterised by text, type, attributes, a unique identifier, and other internal information (e.g. from which ML model the entity was extracted). The attributes are stored as a key-value and mainly contain information extracted by the material parser, such as the chemical formula, the structured composition, the material class, etc. The layout tokens contain low-level information coming from the PDF document: font size, font face, superscript, subscript, bold, italic, and coordinates within the PDF document. The coordinates are pairs of "x" and "y" numbers that are used to build annotation "boxes" to encapsulate each token independently (see Figure 8.4 in the following chapter as an example).

The "Aggregation Task" takes the structured document as input and reduces it to a table format where each row (referred to as a *summarised record*, or, *record*) pivots around the relation materials-Tc and attaches additional elements to it. The number of aggregated records can increase when large entities are extracted and may contain condensed information referring to multiple materials. For example, the raw material "Zn and CU doping La Fe B" will be aggregated as two records (doping: Zn, formula: La Fe B) and (doping: Cu, formula: La Fe B).

In the following listing, we illustrate an example of an aggregated record: lines 2-21 contain the record information such as material name, T_c, pressure, etc. Lines 22-34 represent the span list, which is the original annotation information. They are used to link the aggregated records back to the original structure document information. Line 35, 'hash' is a unique signature of the original document 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. 'type', 'timestamp' and 'status' are internal workflow information

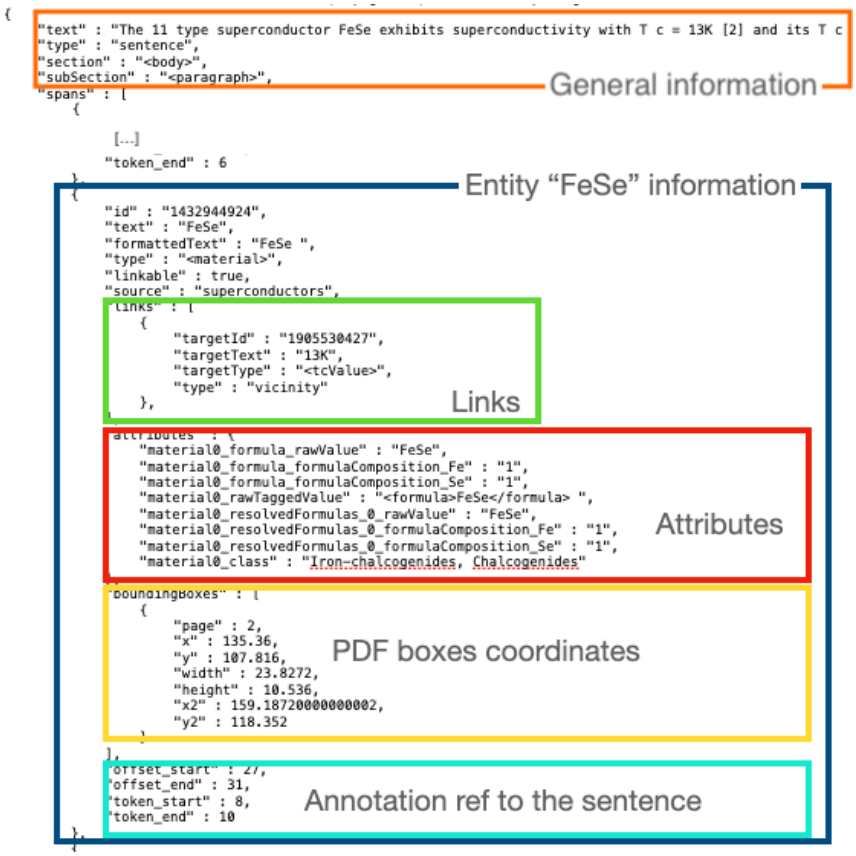


Figure 7.2: Example of the information from one single entity from a passage extracted in the "Extraction Task". The different structured information is highlighted: links, attributes, PDF entities coordinate (to visualise annotations on the PDF document), and annotations references within the passage (to visualise annotations on text).

discussed in the next chapter. The rest of the lines contain the bibliographic data (39-42).

Listing 7.1: Example of record related to the FeSe material after the aggregation.

```
{
  "_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
}
```

7.3 Results

Compared to the original SuperCon, SuperCon^2 was collected in a few days. There is no information about how SuperCon was constructed, only that it started in 1987 [7]. When we built the new process to obtain SuperCon^2, following domain-expert guidance, we considered two pieces of information: "applied pressure" and "measurement method" that have not been collected systematically in SuperCon. "Applied pressure" is the pressure applied to make the material a superconductor and has gained attention because it can radically change the physical structure of a material. On the other hand, the "measurement method" is how scientists measured the superconducting transition temperature T_c and can be used to semantically recognise multiple T_c obtained from the same material or sample. In particular, identifying calculated properties is fundamental when providing data for predictions. These properties were present only for a minority of records in SuperCon and, we hypothesise, were not considered in the initial design. The fact that applied pressure was recorded in different database fields suggests that such action was more a curator's decision than a methodological change. The resulting SuperCon^2 counted 2052 triplets with applied pressure (*material-T_c-pressure*), and 3602 records with an explicit measurement method (*material-T_c-measurement method*). The SuperCon^2 schema is discussed in 7.3 with examples in Table 7.2.

Our TDM process is limited to text, whereas the manual process focuses on plots and tables. This limitation reduces the density of the collected data, as observed by the amount of information (33,000 records) extracted from only 7227 articles. Such additions should be built as separate projects. However, the automatic pro-

cess cannot be used without manual validation, given the high-quality constraints of SuperCon. This allows combining the TDM process with data curation in a homogeneous flow described in Chapter 8.

Category	**SuperCon**	**SuperCon^2**
Size (records)	∼33000	40324
Size (papers)	7227	37700
# records with applied pressure	6	2052
# records with measurement method	600	3602
Process	Manual	Automatic
Time for process	N/A	A few days
Scope	Text, plots, tables	Text

Table 7.1: Comparison in volumes from Supercon and SuperCon^2.

7.4 Conclusions

SuperCon^2 Database represents the result of our effort to build an automatic TDM process to extract experimental data automatically. As a consequence, this is the main contribution of this dissertation. Applying techniques and methodologies described in the previous chapters allowed us to collect a large database in a few days. SuperCon^2 Database was composed of 40324 records of superconductor materials and properties, including additional properties such as the applied pressure and the T_c measurement method, which were not systematically collected in SuperCon. This database is available in various formats at https://github.com/lfoppiano/supercon.

Table 7.2: Summary and description of the SuperCon^2 schema. "Internal information" is technical information not accessible to the users.

Field name	**Description**	**Examples**
	Material information	
Raw material	The material or sample as it appears in the text	
Name	Canonical name of a material	PCCO, PCO, Metal diboride, hydrogen, carbon
Formula	Material expressed as chemical formula. This includes also formulas with stochiometric variables	$Pr_{1.869}Ce_{0.131}CuO_4 - \delta$, MgB_2, $La_{2-x}Sr_xCuO_4$
Doping	Doping ratio and doping materials that might be adjoined to the material	Overdoped, underdoped, optimally doped, bulk, pure, 1% Zn, Zn (from Zn-doped XYZ)
Shape	The shape of the material or the sample	Single crystal, polycrystal, wire, powder, film
Variables	Variables that can be substituted in the formula	x = 0, RE=Ln,St
Class	Material classification according to the domain-experts taxonomy	cuprates, oxides, and alloys
Fabrication	All the information that does not belong to any of the previous tags	Intercalated, synthesized by MBE method, electron-doped, hole-doped
Substrate	Substrate material described in the raw material	PCCO films onto $Pr_2CuO_4(PCO)/SrTiO_3$
	Properties	
Critical Temperature	Superconducting critical temperature	
Applied Pressure	Pressure applied when measuring the superconducting critical temperature	
Measurement Method	Method for measurement of the superconducting critical temperature	Magnetic susceptibility, specific heat, calculation, prediction, resistivity
	Document bibliographic information	
Section	The main body section of the paper	Header, body, annex
Subsection	The secondary segmentation area of the paper	Paragraph, table caption, figure caption, title, abstract
Authors, Title, DOI, Publisher, Journal, Year	Bibliographic information of the document	
	Internal information	
Hash, Timestamp	Hash calculated on the binary content of the original PDF document and the timestamp when the document was processed.	

Chapter 8

Reducing the impact of manual curation

8.1 Introduction

The SuperCon database was built manually from 1987 [7] by the National Institute for Materials Science (NIMS) in Japan, and it is considered a reliable source of experimental data on superconductors [9, 135–137]. However, the updates of SuperCon have become increasingly challenging due to the high publication rate. In previous chapters of this dissertation, we have described our approach to creating a reliable automatic process that allows the rapid collection of experimental data from scientific literature (Chapter 3, 4, 6). We created "SuperCon^2 Database", a comprehensive database of superconductors containing around 40000 entries within an operational duration of just a few days (Chapter 7). 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 use the term *curation* to describe the overall process of reviewing and validating database records, while *correction* refers to the specific action of altering the values of one or more properties within an individual record. When writing this article, we are unaware of any other curation tool focusing on structured databases of extracted information. Several tools for data annotation, such as Inception [138] and Doccano [139], concentrate on text labelling and classification.

In this contribution, we designed and developed a workflow with a user interface, "SuperCon^2 Interface", crafted to produce structured data of superior quality and efficiency to the one obtained by the "traditional" manual approach consisting of

reading documents and noting records, usually on an Excel file.

This contribution comprises three main novelties:

- We developed a workflow and a user interface that allow the curation of a machine-collected database. We demonstrate that using it for data correction resulted in higher quality than the "traditional" (manual) approach.
- We devise an anomaly detection process for incoming data with a lower rejection rate (false positive rate) from domain experts.
- We propose a mechanism that selects training data based on corrected records, and we demonstrate that such selections are rapidly improving the ML models.

8.2 Curation workflow

The curation of the SuperCon^2 Database acts as a workflow where user actions result in database records state transitions (Figure 8.1). Allowed manual actions include a) *mark as valid* (validation) when a record is considered correct or corrected by someone else. When a record is not valid, users can: b) *mark as invalid* when considered "potentially" invalid (or the curator is not confident), c) perform *manual correction* to update it according to the information from the original PDF document, and d) *remove* the record when it was not supposed to be extracted.

In addition to manual operations by users, this workflow also supports automatic actions: "anomaly detection" for pre-screening records (Section 8.2.2) and the "training data collector" for accumulating training data to improve ML models (Section 8.2.3).

Although only the most recent version of a record can be viewed in this system, the correction history is recorded (Section 8.3.3).

8.2.1 Workflow control

The "curation status" determines the workflow state (Section 8.2.1), the user action, and the error type (Section 8.2.1).

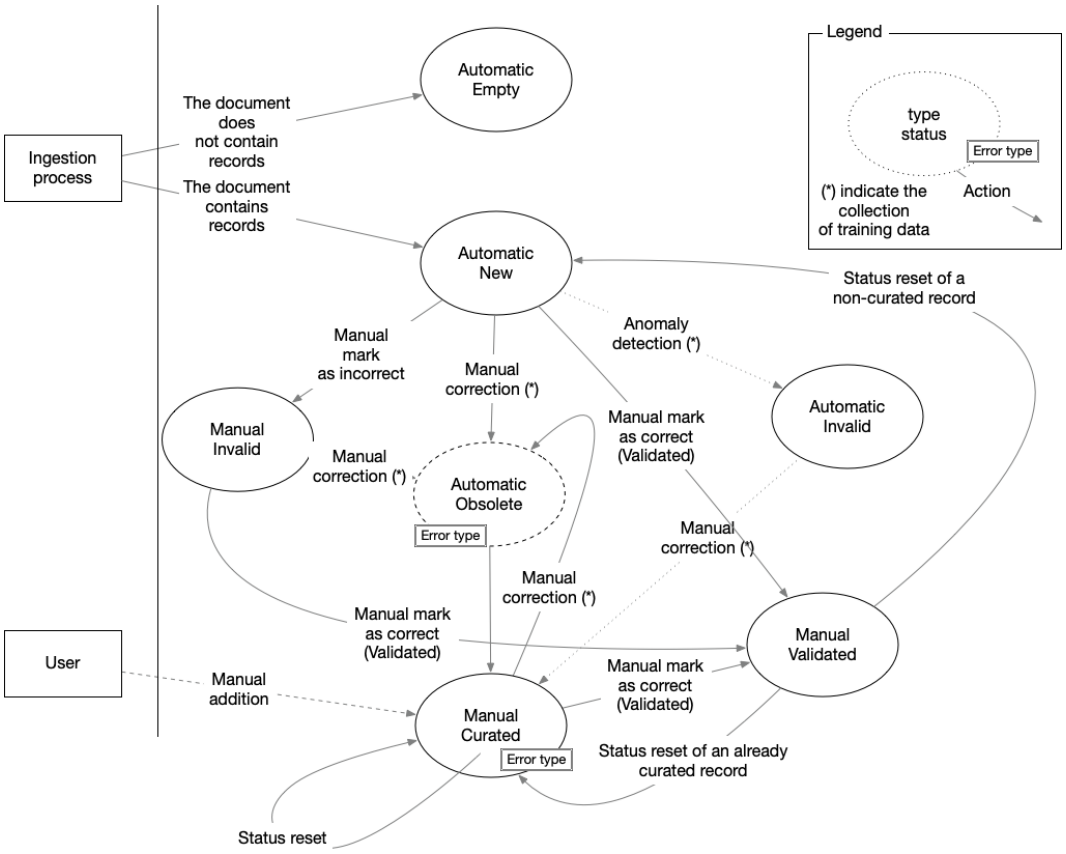


Figure 8.1: Schema of the curation workflow. Each node has type and status properties (Section 8.2.1). Each edge indicates one action. The workflow starts on the left side of the figure. The new records begin with "Automatic, New". Changes of state are triggered by automatic (Section 8.2.2) or manual operations (update, mark as valid, etc.. Section 8.3.1) and results in changes of the properties in the node. Each combination of property values identifies each state. "(*)" indicates a transition for which the training data are collected (Section 8.2.3)

Curation status

The curation status (Figure 8.1) is defined by *type* of action, manual or automatic, and *status*, which can assume the following values:

- **new**: default status when creating a new record.
- **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[1]).
- **removed**: the record has been removed by a curator (internal status).

Error types

We first introduced the error types in Section 4.3.4. However, this work extended its scope to consider data curation and anomaly detection. The additional error type values are as follows:

- **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 reading the paper. This includes when data is from tables
- **Anomaly detection**: Anomaly detection has modified the data, facilitating their retrieval from the interface.
- **Curation amends**: The curator updates the data, which does not present issues due to the automatic system.

Users are required to select one *Error Type* at every record update or removal. This information is stored in the "original" record and can differ at every modification.

[1] "internal status" indicates that their records should be hidden in the interface

8.2.2 Anomaly detection

Anomaly detection is identifying unusual events or patterns in the data. In our context, this means pinpointing data that is 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 [27], and text2chem [16]
- the extracted applied pressure cannot be parsed or falls outside the range 0 - 250 GPa.

Records identified as anomalies have *status* "invalid" and *error type* "anomaly detection" for easy identification. Since this process may find false positives, its output requires validation from curators. For example, in specific contexts, T_c values above room temperature or applied pressure up to 500 GPa may be valid in researchers' hypotheses, calculations or simulated predictions.

We ran the anomaly detection on the full SuperCon^2 Database (40324 records [24]). The anomaly detection identified 1506 records with invalid T_c, 5021 records with an incomplete chemical formula, 304 with invalid applied pressure, and 1440 materials linked to multiple T_c values. Further analysis and cross-references with contrasting information may be added in future.

8.2.3 Automatic training data collector

The curation process is a valuable endeavour demanding significant knowledge and human effort. To maximise the use of this time for collecting as much information as possible. We integrated an automatic procedure in the curation process that, for every correction, accumulates the related data examples that can be used to improve the underlying ML models.

Training data collection

In the event of a correction (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 when multiple records of 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.

Training data management

We designed a specific page of the interface (Section 8.3) to manage the collected data (Figure 8.2) in which each row corresponds to a training example composed by the decorated text showing the identified entities, the document identifier, and the status. The users can examine the data, delete it, send it to the annotation tool to be corrected, and then export it. We integrated our interface with Label-studio [3] to correct the collected training examples. Label-studio is an open-source, python-based, modern interface supporting many TDM tasks (NER, topic modelling, image recognition, etc.).

Figure 8.2: Screenshot of the training data management page in the SuperCon^2 interface. Each row contains one potential training data example. Each example comprises a sentence and its extracted entities (highlighted in colour) with potential annotation mistakes that need to be corrected using an external tool: we used Label-Studio [3]. The column "Status" indicates whether the example has been sent to the external tool.

8.3 Curation interface

The workflow is operated through the user interface, which offers several key features to facilitate the data curation process (Figure 8.1). It provides a comprehensive view of materials and their properties as a table that includes search, filter, and sorting functionality (Figure 8.3). The detailed schema, including examples, is reported in our previous work [24].

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 a table with the extracted records in the same window and the original PDF document decorated with annotations that identify the extracted materials and properties (Figure 8.4).

8.3.1 Manual curation approach

In this section, we discuss our strategy regarding manual curation, which is still indispensable for developing high-quality structures.

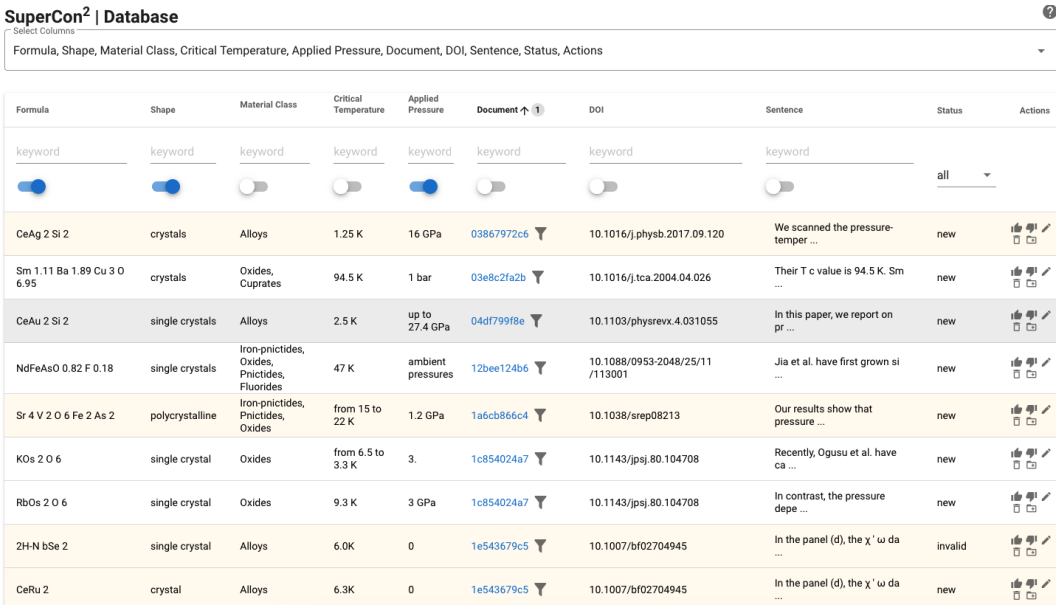


Figure 8.3: Screenshot of SuperCon^2 interface showing the database. Each row corresponds to one material-T_c pair. On the top, there are searches by attribute, sorting and other filtering operations. Curation controls (mark as valid, update, etc.) are on the right (last column). Records are grouped by document with alternating light yellow and white.

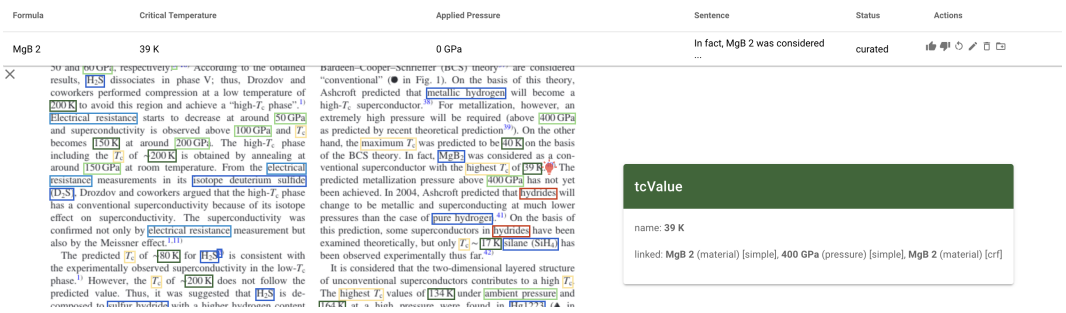


Figure 8.4: PDF document viewer showing an annotated document. The table on top is linked through the annotated entities. The user can navigate from the record to the exact point in the PDF, with a pointer (the red bulb light) identifying the context of the examined entities.

We selected curators from superconductor domain experts to certify sufficient data quality. Nevertheless, as confirmed from our experiment in Section 8.4.3, each individual's experience may impact the final result. We followed two principles to guarantee robustness in the curation process. First, we built solid curation documentation as a form of example-driven guidelines with an iterative approach we first introduced in [4]. Then, we used a double-round validation approach, in which the data was initially corrected by one person and validated in a second round by a different individual.

8.3.2 Curation guidelines

The guidelines consist mainly of the general principles and the correction rules with examples of solutions. The guidelines are designed to provide general information applied to corrections and essential explanations containing illustrations for a faster understanding (e.g. the meaning of the colours of the annotations). Differently from our previous work [4], these guidelines are divided into examples for different scenarios based on the error types mentioned in Section 8.2.1. Each example described the initial record, its context, the expected corrected record and a brief explanation, as illustrated in Figure 8.5.

8.3.3 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 new data is ingested, and it allows us to explain why certain documents haven't been processed (Figure 8.6 top). For example, sometimes documents fail because they don't contain any text (image PDF documents) or are too big (more than 100 pages).

The curation log provides a view of what, when and how a record has been corrected (Figure 8.6 bottom).

8.4 Results and evaluation

This section illustrates the experiments we have run to evaluate our work. The evaluation is composed of three sets of results. The anomaly detection rejection rate (Section 8.4.1) indicates how many anomalies were rejected by curators after

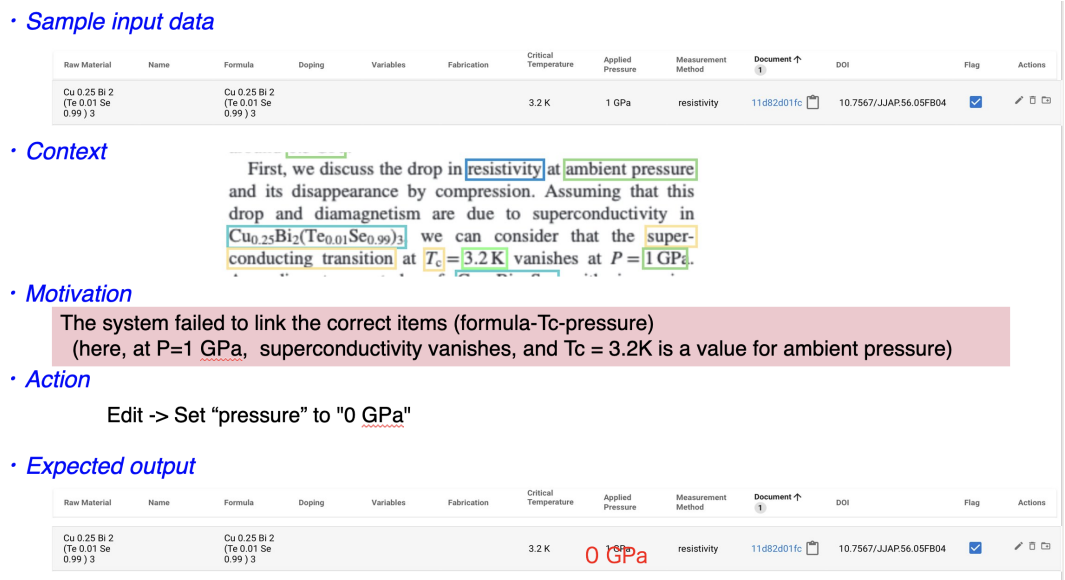


Figure 8.5: Sample curation sheet from the curation guidelines. The sheet is composed of the following information: a) Sample input data: a screenshot of the record from the “SuperCon^2 Interface”, b) *Context* represented by the related part of the annotated document referring to the record in exams. c) The *Motivation*, describing the issue, d) the *Action* to be taken, and the *Expected output*.

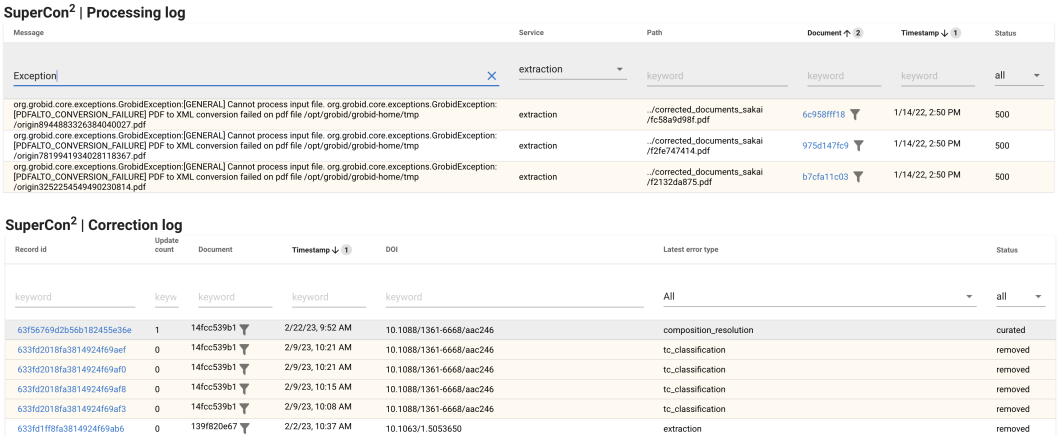


Figure 8.6: Top: *Processing log*, showing the output of each ingestion operation and the outcome with the detailed error that may have occurred. Bottom: *Correction log*, indicating each record, the number of updates, and the date/time of the last updates. Clicking on the “Record id”shows the latest record values.

validation. Then, we demonstrate that the training data automatically selected contributed to improving the ML model with a small set of examples (Section 8.4.2) 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 8.4.3, we find that using the interface improves the quality of the curated data by reducing missing experimental data.

8.4.1 Anomaly detection rejection rate

We evaluated the anomaly detection by observing the "rejection rate", which consists of the number of detected anomalies rejected by human validation. Running the anomaly detection on a database subset with 667 records found 17 anomalies in T_c, one anomaly in applied pressure, and 16 in the chemical formulas. Curators examined each reported record and rejected 4 (23%) anomalies in T_c, six anomalies (37%) in chemical formulas and 0 anomalies in applied pressure. This indicates an appropriately low rate of false positives, although a study with a larger dataset might be necessary.

8.4.2 Training data generation

We selected around 400 records in the Supercon[2] Database that were marked as invalid by the anomaly detection process, and we corrected them following the curation guidelines (Section 8.3.2). Then, we examined the corresponding training data corrected by the interface (Section 8.2.3) and obtained a set of 352 training data examples for our ML models. We call the obtained dataset *curation* to be distinguished from the original SuperMat dataset, referred to as *base*.

We prepared our experiment using SciBERT [67] that we fine-tuned for our downstream task as in [24]. We trained five models we evaluated using a fixed holdout dataset from SuperMat, averaging the results to smooth out the fluctuations. We use the DeLFT (Deep Learning For Text) [110] library for training, evaluating, and managing the prediction models. 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 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 tiny compared to "base", and we want to avoid catastrophic forgetting [140] or overfitting.

Table 8.1: F1-score from the evaluation of the fine-tuned SciBERT models. The training is performed with three different approaches. The *base* dataset is the original dataset described in [4], 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 uses the same holdout dataset from SuperMat [4]. The results are averaged over 5 runs of train and evaluation.

	base(s)	**(base+curation)(s)**	**base(s)+(base+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**

The trained models are then tested using a fixed holdout dataset that we designed in our previous work [24], and the evaluation scores are shown in Table 8.1.

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 8.2), we obtain an improvement of F1-score from 76.67%[2]

[2]In our previous work [24] we reported 77.03% F1-score. There is a slight decrease in absolute scores between DeLFT 0.2.8 and DeLFT 0.3.0. One cause may be using different hyperparameters

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

Table 8.2: Data support, the number of entities for each label in each dataset used for evaluating the ML models. The *base* dataset is the original dataset described in [4], and the *curation* dataset is automatically collected based on the database corrections by the interface and manually corrected.

	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 gives interesting insight relative to the positive impact of selecting the training data. However, there are some limitations: the *curation* dataset is small compared to the *base* dataset. This issue could be verified by correcting all the available training data, 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, particularly the learning rate and batch size, could still be better tuned to obtain better results with the second and third training protocols.

8.4.3 Data quality

We experimented with evaluating the effectiveness and accuracy of data curation using two methods: a) the user interface (*interface*), and b) the "traditional" manual approach consisting of reading PDF documents and populating an Excel file (*PDF documents*).

We selected a dataset of 15 papers, which we assigned to three curators — a senior researcher (SD), a PhD student (PS), and a master's student (MS). Each curator received ten papers: half to be corrected with the *interface* and half with the *PDF Document* method. Overall, each pair of curators had five papers in common, which they had to process using opposite methods. For instance, if curator A receives

in version 0.3.0, such as batch size and learning rate. However, the most probable cause could be the impact of using the Huggingface tokenizers library which is suffering from quality issues https://github.com/kermitt2/delft/issues/150.

paper 1 to be corrected with the *interface*, curator B, who gets the same paper 1, will correct it with the *PDF document* method. After curation, a fourth individual manually reviewed the curated content. The raw data is available in Tables 8.3 and 8.4.

We evaluated the curation considering a double perspective: time and correctness. Time was calculated as the accumulated minutes required using each method. Correctness was assessed using standard measures such as precision, recall, and the F1-score. Precision measures the accuracy of the extracted information, while recall assesses the ability to capture all expected information. F1-Score is a harmonic means of precision and recall.

Discussion

Overall, both methods required the same accumulated time: 185 minutes using the *interface* and 184 minutes using the *PDF Document* method. Not all the curators were familiar with the *interface* method when the experiment was conducted. Although they had access to the user documentation, they had to get acquainted with the user interface, thus the accumulated 185 minutes included such activities.

We examined the quality of the extracted data, and we observed an improvement of +5.55% in precision and a substantial +46.69% in recall when using the *interface* as compared with the *PDF Document* method (Table 8.5). The F1-score improved by 39.35%.

The disparity in experience significantly influenced curation accuracy, particularly regarding high-level skills. Senior researchers consistently achieved an average F1-Score approximately 13% higher than other curators (see Table 8.6). Furthermore, we observed a modest improvement between master's and PhD students. These findings may suggest 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 8.3.1). However, a new experiment with more people should be conducted to confirm this hypothesis.

Finally, the collected data suggest that all three curators had overall more corrected results using the interface as illustrated in Table 8.7.

The results of this experiment confirmed that our curation interface and workflow significantly improved the quality of the extracted data, with an astonishing improvement in recall, thus preventing curators from overlooking important information.

Table 8.3: Timetable recording the time spent for each of the 15 articles. Each row indicates the time and the event (Start, Finish) from each of the curators: Master Student (MD), PhD Student (PD), and Senior Researcher (SR). Duration is expressed in minutes.

Time	Event	Document ID	Curator	Duration (mins)
14:40	Start	02bf1b3db9	PS	0
14:49	Finish	02bf1b3db9	PS	9
14:53	Start	00b50fc0a8	PS	0
14:58	Finish	00b50fc0a8	PS	5
14:37	Start	0aa1b3161f	MS	0
14:50	Start	0454e07f64	SR	0
14:58	Finish	0454e07f64	SR	8
15:01	Start	02cbc58819	PS	0
15:06	Start	00c32076f4	SR	0
15:07	Finish	0aa1b3161f	MS	30
15:08	Finish	02cbc58819	PS	7
15:08	Start	044939701d	PS	0
15:12	Start	0021fd339f	MS	0
15:15	Finish	00c32076f4	SR	9
15:17	Finish	044939701d	PS	9
15:17	Start	08e1cb8f4f	PS	0
15:20	Start	0c7d3163ea	SR	0
15:31	Finish	08e1cb8f4f	PS	14
15:32	Finish	0021fd339f	MS	20
15:32	Start	039105663f	MS	0
15:37	Finish	0c7d3163ea	SR	17
15:53	Finish	039105663f	MS	21
15:55	Start	02c4f00127	MS	0
15:58	Start	0454e07f64	PS	0
16:02	Start	0da5febabf	SR	0
16:08	Finish	0454e07f64	PS	10
16:09	Finish	02c4f00127	MS	14
16:11	Finish	0da5febabf	SR	9
16:11	Start	0012333581	SR	0
16:12	Start	00c32076f4	PS	0
16:18	Start	021c413172	MS	0
16:22	Finish	00c32076f4	PS	10
16:23	Start	0c7d3163ea	PS	0
16:30	Finish	0012333581	SR	19
16:32	Finish	021c413172	MS	14
16:37	Start	02bf1b3db9	MS	0
16:38	Finish	0c7d3163ea	PS	15
17:32	Finish	0021fd339f	SR	12
17:34	Start	039105663f	SR	0
17:55	Finish	039105663f	SR	21
17:56	Start	02c4f00127	SR	0
18:00	Finish	02c4f00127	SR	4
18:00	Start	021c413172	SR	0
18:09	Finish	021c413172	SR	9

Table 8.4: Evaluation scores obtained for each document and method (I: Interface, P: PDF) combination. TP: True positive, FP: False positive, FN: False negative. P: Precision, R: Recall, F1: F1-score

Document ID	**# pages**	**Method**	**# TP**	**# FP**	**# FN**	**P**	**R**	**F1**
Senior Researcher (SR)								
0454e07f64	4	I	6	0	0	100.00	100.00	100.00
00c32076f4	13	P	8	0	0	100.00	100.00	100.00
0c7d3163ea	9	I	13	1	0	92.86	100.00	96.30
0da5febabf	11	P	8	0	1	100.00	88.89	94.12
0012333581	13	I	11	0	0	100.00	100.00	100.00
0aa1b3161f	5	I	9	0	1	100.00	90.00	94.74
0021fd339f	14	P	4	0	8	100.00	33.33	50.00
039105663f	9	I	11	1	0	91.67	100.00	95.65
02c4f00127	13	P	0	0	3	100.00	0.00	0.00
021c413172	5	I	15	0	0	100.00	100.00	100.00
PhD Student (PS)								
02bf1b3db9	7	I	5	0	2	100.00	71.43	83.33
00b50fc0a8	11	P	2	0	7	100.00	22.22	36.36
02cbc58819	4	I	4	0	3	100.00	57.14	72.73
044939701d	12	P	4	0	2	100.00	66.67	80.00
08e1cb8f4f	16	I	5	1	1	83.33	85.71	84.51
0454e07f64	4	P	0	1	5	0.00	16.67	0.00
00c32076f4	13	I	8	0	0	100.00	100.00	100.00
0c7d3163ea	9	P	9	0	5	100.00	64.29	78.26
0da5febabf	11	I	9	0	0	100.00	100.00	100.00
0012333581	13	P	4	4	3	50.00	72.73	59.26
Master Student (MS)								
0aa1b3161f	5	P	1	0	9	100.00	10.00	18.18
0021fd339f	14	I	12	3	3	80.00	100.00	88.89
039105663f	9	P	4	1	7	80.00	41.67	54.79
02c4f00127	13	I	3	1	1	75.00	100.00	85.71
021c413172	5	P	7	1	7	87.50	53.33	66.27
02bf1b3db9	7	P	2	0	5	100.00	28.57	44.44
00b50fc0a8	11	I	7	2	0	77.78	100.00	87.50
02cbc58819	4	P	5	0	2	100.00	71.43	83.33
044939701d	12	I	5	0	1	100.00	83.33	90.91
08e1cb8f4f	16	P	1	0	6	100.00	14.29	25.00

Table 8.5: 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*).

Method	**P (%)**	**R (%)**	**F1 (%)**	**# docs**
PDF document	87.83	45.61	52.67	15
Interface	**93.38**	**92.51**	**92.02**	15

Table 8.6: Evaluation scores (P: precision, R: recall, F1: F1-score) aggregated by experience (MS: master student, PD: PhD student, SR: senior researcher). Each person corrected ten documents.

Experience	**P (%)**	**R (%)**	**F1 (%)**	**# docs**	**# pages**
MS	90.03	60.26	64.50	10	96
PD	83.33	65.69	69.45	10	100
SR	**98.45**	**81.22**	**83.08**	10	96

Table 8.7: Evaluation scores (P: precision, R: recall, F1: F1-score) listed by experience (MS: master student, PD: PhD student, SR: senior researcher), and method (PDF document, Interface).

Experience	**Method**	**P (%)**	**R (%)**	**F1 (%)**	**# docs**	**# pages**
MS	PDF Document	94.58	36.55	48.67	6	46
	Interface	83.19	95.83	88.25	4	50
PD	PDF Document	70.00	48.51	50.78	5	49
	Interface	96.67	82.86	88.11	5	51
SR	PDF Document	**100.00**	55.56	61.03	4	51
	Interface	97.42	**98.33**	**97.78**	6	45

8.5 Code availability

This work is available at https://github.com/lfoppiano/supercon2. The repository contains the code of the SuperCon^2 interface, the curation workflow, and the ingestion processes for harvesting the SuperCon^2 Database of materials and properties. The guidelines are accessible at https://supercon2.readthedocs.io.

8.6 Conclusions

In this contribution, we presented a semi-automatic ”staging area“ to validate efficiently new experimental records automatically collected from superconductor research articles (SuperCon^2 Database [24]) before they are ingested into the existing, manually-build database of superconductors, SuperCon [7]. The system provides a curation workflow and a user interface (SuperCon^2 Interface) tailored to efficiently support domain experts in data correction and validation with fast context switching and an enhanced PDF viewer. Under the hood, the workflow ran “anomaly detection” to automatically identify outliers and a “training data collector” based on human corrections to efficiently accumulate training data to be feedback to the ML model. Compared with the traditional manual approach of reading PDF documents and extracting information from an Excel file, SuperCon^2 significantly improves the quality of curation by approximately 6% and +47% for precision and recall, respectively.

Chapter 9

Conclusion

We propose an end-to-end pipeline for extracting material information from the scientific literature to improve the efficiency and quality of materials databases. This work describes the end-to-end construction of a comprehensive TDM process to extract databases of experimental data automatically. This work represents a novel application of information extraction in materials science. Our effort resulted in the creation of SuperCon^2 Database (Chapter 7), consisting of 40324 materials and properties records collected from 37000 articles.

In Chapter 3 we describe the automatic system that reads PDF documents, extracts relevant information (Chapter 4) and stores them in a tabular format where each entry represents a material and its related properties (T_c, applied pressure, measurement methods, etc.) using a combination with Grobid-quantities: a general system for identification and standardisation of physical quantities and measurements, described in Chapter 6. ML models have been trained and evaluated using SuperMat (Chapter 5), a dataset we developed with domain experts that provides annotations and relations between entities in 164 scientific documents from superconductor research. Material expressions are carefully managed with a specific material parser that combines different methods to decompose mixed information (doping, shape, formula, name, etc.). In Chapter 8, we describe a staging area using the obtained "SuperCon^2 Database" with machine-collected entities. Using a user interface, "SuperCon^2 Interface", we allow domain experts to examine and correct the data by accessing the exact location in the original PDF document decorated with the extracted information. Our interface significantly improves the curation quality by increasing precision and recall by approximately 6% and 47%, respectively.

This work can be expanded to permanent magnetic materials, spintronics, and

thermoelectric research domains. The SuperMat dataset can be expanded to support more properties, such as current, or by adding documents from related domains, for example, magneto-caloric materials research. Finally, with the advent of Large Language Models (LLM), we can replace the relation extraction rule-based algorithm with integration with fine-tuned LLM models, which we have demonstrated promising capabilities in material-properties interaction [25].

Bibliography

[1] G Gajda, A Morawski, K Rogacki, T Cetner, A J Zaleski, K Buchkov, E Nazarova, N Balchev, M S A Hossain, R Diduszko, K Gruszka, P Przysłupski, Ł Fajfrowski, and D Gajda. Ag-doped $FeSe_{0.94}$ polycrystalline samples obtained through hot isostatic pressing with improved grain connectivity. *Superconductor Science and Technology*, 29(9):095002, jul 2016.

[2] Yusuke Shibata, Shintaro Nomura, Ryosuke Ishiguro, Hiromi Kashiwaya, Satoshi Kashiwaya, Yusuke Nago, and Hideaki Takayanagi. Magnetic field imaging of a tungsten carbide film by scanning nano-SQUID microscope. *Superconductor Science and Technology*, 29(10):104004, aug 2016.

[3] 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.

[4] 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.

[5] Evgeny Blokhin and Pierre Villars. The pauling file project and materials platform for data science: From big data toward materials genome. 2018.

[6] Yukari Katsura, Masaya Kumagai, Takushi Kodani, Mitsunori Kaneshige, Yuki Ando, Sakiko Gunji, Yoji Imai, Hideyasu Ouchi, Kazuki Tobita, Kaoru Kimura, and Koji Tsuda. Data-driven analysis of electron relaxation times in pbte-type thermoelectric materials. *Science and Technology of Advanced Materials*, 20(1):511–520, 2019.

[7] 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.

[8] Timo Sommer, Roland Willa, Jörg Schmalian, and Pascal Friederich. 3dsc–a new dataset of superconductors including crystal structures. *arXiv preprint arXiv:2212.06071*, 2022.

[9] 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, 09 2017.

[10] Thanh Dung Le, Rita Noumeir, Huu Luong Quach, Ji Hyung Kim, Jung Ho Kim, and Ho Min Kim. Critical temperature prediction for a superconductor: A variational bayesian neural network approach. *IEEE Transactions on Applied Superconductivity*, 30(4):1–5, 2020.

[11] James J. Hamlin. Superconductivity near room temperature. *Nature*, 569:491–492, 2019.

[12] Luca Foppiano, M. Dieb Thaer, Akira Suzuki, and Masashi Ishii. Proposal for Automatic Extraction Framework of Superconductors Related Information from Scientific Literature. In *Letters and Technology News, vol. 119, no. 66, SC2019-1 (no.66)*, volume 119, pages 1–5, Tsukuba, May 2019. ISSN: 2432-6380.

[13] Duff Johnson. Pdf statistics – the universe of electronic documents, 2018-05-14.

[14] Callum J Court and Jacqueline M Cole. Auto-generated materials database of curie and n el temperatures via semi-supervised relationship extraction. *Scientific Data*, 5:180111, 12 2018.

[15] Callum J. Court and Jacqueline M. Cole. Magnetic and superconducting phase diagrams and transition temperatures predicted using text mining and machine learning. *npj Computational Materials*, 6(1):18, March 2020.

[16] 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.

[17] GROBID contributors. Grobid (generation of bibliographic data). https://github.com/kermitt2/grobid, 2008 — 2019. swh:1:dir:6a298c1b2008913d62e01e5bc967510500f80710.

[18] Mikael Laakso, Patrik Welling, Helena Bukvova, Linus Nyman, Bo-Christer Björk, and Turid Hedlund. The development of open access journal publishing from 1993 to 2009. *PLOS ONE*, 6(6):1–10, 06 2011.

[19] Mikiko Tanifuji, Asahiko Matsuda, and Hideki Yoshikawa. Materials data platform - a fair system for data-driven materials science. In *2019 8th International Congress on Advanced Applied Informatics (IIAI-AAI)*, pages 1021–1022, 2019.

[20] Kyosuke Yamaguchi, Ryoji Asahi, and Yutaka Sasaki. SC-CoMIcs: A superconductivity corpus for materials informatics. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6753–6760, Marseille, France, May 2020. European Language Resources Association.

[21] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[22] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, K. Kawakami, and Chris Dyer. Neural architectures for named entity recognition. *ArXiv*, abs/1603.01360:260–270, June 2016.

[23] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, pages 4171–4186, June 2018.

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

[25] Luca Foppiano, Guillaume Lambard, Toshiyuki Amagasa, and Masashi Ishii. Mining experimental data from materials science literature with large language models. *arXiv preprint arXiv:2401.11052*, 2024.

[26] Sae Dieb, Luca Foppiano, Kensei Terashima, Pedro Baptista de CASTRO, Yoshihiko Takano, and Masashi Ishii. Superconductor research papers clustering using weighted annotated information. In *Proceedings of the National Conference of the Japanese Society for Artificial Intelligence, 36th (2022)*, pages 1S5IS2a03–1S5IS2a03. Japan Society for Artificial Intelligence, 2022.

[27] 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.

[28] Kento Mitsui, Yutaka Sasaki, and Ryoji Asahi. Automatic knowledge acquisition from superconductivity information in literature. *Science and Technology of Advanced Materials: Methods*, 3(1):2206532, 2023.

[29] Sang-Hoon Park, Baekjun Kim, Sihoon Choi, Peter D. W. Boyd, Berend Smit, and Jihan Kim. Text mining metal–organic framework papers. *Journal of Chemical Information and Modeling*, 2018.

[30] Kevin Cruse, Amalie Trewartha, Sanghoon Lee, Zheren Wang, Haoyan Huo, Tanjin He, Olga Kononova, Anubhav Jain, and Gerbrand Ceder. Text-mined dataset of gold nanoparticle synthesis procedures, morphologies, and size entities. 2022.

[31] Satanu Ghosh and Kun Lu. Band gap information extraction from materials science literature – a pilot study. *Aslib Journal of Information Management*, 2022.

[32] K. Choudhary and M. L. Kelley. Chemnlp: a natural language processing based library for materials chemistry text data. 2022.

[33] John A. Keith, Valentin Vassilev-Galindo, Bingqing Cheng, Stefan Chmiela, Michael Gastegger, Klaus-Robert Müller, and Alexandre Tkatchenko. Combining machine learning and computational chemistry for predictive insights into chemical systems. *Chemical Reviews*, 2021.

[34] Filipe Rodrigues, Francisco C. Pereira, and Bernardete Ribeiro. Sequence labeling with multiple annotators. *Machine Learning*, 2013.

[35] Buzhou Tang, Ying Feng, Xiaolong Wang, Yonghui Wu, Yaoyun Zhang, Min Jiang, Jingqi Wang, and Hua Xu. A comparison of conditional random fields and structured support vector machines for chemical entity recognition in biomedical literature. *Journal of Cheminformatics*, 2015.

[36] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.

[37] Usman Naseem, Matloob Khushi, Shah Khalid Khan, Kamran Shaukat, and Mohammad Ali Moni. A comparative analysis of active learning for biomedical text mining. *Applied System Innovation*, 2021.

[38] Xi Yang, Jiang Bian, Yan Gong, William R. Hogan, and Yonghui Wu. Madex: A system for detecting medications, adverse drug events, and their relations from clinical notes. *Drug Safety*, 2019.

[39] Alex Graves, Abdelrahman Mohamed, and Geoffrey E. Hinton. Speech recognition with deep recurrent neural networks. 2013.

[40] Youhui Tian. Artificial intelligence image recognition method based on convolutional neural network algorithm. *Ieee Access*, 2020.

[41] Hasim Sak, Andrew W Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. 2014.

[42] K Cho, B Van Merriënboer, C Gulcehre, et al. Learning phrase representations using rnn encoder–decoder for statistical machine translation,‖ in proceedings of the 2014 conference on empirical methods in natural language processing (emnlp), 2014. *Qatar1724–1734*, 2001.

[43] D. Kim. Research on text classification based on deep neural network. *International Journal of Communication Networks and Information Security (IJCNIS)*, 14:100–113, 2022.

[44] S. Lyu and J. Liu. Combine convolution with recurrent networks for text classification. 2020.

[45] J. Li, C. Wang, and R. Cai. Channel attention convolutional recurrent neural network on street view symbol recognition. *Highlights in Science, Engineering and Technology*, 9:390–397, 2022.

[46] T. Siswantining, S. Pratama, and D. Sarwinda. Spratama model for indonesian paraphrase detection using bidirectional long short-term memory and bidirectional gated recurrent unit. *Media Statistika*, 15:129–138, 2023.

[47] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

[48] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. 2013.

[49] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*, 2016.

[50] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[51] M. J. Peters, M. E. Neumann, M. Iyyer, M. Gardner, C. M. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. *Proceedings of the 2018 Conference of the North American Chapter Of the Association for Computational Linguistics: Hu*, 2018.

[52] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. 2017.

[53] M. Joshi, O. Levy, L. Zettlemoyer, and D. S. Weld. Bert for coreference resolution: baselines and analysis. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conferen*, 2019.

[54] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36:1234–1240, 2019.

[55] R. Joshi, P. Goel, and R. Joshi. Deep learning for hindi text classification: a comparison. *Intelligent Human Computer Interaction*, pages 94–101, 2020.

[56] Joon-Young Jung. Dg-based spo tuple recognition using self-attention m-bi-lstm. *Etri Journal*, 2021.

[57] Walid Hafiane, Joël Legrand, Yannick Toussaint, and Adrien Coulet. Experiments on transfer learning architectures for biomedical relation extraction. 2020.

[58] Young Min Kim and Tae Hoon Lee. Korean clinical entity recognition from diagnosis text using bert. *BMC Medical Informatics and Decision Making*, 2020.

[59] Hu Ng, Glenn Jun Weng Chia, Timothy Tzen Vun Yap, and Vik Tor Goh. Modelling sentiments based on objectivity and subjectivity with self-attention mechanisms. *F1000research*, 2022.

[60] S Shreyashree, Pramod Sunagar, S Rajarajeswari, and Anita Kanavalli. Bert-based hybrid rnn model for multi-class text classification to study the effect of pre-trained word embeddings. *International Journal of Advanced Computer Science and Applications*, 2022.

[61] Mohamed Taha, Hala H. Zayed, Marina Azer, and Mahmoud Gadallah. Automated covid-19 misinformation checking system using encoder representation with deep learning models. *Iaes International Journal of Artificial Intelligence (Ij-Ai)*, 2023.

[62] Ramchandra Joshi. Evaluation of deep learning models for hostility detection in hindi text. 2021.

[63] Abhishek Velankar, Patil Hrushikesh, Amol Gore, Shubham Salunke, and Joshi Raviraj. Hate and offensive speech detection in hindi and marathi. 2021.

[64] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[65] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[66] G. Lample and A. Conneau. Cross-lingual language model pretraining. 2019.

[67] 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.

[68] M. Yasunaga, J. Leskovec, and P. Liang. Linkbert: pretraining language models with document links. 2022.

[69] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing, 2020.

[70] Zhi Hong, Aswathy Ajith, Gregory Pauloski, Eamon Duede, Carl Malamud, Roger Magoulas, Kyle Chard, and Ian Foster. Scholarbert: Bigger is not always better, 2022.

[71] P. Su and K. Vijay-Shanker. Investigation of improving the pre-training and fine-tuning of bert model for biomedical relation extraction. 2021.

[72] S. Pranav, R. A. Chitteth, K. Christopher, G. Sonkakshi, P. L. Prerana, H. Lauren, and R. Rampi. A general-purpose material property data extraction pipeline from large polymer corpora using natural language processing. *npj Comput Mater*, 9, 2023.

[73] T. D. Huan, A. Mannodi-Kanakkithodi, V. Sharma, G. Pilania, and R. Ramprasad. A polymer dataset for accelerated property prediction and design. *Sci Data*, 3, 2016.

[74] S. Huang and J. M. Cole. Batterybert: a pretrained language model for battery database enhancement. *Journal of Chemical Information and Modeling*, 62:6365–6377, 2022.

[75] Jiang Guo, A. Santiago Ibanez-Lopez, Hanyu Gao, Victor Quach, Connor W. Coley, Klavs F. Jensen, and Regina Barzilay. Automated chemical reaction extraction from scientific literature. *Journal of Chemical Information and Modeling*, 62(9):2035–2045, 2022. PMID: 34115937.

[76] Hidir Aras, René Hackl-Sommer, Michael Schwantner, and Mustafa Sofean. Applications and challenges of text mining with patents. In *IPaMin@ KONVENS*, 2014.

[77] A. S. Maiya, D. W. Visser, and A. C. Wan. Mining measured information from text. *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2015.

[78] Milan Agatonovic, Niraj Aswani, Kalina Bontcheva, Hamish Cunningham, Thomas Heitz, Yaoyong Li, Ian Roberts, and Valentin Tablan. Large-scale, parallel automatic patent annotation. In *Proceedings of the 1st ACM workshop on Patent information retrieval*, pages 1–8. ACM, 2008.

[79] Yury Hetsevich and Alena Skopinava. Processing of quantitative expressions with measurement units in the nominative, genitive, and accusative cases for belarusian and russian. In *Text, Speech and Dialogue: 17th International Conference, TSD 2014, Brno, Czech Republic, September 8-12, 2014. Proceedings 17*, pages 101–107. Springer, 2014.

[80] Soumia Lilia Berrahou, Patrice Buche, Juliette Dibie-Barthélemy, and Mathieu Roche. How to extract unit of measure in scientific documents?.

[81] George H John and Pat Langley. Estimating continuous distributions in bayesian classifiers. *arXiv preprint arXiv:1302.4964*, 2013.

[82] Yanna Shen Kang and Mehmet Kayaalp. Extracting laboratory test information from biomedical text. *Journal of pathology informatics*, 4:23–23, August 2013.

[83] Thaer M Dieb, Masaharu Yoshioka, Shinjiro Hara, and Marcus C Newton. Framework for automatic information extraction from research papers on nanocrystal devices. *Beilstein journal of nanotechnology*, 6(1):1872–1882, 2015.

[84] Kyle Hundman and Chris A Mattmann. Measurement context extraction from text: Discovering opportunities and gaps in earth science. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017.

[85] T. Hao, H. Liu, and C. Weng. Valx: a system for extracting and structuring numeric lab test comparison statements from text. *Methods Inf Med*, 55:266–275, 2016.

[86] S. Roy, T. Vieira, and D. Roth. Reasoning about quantities in natural language. *Transactions of the Association for Computational Linguistics*, 3:1–13, 2015.

[87] C. Taha and W. C. K. I. Identifying the units of measurement in tabular data. 2021.

[88] Vinh Thinh Ho, Koninika Pal, and Gerhard Weikum. Qute: Answering quantity queries from web tables. In *Proceedings of the 2021 International Conference on Management of Data*, SIGMOD '21, page 2740–2744, New York, NY, USA, 2021. Association for Computing Machinery.

[89] V. T. Ho, K. Pal, S. Razniewski, K. Berberich, and G. Weikum. Extracting contextualized quantity facts from web tables. *Proceedings of the Web Conference 2021*, 2021.

[90] Dominika Tkaczyk, Paweł Szostek, Mateusz Fedoryszak, Piotr Jan Dendek, and Łukasz Bolikowski. Cermine: automatic extraction of structured metadata from scientific literature. *International Journal on Document Analysis and Recognition (IJDAR)*, 18:317–335, 2015.

[91] Isaac G Councill, C Lee Giles, and Min-Yen Kan. Parscit: an open-source crf reference string parsing package. In *LREC*, volume 8, pages 661–667, 2008.

[92] Dominika Tkaczyk, Andrew Collins, Paraic Sheridan, and Jöran Beel. Evaluation and comparison of open source bibliographic reference parsers: A business use case. *CoRR*, abs/1802.01168, 2018.

[93] Grobid Astro contributors. Grobid-astro: A machine learning software for extracting astronomical entities from scholarly documents. https://github.com/kermitt2/grobid-astro, 2017–2022.

[94] Mohamed Khemakhem, Luca Foppiano, and Laurent Romary. Automatic Extraction of TEI Structures in Digitized Lexical Resources using Conditional Random Fields. In *electronic lexicography, eLex 2017*, Leiden, Netherlands, September 2017.

[95] GROBID Contributors. software-mentions: Grobid module to recognize in textual documents and pdf any mentions of software. https://github.com/Impactstory/software-mentions, 2018 — 2019. [Online; accessed 18-April-2019].

[96] Phil Gooch and Kris Jack. How well does mendeley's metadata extraction work? https://krisjack.wordpress.com/2015/03/12/how-well-does-mendeleys-metadata-extraction-work/.

[97] Josh M Nicholson, Milo Mordaunt, Patrice Lopez, Ashish Uppala, Domenic Rosati, Neves P Rodrigues, Peter Grabitz, and Sean C Rife. Scite: A smart citation index that displays the context of citations and classifies their intent using deep learning. *Quantitative Science Studies*, 2(3):882–898, 2021.

[98] Nipun Sadvilkar and Mark Neumann. Pysbd: Pragmatic sentence boundary disambiguation, 2020.

[99] Edward Loper and Steven Bird. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, page 63–70, USA, 2002. Association for Computational Linguistics.

[100] Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. Scispacy: Fast and robust models for biomedical natural language processing. 2019.

[101] Dan Gillick. Sentence boundary detection and the problem with the u.s. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, NAACL-Short '09, page 241–244, USA, 2009. Association for Computational Linguistics.

[102] Jonathon Read, Rebecca Dridan, Stephan Oepen, and Lars Jørgen Solberg. Sentence boundary detection: A long solved problem? *Computational Linguistics*, pages 985–994, December 2012.

[103] Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. https://github.com/explosion/spaCy, 2017. To appear.

[104] Beyond Language Understanding Bling Team, Microsoft. Blingfire : A lightning fast finite state machine and regular expression manipulation library. Github: BlingFire sbd repo, 2020.

[105] M. Jiang, Y. Chen, M. Liu, S. T. Rosenbloom, S. Mani, J. C. Denny, and H. Xu. A study of machine-learning-based approaches to extract clinical entities and their assertions from discharge summaries. *Journal of the American Medical Informatics Association*, 18:601–606, 2011.

[106] B. Tang, Y. Wu, M. Jiang, and H. Xu. Recognizing clinical entities in hospital discharge summaries using structural support vector machines with word representation features. *BMC Medical Informatics and Decision Making*, 13, 2013.

[107] T. Isazawa and J. M. Cole. Single model for organic and inorganic chemical named entity recognition in chemdataextractor. *Journal of Chemical Information and Modeling*, 62:1207–1213, 2022.

[108] Patrice Lopez, Caifan Du, Johanna Cohoon, Karthik Ram, and James Howison. *Mining Software Entities in Scientific Literature: Document-Level NER for an Extremely Imbalance and Large-Scale Task*, page 3986–3995. Association for Computing Machinery, New York, NY, USA, 2021.

[109] Thomas Lavergne, Olivier Cappé, and François Yvon. Practical very large scale CRFs. In *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 504–513. Association for Computational Linguistics, July 2010.

[110] DeLFT contributors. Delft. https://github.com/kermitt2/delft, 2018. [Online; accessed 16-May-2019].

[111] Matthew C. Swain and Jacqueline M. Cole. Chemdataextractor: A toolkit for automated extraction of chemical information from the scientific literature. *Journal of Chemical Information and Modeling*, 56(10):1894–1904, 2016. PMID: 27669338.

[112] Masashi Yoshikawa, Hiroshi Noji, and Yuji Matsumoto. A* ccg parsing with a supertag and dependency factored model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 277–287. Association for Computational Linguistics, 2017.

[113] Aryeh Tiktinsky, Yoav Goldberg, and Reut Tsarfaty. pybart: Evidence-based syntactic transformations for ie. In *ACL*, 2020.

[114] Swabha Swayamdipta, Sam Thomson, Chris Dyer, and Noah A. Smith. Frame-Semantic Parsing with Softmax-Margin Segmental RNNs and a Syntactic Scaffold. *arXiv preprint arXiv:1706.09528*, 2017.

[115] Junru Zhou and Hai Zhao. Head-driven phrase structure grammar parsing on Penn treebank. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, July 2019. Association for Computational Linguistics.

[116] Yankai Lin, Suhung Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. Neural relation extraction with selective attention over instances. 2016.

[117] Vishnu Hariharan and M. Anand Kumar. Relation extraction using convolutional neural networks. 2019.

[118] Hiroyuki Oka, Atsushi Yoshizawa, Hiroyuki Shindo, Yuji Matsumoto, and Masashi Ishii. Automatic extraction of solvent-names on polymer's solubility from academic articles. In *The 66th JSAP Spring Meeting 2019*, Tokyo, Japan, March 2019. Japan Society of Applied Physics.

[119] Luca Foppiano, Laurent Romary, Masashi Ishii, and Mikiko Tanifuji. Automatic identification and normalisation of physical measurements in scientific literature. In *Proceedings of the ACM Symposium on Document Engineering 2019*, DocEng '19, New York, NY, USA, 2019. Association for Computing Machinery.

[120] Clarissa F. D. Carneiro, Victor G. S. Queiroz, Thiago C. Moulin, Carlos A. M. Carvalho, Clarissa B. Haas, Danielle Rayêe, David E. Henshall, Evandro A. De-Souza, Felippe E. Amorim, Flávia Z. Boos, Gerson D. Guercio, Igor R. Costa, Karina L. Hajdu, Lieve van Egmond, Martin Modrák, Pedro B. Tan, Richard J. Abdill, Steven J. Burgess, Sylvia F. S. Guerra, Vanessa T. Bortoluzzi, and Olavo B. Amaral. Comparing quality of reporting between preprints and peer-reviewed articles in the biomedical literature. *Research Integrity and Peer Review*, 5(1):16, December 2020.

[121] Klaus Krippendorff. Reliability in Content Analysis: Some Common Misconceptions and Recommendations. *Human Communication Research*, 30(3):411–433, 01 2006.

[122] Antonia Zapf, Stefanie Castell, Lars Morawietz, and André Karch. Measuring inter-rater reliability for nominal data – which coefficients and confidence intervals are appropriate? *BMC Medical Research Methodology*, 16(1):93, August 2016.

[123] Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. The inception platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9. Association for Computational Linguistics, June 2018.

[124] Richard Eckart de Castilho, Éva Mújdricza-Maydt, Seid Muhie Yimam, Silvana Hartmann, Iryna Gurevych, Anette Frank, and Chris Biemann. A web-based tool for the integrated annotation of semantic and syntactic structures. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, pages 76–84, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[125] Karën Fort and Benoît Sagot. Influence of pre-annotation on POS-tagged corpus development. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 56–63, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[126] Aurélie Névéol, Rezarta Islamaj Dogan, and Zhiyong Lu. Semi-automatic semantic annotation of pubmed queries: A study on quality, efficiency, satisfaction. *Journal of biomedical informatics*, 44 2:310–8, 2011.

[127] Todd Lingren, Louise Deleger, Katalin Molnar, Haijun Zhai, Jareen Meinzen-Derr, Megan Kaiser, Laura Stoutenborough, Qi Li, and Imre Solti. Evaluating the impact of pre-annotation on annotation speed and potential bias: natural language processing gold standard development for clinical named entity recognition in clinical trial announcements. *Journal of the American Medical Informatics Association : JAMIA*, 21(3):406–413, 2014.

[128] James Pustejovsky and Amber Stubbs. *Natural Language Annotation for Machine Learning: A guide to corpus-building for applications.* " O'Reilly Media, Inc.", 2012.

[129] Thaer M. Dieb, Masaharu Yoshioka, and Shinjiro Hara. Nadev: An annotated corpus to support information extraction from research papers on nanocrystal devices. *Journal of Information Processing*, 24:554–564, jan 2016.

[130] Martin Krallinger, Obdulia Rabal, Florian Leitner, Miguel Vazquez, David Salgado, Zhiyong Lu, Robert Leaman, Yanan Lu, Dong-Hong Ji, Daniel M. Lowe, Roger A. Sayle, Riza Theresa Batista-Navarro, Rafal Rak, Torsten Huber, Tim Rocktäschel, Sérgio Matos, David Campos, Buzhou Tang, Hua Xu, Tsendsuren Munkhdalai, Keun Ho Ryu, S. V. Ramanan, P. Senthil Nathan, Slavko Zitnik, Marko Bajec, Lutz Weber, Matthias Irmer, Saber A. Akhondi, Jan A. Kors, Shuo Xu, Xin An, Utpal Kumar Sikdar, Asif Ekbal, Masaharu Yoshioka, Thaer M. Dieb, Miji Choi, Karin M. Verspoor, Madian Khabsa, C. Lee Giles, Hongfang Liu, K. E. Ravikumar, Andre Lamurias, Francisco M. Couto, Hong-Jie Dai, Richard Tzong-Han Tsai, Caglar Ata, Tolga Can, Anabel Usie, Rui Alves, Isabel Segura-Bedmar, Paloma Martínez, Julen Oyarzábal, and Alfonso Valencia. The chemdner corpus of chemicals and drugs and its annotation principles. *Journal of Cheminformatics*, 7:S2 – S2, 2015.

[131] Christian M. Meyer, Margot Mieskes, Christian Stab, and Iryna Gurevych. DKPro agreement: An open-source Java library for measuring inter-rater agreement. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations*, pages 105–109, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics.

[132] Skopinava AM and Lobanov BM. Processing of quantitative expressions with units of measurement in scientific texts as applied to belarusian and russian text-to-speech synthesis. 2013.

[133] Units of measurement. https://github.com/unitsofmeasurement.

[134] Luca Foppiano, Akira Suzuki, Thaer M Dieb, Masashi Ishii, and Mikiko Tanifuji. Leveraging segmentation of physical units through a newly open source

corpus. In *JSAP Annual Meetings Extended Abstracts The 80th JSAP Autumn Meeting 2019*, pages 4097–4097. The Japan Society of Applied Physics, 2019.

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

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

[137] 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.

[138] Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. The INCEpTION platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9, Santa Fe, New Mexico, 2018.

[139] 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.

[140] 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.

List of Publications

Refereed Journal Paper

1. Luca Foppiano, Pedro Baptista Castro, Pedro Ortiz Suarez, Kensei Terashima, Yoshihiko Takano & Masashi Ishii (2023) Automatic extraction of materials and properties from superconductors scientific literature, Science and Technology of Advanced Materials: Methods, 3:1, DOI: https://doi.org10.1080/27660400.2022.2153633

2. 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 & Masashi Ishii (2021) SuperMat: construction of a linked annotated dataset from superconductors-related publications, Science and Technology of Advanced Materials: Methods, 1:1, 34-44, DOI: https://doi.org/10.1080/27660400.2021.1918396

3. Luca Foppiano, Tomoya Mato, Kensei Terashima, Pedro Ortiz Suarez, Taku Tou, Chikako Sakai, Wei-Sheng Wang, Toshiyuki Amagasa, Yoshihiko Takano & Masashi Ishii (2023) Semi-automatic staging area for high-quality structured data extraction from scientific literature, Science and Technology of Advanced Materials: Methods, 3:1, DOI: https://doi.org/10.1080/27660400.2023.2286219

Refereed Conference Papers

1. Luca Foppiano, Laurent Romary, Masashi Ishii, and Mikiko Tanifuji. 2019. Automatic Identification and Normalisation of Physical Measurements in Scientific Literature. In Proceedings of the ACM Symposium on Document Engineering 2019 (DocEng '19). Association for Computing Machinery, New York, NY, USA, Article 24, 1–4. https://doi.org/10.1145/3342558.3345411