Data & Knowledge Engineering 163 (2026) 102574

Contents lists available at ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

CCASL: Counterexamples to comparative analysis of scientific literature — Application to polymers

Aymar Tchagoue [a,b,*], Véronique Eglin [a], Sébastien Pruvost [b], Jean-Marc Petit [a], Jannick Duchet-Rumeau [b], Jean-Francois Gerard [b]

[a] INSA Lyon, UCBL, CNRS, LIRIS UMR 5205, Villeurbanne, 69100, France
[b] INSA Lyon, UCBL, Université Jean Monnet, CNRS, IMP UMR 5223, Villeurbanne, 69100, France

ARTICLE INFO

Dataset link: github, zonodo

Keywords:
Functional dependency
Comparative analysis
Tables analysis
Machine learning
Counterexample
Polymers

ABSTRACT

The exponential growth of scientific publications has made the exploration and comparative analysis of scientific literature increasingly complex. For instance, identifying pairs of publications that diverge on widely accepted concepts within a domain is extremely difficult, if not impossible, at a large scale. Our work aims to automatically detect such discrepancies using recent artificial intelligence techniques. Given a particular scientific domain, we propose to capture domain knowledge through the definition of arbitrary functions expressed as relaxed functional dependencies (RFDs), and then focus on the large-scale analysis of tables in the publications related to these RFDs.

In this context, we propose a four-step method called Counterexamples to Comparative Analysis of Scientific Literature (CCASL), which consists of the following steps: (1) Modeling the domain knowledge with functions expressed as RFDs, (2) Acquiring a corpus of related publications, (3) Analyzing all tables in the PDF documents and producing a consolidated table, (4) Detecting counterexamples of the RFDs in the consolidated table and conducting a comparative analysis of the pairs of papers containing the detected counterexamples.

We have applied CCASL to a subfield of polymer research by identifying an arbitrary function relating the storage modulus, the polymer structure, and the glass transition temperature. Based on this function, we implemented the four steps of CCASL for large-scale bibliographic confrontation in polymer science, which enabled us to detect several counterexamples. After detailed analysis, these counterexamples were found to originate from two main sources: typographical errors and methodological inconsistencies. The latter led to an update of the initial arbitrary function, specifying that it is valid only for fully reacted mixtures.

1. Introduction

The exponential growth of scientific publications has made the exploration and comparative analysis of scientific literature increasingly complex and challenging. In this continuously evolving scientific landscape, knowledge quickly becomes obsolete, resulting in publications that may appear to conflict with widely accepted concepts or notions. Detecting counterexamples in scientific publications is therefore crucial for validating existing knowledge, synthesizing dispersed information, uncovering exceptions that may inspire new hypotheses, and ultimately supporting both reproducibility and the standardization of scientific

* Corresponding author at: INSA Lyon, UCBL, CNRS, LIRIS UMR 5205, Villeurbanne, 69100, France.
E-mail address: aymar-lebeau.tchagoue-tchagoue@insa-lyon.fr (A. Tchagoue).

https://doi.org/10.1016/j.datak.2026.102574
Received 13 June 2025; Received in revised form 2 February 2026; Accepted 3 February 2026
Available online 4 February 2026

Table 1
Introduction example.

Index	Flow	Head	Power
a	2.5	10.1	22
b	2.5	10.1	25
...	...	...	...
c	3.00	15.00	35
d	3.01	15.01	30

research. Moreover, the increasing accessibility of Natural Language Processing (NLP) and other AI techniques now enables the processing of large volumes of text, making it possible to conduct large-scale knowledge synthesis [1] and meta-analysis [2].

In this context, one of the unique aspects of our study is its focus on the large-scale analysis of tables present in PDF files. Tables represent valuable sources of structured and numerical information [3] and often contain synthesized knowledge that cannot be effectively represented in paragraph form due to the presence of multiple instances, such as polymer characterization data (see example in Fig. 3). This structured information, scattered throughout the scientific literature, encodes a wealth of underlying knowledge that is not explicitly conveyed through textual semantics.

To capture such structured domain knowledge, we rely on the notion of functional dependencies (FDs), introduced in the 1970s within the database community [4] as a fundamental constraint of the relational model. Over the years, FDs have proven essential to database design and data cleaning, while also establishing strong connections with other areas of computer science, including Galois lattices [5] and probabilistic reasoning [6].

Conceptually, as their name suggests, FDs express the *existence of a function* from a domain to a co-domain: for any given value in the domain, there exists exactly one corresponding value in the co-domain. Importantly, this concept does not require explicit knowledge of the function itself, but rather only the awareness that such a functional relationship exists. In database terms, the notions of domain and co-domain are captured through a set of *attributes* on the left-hand side and right-hand side of the FD, respectively. Consequently, given a specific vocabulary represented by a set of attributes, FDs offer a natural way to encode domain knowledge as high-level functions, thereby capturing relationships that are otherwise scattered across the literature.

Example 1. In Table 1, we present a relation with three attributes representing measurements from a run-of-river hydroelectric turbine: the generated power (kW), the corresponding flow (m^3/s), and the head (m). Assume the following FD is defined to express domain knowledge known by experts: $flow, head \rightarrow power$, i.e., the power is a function of the flow and the head. Even if we do not know the function itself, this FD allows us to express its existence. For a given pair of values of $flow$ and $head$, if two rows share the same values, then they must have the very same value for $power$. Otherwise, we have so-called counterexamples of the FD in the relation (e.g., the pair of tuples (a,b) constitutes counterexamples).

Moreover, FDs have been generalized to handle approximate cases in order to account for dirty or imprecise data. In this context, strict equality can be relaxed through predicates, leading to relaxed FDs (RFDs). For instance, measurement precision or uncertainties on attributes such as "flow" and "head" can be taken into account; in this case, the pair of tuples (c,d) would also constitute a counterexample if 3.00 and 3.01 (and 15.00 and 15.01) are close enough to be considered as equal.

This simple example illustrates how FDs, and their relaxed forms, can be used to encode domain knowledge and detect inconsistencies in structured data. Building on this principle, our work extends the use of RFDs beyond traditional databases to the large-scale analysis of scientific literature, where tables can be interpreted as fragments of domain knowledge dispersed across multiple documents. To the best of our knowledge, there is currently no systematic, data-driven approach for assessing the consistency of published experimental tables against established domain knowledge. Therefore, our objective is to automatically identify contradictions among these fragments by expressing domain knowledge through relaxed functional dependencies (RFDs).

Problem statement. Given a specific scientific domain, a collection of related PDF documents, and a set of domain knowledge modeled as functions and expressed as RFDs, how can we automatically identify pairs of documents that contradict this domain knowledge? To tackle this question, three interconnected problems must be addressed. First, *modeling domain knowledge* remains a long-standing issue in AI. Scientific knowledge is often fragmented across multiple sources [7], and the main challenge lies in developing accurate models that integrate these diverse and evolving pieces of information. Second, *analyzing large-scale scientific literature* is increasingly feasible thanks to recent advances in AI [8,9], yet efficiently extracting relevant data from vast and heterogeneous document collections remains a difficult task. Finally, *conducting precise bibliographic confrontation* poses significant difficulties due to inconsistent data reporting, varying terminology, and the sheer volume of available publications [10]. By addressing these three challenges together, this paper aims to contribute to the development of tools and methodologies that enhance researchers' ability to navigate, understand, and leverage the wealth of information contained in scientific literature.

Contributions. The primary contribution of this work is the *CCASL method*, illustrated in Fig. 1, which provides a comprehensive solution to the challenges of domain knowledge modeling, large-scale literature analysis, and bibliographic confrontation through counterexample detection. To achieve this, our contribution is structured around three main components. First, *domain knowledge modeling with arbitrary functions.* Unlike ontology-based approaches [11], we propose modeling domain knowledge using arbitrary functions expressed as RFDs. This formulation enables researchers to formalize and structure complex scientific knowledge, thereby

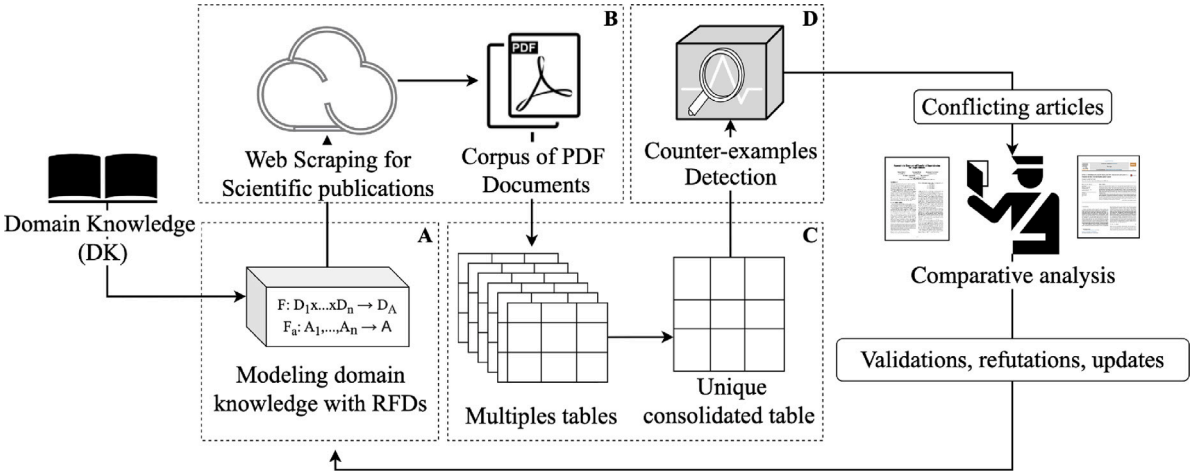

Fig. 1. Overview of CCASL.

improving the understanding of domain-specific relationships. Second, we focus on *consolidating tables in the selected PDF documents*, developing a dedicated pipeline that combines AI-based methods [12,13] with rule-based approaches [14] to extract and merge relevant tables from scientific publications. The vast heterogeneity of table layouts and attribute names in PDF documents represents a major challenge for both extraction and consolidation. To address this, we normalize the extracted tables and align their attributes with those used to express domain knowledge as RFDs. Specifically, we design a robust rule-based and human-in-the-loop process to analyze tables with irregular or multi-level headers, thereby preventing potential error propagation during normalization. Finally, we address *counterexample detection and comparative analysis*, a process designed to uncover inconsistencies within scientific literature based on domain knowledge expressed as RFDs. Specifically, when two rows in the consolidated table agree on the left-hand side of an RFD but differ on the right-hand side, they form a counterexample, indicating a contradiction between the corresponding publications.

Application to polymers. CCASL is not limited to a specific application domain, as it is based on fundamental, domain-independent principles. However, applying the method requires data extraction processes that are domain-specific. We illustrate its application in the specific context of polymer science. We focus on a specific sub-field of polymers called Epoxy-Amine networks (EA) [15]. These materials are commonly used in the composites industry, particularly in aeronautics, transportation, and energy. To the best of our knowledge, there is no structured, open-source database at a national or international level on EA structures and their physicochemical properties. This lack of resources significantly hinders the advancement of data science in this field, leaving many theories based on intuition and experience yet to be rigorously studied. Given that publications related to EAs continuously introduce new materials (design-characterization), the literature is becoming an important source of data. In collaboration with materials scientists, we use CCASL to model EA domain knowledge and identify contradictory publications with respect to specific functions, enabling relevant comparative analyses.

Paper organization. The rest of the paper is organized as follows: Section 2 clarifies the notations used throughout the paper; Section 3 presents the CCASL method, as illustrated in Fig. 1, including (A) modeling domain knowledge with RFDs, (B) web-scraping for PDF documents acquisition, (C) consolidating tables in the selected PDF documents, and (D) counterexamples detection and comparative analysis. Section 4 applies the CCASL method to the polymers scientific domain. Section 5 reviews related work by comparing CCASL to other approaches regarding relaxed FDs, meta-analysis, information extraction, and schema matching. Section 6 discusses the results, their limitations, and future research perspectives. Section 7 concludes the study by summarizing the main contributions, and Data and code availability provides details on accessing the data and software developed for this study.

2. Preliminaries

In this section, we provide the notations used in CCASL. It is assumed that the reader is familiar with database notations [16]. Let $\mathcal{U}_{att}$ be a set of attributes, $\mathcal{D}$ a domain, and $\mathcal{U}_{unit}$ a set of reference units such that: $\mathcal{U}_{att} \cap \mathcal{U}_{unit} = \emptyset$, $\mathcal{U}_{att} \cap \mathcal{D} = \emptyset$, and $\mathcal{U}_{unit} \cap \mathcal{D} = \emptyset$.

Example 2. In line with our example, we say that an attribute (e.g. flow, head and power) is defined over a domain (e.g. real numbers) with a reference unit (e.g. m^3/s, m and kW).

A relation schema R is defined as a nonempty subset of $\mathcal{U}_{att}$. A tuple t over R is a function from R to $\mathcal{D}$. A relation r over R is a finite set of tuples over R. Let r be a relation over R and $X \subseteq R$. $t[X]$ is the restriction of t to X. The projection of r over X, denoted $\pi_X(r)$, is defined as usually $\pi_X(r) = \{t[X] | t \in r\}$. We adopt a set semantics for relational algebra expressions. The size of a finite set E is denoted by $|E|$.

Functional dependencies. Let R be a relation schema, $X \subseteq R$, $A \in R$, and r a relation over R. A functional dependency over R is denoted by $R : X \rightarrow A$. Its semantic is defined as usually: $R : X \rightarrow A$ is satisfied in r, denoted by $r \vDash X \rightarrow A$, if and only if:

$\forall t_1, t_2 \in r$, if $\forall B \in X : t_1[B] = t_2[B]$ then $t_1[A] = t_2[A]$.

A classical measure to approximate the satisfaction of FDs is known as the g_3 measure [17]. It corresponds to the smallest proportion of tuples to be removed form the relation r for $X \rightarrow Y$ to hold in r. More formally, we have:

$$g_3(r, X \rightarrow A) = 1 - \frac{\max(|r'| \mid r' \subseteq r, r' \vDash X \rightarrow A)}{|r|} \tag{1}$$

Example 3. Continuing the previous example, suppose the relation in Table 1 contains a total of n tuples ($n = |r|$). If we consider that the pair (a, b) is the only counterexample to the FD, then the FD would be satisfied by removing one of the tuples in this pair. Consequently, the g_3 measure takes the value:

$$g_3(r, X \rightarrow A) = 1 - \frac{n-1}{n} = \frac{1}{n}$$

Predicates to soften equality. A classic approach to relaxing FD is to replace equality with predicates, as proposed in [18–20]. We suppose to have a set of predicates Φ, one for each attribute in R. Let us consider $A \in R$ such that the domain of A is the real number. An example of predicate Φ_A is as follows: Given two real values u, v:

$$\Phi_A(u, v) = \begin{cases} TRUE \ if \ |u - v| \ \leq \ \epsilon \\ FALSE \ otherwise \end{cases} \tag{2}$$

where ϵ is a threshold for Φ_A. The satisfaction with predicates is defined accordingly: let (R, Φ) be a relation schema with predicates, r a relation over R and $X \rightarrow A$ a FD over R. The relation r satisfies $X \rightarrow A$ with respect to Φ, denoted by $r \vDash_{\Phi} X \rightarrow A$, if for every pair of tuples (t_1, t_2) of r, the following formulas holds [21]:

$$\bigwedge_{B \in X} \Phi_B(t_1[B], t_2[B]) \implies \Phi_A(t_1[A], t_2[A])$$

In the sequel, we refer to such FDs as "Relaxed FDs". A counterexample of the RFD $X \rightarrow A$ in r, is a couple of tuples $(t_1, t_2) \in r$ such that $\Phi_B(t_1[B], t_2[B])$ is true and $\Phi_A(t_1[A], t_2[A])$ is false.

The g_3-error of an RFD $X \rightarrow A$ with respect to r, extends as follows:

$$g_3^{\Phi}(r, X \rightarrow A) = 1 - \frac{\max(|r'| \mid r' \subseteq r, r' \vDash_{\Phi} X \rightarrow A)}{|r|} \tag{3}$$

Its implementation is available with the open-source code $fast\ g_3$ [22].

Example 4. Continuing the previous example, consider the RFD with the previously defined predicate on all attributes and $\epsilon = 0.1$. In this case, in addition to the pair (a, b), the pair (c, d) is also conflicting, since the "flow" values ($|3.00 - 3.01| \leq 0.1$) and the "head" values ($|15.00 - 15.01| \leq 0.1$) are considered equal, while the "power" values ($|35 - 30| \not\leq 0.1$) differ. The RFD would be satisfied by removing one tuple from each conflicting pair. Consequently, since there are two counterexamples, the g_3^{Φ} measure takes the value:

$$g_3^{\Phi}(r, X \rightarrow A) = 1 - \frac{n-2}{n} = \frac{2}{n}$$

3. CCASL method

As indicated in Fig. 1, the CCASL method can be broken down into four main steps, which are: (A) Modeling domain knowledge with RFDs, (B) web-scraping for PDF documents acquisition, (C) consolidating tables in the selected PDF documents, and (D) counterexamples detection and comparative analysis.

3.1. Modeling domain knowledge with RFDs

For a given domain, we propose modeling domain knowledge (DK) as arbitrary functions, formulated by domain experts based on their knowledge, experience, and intuition. The idea is to let them express their DK as functions over a domain and a co-domain. Usually, this is done with a specific vocabulary defining variables.

Example 5. In line with the previous example, in fluid mechanics, `flow`, `head`, and `power` constitute a (simplified) specific vocabulary, from which variables can be easily defined and used for example to express RFDs.

Whenever DK is expressed as a function, it turns out that a FD can be deduced. In database terms, the domain and co-domain of a function can be represented as a cartesian product of some domains. Let $A_1, \ldots, A_n$ and A be some attributes defined respectively over the domains $D_1, \ldots, D_n$ and D_A. Moreover, each attribute A_i is associated with a reference unit U_i, for instance, the $flow$ is measured in ($m^3\ s^{-1}$).

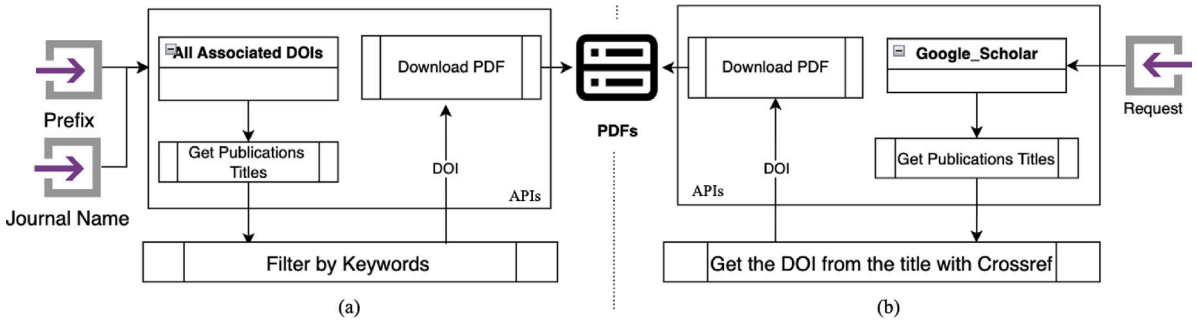

Fig. 2. PDF Corpus Acquisition with two techniques.

We denote f as an arbitrary function declared as follows: $f : D_1 \times \cdots \times D_n \rightarrow D_A$, where $D_1 \times \cdots \times D_n$ is the domain and D_A is the co-domain such that for any value in the domain, there is a corresponding unique value in the co-domain by f. The associated FD to f, denoted by f_d, is easily defined as follows:

$$f_d : A_1, \ldots, A_n \rightarrow A$$

We denote by $\mathcal{U}_{att} = \{A_1, \ldots, A_n, A\}$ as a set of all attributes, and by $\mathcal{U}_{unit} = \{u_1, \ldots, u_n, u\}$ as the set of the corresponding reference units.

Example 6. Continuing the previous example, the power function $f_p : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ can be modeled using three attributes: power, flow, and head, all defined over the domain $\mathbb{R}$. Using the predicate in Eq. (2), we can express the RFD as follows:

$$f_{pd} : flow, head \rightarrow power$$

3.2. Web-scraping for PDF documents acquisition

We now have to identify some scientific domain literature associated with the specific vocabulary identified to define the function. In this context, the main difficulty is to acquire relevant scientific documents, that is, papers containing the modeled knowledge. We use two complementary techniques of web-scraping in the document extraction pipeline.

The first technique focuses uniquely on targeted scientific journals that predominantly deal with the considered scientific domain. This helps reduce noise by retrieving documents from a specific set of publications. The process is illustrated in Fig. 2a as follows: (1) First, using the name and prefix of a journal, we target and retrieve its list of Digital Object Identifiers (DOI). (2) Second, by searching the DOIs in cross-ref libraries we obtain the titles of the publications. (3) Third, using a list of keywords derived from the attributes $\mathcal{U}_{att}$, we filter and download the publications whose titles contain at least one of these keywords.

The second technique leverages the Google Scholar search engine, which aggregates data from scientific publishers' directories, academic databases, and university websites. It ranks documents based on several parameters, such as the *full text*, *where* they were published, *who* wrote them, how *often* and how *recently* they have been *cited* in other scholarly literature [23]. The workflow for this technique is illustrated in Fig. 2b as follows: (1) First, a query is made on Google Scholar using the keywords composed of elements derived from the attributes $\mathcal{U}_{att}$. (2) Second, the result of the query, which is a list of article titles, is retrieved. (3) Third, these titles allow us to find the DOIs of the publications via Cross-ref, facilitating their download.

Unlike the previous technique, this one retrieves documents that are relevant not only based on their titles but also on their content. However, given the wide variety of sources, the resulting documents are likely to exhibit more diverse content. At the end we merge the results of the two techniques.

3.3. Consolidating tables in the selected PDF documents

In scientific publications, it is common practice to record large amounts of results in tables for practical reasons. This observation leads us to focus on the tables found in these publications. Fig. 3 presents an example of such a table in materials science.

3.3.1. Tables extraction from a PDF file

Image analysis is one of the most commonly used methods for extracting tables from documents, particularly when tables have heterogeneous formats. For instance, some tables include borders, while others do not, and there are significant differences in layout, such as varying levels of hierarchical attributes. Furthermore, each author tends to apply their own unique formatting style, which adds to the complexity of the extraction process.

To address this issue, we have developed a table extraction approach based on [12,13,25], summarized in Algorithm 1, and described in the following five stages. (1) To standardize the analysis, each page of a PDF file is first converted into an image I. (2)

Table 2. Storage modulus, glass transition temperature (T_g) and flexural properties of epoxy glass-fiber composites.

Sample	Storage Modulus at 40 °C (GPa)	Storage Modulus at 200 °C (GPa)	T_g (°C)	Flexural Modulus at 25 °C (GPa)	Flexural Strength at 25 °C (MPa)	Strain at Break (%)
RT + 70 + 140	13.8 ± 0.1	3.8 ± 0.3	165	20.6 ± 1.4	354 ± 17.7	1.9 ± 0.1
70 + 140	13.4 ± 0.1	3.7 ± 0.2	164	20.1 ± 0.6	350.4 ± 35.1	2 ± 0.2
140	12.5 ± 0.3	3.6 ± 0.4	165	20.2 ± 1.2	343.1 ± 41.0	2.1 ± 0.2

Fig. 3. Example of a table in a materials science (Polymers) publication [24].

Building on this, the LayoutParser [12] model is employed to detect and extract tables within these images. This open-source library can be applied without additional training on scientific publication pages, as the model is pre-trained on five datasets, including over one million scientific articles from the PubLayNet dataset [26]. Once applied, the model outputs the coordinates of the borders, and each table is extracted as a sub-image S. To assess its performance in our context, we evaluated the model on 20 PDFs containing 51 tables, of which 47 were correctly identified, yielding an accuracy of 92.2%. (3) Following table extraction, a corresponding bag of words for each table image S is obtained using Pytesseract [25] Optical Character Recognition (OCR). (4) Given that scientific publications may contain multiple tables, a preliminary keyword-based classification is then applied to retain only the tables whose bag of words contains at least one term referring to one of the attributes $\mathcal{U}_{att}$. (5) The process concludes with the conversion of the relevant table images into CSV format using AWSTextract [13], which exhibits a low median error rate of 17.3% [27] on a historical dataset of 180,000 pages from Old Bailey [28].

Algorithm 1 Table Extraction from PDFs with Keyword Filtering

Input:
A PDF file P with n pages
$list\_key = \mathcal{U}_{att}$ plus additional terms from the domain expert
Output: A set of CSV files $\{C_1, C_2, ..., C_k\}$
for $p \in P$ **do**
  $I = convert\_to\_image(p)$
  $Bounding\_box = LayoutParser(I)$
  for $b \in Bounding\_box$ **do**
    Extract the sub-image S corresponding to b
    $bag\_of\_words = pytesseract(S)$
    for $l \in$ list_key **do**
      if $l \in bag\_of\_words$ **then**
        $C = AWSTextract(S)$
        Save the CSV file C
      end if
    end for
  end for
end for

3.3.2. Table cleaning

Headers in CSV files can be irregular and multilevel, posing several challenges such as column name identification, different units for the same information (for example, temperature may be recorded in Kelvin K or Celsius °C), and the presence of domain-specific metadata.

Part of the techniques to be used are domain-specific and may require assistance of a domain expert, such as providing a set of relevant synonyms for the attributes. In the sequel, we briefly introduce five functions used in Algorithm 2, which contains the main steps needed to handle the variety of headers that may occur in CSV files.

The first function, **"Keyword_analysis"**, performs keyword detection at several scales in our pipeline, considering a detection valid when the semantic similarity using SpaCy [29] or the character-based proximity with Levenshtein [30] between a keyword and the target text exceeds a 90% threshold. This function is used in the functions *"findHeader"*, *"decompose_ColumnName"*, *"attributes_schema_matching"*, and *"Standardize_DataValues"* to search for keywords, disambiguate word variations, correct OCR errors, and capture synonyms.

The second function, **"findHeader"** (Algo 2, line 6), is a rule-based method designed to reconstruct irregular and multi-level headers into a standardized table header consisting of a single, unique row. This function intervenes in two cases: (1) when the

Algorithm 2 Sketch of the normalization process

1: **Input:**
2: A list C_{su} of CSV files
3: A set of attributes $\mathcal{U}_{att}$ and corresponding reference units $\mathcal{U}_{unit}$
4: **Output:** A list C_{su}_n of normalized CSV files
5: **for** each $file$ in C_{su} **do**
6: $file = findHeader(file)$
7: **for** each column name col_name in $file$ **do**
8: $A_d, u_d, m_d = decompose\_ColumnName(col\_name)$
9: $A, u = attributes\_schema\_matching(col\_name, A_d, \mathcal{U}_{att}, \mathcal{U}_{unit})$
10: $file = file.rename(columns = \{col\_name : A\})$
11: $file, outliers = Standardize\_DataValues(file, A, u, u_d, m_d)$
12: **end for**
13: **end for**

header contains hierarchical and/or merged attributes (Fig. 6b), which often causes issues during OCR reconstruction, as some column names may get split across different columns, leading to errors; and (2) when headers are transposed vertically or include both vertical and horizontal headers, which is detected and corrected to create a standardized format with only one single horizontal header. This rule-based approach is designed to recognize multiple patterns that appear in tables. For example, to identify the header, we identify all the rows and columns that have the highest ratio of alphabetic/numeric strings. Combining this information with the presence or absence of measurement units allows us to find the header.

The third function, **"decompose_ColumnName"** (Algo 2, line 8) addresses the issue that, once the headers of the obtained CSV files are identified and reorganized, different column names with various units may refer to the same attribute. We propose, for a CSV file C, to model each column name "col_name" as a triplet (A_d, u_d, m_d), representing, respectively, the detected attribute, unit, and metadata. Moreover for a given A_d, we assume that there exists a corresponding matching reference attribute ($A \in \mathcal{U}_{att}$) and an associated unit ($u \in \mathcal{U}_{unit}$).

Example 7. Let us suppose there are the following two column names in the extracted CSV files: "flow rate at ($m^3\ s^{-1}$)" and "stream $\times 10^6$ ($L\ s^{-1}$)". Their associated triplet would respectively be: (flow rate, $m^3\ s^{-1}$, 1) and (stream, $L\ s^{-1}$, 10^6). In this case, their corresponding reference attribute and unit are: $A = flow$ and $u = m^3\ s^{-1}$, and the metadata is a multiplicative factor.

The identification of these elements is crucial for schema matching [31] and for subsequently converting the column values to their reference units. The designed function "$decompose\_ColumnName$" identifies the elements of the triplet (A_d, u_d, m_d) by using rule-based methods and a set of expected values. For A_d, we use keyword-based detection to find, in the column names, the group of 1, 2, or 3 words that best matches the attributes in $\mathcal{U}_{att}$. We detect the unit u_d and the metadata m_d by analyzing regular expressions. Moreover, the metadata depends on the case study; in materials science applications, it usually corresponds to experimental conditions mentioned in the column name, which can be detected using keywords, as illustrated by: "Modulus **at 12°C**" or "Q **x10**^{6} ($L.s^{-1}$)". We also use ChemDataExtractor [32] to extract the meaning of specific symbols and abbreviations used in the column name and present in the PDF text (e.g., "Q" is the abbreviation for flow).

The fourth function, **"attributes_schema_matching"** (Algo 2, line 9), has been designed using two approaches. The first one, a *Rule-based approach*, consists of replacing the column name "col_name" in the CSV file with the corresponding reference attribute $A \in \mathcal{U}_{att}$. The result of this attribute normalization is saved in a file, which is manually corrected by the expert to fine-tune the results.

Example 8. In the toy example, "flow rate", and "stream" will be replaced by "flow" in their tables, as they are synonyms.

The second one, a *Machine Learning-based approach*, consists of using the verified data obtained from the rule-based approach to create a Recurrent Neural Network (RNN) model that normalizes column names. This model takes as input the list of column names from each table and outputs the corresponding normalized attributes. However, due to the limited amount of representative training data, the model initially have lower accuracy than the rule-based approach. For this reason, we propose to combine both approaches using the rule-based approach with an expert in the loop until sufficient data is acquired to enable reliable machine learning generalization. The RNN encoding is performed as follows: the attributes are encoded as vectors of 50 dimensions, where each dimension can take a value between 0 and 165. The vector length is determined by the fact that the longest attribute contains 50 characters, meaning 50 positions. The 166 possible values correspond to all the distinct characters used to write the attributes (the alphabet, numbers, exponents, space, punctuation, etc.).

The fifth function, **"Standardize_DataValues"** (Algo 2, line 11), addresses the normalization of data values. For each column, we first detect the data in the cells, which can be noisy due to extra information such as notes in indices or exponents, uncertainties, or descriptive annotations (see example in Fig. 6). We then convert these values to the reference unit u using the detected unit u_d and its multiplication factor m_d. Furthermore, to incorporate human oversight, we flag outlier data using a quantile-based outlier detection [33].

Algorithm 2 presents an overview of our data normalization process. Based on the five functions defined above, each CSV file first undergoes header reconstruction using the "findHeader" function, which standardizes irregular or multi-level headers to accurately identify column names. Each column name is then decomposed into its detected attribute, unit, and metadata using the "decompose_ColumnName" function. This triplet of information is subsequently used for schema matching with the reference attributes and their corresponding units through the "attributes_schema_matching" function. Once the column names have been aligned with the reference attributes across the different CSV files, the "Standardize_DataValues" function is applied to normalize the data values by adjusting their scales, including unit conversions. Finally, an outlier analysis is performed to detect potentially erroneous or suspicious values, for instance, cases where notations such as "10[3]" might be misread as "10131". This overall method has significantly improved the normalization process, and we further discuss its performance in Section 4.3.2.

3.3.3. Building a unique consolidated table from the discovered tables

The overall approach to create a unique consolidated table is presented in Algorithm 3. The integration process consists of successively merging the normalized CSV files into a single consolidated table D_s, defined over the reference schema $\mathcal{U}_{att}$. For each normalized CSV file C with its set of attributes C_{att}, we identify the subset $C'_{att} = C_{att} \cap \mathcal{U}_{att}$ that corresponds to the attributes shared with the reference schema. The corresponding tuples C_{corr}, restricted to these common attributes, are then extracted and appended to D_s.

It is important to clarify that this step depends on the specific case study, as additional rules based on metadata and domain knowledge have to be defined. We provide specific details in the application to Epoxy-Amine networks in Section 4.3.

Algorithm 3 Integration : creation of a unique consolidated table

Input:
set $\mathcal{U}_{att}$ of Attributes.
A list C_{sv} of normalized CSV files.
Output: A unique table Ds.
Create Ds over $\mathcal{U}_{att}$.
$Ds = \emptyset$.
for each table $C \in C_{sv}$ **do**
  C_{att} = set of attributes of C
  $C'_{att} = C_{att} \bigcap \mathcal{U}_{att}$ # set of attributes common to C_{att} and $\mathcal{U}_{att}$.
  C_{corr} = tuples from C restricted to C'_{att} # corresponding data.
  if $C_{corr} \neq \emptyset$ **then**
    $Ds = Ds \bigcup C_{corr}$ # merging Ds with the corresponding data.
  end if
end for

3.4. Counterexamples detection and comparative analysis

We have obtained so far a unique consolidated table D_s. Now we are ready to exhibit counterexample of the function previously identified. The analysis of these counterexamples is carried out through a comparative process, which may reveal errors in the publications, such as typographical, methodological, or calculation errors. Alternatively, the process may lead to an update of the initial function f to better capture domain knowledge.

Example 9. Continuing our running example with $f_{p_d} : flow, head \rightarrow power$, let us consider the unique consolidated table D given in Table 2. We suppose that each attribute has a predicate defined in Eq. (2) with a threshold $\epsilon = 0.1$. The following conclusions can be made:

- Clearly, the pair of tuples highlighted in Table 2 with index: (3, 513), contradicts f_{p_d}.
- There exists at least one erroneous tuple in the conflicting pair. Thus, removing either 3 or 513 from D is sufficient for f_{p_d} to be satisfied.
- Assuming that only one value needs to be eliminated to satisfy f_{p_d}, the value of g_3 is then $\frac{1}{513}$ i.e 0.2% (see Eq. (3)). The data scientist may conclude that the domain knowledge remains valid as $g_3^{\Phi}(D, X \rightarrow A) \simeq 0$.
- A comparative analysis between the scientific documents, identified by their Ref: 423374_20_3 and 436737_13_0, is necessary to understand why they contradict each other. Many reasons may be given, such as: one tuple is erroneous; there is a measurement error; or it is a typographical error.
- The threshold ϵ can be adjusted by the domain expert. As an illustration, in this toy example, the expert can explain the counterexample as being due to temporary perturbations in the turbine caused by obstacles and decide to raise the threshold. In this case, if $\epsilon = 0.2$, then one additional conflicting tuple pair, (2, 3), appears.
- If too many counterexamples appear and new domain knowledge insights arise, f_{p_d} can be updated or the threshold redefined.

This illustrative example demonstrates how CCASL verifies the truthfulness of an arbitrary function f_p while proposing counterexample documents for comparative analysis.

Table 2
Example of a unique consolidated table D for f_{pd}.

Index	Flow	Head	Power	Ref
1	2.5	10.1	22.8	234653_13_2
2	2.8	10.4	23.2	236737_43_0
3	**2.6**	**10.4**	**24.0**	423374_20_3
...	...	...	...	...
511	3.0	10.2	23.3	234082_12_0
512	2.6	9.1	23.1	135114_11_2
513	**2.5**	**10.3**	**22.9**	436737_13_0

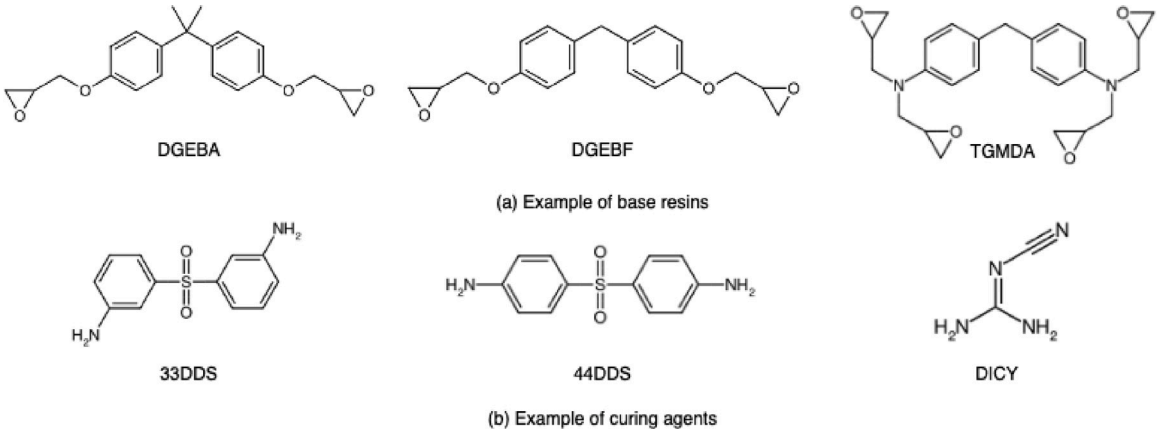

Fig. 4. Example of epoxy and amine prepolymers.

4. Application to epoxy-amine networks

The CCASL method has been applied within the framework of an ongoing project called AMETHYST [34], at the crossroads of computer science and materials science. In the following, we focus on **Epoxy-Amine networks (EA)**, which are crucial materials used in structural composites for aeronautics, automobiles, and wind turbines. They are synthesized through the polymerization of **epoxy** prepolymer (Fig. 4.a) with a hardener (Fig. 4.b), which can be an anhydride or, more commonly, an **amine**.

4.1. EA domain knowledge modeling with arbitrary functions

From discussions with domain experts, we highlight a new arbitrary function related to the structural composition of epoxy-amine (EA) networks, their glass transition temperature (T_g), and their storage modulus (SM). Let $\mathbf{V_{EA}}$ denote the structural composition of the EA network. This relationship suggests the existence of a function expressed as $T_g = f_g(V_{EA}, SM)$, linking the network structure to its thermal and mechanical properties. Below, we outline the main concept to precisely define this function.

EA sample nomenclature. We have designed an EA sample nomenclature, denoted as V_{EA}, detailing the sample composition. This vector highlights each component C_i and its weight p_i within the sample. For an EA with n components V_{EA} is expressed as follows:

$$V_{EA} = [C_1(p_1),\ C_2(p_2),\ ...C_n(p_n)] \tag{4}$$

Example 10. For example, an Epoxy-Amine network (EA_1) could be represented as:

$$V_{EA_1} = [DGEBA(72,3\%),\ 33DDS(25,7\%),\ Clay(2\%)]$$

Where, $DGEBA$ refers to *Bisphenol A diglycidyl ether* and $33DDS$ to *3,3' diaminodiphenylsulfone*. Both are the main EA matrix components [35], and $Clay$ is the nanocomposite with a weight percentage of 2%. The weight (p_i) can be expressed as a percentage or as the exact value of the weight of the component.

Glass transition temperature (tg). The T_g is the temperature at which an amorphous polymer transitions from a hard glassy state to a soft rubbery state, or vice versa.

Storage modulus (SM). The SM is a mechanical property obtained through the Dynamic Mechanical Analysis (DMA) characterization method [36]. An example is shown in Fig. 5, where SM is represented as a function of the temperature (t) [37].

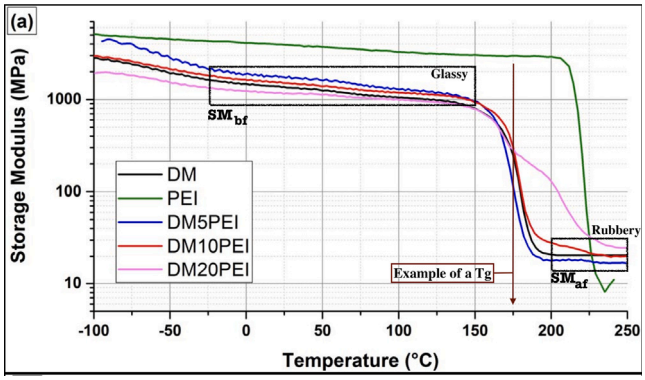

Fig. 5. Storage Modulus E' of Sample : DM, PEI, DM5PEI, DM10PEI and DM20PEI [37].

Table 3
Number of PDFs downloads.

Technique-1	Technique-2	Total
3400	682	4055
Common to both		27

Table 4
Confusion matrix.

	Relevant	Irrelevant
Relevant	$TP = 20$	$FP = 17$
Irrelevant	$FN = 2$	$TN = 170$

Modeling domain knowledge with a function. As a first attempt, we may sketch a function from V_{EA}, SM to T_g. However, from domain knowledge, it appears that the storage modulus could be decomposed into two parts: before and after the glass transition temperature. We denote by SM_{bf} and SM_{af} the storage modulus before and after T_g, respectively. The set of relevant attributes is $\{V_{EA}, SM_{bf}, SM_{af}, T_g\}$. Their domains are text for V_{EA} and $\mathbb{R}$ for the others. Through one month of intensive collaboration, we formulated this domain knowledge into a function, expressed as the following relaxed functional dependency:

$$f_{EA} : V_{EA}, SM_{bf}, SM_{af} \rightarrow T_g \tag{5}$$

4.2. Web-scraping for EA PDF documents acquisition

We have applied the two complementary techniques presented in Section 3.2. Firstly, in *technique-a* (Fig. 2), we targeted 11 polymer journals containing a total of 158 531 DOIs. We filtered them using the keywords: "epoxy, resin, amine, storage modulus, and T_g". As a result, we downloaded a total of **3 400** PDF documents of relevant scientific articles. Secondly, in *technique-b* (Fig. 2), we used the link generated from the search query "Epoxy amine, T_g, Storage modulus" on Google Scholar. Since Google Scholar restricts access to no more than 1000 results per query, we were able to obtain **682** PDFs, covering 68.2% of the expected total. The results obtained by both techniques are then merged, giving a total of 4055 PDF, a summary is given in Table 3.

4.3. Building a unique consolidated table from EA tables occurring in the selected EA papers

4.3.1. EA tables extraction from PDF files

We have applied Algorithm 1, to detect, extract, and filter the relevant tables present in the 4055 documents. Based on domain knowledge, we defined the following keywords used in the algorithm: "Sample, T_g, Storage, Modulus, Tensile, E', G', Rubbery, Glassy, MPa, GPa, °C, °K". The performance of this method was verify on a manually classified subset of 209 table images. Its accuracy, calculated as $\frac{TP+TN}{\text{Total}}$, and its recall, calculated as $\frac{TP}{TP+FN}$, are both 90.90%. The associated confusion matrix is provided below in Table 4.

From the **4055** PDFs analyzed, a total of **5000** table images were identified. Among these, the classification process retained **1400** tables, representing **28%** of the total. We then processed these images with AWSTextract [13] to obtain their CSV versions.

(a)

Composition Epoxy-Amine	Tg(°C)	E'glassy (Pa)	E'rubbery (Pa)
DGEBA-33DDS	117	$1.3.10^8$	$1.70.10^7$
DGEBA-DETA	100	$1.6.10^9$	$1.28.10^7$

(b)

Structures	Tg (°C)	Storage modulus (GPa)	
		20°C	Tg+50°C
DGEBF/DICY	274	1.9	0.01
TGMA/33DDS	276	2.21	0.02

(c.1)

Samples	Storage modulus (300°C)(MPa)	Storage modulus (30°C)(MPa)
DBDBBB/33DDS	40± 4	2740±40
DBDBBB/44DDS	41±3.3	2945±30

(c.2)

Samples	Tg (°C)	G' (GPa)
DBDBBB/33DDS	274	3.72
DBDBBB/44DDS	276	3.41

Fig. 6. Example of relevant table structures extracted from PDF documents.

4.3.2. EA tables cleaning

The CSV files are subsequently processed using Algorithm 2 to generate clean tables with normalized headers. The total number of distinct attributes across all tables before normalization was 2090, and after normalization, it was reduced to 1646, representing a **21.2%** reduction. To achieve this result, we had to customize the algorithm to fit the EA specifications by defining the reference attribute $\mathcal{U}'_{att}$ and its corresponding reference units $\mathcal{U}'_{unit}$ as follows:

$$\mathcal{U}'_{att} = \{V_{EA},\ SM(t),\ T_g\}$$

$$\mathcal{U}'_{unit} = \{\varnothing,\ MPa,\ °C\}$$

In this representation, SM(t) is the SM at the temperature (t). Furthermore, we distinguish three specific table layouts that influenced our normalization process, namely: The first layout concerns tables where the temperature (t) of the SM are indicated, as shown in Fig. 6b. These attributes are decomposed according to Algorithm 2 into (A_d, u_d, m_d), such as: (Storage modulus (20 °C), GPa, 1) and (Storage modulus (Tg+50 °C), GPa, 1). Based on this information, all values are converted to MPa. The second layout concerns tables where only extreme values (smallest and biggest) are indicated, as shown in Fig. 6a. In this case, the SM data recorded in the table are only those before and after T_g, identifiable by the keywords (glassy and rubbery), the columns names are therefore replaced by SM_{bf} and SM_{af}, respectively. The third layout concerns equivalences of quantities. For instance G' in Fig. 6c.2 is related to SM, sometimes noted as E', through the Poisson ratio ν, as follows:

$$E' = 2(1+\nu)G' \tag{6}$$

4.3.3. Building a unique consolidated table

To create a unique consolidated table with Algorithm 3, we define the following reference attributes of f_{EA}, $\mathcal{U}_{att} = \{V_{EA},\ SM_{bf},\ SM_{af},\ T_g\}$. However, we need to customize the algorithm to account for domain knowledge at two levels: Firstly, we use a partial integration approach, as tables from the same document can mutually enrich each other. Therefore, they are merged before the final integration process.

Example 11. Let us consider the two tables in Fig. 6 (c.1 & c.2) taken from the same document. Their column name "Sample" can serve as a key to group information, such as the "T_g" from c.2 and the corresponding "SM" from c.1.

Secondly, we compare the SM temperatures (t) with the T_g to determine whether it is SM_{bf} or SM_{af}. In cases of multiple values, the most extreme values are considered. The integration results in the creation of a unique consolidated table D_{EA} with **510** tuples over $\mathcal{U}_{att}$. However, it is a very ambitious and difficult task to gather data from the EA domain due to the lack of standardization in this field, thus the result has some inconveniences. We count **26,7%** of missing cells, accounting for 545 out of 2040 values. These missing values arise from two main factors: (1) not all publications address all the relevant attributes, some may include only one or two, and (2) in some cases, certain values are not found in tables but occasionally in graphs. After manually completing the data using additional information from the documents, we select only the best tuples that contain all the required attributes $\mathcal{U}_{att}$, resulting in a final usable subset of D_{EA} of **198** tuples over **4** attributes. The dataset is provided in Data and code availability.

4.4. EA counterexample detection and comparative analysis

We have obtained so far the table D_{EA}, illustrated in Table 5. Let us assume that for each attribute A_i of f_{EA} whose domain is $\mathbb{R}$, a predicate can be defined as in Eq. (2). In this formulation, the tolerance ϵ is expressed as $\epsilon = \alpha \cdot \text{avg}(A_i)$, where $\text{avg}(A_i)$ represents the average value of the data in the column corresponding to A_i, and $\alpha \in [0, 1]$ defines the proportion of this average used to set the tolerance threshold.

Table 5
Relation D_{EA} for f_{EA}.

Index	V_{EA}	SM_{bf}	SM_{af}	Tg	Ref
…	…	…	…	…	…
17	[Neat]	2075.32	35	85	[38]
90	[Neat]	2105	27.2	174.9	[39]
28	[DGEBA,DDM]	1620.7	7.6	143	[40]
93	[DGEBA,DDM]	1930	14	−16	[41]
30	[DGEBA,DDM]	2630	43	173	[42]
…	…	…	…	…	…

By setting $\alpha = 10\%$, we have the following thresholds: $\epsilon_{bf} = 330\,\text{MPa}$, $\epsilon_{af} = 8\,\text{MPa}$, and $\epsilon_{Tg} = 12°\text{C}$. We obtain the pairs of counterexample tuples, indicated by their index in Table 5: `(28, 93)` and `(17, 90)`. Assuming that only one tuple from each pair of counterexample tuples has to be eliminated to satisfy f_{EA} over D_{EA}, the g_3 value is $\frac{2}{198}$, i.e., **1%**, which is significantly low and reassuring, the domain knowledge is indeed valid. However, let us analyze the counterexamples one by one to conduct a comparative analysis of the related scientific paper:

Pair `(28, 93)`: We explain the discrepancies between these tuples by the fact that the T_g of the sample in the publication `(93)` [41] is measured in an unreacted state, in contrast to the sample in the publication `(28)` [40], which is a reacted sample. We decide to keep the reacted sample and exclude the other tuple. Furthermore, this counterexample leads us to manually optimize our initial relaxed FD as follows:

$$f_{EA} : V_{EA}, SM_{bf}, SM_{af} \rightarrow Tg_R \tag{7}$$

Where Tg_R is the T_g of the reacted EA samples.

Pair `(17, 90)`: We explain the discrepancies between these tuples by the fact that the values of their attribute V_{EA} are ambiguously stated in their table, where both are $V_{EA} = [Neat]$. In reality, we find that the exact values in their PDF text are, respectively:

$$V_{17} = [Bio-based\ Polymer,\ Super\ Sap\ 100/1000]\ [38]$$

$$V_{90} = [DGEBA(73\%),\ 33DDS(27\%)]\ [39]$$

Furthermore, we observe a discrepancy in the publication `(90)` [39], where the T_g is obtained using the DMA technique, which may introduce a methodological limitation. Thus, we remove the tuple `(90)` [39] from D_{EA} and update the value V_{EA} of the tuple `(17)` [38].

We can detect other counterexamples by varying α. However, based on our domain knowledge, we assume that 10% of the average value is a good threshold to capture relevant discrepancies in the data. Moreover, an effective strategy to find more counterexamples is to raise the threshold on the RFD's left-hand side attributes and lower or keep the threshold for attributes on the right-hand side. For example, for: $\epsilon_{bf} = 700\,\text{MPa}$, $\epsilon_{af} = 29\,\text{MPa}$, and $\epsilon_{Tg} = 12°\text{C}$, the tuple pair `(93, 30)` becomes a counterexample. However, based on domain knowledge, the selected thresholds are too high to indicate a meaningful contradiction between experiments.

We have thus proven that our relaxed FD is very consistent with our data, as the g_3 value is very low ($g_3 \simeq 0$). Moreover, the comparative analyses conducted on the counterexamples have highlighted, on the one hand, a precision on the characterized sample and, on the other hand, potential ambiguities that may arise in the papers tables.

5. Related work

This work draws on several research domains, including functional dependencies, meta-analysis, information extraction, and schema matching. The following review highlights the most relevant contributions in each area, emphasizing their connections and differences with the proposed CCASL framework.

The use of **Relaxed FDs** to model domain knowledge has been explored in prior work, notably for handling noisy data and identifying counterexamples. For instance, [21] formalizes counterexample analysis using the g_3-error metric, and provides open-source implementations in the FASTG3 library [22]. However, most existing works focus on discovering FDs and variants such as RFDs, top-k FDs from data [43–46], while approaches like CCASL reverses the perspective by starting from a meaningful RFD derived from expert knowledge, then confronting it with extracted data to identify contradictions.

Analyzing and detecting these contradictions can be framed as a **meta-analysis**, which is a statistical approach for combining quantitative data from multiple independent studies that address a common research question, thereby enhancing statistical power and reliability [2]. The CCASL method follows a similar rationale, aggregating and comparing findings across studies, but does so at a larger scale and in an almost automated manner. Traditional meta-analyses are often domain-specific and manually curated. For instance, studies have aggregated medical findings such as diagnostic test accuracy for COVID-19 [47] (16 publications) or the impact of Zn-doped synthetic polymers on bone regeneration [48] (10 publications). In contrast, our application to polymer science leverages 4055 publications, far beyond the scope of traditional manual efforts. Other studies perform large-scale text-based meta-analyses (e.g., ChatGPT's early impact [49]), yet none exploit structured tabular data, which is at the core of CCASL.

Table 6
Overview of representative works and their relationship to CCASL.

Reference	Domain	Data type	Main insights/Comparison to CCASL
[6,21,22]	RFDs analysis	Relational data	Models uncertainty and detects inconsistencies./CCASL applies RFDs to extracted scientific data.
[2,47–49]	Meta-analysis	Text/numeric	Combines results across studies; *manual*, domain-specific./CCASL *automates* large-scale meta-analysis using tabular data.
[9,50–52]	Text-based IE (NLP, LLM)	Unstructured text	Extracts entities and relations from text./CCASL uses NLP, as LLMs require large datasets.
[3,26,56]	Table-based IE (AI)	Structured tables	Captures numeric knowledge but requires a large amount of labeled data./CCASL combines AI and rules to handle noisy tables.
[31,53–55]	Schema matching (LLM, hybrid)	Relational schemas	Aligns heterogeneous datasets but needs large training corpora./CCASL uses rule-based matching for low-resource domains.

In the absence of structured data, extracting relevant information directly from documents becomes essential. **Information extraction (IE)** aims to transform unstructured or semi-structured text into structured data [14]. Existing IE methods mainly rely on three approaches: (i) NLP-based extraction [9,50,51], (ii) table-based analysis [3], and (iii) LLM-based extraction [52]. CCASL combines (i) and (ii) to jointly leverage textual and tabular data. While LLMs show promise, their effectiveness depends on large annotated corpora, resources currently unavailable in our experimental domain, hence our reliance on a hybrid AI and rule-based strategy.

An essential step for integrating the extracted information is **schema matching** [31], which aligns semantically related attributes across heterogeneous schemas. Recent studies have leveraged LLMs to improve this task in specialized domains. For example, [53] experiments with different prompting strategies for schema matching in healthcare, while [54] introduces *Matchmaker*, a compositional LLM-based program. Similarly, [55] proposes *Magneto*, combining small and large models for cost-efficient matching. Although effective, these approaches require large supervised datasets for fine-tuning. In contrast, CCASL uses a rule-based and human-in-the-loop strategy tailored to the polymer (epoxy-amine) domain, ensuring robustness despite limited training data.

In summary, while existing works provide essential building blocks, meta-analysis for cross-study synthesis, information extraction for data extraction, RFDs for modeling uncertainty, and schema matching for structural alignment, none integrate these elements into a unified pipeline for large-scale bibliographic confrontation. As summarized in Table 6, each family of methods addresses a specific aspect of the problem but remains limited when considered in isolation. The proposed CCASL framework bridges this gap by combining these concepts into a quasi automated approach capable of detecting contradictions across scientific publications.

6. Discussion

The CCASL method was conceived through close collaboration between epoxy-amine chemists and computer scientists. One of the main characteristics of this chemical research field is the scarcity of structured and openly available digital data. Experimental results are typically obtained through "bench-top" processes, with limited attention to systematic digital recording. Nevertheless, numerous research articles exist, often containing valuable tabular summaries within their bodies. These tables represent a rich yet unexploited source of structured information.

To address this, we leveraged recent advances in table detection and analysis from the NLP domain to automatically extract and consolidate tabular data from scientific publications. A major challenge in this process was the alignment of heterogeneous attribute names across different tables, sometimes even within the same paper, due to the lack of standardized notations and abbreviations in the chemical literature. Despite these difficulties, the proposed pipeline successfully produced a unified table in the epoxy-amine domain, consolidating 510 tuples from 4055 PDFs, as detailed in Section 4.

The formulation of RFD allows the identification of *counterexamples*, i.e., tuples that violate expected relationships between attributes. To the best of our knowledge, this is the first time that such a relational approach has been applied to scientific literature to uncover contradictions between independent studies. Our experiments revealed two main categories of counterexamples. The first one (i) corresponds to typographical inconsistencies, often resulting from notational variations, minor transcription errors, or ambiguous terminology. These cases can typically be resolved by correcting or harmonizing the extracted data. The second one (ii) is a more insightful category since it comprises methodological inconsistencies, where conflicting results emerge from different experimental protocols or analytical methods across publications. Such findings require domain experts to interpret the discrepancies, which can ultimately lead to refinement or revision of existing domain knowledge. For example, we found that the RFD we studied for epoxy–amine systems was valid only for Tg measured on already-reacted mixtures. This clarification allowed us to further refine the established RFD. In this sense, CCASL acts as an AI-based assistant for scientific validation and knowledge evolution.

The strength of CCASL lies in its integrative nature: it unifies four traditionally distinct research directions, meta-analysis, information extraction, schema matching, and counterexample analysis, into a coherent framework for large-scale bibliographic confrontation. As summarized in Table 6, while prior works focus on isolated aspects (e.g., manual meta-analysis, data aggregation or structural matching), CCASL bridges these elements into a single, knowledge-driven process capable of detecting contradictions directly from the literature.

Limitations. The current pipeline requires domain expert involvement at multiple stages: formulating appropriate RFDs, verifying the data normalization process, and interpreting the detected counterexamples. Although such involvement ensures accuracy and domain relevance, it restricts full automation. As a result, for a given application, the relevance of the approach depends on both the quality of the data and the expertise of the domain expert.

Perspectives. Beyond the polymer domain, the proposed approach is generalizable to other scientific fields where domain knowledge can be expressed as a function, and where relevant tabular data can be retrieved from documents. For instance, CCASL could be applied in medicine to identify conflicting patient records based on associated RFDs. Future work could explore the integration of multimodal data (e.g., images, graphs) to extend CCASL's applicability, as well as the development of semi-automated RFD suggestion mechanisms to reduce the burden on domain experts.

7. Conclusion

This work introduced CCASL, a novel methodology designed to detect and analyze contradictions across scientific publications by leveraging structured information contained in tables. Developed through close collaboration between epoxy–amine chemists and computer scientists, CCASL bridges the gap between domain expertise and automated document analysis. The approach relies on four main stages: (1) modeling domain knowledge with RFDs, (2) acquiring the related publications, (3) extracting and consolidating tabular data from those publications, and (4) detecting counterexamples that violate the modeled relationships and enabling comparative analysis of the corresponding publications.

Applied to the epoxy–amine domain, CCASL demonstrated its ability to process a large corpus of polymer literature. Based on an intuitive function modeled as an RFD, more than 4000 PDFs were downloaded, selected from over 158,531 DOIs obtained from scientific journals. From these PDFs, approximately 5000 tables were detected, of which around 1400 were classified as relevant, i.e., containing the attributes of interest. This process led to a consolidated dataset of 510 tuples with normalized attributes. Despite the inherent heterogeneity and incompleteness of the extracted data, CCASL successfully identified pairs of counterexamples highlighting inconsistencies between studies. These findings, validated by domain experts, revealed two main categories of inconsistencies, typographical and methodological, confirming the method's capacity to contribute to scientific validation and knowledge refinement. Although ambitious and challenging, the approach addresses the difficulties of data acquisition and normalization from heterogeneous sources, emphasizing the importance of close collaboration with domain experts during table extraction, cleaning, and schema matching.

Future work will focus on extending CCASL in several directions. We aim to enhance the robustness of table extraction and schema alignment through multi-modal learning and large language models. In addition, we plan to generalize the approach to other scientific fields such as medicine or materials informatics, where tabular data and functional dependencies are prevalent. Finally, we intend to explore the automated discovery of RFDs to reduce dependence on manual expert input and to transform CCASL into an intelligent assistant capable of uncovering new scientific hypotheses. In summary, CCASL demonstrates how domain knowledge, when formalized as a function and combined with large-scale document analysis, can reveal both data inconsistencies and new research insights, marking a promising step toward AI-assisted scientific validation.

CRediT authorship contribution statement

Aymar Tchagoue: Writing – review & editing, Writing – original draft, Methodology, Formal analysis, Data curation, Conceptualization. **Véronique Eglin:** Writing – review & editing, Supervision, Methodology. **Sébastien Pruvost:** Writing – review & editing, Supervision. **Jean-Marc Petit:** Writing – review & editing, Supervision, Methodology, Conceptualization. **Jannick Duchet-Rumeau:** Supervision, Funding acquisition. **Jean-Francois Gerard:** Supervision, Project administration.

Funding

This work benefited from the state aid managed by the National Research Agency, France under the France 2030 program, bearing the reference ANR-22-PEXD-0004 [34].

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Aymar TCHAGOUE reports financial support and administrative support were provided by National Institute of Applied Sciences of Lyon. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The code used in this study is available on github and the associated dataset is publicly available on zonodo.

References

[1] R. Whittemore, A. Chao, M. Jang, K.E. Minges, C. Park, Methods for knowledge synthesis: An overview, Hear. Lung: J. Acute Crit. Care 43 (5) (2014) 453–461, http://dx.doi.org/10.1016/j.hrtlng.2014.05.014.

[2] M. Crowther, W. Lim, M.A. Crowther, Systematic review and meta-analysis methodology, Blood 116 (17) (2010) 3140–3146, http://dx.doi.org/10.1182/blood-2010-05-280883, Review article.

[3] M. Kasem, et al., Deep learning for table detection and structure recognition: A survey, 2022, http://dx.doi.org/10.48550/arXiv.2211.08469, arXivPreprint. arXiv:2211.08469.

[4] R. Fagin, Functional dependencies in a relational database and propositional logic, IBM J. Res. Dev. 21 (6) (1977) 534–544, http://dx.doi.org/10.1147/rd.216.0534.

[5] S. Lopes, J. Petit, L. Lakhal, Functional and approximate dependency mining: database and FCA points of view, J. Exp. Theor. Artif. Intell. 14 (2–3) (2002) 93–114, http://dx.doi.org/10.1080/09528130210164143.

[6] Y. Chen, A. Darwiche, Identifying causal effects under functional dependencies, "Entropy" 26 (12) (2024) 1061, http://dx.doi.org/10.3390/e26121061, Author to whom correspondence should be addressed. URL https://www.mdpi.com/1099-4300/26/12/1061.

[7] M.E. Deagen, J.P. McCusker, T. Fatey, S. Stouffer, L.C. Brinson, D.L. McGuinness, L.S. Schadler, FAIR and interactive data graphics from a scientific knowledge graph, Sci. Data 9 (2022) 239, http://dx.doi.org/10.1038/s41597-022-01352-z.

[8] L. Schmidt, A.N. Finnerty Mutlu, R. Elmore, et al., Data extraction methods for systematic review (semi)automation: Update of a living systematic review [version 3; peer review: 3 approved], F1000Research 10 (2025) 401, http://dx.doi.org/10.12688/f1000research.51117.2.

[9] S.V. Mahadevkar, S. Patil, K. Kotecha, et al., Exploring AI-driven approaches for unstructured document analysis and future horizons, J. Big Data 11 (2024) 92, http://dx.doi.org/10.1186/s40537-024-00948-z.

[10] H. Morgan, Conducting a qualitative document analysis, Qual. Rep. 27 (1) (2022) 64–77, http://dx.doi.org/10.46743/2160-3715/2022.5044.

[11] I. Mendil, A framework for explicit modelling of domain knowledge in state-based formal methods: the case of interactive critical systems (Ph.D. thesis), Institut National Polytechnique de Toulouse - INPT, Toulouse, France, 2023, English. NNT: 2023INPT0074, tel-04611765.

[12] Z. Shen, R. Zhang, M. Dell, B.C.G. Lee, J. Carlson, W. Li, LayoutParser: A unified toolkit for deep learning based document image analysis, in: J. Lladós, D. Lopresti, S. Uchida (Eds.), Document Analysis and Recognition – ICDAR 2021, in: Lecture Notes in Computer Science, vol. 12821, Springer, 2021, http://dx.doi.org/10.1007/978-3-030-86549-8_9, ICDAR 2021 Conference Proceedings.

[13] A. Biswas, L. Reddi, E. Belval, Amazon textract's new layout feature introduces efficiencies in general purpose and generative AI document processing tasks, Technical Report ML-15397-Textract-Layout, Amazon, 2023, http://dx.doi.org/10.13140/RG.2.2.34815.62881.

[14] G. Zaman, H. Mahdin, K. Hussain, A.-u.-R. Rahman, Information extraction from semi and unstructured data sources: a systematic literature review, ICIC Express Lett. 14 (6) (2020) 593–603, http://dx.doi.org/10.24507/icicel.14.06.593.

[15] H. Pulikkalparambil, S.M. Rangappa, S. Siengchin, J. Parameswaranpillai, Introduction to epoxy composites, in: Epoxy Composites, Wiley, 2021, http://dx.doi.org/10.1002/9783527824083.ch1.

[16] M. Levene, G. Loizou, A Guided Tour of Relational Databases and Beyond, first ed., Springer London, 2012, p. XIV + 625, http://dx.doi.org/10.1007/978-0-85729-349-7.

[17] H.M. J. Kivinen, Approximate inference of functional dependencies from relations. theoretical computer science, 149(1):129–149, 1995, http://dx.doi.org/10.1016/0304-3975(95)00028-U.

[18] L. Caruccio, V. Deufemia, G. Polese, Relaxed functional dependencies—A survey of approaches, IEEE Trans. Knowl. Data Eng. 28 (1) (2016) 147–165, http://dx.doi.org/10.1109/TKDE.2015.2472010.

[19] B. Chardin, E. Coquery, M. Pailloux, J.-M. Petit, RQL: A query language for rule discovery in databases, Theoret. Comput. Sci. 658 (Part B) (2017) 357–374, http://dx.doi.org/10.1016/j.tcs.2016.11.004.

[20] S. Song, F. Gao, R. Huang, C. Wang, Data dependencies extended for variety and veracity: A family tree, IEEE Trans. Knowl. Data Eng. 34 (10) (2022) 4717–4736, http://dx.doi.org/10.1109/TKDE.2020.3046443.

[21] P. Faure-Giovagnoli, Domain knowledge and functions in data science, application to hydroelectricity production (Ph.D. thesis), INSA de Lyon, Lyon, France, 2023, Databases [cs.DB], English, NNT: 2023ISAL0093, tel: 04519659.

[22] P. Faure-Giovagnoli, J.-M. Petit, V.-M. Scuturici, Fastg3 - a python library for computing the g3 indicator efficiently, 2022, Software, HAL preprint: hal-04118253.

[23] Google Scholar, About, how are documents ranked?, 2024, Website. URL https://scholar.google.com/intl/fr/scholar/about.html.

[24] A. Patel, O. Kravchenko, I. Manas-Zloczower, Effect of curing rate on the microstructure and macroscopic properties of epoxy fiberglass composites, Polymers 10 (2) (2018) 125, http://dx.doi.org/10.3390/polym10020125.

[25] A. Kay, Tesseract: An open-source optical character recognition engine, Linux J. 2007 (159) (2007) 2.

[26] X. Zhong, J. Tang, A.J. Yepes, PubLayNet: Largest dataset ever for document layout analysis, 2019, arXiv:1908.07836 [cs.CL].

[27] W. Ughetta, B.W. Kernighan, The old bailey and OCR: Benchmarking AWS, azure, and GCP with 180,000 page images, in: Proceedings of the ACM Symposium on Document Engineering (DocEng '20), 2020, pp. 1–4, http://dx.doi.org/10.1145/3395027.3419595.

[28] T. Hitchcock, R. Shoemaker, C. Emsley, S. Howard, J. McLaughlin, et al., The old bailey proceedings online, 1674-1913, 2023, Version 9.0, Autumn 2023, Online resource.

[29] X. Schmitt, S. Kubler, J. Robert, M. Papadakis, Y. LeTraon, A replicable comparison study of NER software: Stanfordnlp, NLTK, opennlp, spacy, gate, in: Proceedings of the Sixth International Conference on Social Networks Analysis, Management and Security, SNAMS, Granada, Spain, 2019, pp. 338–343, http://dx.doi.org/10.1109/SNAMS.2019.8931850.

[30] V.I. Levenshtein, Binary codes capable of correcting deletions, insertions, and reversals, Sov. Phys. Dokl. 10 (8) (1966) 707–710.

[31] E. Rahm, P.A. Bernstein, A survey of approaches to automatic schema matching, VLDB J. 10 (2001) 334–350, http://dx.doi.org/10.1007/s007780100057.

[32] M.C. Swain, J.M. Cole, ChemDataExtractor: A toolkit for automated extraction of chemical information from the scientific literature, J. Chem. Inf. Model. 56 (10) (2016) 1894–1904, http://dx.doi.org/10.1021/acs.jcim.6b00207.

[33] A. Mazarei, R. Sousa, J. Mendes-Moreira, et al., Online boxplot derived outlier detection, Int. J. Data Sci. Anal. 19 (2025) 83–97, http://dx.doi.org/10.1007/s41060-024-00559-0.

[34] PEPR DIADEM, Projet AMETHYST, France 2030 PEPR DIADEM, 2025, URL https://www.pepr-diadem.fr/projet/amethyst/.

[35] W. Jilani, N. Fourati, C. Zerrouki, P.-A. Faugeras, A. Guinault, R. Zerrouki, H. Guermazi, Exploring the structural properties and enhancement of opto-electrical investigations for the synthesized epoxy-based polymers with local nanoscale structures, Mater. Res. Express 7 (3) (2020) 035305, http://dx.doi.org/10.1088/2053-1591/ab7b2a.

[36] J.D. Menczel, R.B. Prime, Thermal Analysis of Polymers: Fundamentals and Applications, Wiley, 2008, http://dx.doi.org/10.1002/9780470423837.

[37] H. Nour, J.-L. Auge, O. Gain, S. Pruvost, Epoxy/poly-etherimide blends for electrical insulation, in: 2016 IEEE Electrical Insulation Conference, EIC, Montreal, QC, Canada, 2016, pp. 313–316, http://dx.doi.org/10.1109/EIC.2016.7548590.
[38] B.J. Tiimob, S. Jeelani, V.K. Rangari, Eggshell reinforced biocomposite—An advanced "green" alternative structural material, J. Appl. Polym. Sci. 133 (11) (2015) http://dx.doi.org/10.1002/app.43124.
[39] J.M. Misasi, Q. Jin, K.M. Knauer, S.E. Morgan, J.S. Wiggins, Hybrid POSS-hyperbranched polymer additives for simultaneous reinforcement and toughness improvements in epoxy networks, Polymer 117 (2017) 54–63, http://dx.doi.org/10.1016/j.polymer.2017.04.007.
[40] H. Nabipour, X. Wang, L. Song, Y. Hu, A high performance fully bio-based epoxy thermoset from a syringaldehyde-derived epoxy monomer cured by furan-derived amine, Green Chem. 23 (2021) 501–510, http://dx.doi.org/10.1039/d0gc03451g.
[41] M.G. Prolongo, F.J. Martínez-Casado, R.M. Masegosa, C. Salom, Curing and dynamic mechanical thermal properties of epoxy/clay nanocomposites, J. Nanosci. Nanotechnol. 10 (4) (2010) 2870–2879, http://dx.doi.org/10.1166/jnn.2010.1385.
[42] S. Grishchuk, Z. Mbhele, S. Schmitt, J. Karger-Kocsis, Structure, thermal and fracture mechanical properties of benzoxazine-modified amine-cured DGEBA epoxy resins, Express Polym. Lett. 5 (3) (2011) 273–282, http://dx.doi.org/10.3144/expresspolymlett.2011.27.
[43] J. Liu, J. Li, C. Liu, Y. Chen, Discover dependencies from data—A review, IEEE Trans. Knowl. Data Eng. 24 (2) (2012) 251–264, http://dx.doi.org/10.1109/TKDE.2010.197.
[44] H. Yao, H.J. Hamilton, Mining functional dependencies from data, Data Min. Knowl. Discov. 16 (2) (2008) 197–219, http://dx.doi.org/10.1007/s10618-007-0083-9.
[45] P. Mandros, M. Boley, J. Vreeken, Discovering reliable approximate functional dependencies, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'17, 2017, pp. 355–363, http://dx.doi.org/10.1145/3097983.3098062.
[46] J. Liu, F. Ye, J. Li, J. Wang, On discovery of functional dependencies from data, Data Knowl. Eng. 86 (2013) 146–159, http://dx.doi.org/10.1016/j.datak.2013.01.008.
[47] B. Böger, M.M. Fachi, R.O. Vilhena, A.F. Cobre, F.S. Tonin, R. Pontarolo, Systematic review with meta-analysis of the accuracy of diagnostic tests for COVID-19, Am. J. Infect. Control 49 (1) (2021) 21–29, http://dx.doi.org/10.1016/j.ajic.2020.07.011.
[48] S. Jiang, Y. Zhang, F. Alsaikhan, A.T. Jalil, M.K. Gol, A. Tarighatnia, A meta-analysis review of the effect of Zn-doped synthetic polymer materials on bone regeneration, J. Drug Deliv. Sci. Technol. 76 (2022) 103792, http://dx.doi.org/10.1016/j.jddst.2022.103792.
[49] C. Leiter, R. Zhang, Y. Chen, J. Belouadi, D. Larionov, V. Fresen, S. Eger, ChatGPT: A meta-analysis after 2.5 months, Mach. Learn. Appl. 16 (2024) 100541, http://dx.doi.org/10.1016/j.mlwa.2024.100541.
[50] E.A. Olivetti, J.M. Cole, E. Kim, O. Kononova, G. Ceder, T.Y.-J. Han, A.M. Hiszpanski, Data-driven materials research enabled by natural language processing and information extraction, Appl. Phys. Rev. 7 (2020) 041317, http://dx.doi.org/10.1063/5.0021106.
[51] K. Lo, J.C. Chang, A. Head, J. Bragg, A.X. Zhang, C. Trier, C. Anastasiades, T. August, R. Authur, D. Bragg, et al., The semantic reader project: Augmenting scholarly documents through AI-powered interactive reading interfaces, Commun. ACM 67 (10) (2024) 50–61, http://dx.doi.org/10.1145/3659096.
[52] J. Dagdelen, A. Dunn, S. Lee, N. Walker, A.S. Rosen, G. Ceder, et al., Structured information extraction from scientific text with large language models, Nat. Commun. 15 (2024) 1418, http://dx.doi.org/10.1038/s41467-024-45563-x.
[53] M. Parciak, B. Vandevoort, F. Neven, L.M. Peeters, S. Vansummeren, Schema matching with large language models: An experimental study, 2024, http://dx.doi.org/10.48550/arXiv.2407.11852, arXiv Preprint.
[54] N. Seedat, M. van der Schaar, Matchmaker: Self-improving large language model programs for schema matching, 2024, http://dx.doi.org/10.48550/arXiv.2410.24105, arXiv Preprint.
[55] Y. Liu, E. Pena, A. Santos, E. Wu, J. Freire, Magneto: Combining small and large language models for schema matching, 2024, http://dx.doi.org/10.48550/arXiv.2412.08194, arXiv Preprint.
[56] A. Gemelli, E. Vivoli, S. Marinai, Graph neural networks and representation embedding for table extraction in PDF documents, 2022, http://dx.doi.org/10.1109/ICPR56361.2022.9956590.
[57] A. Tchagoue, V. Eglin, J.-M. Petit, S. Pruvost, J. Rumeau, J.-F. Gerard, A fine-tuned language model approach for accurate polymer glass transition temperature prediction, Journal of Chemical Information and Modeling (2025) http://dx.doi.org/10.1021/acs.jcim.5c02469.
[58] F.c. Wieckowiak, V. Eglin, T. Bonnet, S. Bres, L. Rousseau, A multimodal evaluation pipeline for mathematical expression recognition: comparisons of datasets, metrics, and models, 16026, Springer, 2026, pp. 120–136, http://dx.doi.org/10.1007/978-3-032-04627-7_7,
[59] M. Marcy, J.-M. Petit, V.-M. Scuturici, J. Bonjour, C. Fertel, G. Cavalier, Can surrogate keys negatively impact data quality?, Proceedings of the VLDB Endowment 18 (12) (2025) 5279–5282, http://dx.doi.org/10.14778/3750601.3750651.
[60] L. Nourine, J.-M. Petit, S. Vilmin, Towards declarative comparabilities: application to functional dependencies, Journal of Computer and System Sciences 146 (2024) 103576, http://dx.doi.org/10.1016/j.jcss.2024.103576.
[61] A. Topalian, F.c. Méchin, J. Duchet-Rumeau, R. Klucker, J.-F. Gérard, Effect of the formation of hydantoin in aspartate-based polyurea networks, Progress in Organic Coatings 200 (2025) 109102, http://dx.doi.org/10.1016/j.porgcoat.2025.109102.

Aymar Tchagoue is a Ph.D. candidate at LIRIS (INSA Lyon, CNRS UMR 5205), working within the Imagine and BD teams. His doctoral research focuses on applying artificial intelligence to polymers, including information extraction, property prediction, and chemical formulation generation. In 2023, he presented his work at the GFP National Polymer Group Conference: "Towards AI-based Polymer Discovery: An Optimized Data Collection System from Epoxy/Amine Tables" (Colloque National du GFP, Bordeaux). He recently published a study on a novel method for predicting glass transition temperature, significantly improving prediction results [57] (JCIM, 2025).

Véronique Eglin is a Professor at LIRIS (INSA Lyon, CNRS UMR 5205) and leads the Imagine team. Her research expertise includes document image analysis, classification, information retrieval in images, and low-level image processing. Her recent work spans multimodal information extraction; for example, Wieckowiak et al. [58] proposed a multimodal evaluation pipeline for mathematical expression recognition (ICDAR 2025).

Jean-Marc Petit is a senior researcher at LIRIS (INSA Lyon, CNRS UMR 5205), specializing in knowledge representation, schema matching, and data integration across heterogeneous scientific data sources. His work contributes to bridging structured data (like tables) and semantic knowledge in scientific literature [59,60].

Sébastien Pruvost is a Professor at IMP (INSA Lyon, CNRS UMR 5223). He studies the relationship between polymer structure, morphology, and physical properties, with a focus on dielectric behavior, molecular mobility, and conductivity in polymer networks. He also coordinates the high throughput platform for polymer composite synthesis and characterization at IMP.

Jannick Duchet-Rumeau is Director of the IMP laboratory (INSA Lyon, CNRS UMR 5223). Her research explores the nanostructuring of polymeric materials, both homogeneous and multiphase, with emphasis on interfaces and structure–property relationships. She is deeply engaged in sustainable polymer design,

developing materials with controlled degradation or recyclability. She co-authored work on the formation of hydantoin in aspartate-based polyurea networks [61].

Jean-Francois Gerard is a senior researcher at IMP (INSA Lyon, CNRS UMR 5223), specializing in the synthesis and characterization of polymer networks and thermosets. He has co-authored work on advanced epoxy thermosets derived from ionic liquid monomers. His collaborations with Jannick Duchet-Rumeau contribute to the development of functional and sustainable polymer materials.